TESIS DOCTORAL

STUDY AND IMPLEMENTATION OF NEUTRONIC-THERMOHYDRAULIC COUPLING METHODS FOR DETAILED ANALYSIS OF TRANSIENTS IN NUCLEAR REACTORS FOR SAFETY MARGINS OPTIMIZATION

TESIS DOCTORAL
PARA LA OBTENCIÓN DEL
TÍTULO DE **DOCTOR**

Septiembre 2019

Adrián Sabater Alcaraz

DIRECTOR DE LA TESIS DOCTORAL:
Diana Cuervo Gómez
Carolina Ahnert Iglesias

A mis padres, Andrés y Fini

Abstract

Understanding the performance of a nuclear reactor is a complex task that covers different physical fields as the neutron behavior or fluid dynamics. These fields are not isolated, and there is a feedback among them.

Along nuclear engineering history, different codes have been developed to analyze each of the physics in which the problem can be split in an isolated way. An example of this are the neutronic or the thermal-hydraulic codes. To study the performance of the reactor in a realistic way, it is necessary to take into account the interaction among the physics, or in other words, the codes have to be coupled.

Several methodologies or models can be applied to develop a coupled system. A common way to classify the coupling is defining two types communication: internal and external. The external couplings are developed employing an external platform to exchange data among the codes. The external platform adapts the information from the source code to the requirements of the target code. The internal couplings do not require an external structure to exchange information. A set of subroutines are developed to allow the coupling. These subroutines are designed for specific codes. Modifying the code implicates to modify these subroutines.

The multi-physics platforms integrate several codes developed to analyze different fields of the physics involved in the nuclear reactor. These codes are coupled in different ways to supply a more realistic view of the performance of the system. Integration of the codes into these platforms may become a complex task depending on the code structure that was initially developed to be executed stand-alone.

At present, different kinds of multi-physics platforms exist that has been developed in the past, like SEANAP, or are being developed now, like NURESIM, CASL, MOOSE, etc. One of these platforms, NURESIM, is the result of the work performed by different European organizations through three European projects being the last one called NURESAFE. The work described in this Thesis was developed partially as part of the NURESAFE project but can also be considered as one of the applications of the product of this international collaboration.

As a result of previous European projects, the in-house UPM neutronic code, COBAYA, was integrated in the NURESIM platform although not yet with all its capabilities. Through the work performed for the last project, NURESAFE, the code was able to fully use both solvers included on it to perform either nodal or Pin-by-Pin diffusion calculations.

The Thesis has been carried out in the framework of the mentioned project with the objective of taking advantage of the capabilities provided by the multi-physics platform to coupled COBAYA with different codes and to use different spatial discretizations. This

has allowed to study deeply the effects that different types of spatial discretizations or coupling schemes may produce in the results obtained for different transient scenarios that are related with the plant safety.

Regarding the first topic, spatial discretization analysis, the Thesis has focused on analyzing the differences obtained when nodal or pin cell diffusion analysis are carried out in the neutronic field in combination with average channel or subchannel thermal-hydraulic calculations. Within the NURESIM European Platform, the data exchange and the execution are controlled by an external system, the computational platform SALOME. Each one of the codes coupled in the platform creates its own mesh or meshes to store the fields in order to exchange the information with the other codes. SALOME interpolates the fields to exchange among the several meshes created by each code. This capability of the platform has allowed to study the effect exerted by different combinations of meshes for the neutronic and the thermal-hydraulic solvers. This effect has importance in safety analysis where it is a common approach that both calculations differ in the node size definition what has an impact in the safety variables values obtained like fuel or clad temperature.

Concerning the other aspect studied, the effect that the temporal coupling can also have on the results obtained for the safety variables, the external coupling used with COBAYA code allows implementing different temporal coupling schemes without significant modifications in the codes. This type of study was firstly proposed in a previous European project, NURISP, that was also devoted to the development of the European platform but the study was not possible because the integrated codes did not fulfill at that moment all the requirements to allow the definitions of different temporal schemes. Therefore, it has been with the work of this Thesis when one of the objectives of this series of EU projects has been achieved.

The transient analyses have been performed employing four different coupling schemes. Three of them are improvements of the Operator Splitting method, an explicit coupling method that is widely used for temporal code coupling in multiphysics platforms where implicit coupling is not possible. The fourth one is the extrapolated method, a semi-implicit method that was previously used by the COBAYA code for an internal coupling with COBRA-III and COBRA-TF codes but that has not been used yet in the external coupling within the NURESIM platform.

In conclusion, this Thesis has made use of the capabilities included in the NURESIM European Platform through the coupled system COBAYA/COBRA-TF to carry out detailed analysis regarding safety variables for the plant operation. Further studies can be performed using other codes integrated in the Platform to capitalize the efforts made by the organizations involved.

Resumen

Entender el comportamiento de un reactor nuclear es una tarea complicada que abarca diferentes aspectos de la física, como por ejemplo la neutrónica y la termohidráulica. Además, cada uno de los campos de la física que aparecen en un reactor nuclear no están aislados y se van modificando unos a otros.

Tradicionalmente, cada uno de los campos de la física que están involucrados en un reactor nuclear se han modelizado con códigos que resuelven solo una parte del problema. Un ejemplo podría ser la neutrónica, hay software diseñado para resolver únicamente este aspecto del problema. Para poder estudiar el comportamiento de un reactor nuclear de la manera más realista posible, es necesario que estas físicas interactúen o, dicho de otra manera, que los códigos estén acoplados.

Los acoplamientos se pueden desarrollar siguiendo diferentes metodologías y distintos modelos. Lo más común es separarlos en dos grupos, acoplamientos externos e internos. Los acoplamientos externos se realizan entre dos o más códigos utilizando una plataforma entre ellos que se encarga de transformar la información que se obtiene de uno de los códigos al formato que necesita el código receptor. Por otro lado, los acoplamientos internos. Este tipo de acoplamiento no requiere de ninguna estructura externa. Para permitir el acoplamiento se desarrollan un conjunto de subrutinas específicas para los códigos que se quiere acoplar.

Los reactores nucleares cuentan con una fenomenología compleja. Al ir aumentando las capacidades computacionales, se han ido refinando los cálculos, y se han ido perfeccionando y aumentando los tipos de análisis que se realizan en un reactor nuclear. Estos cálculos van desde cálculos neutrónicos, hasta análisis de CRUD o el cálculo de los coeficientes de pérdida de carga de las rejillas mezcladoras con CFD. Para analizar cada vez más aspectos de los reactores nucleares y de los sistemas de seguridad de un reactor surgen las plataformas multifísica.

Las plataformas multifísica integran diferentes códigos, desarrollado para analizar aspectos aislados de un reactor nuclear, que trabajan de manera conjunta dando una visión lo más completa posible de este. Construir este tipo de plataformas suele ser una tarea bastante compleja, porque cada uno de los códigos tiene diferente origen y están diseñados para ser ejecutados de manera aislada, sin realimentación con otros códigos.

Hay varias de estas plataformas multifísicas como, por ejemplo: NURESIM, CASL, MOOSE, o SEANAP. EL trabajo descrito en esta tesis está realizado dentro del proyecto NURESAFE. NURESAFE es la tercera y última parte del proyecto NURESIM (la segunda fue NURISP). Durante NURISP, el equipo de la Universidad Politécnica de Madrid integró parcialmente COBAYA en la plataforma NURESIM, permitiendo realizar cálculos acoplado a nivel nodal. En la última fase, durante el proyecto NURESAFE, se integró COBAYA dentro de la plataforma NURESIM completamente. A partir de este punto, se podían ejecutar cálculos nodales y PbP dentro de la plataforma NURESIM, acoplado con cualquiera de los tres códigos termohidráulicos integrados, FLICA, SUBCHNAFLOW o COBRA-TF.

La tesis se ha realizado dentro del proyecto NURESAFE con el fin de aprovechar las ventajas que ofrece el acoplamiento externo desarrollado en la plataforma NURESIM. El intercambio de información y la ejecución de los códigos está controlado por un código externo, en el caso de NURESIM es SALOME. Cada uno de los códigos acoplados genera unas mallas con diferente nodalización geométrica donde se almacenan los resultados y SALOME interpola entre las diferentes mallas. El acoplamiento externo permite hacer cambios en el modelo matemático de acoplamiento temporal sin necesidad de introducir modificaciones significativas en los códigos.

Otro de los aspectos analizados es el efecto de la nodalización geométrica sobre el cálculo. Cada uno de los códigos tiene una nodalización radial diferente. SALOME interpola los campos que se van a intercambiar entre las mallas de cada uno de los códigos. Esto permite realizar cálculos con un mallado muy refinado con uno de los códigos mientras que, el resto de los códigos pueden resolver el problema con unos mallados más sencillos. Para poder analizar este aspecto del acoplamiento espacial se prepararon varios casos ficticios que generan un fuerte gradiente en el flujo neutrónico. Esto permitió analizar el efecto que ejerce la interpolación sobre los cálculos de criticidad y los transitorios. Este efecto tiene importancia en el análisis de seguridad porque los resultados de temperaturas de combustible y vaina son diferentes según las mallas que se combinen.

El acoplamiento externo desarrollado entre COBAYA y COBRA-TF a través de SALOME permite implementar diferentes modelos de acoplamiento temporal y analizar el efecto que ejercen sobre el transitorio. Este tipo de estudios fueron propuestos previamente durante el proyecto Europeo NURISP, el análisis no pudo llevarse a cabo durante el proyecto NURISP. En la última parte de la tesis se ha realizado justamente lo que se pensó para el proyecto NURISP, se han implementado varios modelos de acoplamientos temporal y analizar el efecto que estos tienen en la evolución de diferentes tipos de transitorios. Este trabajo ha permitido alcanzar uno de los objetivos del proyecto Europeo.

El análisis de los transitorios se ha realizado con cuatro modelos diferentes de acoplamiento. Todos ellos tienen como base el *Operator Splitting* (OS). Este método divide el problema en tantas partes como códigos hayan acoplados y es uno de los más utilizados para el acoplamiento entre códigos, los códigos se ejecutan siguiendo una

secuencia. Además, se analizan tres modelos más de acoplamiento, todos ellos mejoras del OS. Dos son modelos explícitos y el cuarto, el modelo extrapolado, es un modelo semi-implícito. Este último modelo fue utilizado anteriormente para el acoplamiento interno entre COBAYA con COBRAIII y COBRA-TF, pero no ha sido utilizado en el acoplamiento externo. Con los cuatro modelos de acoplamiento se analizan dos transitorios, buscando los efectos que tiene en la evolución de los transitorios cada uno de los acoplamientos.

Como conclusión, en esta tesis se ha podido realizar un estudio de algunas de las variables relacionadas con la seguridad de la planta, a través del acoplamiento externo entre COBAYA y COBRA-TF dentro de la plataforma NURESIM. En el futuro se podrían realizar los mismos estudios utilizando cualquier otro código integrado en la plataforma NURESIM y aprovechar el esfuerzo realizado por los equipos integrantes en el proyecto NURESIM.

Acknowledgements

Quiero dar las gracias a todas las personas que han hecho posible este trabajo y a las personas que he conocido de las que me llevo muy buenos recuerdos, si dejo a alguien, le pido perdón por adelantado. De antemano digo que escribir no es lo mío, prefiero FORTRAN.

En primer lugar, mi directora de tesis, Diana Cuervo. Le quiero agradecer que me diera la oportunidad de formar parte del equipo de Ciencia y Tecnología de Sistemas Avanzados de Fisión Nuclear, por haber sabido ayudarme cuando estoy perdido y por todo el esfuerzo que ha puesto en este trabajo. Para mí ha sido toda una experiencia a su lado y sin ella este trabajo no sería posible. A parte de una directora me llevo una amistad.

Tampoco me puedo olvidar de Carolina Ahnert, mi otra directora. Siempre que ha hecho falta su ayuda, ahí estaba. Muchas gracias por darme la oportunidad de estar en vuestro equipo.

Por supuesto quiero dar las gracias a mis padres, Andrés y Fini. Su trabajo va más allá de los años de tesis, desde antes de que empezara a estudiar en Madrid. Todo lo que he conseguido no habría sido posible sin vosotros. Cómo no, a mi hermana Andrea, mi sister y su tete, la que siempre está ahí para poder charlar sobre mis problemas.

Y por supuesto, le quiero dar las gracias a Edith. Ella ha sido la que día a día ha soportado mis agobios y mis problemas, la que me ha animado en el final, cuando encender el ordenador a veces suponía un gran esfuerzo. Por frases como "venga, que ya queda poco" solo puedo darte las gracias.

Mis compañeros de equipo Emilio, Santiago. Nuestras charlas en el despacho, las discusiones "radiador sí, radiador no", cotilleos, viajes, "esto no compila", etc. De Emilio y Santi solo puedo decir que ha sido un gusto haber podido trabajar con vosotros.

A mis amigos del departamento, Gonzalo, Jose Antonio, Karen, Pablo, Miguel, Emma, Alejandro, Bea, Marta, Maria José, Rafa, Julio, Lenin, Gonzalo, Antonio, Luis Felipe,.... Esos partidos de pádel con Jose Antonio, las salidas a Takos, nuestras discusiones del café en las que arreglábamos el mundo, etc. También a Gonzalo, por su ayuda cuando estás ya al final pero no ves el fin.

No puedo olvidar a Karen, creo que conocerte ha supuesto un cambio radical en mi vida ya que me llevó a conocer a tu hermana. Has pasado de buena amiga a formar parte de mi familia.

Por último, no me quiero dejar a Enusa Industrias Avanzadas y mis compañeros de trabajo. Aunque el último año y medio haya sido muy difícil, solo puedo decir que Enusa

ha facilitado en todo lo posible que pudiera seguir con mis estudios. Además, mis compañeros, Jorge, Paloma, Alfonso, Rosario, Carmen, Lustolde, David y Ruben se han asegurado de que acabara y me han hecho pasar buenos ratos, con unos buenos partidos y con frases como: "le vamos a decir a Edith que te quite el wifi" o "mucho ves tú Juego de Tronos".

Muchas gracias a todos

Contents

ABSTRACT	VI
RESUMEN	IX
ACKNOWLEDGEMENTS	XII
CONTENTS	XIV
LIST OF FIGURES	XVII
LIST OF TABLES	XXI
ABBREVIATIONS	XXII
Technological	xxii
Institutions	xxiii
CHAPTER 1 INTRODUCTION	24
1.1 Thesis origin	24
1.2 Purpose and scope of the thesis	26
1.3 Thesis outline	27
CHAPTER 2 STATE OF THE ART	30
2.1 Introduction	30
2.2 Coupling schemes available for temporal coupling	31
2.2.1 Operator Splitting coupling methods	31
2.2.2 Jacobian-Free Newton-Krylov methods	40
2.2.3 Approximated Block Newton Methods	42
2.3 Internal and external coupling	46
2.4 Multiphysics platforms and code systems	49
2.4.1 NURESIM	49
2.4.2 VERA-CS	51
2.4.3 NEAMS Workbench and MOOSE Applications	53
2.4.4 Universidad Politécnica de Valencia Multi-physics Platform	55
2.4.5 Others???	¡Error! Marcador no definido.
2.5 Codes used and coupled	58
2.5.1 COBAYA4	58
2.5.2 COBRA-TF	62

CHAPTER 3 SALOME PLATFORM DETAILS	68
3.1 SALOME description	68
3.1.1 Platform structure and integration of codes	68
3.1.2 Data flow and coupling schemes definition	70
3.1.3 MedCoupling library and fields interpolation	71
3.1.4 ICoCo standard	75
3.1.5 Integration of previous versions of COBAYA	76
3.2 Modifications in COBAYA integration: COBAYA4	79
 CHAPTER 4 DEVELOPMENTS IN THE INTEGRATION AND COUPLING OF COBAYA FOR STEADY-STATE CALCULATIONS	 83
4.1 Introduction	83
4.2 Initial stand-alone execution	83
4.3 Coupling with COBRA-TF	84
4.4 Spatial coupling using MedCoupling	86
4.5 Validation of the coupling: MCNP6/CTF system	91
4.5.1 MCNP6	91
4.5.2 CTF	91
4.6 Description of case analyzed	92
4.6.1 Fuel assembly specifications	92
4.6.2 Generation of cross sections	93
4.6.3 Differences in the mesh of COBRA-TF	94
4.7 Coupling algorithms used for steady-state calculations	95
4.8 Results and comparison	96
4.8.1 2D standalone calculations at CZP	96
4.8.2 3D standalone calculations at CZP	97
4.8.3 3D coupled calculation at HFP	98
 CHAPTER 5 SPATIAL DISCRETIZATION ANALYSIS	 109
5.1 Introduction	109
5.2 Fuel assembly analysis	110
5.2.1 Problem specification	110
5.2.2 Results and conclusions	112
5.3 Full core analysis	115
5.3.1 Overview of the NURESAFE benchmark	115
5.3.2 Steady-state analysis	116
5.3.3 Transient calculations	116

CHAPTER 6 TRANSIENT ANALYSIS IN NURESIM PLATFORM	127
6.1 Introduction	127
6.2 Temporal coupling with COBRA-TF	128
6.3 Coupling schemes implemented	129
6.3.1 Operator splitting	130
6.3.2 Enhanced methods	139
6.4 Rod ejection transient	146
6.4.1 Reactivity transient with the Operator Splitting	147
6.4.2 Reactivity transient with the first order Crank-Nicolson method	152
6.4.3 Reactivity transient with the second order Crank-Nicolson method	157
6.4.4 Reactivity transient with the Extrapolated Method	162
6.4.5 Comparison of results	168
6.5 MSLB transient	170
6.5.1 Comparison of results obtained with the different coupling schemes	170
6.6 Numerical convergence Rates	175
6.7 Results	180
CHAPTER 7 CONCLUSIONS	182
CHAPTER 8 FUTURE WORK	186
8.1 Effective fuel temperature	186
8.2 Time step calculation	188
8.3 Uncertainty propagation at each coupling scheme	188
8.4 Coupling effect DNB	189
8.5 Convergence algorithm Neutronic-Thermohydraulic	190
8.6 Implementation of Approximated Block Newton Methods	191
BIBLIOGRAPHY	193
ANNEX A	200
CURRICULUM VITAE	210

List of Figures

Figure 2.1 Coupling scheme among neutronic code and thermal-hydraulic and thermo-mechanic	31
Figure 2.2 Operator Splitting.....	33
Figure 2.3 Stagger time grid	34
Figure 2.4 FPI scheme. Both codes iterate until achieve the convergence.....	38
Figure 2.5 Diagram of JFNK method.....	41
Figure 2.6 Internal Coupling	46
Figure 2.7 External coupling.....	48
Figure 2.8 NURESIM platform with all the codes integrate into the platform.....	50
Figure 2.9 Homogenization of the fuel assembly from pin-by-pin to one node per fuel assembly	58
Figure 2.10 Homogenization at pin level. The radial node includes the fuel rod and a section of the subchannel around.....	59
Figure 2.11 Current and fluxes used in COBAYA4 equations	62
Figure 2.12 Radial nodalization of the fuel pellet	66
Figure 3.1 Control levels in NURESIM platform	69
Figure 3.2 Example of a SUPERVISION graph for coupling	71
Figure 3.3 Target and source mesh not overlapping.....	74
Figure 3.4 Evolution of COBAYA.....	77
Figure 3.5 SLAOME 6.6.0 post-proc.....	78
Figure 3.6 Structure of the internal coupling developed between COBAYA3 and the thermal-hydraulic codes	79
Figure 3.7 MedCoupling between the neutronic and any thermal-hydraulic code integrated into NURESIM	80
Figure 3.8 Neutronic thermal-hydraulic coupling in NURESIM platform through SALOME	82
Figure 4.1 Diagram stand-alone calculation.....	84
Figure 4.2 Diagram COBAYA4 and COBRA-TF	86
Figure 4.3 Interpolation between two axial meshes with exact overlap	87
Figure 4.4 Interpolation between two axial meshes without exact overlap.....	88

Figure 4.5 Radial Thermal-hydraulic mesh, blue, radial neutronic mesh, red at PbP sub-channel calculation	89
Figure 4.6 Thermal-hydraulic sub-channel and neutronic pin node	89
Figure 4.7 Neutronic mesh, fluid thermal-hydraulic mesh, and rod thermo-mechanic mesh.....	90
Figure 4.8 Radial view of the fuel assembly	93
Figure 4.9 Neutronic thermal-hydraulic iteration including first cross-sections libraries calculations	94
Figure 4.10 COBRA-TF channel centered cell and CTF rod centered cell	94
Figure 4.11 Diagram MCNP6-COBRA-TF	96
Figure 4.12 Average axial power profile from COBAYA4/COBRA-TF and MCNP6/CTF with Doppler temperature and MCNP6/CTF with average fuel temperature as the effective fuel temperature.....	99
Figure 4.13 Axial distribution of the average fuel temperature.....	100
Figure 4.14 Axial distribution of the average fluid temperature.....	101
Figure 4.15 Average axial density distribution	102
Figure 4.16 2D Radial Power distribution COBAYA4-COBRA-TF.....	103
Figure 4.17 2D radial power distribution from MCNP6-CTF employing the Doppler fuel temperature as an effective fuel temperature	104
Figure 4.18 2D radial power distribution from MCNP6-CTF employing the Average fuel temperature as an effective fuel temperature	105
Figure 4.19 Relative difference between COBAYA4/COBRA-TF and MCNP6/CTF using the average fuel temperature as effective fuel temperature, employing the solution from MCNP6 as a reference solution.....	106
Figure 4.20 Relative error between COBAYA4/COBRA-TF and MCNP6/CTF using the Doppler fuel temperature as effective fuel temperature, employing the solution from MCNP6 as a reference solution.....	107
Figure 5.1 Different meshes used for the three combinations for NK-TH resolutions.	111
Figure 5.2 Radial power distribution with three different radial nodalization and combinations.....	114
Figure 5.3 Power evolution nodal and pin by pin calculation	117
Figure 5.4 Zoom of the power evolution in the power excursion region.....	117
Figure 5.5 Average moderator temperature evolution nodal and pin by pin	118
Figure 5.6 Comparison Fq nodal and PbP integrated to assembly calculation	120
Figure 5.7 Comparison Fxy nodal and PbP integrated to assembly calculation.....	120
Figure 5.8: Evolution of the radial (axially-integrated) peaking factor Fxy , Fz (radially-integrated) peaking factor and Fq along the transient for pin-by-pin and nodal calculations	121
Figure 5.9 Radial power distribution at 95 seconds of transient obtained by the PbP simulation	122

Figure 5.10 Radial power distribution at 95 seconds of transient obtained by the nodal simulation	123
Figure 5.11 Relative difference taking the maximum value of the nodal solution as a reference solution	124
Figure 5.12: Axially-integrated radial power distribution for the pin-by-pin (at the pin level) and the nodal calculation	125
Figure 6.1 Neutronic thermal-hydraulic iteration	129
Figure 6.2 Numerical derivative methods	132
Figure 6.3 Operator Splitting scheme with COBAYA4 and COBRA-TF	135
Figure 6.4 Graphical representation of the predictor technique	137
Figure 6.5 Diagram of the OS	138
Figure 6.6 Staggered mesh scheme performance	139
Figure 6.7 Extrapolated method performance	144
Figure 6.8 Power and Doppler temperature evolution in the rod ejection transient using OS coupling method	148
Figure 6.9 Effect exerted by the time-step size on the transient evolution with OS ...	150
Figure 6.10 Effect exerted by time-step size on the fuel centerline temperature of the hottest fuel assembly employing OS method	151
Figure 6.11 Evolution of the average moderator temperature and density with the OS	152
Figure 6.12 Power and Doppler temperature evolution using first order Crank-Nicolson method	153
Figure 6.13 Effect exerted by the time-step size on the CN1	154
Figure 6.14 Effect exerted by time-step size on the fuel centerline temperature of the hottest fuel assembly employing CN1 method	157
Figure 6.15 Power and Doppler temperature evolution using second order Crank-Nicolson method	158
Figure 6.16 Comparison of the power and Doppler temperature evolution in the transient with the same time step, 0.005 seconds	159
Figure 6.17 Effect exerted by time-step size on the fuel centerline temperature of the hottest fuel assembly employing CN2	160
Figure 6.18 Reactivity and the Doppler temperature values along the transient using CN2, CN1 and OS methods	161
Figure 6.19 Power evolution with the extrapolated method	163
Figure 6.20 Power and average Doppler temperature evolution with EM zoom in the central region	164
Figure 6.21 Extrapolation with a different time step	165

Figure 6.22 Effect exerted by time-step size on fuel centerline temperature on the hottest fuel assembly by the EM.....	167
Figure 6.23 Comparison of the Four coupling schemes with time step 0.005 seconds	168
Figure 6.24 Time to run a rod ejection transient with the time step of 0.01 seconds .	169
Figure 6.25 Power and average Doppler temperature with the four coupling models implemented employing the same time step	171
Figure 6.26 Power and reactivity with the Extrapolated method and the Operator Splitting	172
Figure.6.27 Fuel centerline temperature hottest fuel assembly at the end of the transient with OS and Extrapolated model	173
Figure 6.28 Evolution of the average moderator temperature calculated by CTF and the extrapolate moderator temperature extrapolated.....	174
Figure 6.29 Order of accuracy extrapolated and OS coupling schemes of the total power	177
Figure 6.30 Order of accuracy of the CN first and second order schemes.....	178
Figure 6.31 Zoom order of accuracy Crank-Nicolson	179
Figure 6.32 MSLB employing a long-time step.....	180
Figure 8.1 Neutron capture within the fuel pellet (Santamarina and Bernard 2015) ..	187

List of Tables

Table 4.1 Geometry description of the fuel assembly	92
Table 4.2 Neutronic stand-alone 2D calculations with different codes	97
Table 4.3 Solution from the 3D stand-alone neutronic calculation, MCNP6, and COBAYA4	97
Table 4.4 Comparison of the K-EFF between COBAYA4/COBRA-TF and MCNP6/CTF employing two different effective fuel temperatures	98
Table 4.5 Fz from the three calculations using the value of COBAYA4/COBRA-TF as a reference solution	99
Table 5.1 Multiplication Factor	113
Table 5.2 Radial power peaking factors $F_{\Delta H}$ (axially integrated).....	113
Table 5.3 Axial power peaking Factor, Fz	113
Table 5.4: Comparison of parameters corresponding to the main events in the transient	118
Table 5.5: Peaking factors at the moment of maximum core power and differences in percent with respect to the pin-by-pin solution	119
Table 6.1 Magnitude of the peak power	148
Table 6.2 Comparison of the evolution of the peak power with the time step of CN1 and OS	153
Table 6.3 Differences between the values exchanged by the coupling and the values calculated by each code	155
Table 6.4 Comparison of the peak power values	158
Table 6.5 Differences between Doppler temperatures calculate and transfer through the feedback in a CN2 coupling	160
Table 6.6 Comparison of the power peak values	163

Abbreviations

Technological

ABN	Approximated Block Newton Method
ACMFD	Analytic Coarse-Mesh Finite-Differences. AEF Advanced European Fuel
ARI	All Rods In
ARO	All Rods Out
BEPU	Best-estimate plus uncertainties
BOC	Beginning of Cycle
BP	Burnable Poison
BWR	Boiling Water Reactor
CN	Crank-Nicolson
CRUD	Chalk River Unidentified Deposits
CZP	Cold Zero Power
EM	Extrapolated method
EOC	End Of Cycle
FA	Fuel Assembly
FMFD	Fine-Mesh Finite Differences
FPI	Fixed Point Iteration
HZP	Hot Zero Power
HFP	Hot Full Power
JFNK	Jacobian Free Newton-Krylov
MOX	Mixed Oxide
MSLB	Main Steam Line Break
NK	Neutron-Kinectics
NURESAFE	NUclear REactor SAFETY simulation platform
NURESIM	NUclear REactor SIMulation
NURISP	NUclear Reactor Integrated Simulation Project
OS	Operator Splitting
OTF	On-The-Fly
PbP	Pressurized Water Reactor
SEANAP	Sistema Español de Análisis de Núcleos de Agua a Presión
TH	Thermalhydraulics
TM	Thermomechanics
TMI	Three Mile Island

Institutions

CEA	Commissariat à l'énergie atomique et aux énergies alternatives
CSN	Consejo de seguridad Nuclear
GRS	Gesellschaft für Anlagen- und Reaktorsicherheit
HZDR	Helmholtz-Zentrum Dresden-Rossendorf
IAEA	International Atomic Energy Agency
INL	Idaho National Laboratory
NCSU	North Carolina State University
NEA	Nuclear Energy Agency
OECD	Organisation for Economic Co-operation and Development
ORNL	Oak Ridge National Laboratory
NK	

Chapter 1 Introduction

1.1 Thesis origin

A nuclear reactor is a very complex system where different phenomena are involved. Just to begin, there is a very important interaction between neutrons behavior and coolant dynamics that leads to the fact that Neutron Kinetics (NK) and Thermal-Hydraulics (TH) are highly coupled. These are not the only fields coupled. Fuel performance and coolant chemistry also interact with each other and the others making impossible to predict the behavior of the reactor precisely enough without taking into account at least two of the physical fields.

In the first decades of the nuclear engineering development, due to the lack of enough computing power, the nuclear industry and regulatory bodies employed different codes to simulate different fields imposing as boundary conditions the behavior of the other ones. This approach implies a high inaccuracy in the results, as they have to be obtained from a large amount of hypothesis. Later, simplified models of the most influential phenomena were introduced in the calculation in order to have a better accuracy without increasing greatly the computing overhead. In this way, point kinetic models in system thermal-hydraulics calculations or closed channel TH analysis in neutron physics codes were included what allowed to have a better approximation of the field that was not detailed analyzed.

Another effect of the lack of enough computational capabilities is the common practice of simulating large domains using limited number of nodes. In the case of a NK-TH simulation, this makes that the resolution of the fuel assembly is neglected giving as result the average value in nodes with dimensions equivalent to the assembly pitch. Taking into account that the detailed power distribution within the fuel assembly can impact the results obtained related with the safety limits like DNBR or clad temperature, a methodology for pin by pin power distribution reconstruction was commonly implemented in the NK calculation.

Nowadays it is still common this type of approach that can be defined as “traditional” multiphysics analysis. In (Ivanov et al. 2016) main characteristics of this approach are summarized.

As the computational capabilities were increasing, two lines of development showed up. On one side, the codes coupled became more realistic reducing approximations and increasing the complexity of equations to solve. This was the case for thermalhydraulics, where codes began to solve the two-phase equations in the more simplified combination.

This new approach, that can be defined as “novel” or state of the art, includes high-fidelity coupling on pin, or even subpin, and subchannel level to simulate the reactor core taking into account several physics phenomena such as neutronics, thermal-hydraulics, fuel performance, structural mechanics or chemistry. An important

To carry out this type of analysis, it is also necessary to introduce High Performance Computing (HPC) capabilities in the solvers and integrate them in a manner that a tight coupling may be performed without losing their capabilities.

With this purpose, several simulation platforms have been developed to integrate these new solvers. One of them, the European Platform for Nuclear Reactor Safety Simulation NURESIM has been part of the work performed by several European institutions to produce this type of high-fidelity tools. This platform has been developed along three projects founded by the European Commission; NURESIM, NURISP and NURESAFE.

The group of Reactor Physics at the Universidad Politécnica de Madrid (UPM) has extensive experience in the development of this types of platforms. Beginning on eighties the group developed the codes system SEANAP able to perform the simulation of PWR cores from refueling to operational transients covering all steps of the calculation including cross sections generation. This system included a 2D pin by pin calculation with the first version of COBAYA that supplied the nodal discontinuity factors and hot-pin to node average power ratios to the nodal calculation with code SIMULA (Castro 2018). This system is still used by some utilities to plan operational maneuvers and analyze transients.

With the participation of the group in the European Project NURESIM, COBAYA code was expended to perform fully 3D nodal and pin-by-pin calculations. Also, during this project, the code was integrated in the European Platform NURESIM. This version of the code was denoted as COBAYA3.

The integration was not yet full and only nodal calculations could be performed at this stage of the work. Anyway, this allowed to couple the code with FLICA IV thermalhydraulic code developed by *Commissariat à l'énergie atomique et aux énergies alternatives* (CEA) to perform steady-state and transient analysis although yet the at nodal level and using explicit operator splitting technique for temporal coupling.

During NURISP project, the code and coupling were tested in a boron dilution transient (G. Jimenez et al. 2015) comparing the results obtained with COBAYA3/FLICA-4 DYN3D/FLICA4 and DYN3D/SUBCHANFLOW.

Also during this project, a theoretical analysis of the temporal coupling methodologies that might be applicable using NURESIM platform concept was performed by (Omar

Zerkak, Kozlowski, and Gajev 2015) but it was not applied because the type of integration performed with the codes did not still allowed to establish different temporal coupling techniques within the platform.

The work performed by the UPM research group during NURESAFE project, where this Thesis has been developed, has allowed to use COBAYA code at the pin level as well as to define different temporal coupling techniques, all within the NURESIM platform at the same time that being coupled with a new thermal-hydraulic code integrated in the platform during this project, COBRA-TF.

Along NURESAFE, COBAYA3 was updated and became COBAYA4. COBAYA4 was fully integrated into NURESIM platform, allowing to perform nodal and pin-by-pin calculations in steady state and transient scenarios into NURESIM platform coupled with any of the three core-thermal-hydraulic codes integrated into NURESIM platform, COBRA-TF, SUBCHANFLOW, FLICA-IV. Among these three codes, COBRA-TF is the one chosen by the Universidad Politécnica de Madrid to carry out couple calculations in this project and the capabilities of COBAYA4 and the new thermal-hydraulic coupling with COBRA-TF were tested simulating a Main Steam Line Break (MSLB) scenario defined during the project as benchmarking exercise (Kozmenkov and Kliem 2014).

1.2 Purpose and scope of the thesis

This document includes the work performed by the candidate in order to integrate in a full manner COBAYA code into NURESIM platform, to couple the code with the core-thermalhydraulics code COBRA-TF and to analyze the impact that different spatial mappings and temporal coupling methodologies may have in the results obtained using the capabilities of coupling definition of the European Platform NURESIM.

In order to perform this work, original contributions have been performed. Some of them were part of the research group developments within NURESAFE project but other were carried out after finalizing the project. These all are summarized as follows:

- Full integration of COBAYA code into de NURESIM platform to perform coupled calculations with core-thermal-hydraulics codes available in the platform. COBAYA4 version of the code has been included within the set of codes available into the NURESIM European Platform. The code is now able to perform not only nodal but also pin-by-pin calculations and being coupled with different core-thermal-hydraulic codes as FLICA-IV, SUBCHANFLOW or COBRA-TF. Due to the characteristics of the computing platform, this coupling can be performed using different spatial meshes for the thermalhydraulic and neutronic calculations being the interpolation task transparent for the user. In this way, different resolutions at NK and TH physics can be tested or used for operational or safety analyses (García-Herranz et al. 2017) , (Spasov et al. 2017) , (Kliem et al. 2017).

- Detailed analysis of the effect of using channel averaged core-thermalhydraulics calculations coupled with pin-by-pin neutronic calculations in safety parameters as fuel or clad temperature. Traditionally, in plant safety analysis, the nodes at reactor core used by plant TH codes are including several assemblies but NK nodes are representing only one assembly. As TH variables used are channel averaged, the fluid behavior impact is being smeared when is supplied as feedback for the NK calculation. This effect can be analyzed using the COBAYA4 and COBRA-TF codes integrated in NURESIM as they can be coupled using both same or different resolution in NK or TH field. This study has been performed for a case included in a congress paper for the American Nuclear Society(Sabater, Cuervo, and Sánchez-Cervera 2016)
- Detailed analysis of the impact of using temporal coupling methodologies with different level of implicitness in external coupling for selected Reactivity Insertion Accidents (RIA). In this work, two RIA transients have been analyzed; a Main Steam Line Break (MSLB) and a Rod Ejection Accident (REA). They have been selected due to the fact that the origin of the reactivity insertion is very different. This will be explained in Chapter 5.

1.3 Thesis outline

The thesis is divided into nine chapters. The first chapter is the introduction where the framework in which the work has been performed is established. It includes also a summary of the original contributions of the thesis and the present outline.

Chapter 2 consists of the description of the state of the art regarding main topics of this work. The study described in this thesis concerns the analysis of the coupling among codes. The calculations performed for the thesis have been done employing the multi-physics platform NURESIM. NURESIM platform allows to implement different kinds of temporal coupling and to analyze the effect exerted by the couplings on the transient evolutions. The first step was to understand the different possibilities that could be selected to discriminate the most adequate. This study about the mathematical models for coupling has been described in the first part of the second chapter.

These models are the most widely employed by multi-physics platforms and coupling systems. NURESIM is the platform where COBAY4 have been integrated and coupled but is not the only one being developed and used currently. A short description of other multi-physics platforms is included in second part of Chapter 2 to highlight the importance that outstanding international institutions are giving to the multi-physics analysis using computational platforms.

This chapter includes also a detailed description of the mathematical coupling models implemented between COBAYA4 and COBRA-TF using the NURESIM platform

capabilities. NURESIM platform is ready to couple different codes through an External coupling, the best model for the external coupling is Operator Splitting (OS) technique and some variations of it were discovered adequate to be implemented between COBAYA4 and COBRA-TF. Therefore, a detailed description of these methods including discretization formulism is included in section 2.2.1.

Chapter 3 includes extensive explanation of the characteristics and capabilities of the SALOME computational platform where COBAYA code was integrated and coupled. The structure of the platform is described as well as the standards and physical models used to manipulate and transfer the data that is going to be managed by the coupled codes. Second part of Chapter 3 is devoted to explaining the work performed to integrate the COBAYA4 code into the platform.

Chapter 4 describes the work performed to couple and execute the COBAYA4/COBRA-TF system into the NURESIM European Platform. This included the development of the coupling scripts by mean of Python language, development of the different spatial meshes that will be used along this work and validation of the code by the use of the results obtained with reference system MCNP6/CTF in a collaboration with the Reactor Dynamics and Fuel Management Group at North Carolina State University.

A detailed analysis of the effect of the interpolation of the fields to exchange through the coupling has been included in Chapter 5. NURESIM platform performs the interpolation of the field between COBRA-TF and COBAYA4. The platform allows to carry out calculations combining different accuracy orders in the meshes. The combination of the meshes exert an effect on the calculations that has been investigated. The first part of the chapter analyzes these effects. The second part of the chapter proves the capabilities of COBAYA4 to perform a full core pin-by-pin coupled transient calculations.

Chapter 6 describes the effects exerted on the transient evolutions by the coupling models implemented between COBAYA4 and COBRA-TF and explained previously. The analyses were performed with two different types of reactivity transients; a Rod Ejection event and a Main Steam Line Break scenario. Both reactivity transients have different origin. The rod ejection changes the composition of the core resulting in a power excursion meanwhile, the MSLB modifies the inlet temperature at the core generating a power excursion. Other important difference between both transients is the time scale where they are evolving that goes from 1 second for the first to several minutes or hours (depending the objective of the analysis for the second) that they take. The sensitivity of some important safety variables like fuel temperature to the coupling scheme and other parameters of the coupling is demonstrated in this chapter.

Finally, Chapter 7 and Chapter 8 are the conclusions and future work. It would be very interesting to analyze the effective fuel temperature models used and the effect that this model can have in the results obtained for different transients. Moreover, the

transients have been executed with a fixed time step. It would be very convenient to include an adaptative time-step dependent on the coupling used and on the evolution of the transient. Also new coupling methods can be analyzed and implemented that can improve the COBAYA4/COBRA-TF performance within the NURESIM platform.

Chapter 2 State of the art

2.1 Introduction

As the computational capacities increase, the nuclear safety councils from different countries demand a more accurate analysis of the physics implicated in the behavior of the nuclear reactor. Moreover, accuracy increase of the analysis provides a better knowledge of the behavior of the components in operation and the response of the components in a nuclear accident. This knowledge allows the industry and the regulator to increase the operating margins and to know in a better way how the components of the reactor respond to an accident.

Different physical phenomena have to be studied to simulate a nuclear reactor. The main physics, although not the only ones, implicated in the problem are neutronics, thermal-hydraulic, and thermo-mechanic. The neutronic code calculates the flux and power distribution, the thermal-hydraulic code the moderator temperature and density distribution, and the thermal-mechanics code is able to predict the fuel temperature distribution (Figure 2.1.). The problem should be solved simultaneously to obtain a realistic picture of the processes that are occurring in the reactor. As this is not usually feasible, the general approach is to couple the codes in some way, giving as input of one of them the output of the other. This is performed iteratively for both types of calculations; steady state or transient evolution. When the steady state situation is the one that is being calculated, a convergence criterion in the physical fields values is established in order to check if solution of one code is still changing the results obtained with the other. When the change is below a defined value the solution has been reached.

In the case of analyzing temporal evolution of the system, data is exchanged between codes in different ways depending on the manner the equations are discretized, and the point of exchange that is defined. The methods of temporal coupling can be then classified into three groups, explicit, semi-implicit and implicit.

At each time step, the TH code calculates the new coolant properties, i.e. temperature, pressure, void fraction, phasic densities, and center line and surface fuel temperature. Some of these parameters are passed to the NK code once the thermalhydraulic code achieves the endpoint of the iteration. Then the NK solver updates the cross sections based of the new values of the thermalhydraulic parameters and, performs a new iteration.

Figure 2.1 Coupling scheme among neutronic code and thermal-hydraulic and thermo-mechanic

The temporal coupling implemented between different codes may exert a significant effect on the solution. This should be analyzed carefully to identify which is the optimum option among the ones that are available for the user.

During the work described in this document, several coupling methods were implemented between the NK code COBAYA4 and the TH code COBRA-TF. In order to know the possibilities that were available for implementing, a study of the options was performed and has been included in next section.

2.2 Coupling schemes available for temporal coupling

Different techniques are employed to perform a temporal coupling in the multi-physics analysis. The way this temporal coupling is performed has an impact in the transient evolution and the final safety analysis. In next sections different methods that can be used for this type of coupling are summarized.

2.2.1 Operator Splitting coupling methods

Temporal coupling is employed in the transient calculations. This kind of problems are called multi-physics problems because, in the case of core analysis, these variables are the thermal power, the moderator temperature, and density and the fuel temperature. The different physics involved in the problem are solved separately. The output of one physic is used as a boundary condition of another one. One example is the neutronic and thermal-hydraulic coupling used in the multi-physics core analysis. In one hand, the neutronic diffusion code reads the temperatures and density distribution from the thermal-hydraulic and obtain a new power distribution, and on the other hand, the thermal-hydraulic code reads the thermal power from the neutronic code and calculates the temperatures and density distribution.

2.2.1.1 Operator Splitting coupling technique

As Figure 2.2 shows, the physical problem to be solved can be divided into different sub-problems, as many as physic phenomena participates in the problem.

Different codes solve each one of the issues as a black-box. When one code converges, transfers the information to the other codes and carry out new calculations with the data obtained by the previous code as an input. This methodology to solved the multi-physics problems is called Operator-Splitting (OS) (Mahadevan and Ragusa 2006).

The coupling methods studied in this thesis are focused on the application to transient calculations. OS technique can be applied to analyze the steady-state, and transient calculations (F. Merino 1993). The coupling on the steady state calculations employs an OS and, in some cases a damping to improve the convergence. The damping was requires to achieve the convergence of the steady-state in a MSLB VVER from NURES SAFE project (Spasov et al. 2017).

In the nuclear multi-physics problems, OS technique is widely employed to solve the neutronic and thermal-hydraulic transient calculations. At each iteration, the neutronic code calculates the power distribution and the thermal-hydraulic code gets the moderator temperature and density distribution and the fuel effective temperature distribution.

The OS can be represented by two pairs of partial differential equations coupled (PDEs) together. The Eq (2-1) and (2-2) represent the PDEs systems. Both equations are coupled because the solution of one equation is required to solve the other equation.

$$\frac{\partial u}{\partial t} = g(u, v) \quad (2-1)$$

$$\frac{\partial v}{\partial t} = f(u, v) \quad (2-2)$$

The equations have to be discretized previously to solve the problem. Eq. (2-1) and (2-2) represent a semi-implicit problem. Eq. (2-1) is an explicit equation, but the Eq. (2-2) needs the term u^{n+1} of Eq. (2-1) to solve the problem. This is the reason why the system is semi-implicit.

$$\frac{u^{n+1} - u^n}{\Delta t} = g(u^n, v^n) \quad (2-3)$$

$$\frac{v^{n+1} - v^n}{\Delta t} = f(u^{n+1}, v^n) \quad (2-4)$$

Figure 2.2 Operator Splitting

Figure 2.2 shows graphically the behavior of the OS scheme:

1. The iteration begins with the iterations of sub-problem physics TH. This iteration begins at the point n and with the value u^n as an input the first code calculates v^{n+1} .
2. The parameter v^{n+1} is transfer to NK as a input.
3. Neutronics carries out the iteration taken as input u^{n+1} and beginign from point v^n .
4. The calculation obtains the value of v^{n+1} .
5. The value v^{n+1} is transferred to the thermal-hydraulic through the coupling.

These five points represent one iteration between physics. The process is iteratively repeated up to the end of the calculation.

In the LWR problem, Eq. (2-3 and (2-4 represent each one of the physics involved in the problem (could be more than two, as many as code are employed in the problem). One equation represents the neutronic and, another represents the thermal-hydraulic. Both move forward in time, and the beginning and the end of the iteration of two physics is the same.

In the case analyzed in this document, the neutronic code, COBAYA4, reads the temperatures and densities from the thermal-hydraulic code and computes new power and advance time step. The thermal-hydraulic code, COBRA-TF, reads the value of the power and it computes the new iteration to obtain the values of temperatures and densities.

There are many coupling systems than employ the OS as a coupling model; some of them are:

- DYN3D and SUBCHANFLOW (Daeubler et al. 2015)
- CATHARE2 and FLICA-IV (Kliem et al. 2017)
- DYN3D and COBRA-TF (Périn and Velkov 2017)

OS splitting is the easiest mode to develop coupling. There are other methodologies which are based on the OS and supposes an improvement of it.

This method can be improved in different ways. The two methods used widely are:

- Staggered time grids
- Higher-order treatment of non-linearities

Staggered time grids

The staggered time grid method is a variation of the operator splitting method shown in Figure 2.2. The method was employed firstly to solve the time-dependent Maxwell's equations, (Kane Yee 1966) and, can be employed in the multi-physics problem to coupled the codes together. Figure 2.3 shows a scheme of the method. This kind of method were employed originally to solve the coupled problems.

Figure 2.3 Stagger time grid

This method can be explained, as was before with the OS, through two coupled PDE equations, Eq. (2-1 and (2-2. The two equations represent each one of the physics involved in the problem.

The method can be solved by applying θ -scheme to the Eq. (2-5 and (2-6 (Omar Zerkak, Kozlowski, and Gajev 2015):

$$\frac{u^{n+\frac{1}{2}} - u^{n-\frac{1}{2}}}{\Delta t} = \theta_1 f(u^{n+1}, v^n) + (1 - \theta_1) f(u^n, v^n) \quad (2-5)$$

$$\frac{v^{n+1} - v^n}{\Delta t} = \theta_2 g\left(u^{n-\frac{1}{2}}, v^{n+\frac{1}{2}}\right) + (1 - \theta_2) g\left(u^{n+\frac{1}{2}}, v^{n-\frac{1}{2}}\right) \quad (2-6)$$

The value of parameter θ in the Eq. (2-5 and (2-6 allows to solve the problem employing different methods to discretize the semi-implicit PDEs. The most common values used are:

- $\theta = 1$: explicit method (first order accuracy).
- $\theta = 0$: implicit method (first order accuracy).
- $\theta = 0.5$: Crank-Nicolson scheme (second order accuracy).

Moreover, this technique allows also to introduce other methodologies just by changing the values of the θ_1 and θ_2 parameters. When these parameters are set to values higher than 1, an extrapolation of the variables calculated by one code is passed to the other. A detailed analysis of this technique will be addressed in Chapter 6.

An example of the staggered grid with the value of $\theta_1 = \theta_2 = 1.5$ was analyzed by the Universidad Politécnica de Madrid, coupling SIMTRAN and RELAP-5 and employed to analyze a MSLB (Aragonés et al. 2004) and with COBAYA3 and COBRA-TF (G. Jimenez et al. 2015).

Higher-order treatment of non-linearities

One of the problems is the treatment of the non-linearities terms of the equations. The non-linearities produce loss of accuracy achieved in the methods explained previously (staggered mesh discretization). Different strategies are employed to restore the accuracy order. The high order methods are employed to correct the problem of the non-linearities (Ragusa and Mahadevan 2009).

The problem about the nonlinearities is illustrated by a partial Eq. (2-7 where function the function $f(v, t)$ is split into two terms, $N(t, v)$ that represents the non-linear component and $L(t)$ that represents the lineal component.

$$\frac{\partial v}{\partial t} = f(v, t) = L(t)v + N(t, v) \quad (2-7)$$

Applying a Crank-Nicolson discretization scheme:

$$\begin{aligned} \left[I - \frac{\Delta t}{2} L(t^n) \right] v^n &= \left[I - \frac{\Delta t}{2} L(t^{n-1}) \right] v^{n-1} + \\ &+ \frac{\Delta t}{2} [N(t^n, v^n) + N(t^{n-1}, v^{n-1})] + O(\Delta t^3) \end{aligned} \quad (2-8)$$

Eq. (2-8 is implicit because of the term v^n requires solving the nonlinear term $N(t^n, v^n)$.

There are different options to address the problem. The simplest solution consists of linearizing the term $v^{n,p}$. With the first order linearization is it possible to predict v^n , Eq. (2-9, and the nonlinear term Eq. (2-10).

$$v^{n,p} \approx v^{n-1} + O(\Delta t) \quad (2-9)$$

$$N(t^n, v^n) = N(t^n, v^{n,p}) + \Delta(t) \quad (2-10)$$

This method is easy to implemented but degrades the convergence order of the Eq. (2-8 to first order (Ragusa and Mahadevan 2009). represents the first accuracy order with the linearization of the non-linear term.

$$\begin{aligned} \left[I - \frac{\Delta t}{2} L(t^n) \right] v^n &= \left[I - \frac{\Delta t}{2} L(t^{n-1}) \right] v^{n-1} + \\ &+ \frac{\Delta t}{2} [N(t^n, v^n) + N(t^{n-1}, v^{n-1})] + O(\Delta t^3) \end{aligned} \quad (2-11)$$

Other techniques can be employed to obtain the nonlinear term $N(t^n, v^n)$ and keep the accuracy order like the *High Order Explicit Linearization* method or the *Picard Iterations* also known as Fixed point method.

High order explicit linearization

Predict-Evaluated correct methods (PECM) allow restoring the high accuracy order employing a high order explicit linearization instead a first-order linearization as Eq. (2-10).

The prediction formula of the term $v^{n,p}$ in the Eq. (2-8 can be obtained from the Taylor with a high order (Ragusa and Mahadevan 2009):

$$f(x + h) = f(x) + hf'(x) + \dots + hf^n(x) + O(h^{n+1})$$

$$f(x + h) = f(x) + (f(x) - f(x - h)) + O(h^2)$$

$$v^{n,p} = v^{n-1} + (v^{n-1} - v^{n-2}) + O(h^2)$$

$$v^{n,p} = 2v^{n-1} - v^{n-2} + O(h^2) \tag{2-12}$$

The combination of the Crank-Nicolson scheme in the Eq. (2-8 and the high order prediction of the term $v^{n,p}$ at Eq. (2-12 restores the second accuracy order. This technique will be employed in Chapter 6 to analyze the effect exerted by the order on the transient.

Picard iteration (fixed point method)

Another technique to solve the non-linear term is the introduction of a nested loop to iterate using a fixed-point method. The extra loops iterated between the thermal-hydraulic and neutronics until the difference between the value of the power from the current iteration and the new iteration is lower than a threshold. This technique has been employed previously in other works as

Figure 2.4 FPI scheme. Both codes iterate until achieve the convergence

This method is interesting because allows to carry out calculations with large time step. FPI becomes unnecessary with small time steps (Gomez-Torres et al. 2012). The method is employed in the coupling between DYN3D and SUBCHANFLOW.

Taking the Eq. (2-8), it is possible to set up a Picard iteration with the variable v^n as an argument of the non-linear term as $v^{n,l}$, and the parameter v^n as the next point of the iterations, $v^{n,l}$. The extra loop will estimate the non-linear term at each iteration between time n to time $n+1$ iterating over the variable v . Once the convergence is achieved, it is possible to obtain the value of the non-linear term and calculated the term v^n .

The structure of the inner loop (fixed point iteration) is as follows.

$$\begin{aligned} \left[I - \frac{\Delta t}{2} L(t^n) \right] v^n &= \left[I - \frac{\Delta t}{2} L(t^{n-1}) \right] v^{n-1} + \\ &+ \frac{\Delta t}{2} [N(t^n, v^{n,l}) + N(t^{n-1}, v^{n-1})] \end{aligned} \quad (2-13)$$

Once the convergence is achieved, and the non-linear term is estimated, $N(t^n, v^{n,l})$, it is possible to introduce it in Eq. (2-13) and calculate the values of the next time iteration (not Picard iteration), v^n .

Introducing an extra loop at each iteration increases the computational requirements. It is possible to reduce the calculation time by adding an accelerated procedure as the Aitken technique:

$$\tilde{v}^{n,i} = v^{n,i-2} - \frac{(\Delta u^{n,i-2})^2}{\Delta^2 u^{n,i-2}} \quad (2-14)$$

$$\Delta^2 u^{n,i-2} = u^{n,i} - 2u^{n,i-1} + u^{n,i-2} \quad (2-15)$$

$$\Delta u^{n,i-2} = u^{n,i-1} - u^{n,i-2} \quad (2-16)$$

2.2.2 Jacobian-Free Newton-Krylov methods

The Jacobian-Free Newton-Krylov methods (JFNK) are suitable to solve the nonlinear of algebraic equations (Hales et al., n.d.) and are employed to coupled together codes to study multi-physics phenomena (Gill and Azmy 2009). These systems arise once the systems of PDEs are discretized (Knoll and Keyes 2004).

The method employs a nested loop. The external loop employs a Newton FPI to rewrite the problem into residual way. Once the residual is established, it is solved employing a krylov method by an internal iteration. This method was employed in the coupling between TRACE-PARCS (Watson and Ivanov 2012).

The neutronic thermal-hydraulic transient calculation can be solved employing the JFNK. This method is different than the OS because, in the transient calculation employing OS, the codes do not iterate. With the JFNK there is an iteration among the code at each time step.

The Newton problem is defined as:

$$F(u) = 0 \quad (2-17)$$

Eq (2-17) can be described as a Taylor series at point u^k :

$$F(u^{k+1}) = F(u^k) + (u^{k+1} - u^k)F'(u^k) + \text{higher order terms} \quad (2-18)$$

Neglecting the higher order terms of Eq. (2-17) and setting the term at $F(u^{k+1})$ to 0:

$$-F(u^k) = (u^{k+1} - u^k)F'(u^k) \quad (2-19)$$

Changing F' to the Jacobian the Eq. (2-19) changes to:

$$-F(u^k) = \delta u J(u^k) \quad (2-20)$$

Eq. (2-20) can be employed to solve the coupled system represented by the Eq. (2-21), obtaining the Eq. (2-22):

$$\frac{\partial v}{\partial t} - f(v, t) = \text{res}(u) = 0 \quad (2-21)$$

$$\begin{cases} J^k \delta u^k = -res(u^k) \\ u^{k+1} = u^k + \delta u^k \end{cases} \quad (2-22)$$

Eq.(2-22 is the result of applying the Newton method to the problem Eq. (2-17. Then, a Krylov method is employed to solve the Eq. (2-20. The Ep. (2-22 obtains the value of the field at the next iteration and onset another iteration.

To summarize the JFNK method:

1. The PDEs are rewritten as a residual equation.
2. Newton method scheme is employed to solve the residual and obtain the next time.
3. The equation obtained once the Newton method is applied is solved employing the Krylov method.
4. The different physics involved in the problem are solved simultaneously, instead of splitting the problem

Figure 2.5 Diagram of JFNK method

In order to apply the JFNK method it is necessary to rewrite the algorithm resolving each part of the coupled physical problem to obtain a single set of equations to be solved simultaneously. This task involves significant modifications in the codes and is not convenient for codes that are developed by different organizations.

2.2.3 Approximated Block Newton Methods

The JFNK method solves the coupled problem simultaneously and requires implementing many changes in the codes structure before it is employed. This methods require forming and keeping track of the residuals of all the variables of the coupling simultaneously (Yeckel, Lun, and Derby 2009) .

The Approximated Block Newton Methods try to implement a coupling algorithm keeping the main idea of the JFNK method but reducing the modification that are required to implement in each code.

The Approximates Block Newton Methods (ABN) are based in a different approach. The physics implicated in the problem can be solved in a modular way keeping the solver as a black box, like the OS method.

To describe the ABN method, to solvers, F and G would be coupled. The two letters represent the two solvers. Both of them have been developed isolation.

The method is described employing two solvers, F and G. As in the OS method, the two physics are coupled, and the output of one is the input of the other. But with the ABM, the iteration.

The method of the ABN method is based on Eq. (2-23). The equation represents two codes coupled together. As with the OS, the output from one equation is the input for the other (Mylonakis, Varvayanni, and Catsaros 2017). The Eq.(2-23 includes subscripts. These subscripts display the internal iteration of the inner loop to converge both variables.

$$\begin{aligned} x_{k+1}^{n+1} &= F(x_k^{n+1}, y_k^n) \quad k = 1, 2, \dots \\ y_{k+1}^{n+1} &= G(x_k^{n+1}, y_k^{n+1}) \end{aligned} \tag{2-23}$$

The two Eq. (2-23) are rewritten into residual way, Eq. (2-24):

$$\begin{cases} f(x, y) = x - F(x, y) = 0 \\ g(x, y) = y - G(x, y) = 0 \end{cases} \quad (2-24)$$

The root of the problem is finding the value of x and y . The eq 2.20 is solved employing the Newton Method. And a Block Newton Method is created from Eq.1.20:

$$\begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} f \\ g \end{pmatrix} \quad (2-25)$$

The Eq. (2-25) represents the coupled problem. In the ABN methods, the codes are performing like a black box and it is not possible to obtain the Jacobian of the codes. The Jacobian can be eliminated from the equations using the Jacobian inverse of the subproblem (Yeckel, Lun, and Derby 2009)

$$\begin{pmatrix} \frac{\partial f^{-1}}{\partial x} & 0 \\ 0 & \frac{\partial f^{-1}}{\partial y} \end{pmatrix} \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} \frac{\partial f^{-1}}{\partial x} & 0 \\ 0 & \frac{\partial f^{-1}}{\partial y} \end{pmatrix} \begin{pmatrix} f \\ g \end{pmatrix} \quad (2-26)$$

Employing the Gauss elimination method, the Eq. (2-26) can be represented as:

$$\begin{pmatrix} I & C \\ 0 & S \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} q \\ r \end{pmatrix} \quad (2-27)$$

Where the terms of the matrix, Eq. (2-27) are:

- $C = \left(\frac{\partial f}{\partial x}\right)^{-1} \frac{\partial f}{\partial y}$
- $S = I - \left(\frac{\partial g}{\partial y}\right)^{-1} \left(\frac{\partial g}{\partial x}\right) C$
- $q = \left(\frac{\partial f}{\partial x}\right) f$
- $r = \left(\frac{\partial g}{\partial y}\right)^{-1} g - \left(\frac{\partial g}{\partial y}\right)^{-1} \frac{\partial g}{\partial x} q$

ABN method allows solving each one of the physics separately as it was done with the OS technique. Eq. (2-27) allows to obtain the values for Δy and it can be introduced in the system to obtain the values of Δx .

The main difficulty regarding the application of an ABN method is the obtention of the partial derivatives included in the Eq.(2-23). The solvers are addressed as an isolated black box and so this makes difficult the task of obtaining these partial derivatives. This problem can be removed by applying two methods:

1. Employing the Taylor expansion to define the partial derivative terms of the Eq. (2-23)
2. Solve the equation $S\Delta y = -r$ with an iterative approach (Newton-Krylov)

Employing the Taylor expansion method, the partial derivatives term can be represented as:

$$C = \left(\frac{\partial f}{\partial x}\right)^{-1} \frac{\partial f}{\partial y} \approx \frac{1}{\epsilon} \left(\frac{\partial f}{\partial x}\right)^{-1} [f(x, y + \epsilon) - f(x, y)] \quad (2-28)$$

$$S = I - \left(\frac{\partial g}{\partial y}\right)^{-1} \left(\frac{\partial g}{\partial x}\right) C \approx I - \frac{1}{\epsilon} \left(\frac{\partial g}{\partial y}\right)^{-1} [g(x, y - \epsilon C) - g(x, y)] \quad (2-29)$$

$$r = \left(\frac{\partial g}{\partial y}\right)^{-1} \left(g - \frac{\partial g}{\partial x} q\right) \approx \left(\frac{\partial g}{\partial y}\right)^{-1} g(x + q, y) \quad (2-30)$$

The system can be then expressed with the same pair of coupled equations, similar to the OS method (section 2.2.1).

$$C = \left(\frac{\partial f}{\partial x}\right)^{-1} \frac{\partial f}{\partial y} \approx \frac{1}{\epsilon} \left(\frac{\partial f}{\partial x}\right)^{-1} [f(x, y + \epsilon) - f(x, y)] \quad (2-31)$$

$$S = I - \left(\frac{\partial g}{\partial y}\right)^{-1} \left(\frac{\partial g}{\partial x}\right) C \approx I - \frac{1}{\epsilon} \left(\frac{\partial g}{\partial y}\right)^{-1} [g(x, y - \epsilon C) - g(x, y)] \quad (2-32)$$

$$r = \left(\frac{\partial g}{\partial y}\right)^{-1} \left(g - \frac{\partial g}{\partial x} q\right) \approx \left(\frac{\partial g}{\partial y}\right)^{-1} g(x + q, y) \quad (2-33)$$

The method can be implemented as an OS but, the equation, $S\Delta y = r$ must be (Yeckel, Lun, and Derby 2009) solved employing an iterative method. In the case of the coupling between COBAYA4 and COBRA-TF, it should require implementing a solver in the coupling to calculate the value of y^k . To implement this coupling method, significant modifications would be required in the coupling script.

2.3 Internal and external coupling

An important aspect that must be addressed when codes are coupled for a multi-physics application is, if they will be coupled by an internal or external way.

Internal coupling does not employ extra codes to store and exchange the information between codes. The external coupling does not require significant modifications in the codes. The information is stored within meshes and an auxiliary code, a part of the coupled, is in charge to interpolate the information among the meshes.

In the case of an internal coupling, internal subroutines have to be created to transfer the information. In this case the codes are usually compiled together as one of them is part of the other.

An example is the neutronic thermal-hydraulic coupling in the core analysis. Neutronic calculates the power and the thermal-hydraulic the temperatures, densities, and boron concentrations

If the neutronic code is coupled with more than one thermal-hydraulic code, each one of the couplings requires different subroutines to develop a coupling.

Figure 2.6 Internal Coupling

As can be seen in Figure 2.6, the information is transferred from thermal-hydraulic to the neutronic and vice versa directly, without any intermediate program. Each code includes subroutines that have been created to reshape the information according to the requirements of the code that reads the feedback.

The advantage of the internal coupling:

- Internal couplings are faster than the external coupling because the transfer information does not require any intermediated structures.

The disadvantage of the internal coupling:

- Internal coupling is tight coupling. It is necessary to develop new subroutines for the new thermal-hydraulic code to change the thermal-hydraulic code in a neutronic/thermal-hydraulic coupling.

VIENE DE MCNP6/CTF →

The advantages of the external coupling against the internal coupling are:

- The coupling is carried out by a Python Script. The script contains all the ICOCO functions from both codes. To switch one thermal-hydraulic code by another one integrates into NURESIM platform, it is enough to change the ICOCO functions from the scripts.
- The NK mesh and the TH meshes could have a different precision level. The external coupling interpolates the values of the fields from one mesh to another when they have different accuracy level.

The advantages of the internal coupling about the external coupling are:

- The internal coupling does not create meshes to store the fields; it uses different subroutines developed ah-doc to the codes. The codes are compiled at the same time and creating an executable and libraries. This type of coupling runs faster than the external coupling.

← VIENE DE MCNP6/CTF

The external coupling among different codes is performed employing an extra program that reads the fields and reshapes it adapting to the requirement of the other code. In this case, no extra subroutines are necessary. The data exchange is performed by other extra code. An example of this kind of coupling is the NURESIM platform that will be explained follow. Codes integrate into NURESIM platform are coupled through SALOME. Each code creates one or more meshes to store the field to exchange in the coupling, and MedCoupling library (library including SALOME) interpolate the fields from one mesh to another allowing the perform spatial coupling among meshes with different accuracy level.

Figure 2.7 External coupling

The advantages of the external coupling are:

- The codes, neutronic or thermal-hydraulic, need many fewer modifications than the internal coupling
- Simplify the update of the code because it does not need extra subroutines.

Disadvantages:

- External coupling is slower than the internal coupling.

The two types of coupling can be applied to employ different coupling schemes, OS, JFNK, etc. All the coupling schemes present their advantage and disadvantage and will explain in sections section 2.2

2.4 Multiphysics platforms and code systems

The temporal and spatial coupling can be implemented between two codes in an isolated way or can be implemented through a computing platform.

Multi-physics platforms are those computing platforms where codes solving different physical fields or with different type of approximations to solve the same equations have been integrated. With the help of the computing platform the different codes can be coupled together to solve the system to be analyzed. Therefore, the type of coupling can be defined as external because the computing platform is in charge of building meshes, interpolate information and visualize results. Usually, in this type of platforms the temporal coupling cannot be totally implicit as the codes cannot be solved at the same time.

In the case of codes systems, the coupling is generally internal, and the codes are highly coupled being possible to use temporal coupling methods that solve the equations with higher implicitness. The disadvantages of this type of systems is that the codes must be modified before to be integrated into the platform.

Each one of these codes has been developed to analyze different physics in which the problem can be splitter. The most significant multi-physics platform or systems of codes that nowadays has been developed are summarized next.

2.4.1 NURESIM

NURESIM is the European Platform for Nuclear Reactor Safety Simulation that has been developed through three successive Collaborative Projects among different European laboratories, universities, and companies. The first one, NURESIM, established a first prototype of the multi-physics simulation environment based on the SALOME open-source computational platform (Salome Platform 2015) and integrating some codes.

The second project, NURISP, was the consolidation of the platform structure and the extension of its capabilities towards high-resolution calculations. Finally, the project where this work is immersed, NURESIM, was conceived for validation of the NURESIM platform delivering industry-like applications and establishing the platform as a reference European tool (Chanaron et al. 2015).

NURESIM platform integrates the codes for simulation of different aspects of the nuclear reactor providing the following capabilities (Kliem et al. 2017):

- Accurate representation of physical phenomena in reactor and core physics, two-phase flow thermal hydraulics and fuel modeling
- Multi-scale and multi-physics capabilities of reactor safety computations

- Easy coupling of different codes and solvers by using common data structures and generic functions
- Generic pre-processing and post-processing and supervision functions
- Deterministic and statistical sensitivity and uncertainty analyses

The codes are integrated into a virtual environment generated by the SALOME open-source software.

Figure 2.8 NURESIM platform with all the codes integrate into the platform

Figure 2.8 shows all the codes integrated into SALOME platform and coupling possibilities to simulate the different physical phenomena. All the codes except TRIPOLI4 and CRONOS2 are integrated into NURESIM platform (Chanaron 2017).

In this platform, APOLLO2 generates the cross sections library for the neutronic core simulators. The core simulators integrated into NURESIM platform are:

- COBAYA4 (UPM)
- CRONOS2 (CEA)
- DYN3D (HZDR)

As the codes have been integrated following a standard approach known as ICoCo (see section 3.1.4), it is generally possible to substitute one by the other to carry out the same simulation.

To prove the capability of the platform to perform coupled calculations, several “situation targets” were proposed to be analyzed by participants of the project using different codes or coupling strategies. UPM group participated in simulation of a Main Steam Line Break (MSLB) in a PWR and in a VVER, both proposed in the sub-project 1 “Multiphysics applications involving core physics”.

2.4.2 VERA-CS

The Consortium for Advanced Simulation of Light Water Reactor (CASL) is a project sponsored by the Department of Energy of the United States of America established in 2010 with the goal to provide a multi-physics modeling and simulations capabilities to enhance the fuel management, safety, nuclear energy’s economic improvements and the operation of the nuclear power plants.

CASL is formed by different universities and national laboratories. The goal of the project is to integrate different codes into a multi-physics platform, to solve different physics in the nuclear reactor in a coupled way. The codes integrated into the platform, once they work in a coupled way, predict the performance of the reactor and enhance the modeling and simulation of the nuclear reactor (Turinsky and Kothe 2016).

The platform developed by CASL to carry out a coupled calculation is VERA, i.e, the Virtual Environment for Reactor Applications. VERA is integrated by different codes and provides a pin-resolved, quasi-steady-state prediction of the state of the reactor (Turner et al. 2016).

The physics components integrated into VERA are:

- Neutronics
- Thermal-hydraulics
- Fuel rod heat transfer and mechanics
- Chemical, mechanical and geometric effects

2.4.2.1 Neutronic codes

Vera includes three neutronic solvers, Insilico, MPACT, and Shift. Insilico and MPACT provide deterministic pin-homogenized and pin-resolved transport capabilities respectively, using multi-group discretization. The shift is a continuous-energy Monte Carlo code that solved the problem without energy and geometry discretization.

The neutronic codes supply the power distribution, isotopic depletion, and fission products creation as a function of space and time. This variable is the input for the thermal-mechanic and thermal-hydraulic codes and the calculation of the cross-section.

2.4.2.2 Thermal-hydraulic codes

Thermal-hydraulic codes supply the information about the coolant temperature and coolant density used by the neutronic codes to update the value of the cross-section. The calculation of the temperatures and densities is performed traditionally by sub-channel codes. These codes simplify momentum equations for cross-flow between channels and a variety of models for two-phase fluid flow. The thermal-hydraulic code selected by CASL is CTF, code derived from the original COBRA-TF code and that its been further developed by North Carolina State University and Oak Ridge National Lab.

In addition, three-dimensional CFD codes have been developed under the support of CASL, Hydra-TH and Drekar, but also commercial STAR-CMM+ code has been utilized. CASL applications require both, sub-channel and CFD approach.

Thermal-hydraulic behavior of the coolant depends on the thermomechanical behavior of the fuel rods. Thermal-hydraulic and thermomechanical codes work in a coupled way to perform the coolant temperature and density and the fuel temperature.

Although the CFD codes were initially proposed as the thermal-hydraulic module in the platform, finally CTF was chosen for this function. Although CFD method must be used for some CASL Challenge Problems it is not yet feasible to use them for core-wide analysis.

2.4.2.3 Thermo-mechanic code

The fuel composition and the geometrical properties change along the fuel cycle. The fission products change the thermal properties of the ceramic material and getting worse the thermal conductivity. Some of these fission products are gaseous and diffuse out of the pellet and into the gap, changing the pressure and composition which can reduce the conductivity. Another effect is the ballooning of the pellet. It can take to a phenomenon called pellet-cladding interaction. The pellet and the clad are in touch and increase the thermal conductivity in this area.

All these phenomena change the fuel temperature and the cross-section of the neutronic calculations.

The thermomechanical calculations are performed in CASL by two codes depending on the fidelity level fuel performance, BISON, and COBRA-TF.

BISON is a parallel, finite-element fuel performance code that includes all of the physics of a traditional fuel performance code, and is capable of executing in both 2D and 3D. BISON is used to predict high-resolution solution for fuel performance problems that required high-fidelity predictions of intra-rod physics, such a pellet-clad mechanical interaction, and pellet cladding integrity evaluations during accident conditions(Turner et al. 2016).

Thermal-hydraulic code COBRA-TF has a thermomechanical module developed to analyze the fuel temperature distribution inside the pellet. COBRA-TF obtains the fuel temperature for each one of the radial nodes of the pellet and the centerline fuel temperature extrapolating the temperature of the inner node. The temperatures performance depends on the burnup and the gadolinium.

2.4.2.4 CRUD

CRUD refers to the deposition of products on the clad surface. In areas of the cladding surface where subcooled boiled occurs, particulate dissolved into the water plate out on the clad surface. The deposition of these particles on the clad surface reduces the thermal conductivity and increase the clad surface temperature accelerating the zircaloy corrosion. This phenomenon is called CILC, Crud Induced Localized Corrosion. Another important phenomenon relative is called CIPS (Crud Induced Power Shift). In the PWR, the water includes acid boric in the coolant. The CRUD includes boron and causes anomalies in the core's power distribution and reactivity behavior. (Turinsky and Kothe 2016).

MANBA (MPO Advanced Model for Boron Analysis) simulates 3D growth of CRUD over the outside clad surface and analyze the effect describes before, the reduction of the thermal conductivity and the redistribution of the power due to the boron.

MANBA simulates the growth of CRUD in the radial direction at the clad coolant interphase of the fuel rod and analyzes the two effect describes before, the reduction of the thermal conductivity and the anomalies of the power distribution.

2.4.3 NEAMS Workbench and MOOSE Applications

NEAMS (Nuclear Energy Advanced Modeling and Simulations) is a program developed by the Department of Energy of the United States of America to model and simulate current and future nuclear energy systems. The NEAMS project is organized into three different sections: Fuel products line, reactor product lines and integration product line (B. T. Rearden et al. 2018)

The Multi-Physics Object-oriented Simulation Environment (MOOSE) is a finite element multi-physics developed by the Idaho National Laboratory. As other multi-physics platforms, MOOSE is compound by different codes in charge of different physics, neutronic, thermal-hydraulic, thermomechanical, etc. The difference between MOOSE and other platforms like for example NURESAFE is the temporal coupling method employed. NURESAFE is based on a splitting method while MOOSE uses the Jacobian Free Newton Krylov (JFNK) method in the coupling.

MOOSE, as similar platforms, was developed in a modular way, allowing to plug or unplug one solver, or develop new solvers and integrate them into the platform.

MOOSE's structure is based on a layered approach (D. Gaston et al. 2009) to support the multi-physics simulations. All its components can be classified into two groups, MultiApp and Transfer. The principal parts are MultiApp or kernels. Each one of the kernels represents each one of the physics integrated into the platform, thermal-hydraulic, solid, neutronic, etc. There is a master application to coordinate the rest of the applications. All the kernels have to be adapted to the Jacobian-free Newton-Krylov (JFNK) method used to solve all the partial differential equations simultaneously.

The Multiapps need the exchange of data. This task is carried out by the Transfer through different libraries developed to swap information between the meshes. The Transfer functions are designed to move the information between domains taking account the different accuracy level of the meshes to interpolate the values to exchange.

The equations solve for each physics applications may coexist with MOOSE and be solved simultaneously in a single large Matrix (NEA), it requires a tight coupling among all the codes integrated into the platform.

The platform includes codes to model several aspects of a nuclear reactor to model a LWR: neutron transport, fuel performance, and thermal fluids. These three components in MOOSE are:

- RattleSnake, neutron transport code
- RELAP7, thermal-hydraulics
- BISON, fuel performance

Each one of the components solves one of the physics and transfer the information to another solver through the Transfers, moving data between the MultiApps. The sequence to model an LWR is the next (D. R. Gaston et al. 2015):

1. The neutronic code RattleSnake using the previous time step cladding, fuel and coolant temperatures computes the fission rate.
2. The cladding temperature is transferred from RattleSnake to RELAP-7, it calculates the coolant temperature and gives it back to RattleSnake.
3. Coolant temperature is given back to RattleSnake to be updated.
4. RattleSnake transfers the fission ratio and the coolant temperature to BISON.
5. BISON computes the fuel and cladding temperature and gives them back to RattleSnake.

With all the temperatures RattleSnake update the fission rate and the code moves ahead in time. The new fission rate changes the temperatures and repeat again the loop.

BISON is a thermomechanical code developed using the INL MOOSE. BISON is a governing correlation consist of fully coupled partial differential equation for energy, species and momentum conservation.

BISON has been developed to calculate thermal properties, solid and gaseous fission products swelling, densification, thermal and irradiation creep, fracture via relocation or smeared as a function of the fuel temperature and the burnup.

2.4.4 Universidad Politécnica de Valencia Multi-physics Platform

Universidad Politécnica de Valencia (UPV) is developing a multi-physic platform integrating some codes widely used in the nuclear industry to perform safety analysis in a light water reactor (A. Abarca 2017; Hidalgo et al. 2019).

The codes of the platform are classified into three groups:

- Neutronic
- Thermal-hydraulic
- Thermo-mechanic

Apart from the codes integrated, there are other codes that participate in the calculations, but they are not coupled. They are employed by the multi-physics platform to perform calculations that need to be done one time in the calculation, as the generation of the cross-sections.

2.4.4.1 Thermal-Hydraulic codes

Two thermal-hydraulic codes have been selected to be part of the platform, TRACE, and COBRA-TF.

TRACE (TRAC/RELAP Advanced Computational Engine- formerly called TRAC-M) is a thermal-hydraulic code designed to perform best-estimated analysis of loss of coolant accident, or other optional transient scenarios in a Light Water Reactor, PWR and BWR.

TRACE model includes:

- Two Phases of flow
- Non-equilibrium thermodynamics
- Reactor kinetics
- Reflood
- Level tracking

The other thermal-hydraulic code is COBRA-TF. Details of the code are included in section 2.5.2.

2.4.4.2 Neutronic code

The neutronic code chosen to be part of the platform is 3D-PARCS. PARCS (Purdue Advanced Reactor Core Simulator) is a multigroup neutron diffusion code able to compute steady-state and transient calculations at Cartesian or Hexagonal geometry. Moreover, PARCS is capable to compute the SP3 and power reconstruction to obtain the power distribution at pin level. PARCS is coupled through an internal coupling to TRACE(TRACE/PARCS).

2.4.4.3 Thermo-mechanic codes

The thermo-mechanic codes of the platform are FRAPCON and FRAPTRAN. These codes perform thermo-mechanic analyses of the fuel in a steady state and transient calculations.

2.4.4.4 Coupling

The UPV multi-physics platform is compound by several codes couple by themselves.:

- TRACE with COBRA-TF (TH to TH)
- COBRA-TF and FRAPTRAN
- COBRA-TF and PARCS

The coupling developed are semi-implicit, in time and in space. The goal of the coupling between TRACE and COBRA-TF is to replace the thermal-hydraulic core solver of TRACE by COBRA-TF and performing a more detail calculations.

TRACE is the master code and controls the execution, while COBRA-TF solves the thermal-hydraulic problem in the core. TRACE call to the COBRA-TF solver to update the core pressure.

Once COBRA-TF converges, the solution of COBRA-TF is transferred from one model to another. COBRA-TF core model is more accurate than TRACE so the variables from COBRA-TF will be averaged according to the domains in the TRACE model

COBRA-TF is coupled to with the neutronic code PARCS. The coupled developed was an internal coupling. The coupling scheme is an operator splitting with a staggered time mesh. Once the code converges, it exchanges the necessary information to the other code to carry out new calculations.

COBRA-TF read the power distribution from PARCS and PARCS reads the moderator temperature and density and the boron concentration. COBRA-TF calculates the Doppler temperature too, but the value employed by PARCS is the value of the Doppler temperature calculated by FRAPTRAN-FRAPCON.

The other coupling developed is between COBRA-TF and FRAPTAN. As COBRA-TF/PARC the coupling between COBRA-TF and FRAPTRAN is an internal coupling with a staggered time mesh. The time meshes of FRAPTRAN and PARCS match in time.

The platform is able to perform several transient calculations as a rod ejection or a turbine trip reactor in a PWR or BWR reactor.

2.4.5 Aztlan platform

Aztlan Platform is a Mexican national project lead by the National Institute for Nuclear Research together the Instituto Politécnico Nacional, Universidad Nacional Autónoma de México and the Universidad Nacional Autónoma Metropolitana.

The goal of the project is to integrate neutronic, thermal-hydraulic and thermomechanic codes developed by different Mexican institution integrated into a common platform like SALOME in the NURESIM platform(Gómez Torres, A.M. et al. 2015).

The neutronic modules under development are:

- AZTRAN, neutron transport code to solve the multi-group time independent discrete ordinates neutron transport equation
- AZKIND, 3D diffusion code to solve the time dependent neutron diffusion equation in cartesian geometry
- AZNHEX, 3D diffusion code to solve the time dependent neutron diffusion equation in hexagonal geometry

The thermal-hydraulic module is AZTHECA and the thermomechanic module is FUELHT. Both modules calculates the moderator temperature and density, and the fuel temperature respectively.

The AZTLAN coupling employs an external coupling with an explicit scheme.

2.5 Codes used and coupled

2.5.1 COBAYA4

COBAYA4 is the last version of the neutron diffusion code developed by the Universidad Politécnica de Madrid. This version was modified to be fully integrated into NURESIM Platform.

COBAYA4 is able to perform steady state and transient calculations employing different accuracy level in the geometrical mesh for cartesian and hexagonal geometry (Lozano et al. 2010; Herrero 2012; Ochoa, Herrero, and Garcia-Herranz 2011). The calculation can be nodal, Figure 2.9, or pin-by-pin (PbP), Figure 2.10. In the nodal calculations, each fuel assembly is modeled as one or four nodes. In the PbP calculations, each node is the fuel rod and the part of the sub-channel that surrounds the rod.

Figure 2.9 Homogenization of the fuel assembly from pin-by-pin to one node per fuel assembly

Figure 2.10 Homogenization at pin level. The radial node includes the fuel rod and a section of the subchannel around

COBAYA solves the neutron diffusion equation including the precursor delayed neutron balance. These equations considering g energy groups are as follows.

$$\begin{aligned}
 & \frac{1}{v_g} \frac{\partial \phi_g}{\partial t} + \nabla(-D \nabla \phi_g) + \Sigma_a^g \phi_g + \sum_{\forall g' \neq g} \Sigma_s^{g \rightarrow g'} \phi_{g'} = \\
 & = \sum_{\forall g' \neq g} \Sigma_s^{g' \rightarrow g} \phi_{g'} + \frac{(1-\beta)\chi_p^g}{k} \sum_{\forall g'} v \Sigma_f^{g'} \phi_{g'} + \chi_d^g \sum_{\forall n} \lambda_n C_n + S_g ; g = 1 \dots G
 \end{aligned} \tag{2-34}$$

$$\frac{\partial C_n}{\partial t} = \beta_n \frac{1}{k} \sum_{\forall g'} v \Sigma_f^{g'} \phi_{g'} - \lambda_n C_n ; n = 1 \dots D \tag{2-35}$$

The equation solved by COBAYA4 for transient and criticality calculations are not the same. The temporal terms are removed from the transient diffusion equations becoming:

$$\nabla(-D \nabla \phi_g) + \Sigma_a^g \phi_g + \sum_{\forall g' \neq g} \Sigma_s^{g \rightarrow g'} \phi_{g'} =$$

$$= \sum_{\forall g'=g} \Sigma^{g' \rightarrow g} \phi^{g'} + \frac{\chi^g}{k} \sum_{\forall g'} \nu \Sigma_f^{g'} \phi^{g'} + S_g; g = 1 \dots G \quad (2-36)$$

COBAYA is able to solve the neutron diffusion equation employing two different models for the currents according to the meshes, coarse or fine radial mesh. These two models are Analytic Coarse Mesh Finite Differences (ACMFD) for the nodal analysis and Fine Mesh Finite Differences (FMFD) for the PbP calculations. These two models establish the relationship between the neutron flux in the center of the adjacent nodes. Thus, the diffusion equation is a function of the neutron flux instead of the neutron flux and the currents.

The steady state neutron diffusion equation in Cartesian geometry in the ACMFD method integrated into each volume is define as:

$$\int_V \nabla |J_g(\vec{r})\rangle dV + \int_V \mathbf{DA} |\phi_g(\vec{r})\rangle dV + \int_V |S_g(\vec{r})\rangle dS \quad (2-37)$$

$$\int_V |J_g(\vec{r})\rangle dS + \int_V \mathbf{DA} |\phi_g(\vec{r})\rangle dV + \int_V |S_g(\vec{r})\rangle dS \quad (2-38)$$

The control volume is a fuel assembly or a quarter of a fuel assembly.

The average flux in the node and the average external source:

$$\frac{1}{V} \sum_{l=6}^6 |J_g(\vec{r})\rangle \cdot S_l + \mathbf{DA} |\bar{\phi}(\vec{r})\rangle = |\bar{S}_g\rangle \quad (2-39)$$

The values of the current flux and the external fluxes are represented as a vector with the dimension of the number of energy groups. D is a diagonal matrix. The elements of the diagonal are the diffusion coefficient of each energy group. A is the diffusion matrix in multigroup (Lozano 2009). The diffusion matrix includes the absorption, fission, and scattering:

$$A = \begin{bmatrix} \frac{\Sigma_{a1} + \sum_j \Sigma_{s,1 \rightarrow j} - \frac{\chi_1 v \Sigma_{f1}}{k_{eff}}}{D_1} & \dots & \frac{-\Sigma_{s,j \rightarrow 1} - \frac{\chi_1 v \Sigma_{f1}}{k_{eff}}}{D_i} & \dots & \dots \\ \dots & \ddots & \dots & \dots & \dots \\ \frac{-\Sigma_{s,1 \rightarrow i} - \frac{\chi_i v \Sigma_{f1}}{k_{eff}}}{D_i} & \dots & \frac{\Sigma_{ai} + \sum_j \Sigma_{s,i \rightarrow j} - \frac{\chi_i v \Sigma_{fi}}{k_{eff}}}{D_i} & \dots & \dots \\ \dots & \dots & \dots & \ddots & \dots \\ \dots & \dots & \dots & \dots & \ddots \end{bmatrix} \quad (2-40)$$

Each node is coupled with the nodes around it with the leaks term. To solve the diffusion equation, extra conditions are required to link the current in the interfaces, $|J_g^l\rangle$, the average neutron flux in the interface, $|\phi_g^{sl}\rangle$, and the average neutron flux in the node $|\phi_g^{sl}\rangle$.

The extra conditions employed by COBAYA4 to solve the coupling terms in a nodal analysis is the Analytic Coarse Mesh Finite Differences (ACMFD) (Lozano 2009).

The correlation between the neutron current, the neutron fluxes between the adjacent nodes of the ACMFD is:

$$|J_g^{i+1/2}\rangle = (A_i^j D_i^{-1} + A_{i+1}^j D_{i+1}^{-1})^{-1} \frac{A_i^f |\overline{\phi_g^i}\rangle - A_{i+1}^f |\overline{\phi_g^{i+1}}\rangle}{H/2} \quad (2-41)$$

The neutronic solver at pin level of COBAYA4 is based in the Fine Mesh Finite Differences (FMFD). This solver allows for solving full core problems at fuel rod level (Herrero 2012). The cross sections employed by COBAYA4 are calculated at pin level. Each node has different material. In Chapter 5, deeper analyses will be performed about the PbP calculation.

Figure 2.11 Current and fluxes used in COBAYA4 equations

The relation between the neutron fluxes of two adjacent nodes and the neutron current between is:

$$J = -D_n^g \frac{\partial \phi_n^g}{\partial x} \Big| = D_n^g \frac{\partial \phi_n^g}{\partial x} \Big| \approx \alpha_n^g (f_{nm}^g \phi_n^g - g_{mn}^g \phi_m^g) \quad (2-42)$$

$$\alpha_n^g = \frac{D_n^g d \phi_n^g + D_m^g d \phi_m^g}{D_n^g d \phi_n^g / f_{m \rightarrow n}^g + D_m^g d \phi_m^g / f_{m \rightarrow n}^g} \quad (2-43)$$

The factor f is defined as the ration between the neutron heterogeneous and homogeneous flux in the interface:

$$f = \frac{\phi_s^{het}}{\phi_s^{hom}} \quad (2-44)$$

With these two discretizations, it is possible to solve the diffusion equation, in a nodal and PbP calculation, employing as an only variable the neutron flux in the node.

2.5.2 COBRA-TF

COBRA-TF is a two-phase code for thermal-hydraulics core analysis. It solves the conservation equations using the model of two-fluids and three fields where these fields are vapor, liquid and entrained liquid droplets (Avramova and Salko 2016a). The version used in this work is the one coming from the CASL project (R. Salko, S Palmtag, and B.

Collins 2016) and integrated by GRS in the NURESIM Platform at the beginning of NURESAFE project. The code is able to solve both steady state and transient problems.

The three fields are modeled by a set of nine equations including an equation for incondensable gases. The liquid and droplet fields are in thermal equilibrium and they are modeled with the same energy equation. The equations are defined in a cartesian geometry (Avramova and Salko 2016b).

The mathematical solver employed is called SIMPLE (Semi-implicit Method for Pressure-Linked Equations).

The equations solved are as follows (Avramova 2007):

- 4 mass conservative equations:

$$\frac{\partial}{\partial t} \alpha_v \rho_v + \nabla(\alpha_v \rho_v \vec{U}_v) = \Gamma_v'' + \nabla \vec{G}_v \text{ vapor} \quad (2-45)$$

$$\frac{\partial}{\partial t} \alpha_l \rho_l + \nabla(\alpha_l \rho_l \vec{U}_l) = -\Gamma_l'' - S'' + \nabla \vec{G}_l \text{ liquid} \quad (2-46)$$

$$\frac{\partial}{\partial t} \alpha_e \rho_l + \nabla(\alpha_e \rho_l \vec{U}_e) = -\Gamma_e'' + S'' \text{ entrained liquid} \quad (2-47)$$

$$\frac{\partial}{\partial t} \alpha_g \rho_g + \nabla(\alpha_g \rho_g \vec{U}_g) = \Gamma_g'' + \nabla \vec{G}_g \quad (2-48)$$

- 2 energy conservative equation:

$$\frac{\partial}{\partial t} (\alpha_v \rho_{vg} h_{vg}) + \nabla(\alpha_v \rho_{vg} h_{vg} \vec{U}_v) = \Gamma_v''' h_{vg} + q_{iv} + Q_{vg}''' - \nabla''(\alpha_v q_{vg}^T) \quad (2-49)$$

$$\frac{\partial}{\partial t} ((\alpha_v + \alpha_e) \rho_l h_l) + \nabla(\alpha_l \rho_l h_l \vec{U}_l) = \Gamma_l''' h_l + q_{il} + Q_l''' - \nabla''(\alpha_v q_l^T) \quad (2-50)$$

- 3 momentum equations (solved for each direction):

$$\begin{aligned} & \frac{\partial}{\partial t} (\alpha_v \rho_{vg} \vec{U}_v) + \nabla(\alpha_v \rho_{vg} \vec{U}_v \vec{U}_v) = \\ & = -\alpha_v \nabla P + \alpha_v \rho_{vg} \vec{g} - \vec{\tau}_{wv}'' - \vec{\tau}_{lw}'' - \vec{\tau}_{lev}'' + (\Gamma_v''' \vec{U}) + \nabla(\alpha_v T_{vg}^T) \text{ vapor} \end{aligned} \quad (2-51)$$

$$\begin{aligned}
& \frac{\partial}{\partial t} (\alpha_l \rho_l \vec{U}_l) + \nabla (\alpha_l \rho_l \vec{U}_l \vec{U}_l) = \\
& = -\alpha_l \nabla P + \alpha_l \rho_l \vec{g} - \vec{\tau}_{wl}'' - \vec{\tau}_{lwl}'' - (\Gamma_l''' \vec{U}) - (S''' \vec{U}) + \nabla (\alpha_l T_l^T) \text{ liquid} \quad (2-52)
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial}{\partial t} (\alpha_e \rho_l \vec{U}_e) + \nabla (\alpha_e \rho_l \vec{U}_e \vec{U}_e) = \\
& = -\alpha_e \nabla P + \alpha_e \rho_l \vec{g} - \vec{\tau}_{we}'' - \vec{\tau}_{lve}'' - (\Gamma_e''' \vec{U}) + (S''' \vec{U}) \text{ entrained liquid} \quad (2-53)
\end{aligned}$$

The momentum cell is not the same than energy and mass cells. They are staggered in space. The phase velocities are obtained at the interfaces of the mass and energy cell, while the density, enthalpy, pressure, etc., at the center of this cell.

The equations are solved in three steps:

1. The momentum equations are solved for each cell and estimated, and the new mass flow rate is obtained
2. The velocities are calculated from the momentum equations. These velocities do not satisfy the energy and mass equations, there are some residual errors at each equation
3. The residual error is set to zero using the block Newton-Raphson method. The residual equations are linearizing respect the independent variables to employ the solver.

Once the residual error has been linearized, the system to solve for each cell is:

$$\begin{bmatrix}
\frac{\partial E_{CG}}{\partial \alpha_g} & \frac{\partial E_{CG}}{\partial \alpha_v} & \frac{\partial E_{CG}}{\partial \alpha_v h_v} & \frac{\partial E_{CG}}{\partial [(1-\alpha)_v] h_l} & \frac{\partial E_{CG}}{\partial \alpha_e} & \frac{\partial E_{CG}}{\partial P_j} & \frac{\partial E_{CG}}{\partial P_{i=1}} & \cdots & \frac{\partial E_{CG}}{\partial P_{NCON}} \\
\frac{\partial E_{CL}}{\partial \alpha_g} & \frac{\partial E_{CL}}{\partial \alpha_v} & \frac{\partial E_{CL}}{\partial \alpha_v h_v} & \frac{\partial E_{CL}}{\partial [(1-\alpha)_v] h_l} & \frac{\partial E_{CL}}{\partial \alpha_e} & \frac{\partial E_{CL}}{\partial P_j} & \frac{\partial E_{CL}}{\partial P_{i=1}} & \cdots & \frac{\partial E_{CL}}{\partial P_{NCON}} \\
\frac{\partial E_{CV}}{\partial \alpha_g} & \frac{\partial E_{CV}}{\partial \alpha_v} & \frac{\partial E_{CV}}{\partial \alpha_v h_v} & \frac{\partial E_{CV}}{\partial [(1-\alpha)_v] h_l} & \frac{\partial E_{CV}}{\partial \alpha_e} & \frac{\partial E_{CV}}{\partial P_j} & \frac{\partial E_{CV}}{\partial P_{i=1}} & \cdots & \frac{\partial E_{CV}}{\partial P_{NCON}} \\
\frac{\partial E_{CE}}{\partial \alpha_g} & \frac{\partial E_{CE}}{\partial \alpha_v} & \frac{\partial E_{CE}}{\partial \alpha_v h_v} & \frac{\partial E_{CE}}{\partial [(1-\alpha)_v] h_l} & \frac{\partial E_{CE}}{\partial \alpha_e} & \frac{\partial E_{CE}}{\partial P_j} & \frac{\partial E_{CE}}{\partial P_{i=1}} & \cdots & \frac{\partial E_{CE}}{\partial P_{NCON}} \\
\frac{\partial E_{EL}}{\partial \alpha_g} & \frac{\partial E_{EL}}{\partial \alpha_v} & \frac{\partial E_{EL}}{\partial \alpha_v h_v} & \frac{\partial E_{EL}}{\partial [(1-\alpha)_v] h_l} & \frac{\partial E_{EL}}{\partial \alpha_e} & \frac{\partial E_{EL}}{\partial P_j} & \frac{\partial E_{EL}}{\partial P_{i=1}} & \cdots & \frac{\partial E_{EL}}{\partial P_{NCON}} \\
\frac{\partial E_{EV}}{\partial \alpha_g} & \frac{\partial E_{EV}}{\partial \alpha_v} & \frac{\partial E_{EV}}{\partial \alpha_v h_v} & \frac{\partial E_{EV}}{\partial [(1-\alpha)_v] h_l} & \frac{\partial E_{EV}}{\partial \alpha_e} & \frac{\partial E_{EV}}{\partial P_j} & \frac{\partial E_{EV}}{\partial P_{i=1}} & \cdots & \frac{\partial E_{EV}}{\partial P_{NCON}}
\end{bmatrix}
\begin{Bmatrix}
\delta \alpha_g \\
\delta \alpha_v \\
\delta \delta h_v \\
\delta [(1-\alpha)_v] h_v \\
\delta \alpha_e \\
\delta P_j \\
\delta P_{i=1} \\
\vdots \\
\delta P_{NCON}
\end{Bmatrix}
= - \begin{Bmatrix}
E_{CG} \\
E_{CL} \\
E_{CV} \\
E_{CE} \\
E_{EL} \\
E_{EV}
\end{Bmatrix}
\quad (2-54)$$

Or written as an operator:

$$[R(x)]\{\partial(c)\} = -E \quad (2-55)$$

$R(x)$ is the Jacobian of the system composed of analytics derivatives of each equation with respect to the linear variation of the independent variables.

The system is reduced employing the Gaussian elimination technique to obtain the independent variables, void fraction and the pressure at the current cell and the adjacent:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} & r_{15} & r_{16} & r_{17} & \cdots & r_{1(6+NCN)} \\ 0 & r_{22} & r_{23} & r_{24} & r_{25} & r_{26} & r_{27} & \cdots & r_{(6+NCN)} \\ 0 & 0 & r_{33} & r_{34} & r_{35} & r_{36} & r_{37} & \cdots & r_{3(6+NCN)} \\ 0 & 0 & 0 & r_{44} & r_{45} & r_{46} & r_{47} & \cdots & r_{4(6+NCN)} \\ 0 & 0 & 0 & 0 & r_{55} & r_{56} & r_{57} & \cdots & r_{5(6+NCN)} \\ 0 & 0 & 0 & 0 & 0 & r_{66} & r_{67} & \cdots & r_{6(6+NCN)} \end{bmatrix} \begin{Bmatrix} \delta\alpha_g \\ \delta\alpha_v \\ \partial\delta h_v \\ \delta[(1-\alpha)_v]h_v \\ \delta\alpha_e \\ \delta P_j \\ \delta P_{i=1} \\ \vdots \\ \delta P_{NCN} \end{Bmatrix} = - \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{Bmatrix} \quad (2-56)$$

Or:

$$\delta P_j = a + \sum_{i=1}^{NCN} g_i \delta P_i \quad (2-57)$$

COBRA-TF delivers the fuel temperature profile and the inner and outer clad temperatures and uses the MATPRO-11 conductivity model to solve the thermomechanical problem (M. O. Yilmaz 2014).

To do so, it includes a thermo-mechanic solver for modeling fuel rods, electric heater tubes, and walls. The solver uses a finite-difference form of heat conduction equations which is formulated using the heat balance approach (Avramova and Salko 2016a). The solver calculates the fuel temperature in the inner rings in the fuel pellet, and the inner and outer clad temperature.

Figure 2.12 Radial nodalization of the fuel pellet

The fuel pellet is divided into several nodes or radial region. The clad is also nodalized and the inner and outer temperature are calculated (Mine O. Yilmaz, Avramova, and Andersen 2016).

The general conduction equation solved is (Avramova and Salko 2016b):

$$\frac{d}{dt} \int_V \rho C_p CT = \int_A \bar{n}_k Q_k dA + \int_V Q''' dV - \int_A Q_s dA \quad (2-58)$$

The term on the left side of the equation is the temporal term. The first term of the right side of equation stands for the energy conduction in axial, radial and azimuthal direction. The second term is the volumetric energy generated by fission or electric heating. The third term represent the convective heat transfer term at the cell fluid or vice versa.

The temperature of the different region of the fuel assembly can be estimated as follows (Toptan and Avramova 2016):

The cladding outside temperature:

$$T_{co} = T_f + \frac{q'}{h2\pi(R + t_g + T_c)} \quad (2-59)$$

The cladding inside temperature:

$$T_{ci} = T_{co} + \frac{q' \ln \left(\frac{R + t_g + t_c}{R + t_g} \right)}{2\pi k_c} \quad (2-60)$$

The fuel surface temperature:

$$T_s = T_{ci} + \frac{q'}{2\pi R h_g} \quad (2-61)$$

The fuel centerline temperature, T_{cl} in the case of radially uniform heat generation:

$$\int_{T_s}^{T_{cl}} k_f dT = \frac{f q'}{4\pi} \quad (2-62)$$

The fuel centerline temperature is calculated by COBRA-TF extrapolating the temperature of the inner radial node.

Chapter 3 SALOME platform details

As already stated in section 2.4.1, NURESIM Simulation Platform is based in the SALOME computing platform.

SALOME (Simulation numérique par Architecture Logicielle en Open source et à Methodologies d'Évolution) is a software developed by CEA, EDF and OPENCASCADE. SALOME includes different capabilities; some of them allow the coupling of the codes integrated into the platform. It is also able to show the results in different formats including 3D spatial representation as function of time.

All the codes integrated into NURESIM platform must be written into a modular way. It means that the codes integrated must be able to run some functions isolate, as:

- Initialized the steady state
- Solve the steady state
- Initialize transient calculations

SALOME is a computing platform developed by the CEA, EDF and OpenCascade (O Zerkak et al. 2007) SALOME can be employed as a pre and post process code, and as a platform to integrated and running codes in a coupled way.

The coupling developed among the codes into NURESIM platform are classified as an external coupling. The codes are not connected directly to SALOME through the “component”.

The SALOME platform is coded in C++. Most of the codes as COBAYA4 and TF are written in Fortran. The codes interfaces have to be able to operate with Fortran and C++ libraries (Périn 2016)

3.1 SALOME description

Next sections will present the most important aspects of the platform for the purpose of performing a calculation with coupled codes integrated on it.

3.1.1 Platform structure and integration of codes

SALOME comprises different layers to perform the tasks needed by the coupled codes (Figure 3.1). The lower layer is the main part of SALOME that is in charge of controlling the execution of different codes or libraries included in the platform.

Over it, the communication layer is including the subroutines that generate the meshes, store the fields (as the temperatures and the power) into the meshes and, interpolate the those fields to transfer them from one mesh to the other.

Finally, the wrapper or Application Program Interface (API), that is coded in C++ or Python must be developed particularly for each code that has to be integrated and is the connection between SALOME and the solver. When a code is integrated into the platform is called “component”. Other components different than the codes are the interpolation component INTERP_2.5D, the visualization component SUPERVISION (previously known as YACS) and many others not used in this work.

Figure 3.1 Control levels in NURESIM platform

When a code is to be integrated in the platform, it has to be reshaped lightly in order to connect the main subroutine of the code with the API built in C++ or Python. This API must call to the different high-level functions defined in the calculation process of the code. In order to do that, different API functions, called here “methods”, are defined for each code.

The number of methods defined for each code integration can vary depending on the type of integration and also on the capabilities of the code. If the code integration is soft, the API just include one method that performs all the processes included in the code. In this case the code is like a *black-box* that is called by SALOME and run within the platform but cannot be coupled with other codes in a full way. The other option is a strong integration. In this case, code functions can be called from the platform in other

to perform the different task that are part of the calculation process like initialize the code, read the input, solve steady-state, etc. This type of integration allows a full coupling of the code with others integrated in the same way.

As stated in (Crouzet 2015), the choice of the methods included in a particular component interface is up to the integrator. In first code integrations performed in NURESIM platform including first versions of COBAYA code, the definition of methods included in the API were not defined and were decided by each code developer. During NURESAFE project a standard definition of functions called ICoCo (Interface for Code Coupling) was adopted that allowed an easier code coupling scheme development.

3.1.2 Data flow and coupling schemes definition

SUPERVISION component controls the execution of the different processes that have to be performed when the code or codes are executed. This component can be used by mean of a Graphical User Interface that allows to define graphically the data flow and execution of different code functions defined when integrating the code through the wrapper. This option for using SUPERVISION was thought to be user friendly when defining different types of coupling schemes. But it becomes very complex easily as can be seen in Figure 3.2 (Spasov, Kolev, and Cuervo 2012).

Figure 3.2 Example of a SUPERVISION graph for coupling

Another option to use SUPERVISION is by mean of a Python script that defines the dataflow and coupling scheme. This is a best option when the scheme is becoming complex an also if it will be reused with little modifications.

As it is shown in Figure 3.1, a Python script controls the execution of the codes and call the function to exchange the information from one code the others. The functions in the script are the same in the wrapper. These functions initialized the codes, call the solver, check convergence, etc. Moreover, the script calls the functions to store the information into the meshes and transfer the fields from one code to another (MedCoupling)

3.1.3 MedCoupling library and fields interpolation

In order to use the different capabilities of the platform, SALOME includes several libraries to perform the different task. One of the most important modules is MedCoupling (Crouzet 2014). This MedCoupling is in charge of generating the meshes that will be used by the coupled codes and establish the interpolation strategy to be used according with instructions given by the user.

Usually, interpolation of meshes is a complex topic that has to be addressed when two codes are coupled. In the internal coupling, functions are developed to perform tight coupling among different codes. These functions exchange the information among the codes, and they have been developed to consider the different meshes that each code

employ. To switch the coupling from one code to another or even using the same codes, to change the way they are spatially coupled, requires modifying the functions developed.

The external coupling developed in NURESIM platform presents an important advantage over the internal coupling. Once the code is integrated into NURESIM platform, the information to exchange is stored in one or more meshes created by the code. If the meshes created by different codes have not the same accuracy level or spatial distribution, SALOME, through the MedCoupling library, interpolates the fields from one mesh to the other making this task transparent for the user.

Also, it is usual that this task has to be redone when the code is coupled with a new code or in a spatial way different than the one already designed. Therefore, the interpolation of meshes is one of the advantages of using this platform. It is only necessary to define spatially the meshes to be used by each code and the platform is in charge of transferring fields from one to the other and back.

The coupling developed through SALOME is an external coupling. In this kind of coupling, the codes are not connected directly to the other; there is an intermedia structure to facilitate the feedback. This structure is the MedCoupling library. The codes coupled themselves generated meshes to store the information implicated in the feedback. The meshes could have a different geometrical structure (different nodalization). MedCoupling library interpolates the information from one mesh to another replacing the subroutines developed into the internal coupling.

To switch one code by another with different mesh does not require modifications in the MedCoupling. the library is ready to interpolate the information among different meshes created by the different codes integrated into NURESIM platform. If a new code were integrated, it would require modifications in the code to generate the meshes and readapt the structure of the code to be integrated into NURESIM platform, but the MedCoupling would remain the same and would be able to interpolate the fields of the new code in the new mesh with the meshes of the other codes integrates previously.

For example, if the neutronic code analyzes a full core plus the reflector, the thermal-hydraulic code has to analyze the same configuration. With SALOME, this is not required, the neutronic code can perform calculations with the reflector and the fuel assemblies while the thermal-hydraulic code can model just the fuel assemblies. In the neutronic and thermal-hydraulic calculations, SALOME can set the values of the thermal-hydraulic in the reflector (or in other regions).

Moreover, the MedCoupling library allows to carry out calculations employing different nodalization at each physics. In the neutronic thermal-hydraulic calculations COBAYA4 and COBRA-TF, COBAYA4 could employ a PbP mesh while COBRA-TF a lumped channel mesh.

The variables to interpolated in SALOME platform are classify as P0 or P1. The variables P0 are these cell-based. The variables P1 are point based.

The P0 variables are classify into two groups, extensive and intensive:

- The extensive variables are the integrated value of the fields in the volume of the cell, for example the power, the power stored in a cell is a function of the volume of the cell.
- The intensive variables are the average value of the variable in the cell, for example the temperature or the power density, inside one cell the temperature is the same at each point.

The interpolations employed by SALOME are two types:

- Conservative: The integral of the fields must be conserved.
- Maximum principle: the target fields must remain between the upper and lower bounds of the source fields.

The interpolation is performed from a source mesh (SM) to a target mesh (TM) by cell intersection. The fields are exchanged employing an interpolation matrix:

$$\phi_{TM} = M\phi_{SM} \quad (3-1)$$

In the maximum principle method, the variable interpolated has to be in the range of the variable in the source mesh. For example, if the power is between 30 and 90 W, the power in the interpolation mesh has to be within this range. Another significant aspect is the overlapping. The source and target mesh can be or not overlapping, this means that the area (or volume or length) of the target mesh is within the area of the source mesh in the case of the overlapping or. When the two meshes are not overlapping, just a section of the target and source meshes match (Figure 3.3).

Figure 3.3 Target and source mesh not overlapping

The interpolation method, or the interpolation matrix, employ is a function of the type of variable exchanged, if the variable is extensive or intensive.

At the coupling between COBAYA4 and COBRA-TF, the interpolation method employed with the intensive variables, like the temperature or density, is the conservative method.

$$\begin{aligned}
 \int_{S_i} \phi &= \sum_{S_i \cap T_i = 0} \int_{T_i \cap S_i} \phi_s \\
 \int_{T_i} \phi &= Vol(T_i) \phi_{T_i} \\
 \sum_{S_i \cap T_i = 0} \int_{T_i \cap S_i} \phi_s &= \sum_{S_i \cap T_i = 0} Vol(T_i \cap S_j) \phi_{S_i} \\
 M &= \frac{Vol(T_i \cap S_j)}{Vol(T_i)} \tag{3-2}
 \end{aligned}$$

The same process has to be done with the extensive variables, as the power. The variable is a function of the volume of the cell, for example the power. Is the region

(area, or volume) has a power the double size of the region will have double power. The total power has to remain the same.

$$\begin{aligned}
 \int_{T_i} \phi &= \sum_{S_i \cap T_i = 0} \int_{T_i \cap S_i} \phi_s \\
 \int_{T_i} \phi &= \phi_{T_i} \\
 \sum_{S_i \cap T_i = 0} \int_{T_i \cap S_i} \phi_s &= \sum_{S_i \cap T_i = 0} \frac{\text{Vol}(T_i \cap S_i)}{\text{Vol}(S_i)} \phi_{S_i} \\
 M &= \frac{\text{Vol}(T_i \cap S_i)}{\text{Vol}(S_i)} \tag{3-3}
 \end{aligned}$$

The first interpolation method, Eq. (3-2) is employed with the intensive variables, those that are not function of the volume of the cell, while the second method Ep. (3-3) is employed with the extensive variable, those that are function of the volume of the cell.

3.1.4 ICoCo standard

As one of the objectives of the NURESIM platform is to perform coupled calculations being able to use codes with different types of models, meshes or approximations for the same physical phenomena, it is important that the most general functions that will be used by the codes for data exchange and process execution will fulfill a standard defined previously to be integrated. This standard, initially defined and used by CEA to couple different thermal-hydraulic codes, is called ICoCo (Interface for Code Coupling) (Deville and Perdu 2012).

The methods in the API (Application Program Interface) can be divided into two groups: code control and data exchange. A detailed description can be found in (Crouzet 2015).

The code control methods developed for the COBAYA4 API allow to:

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

Interface for Code Coupling (ICOCO) is one of the most significant advantages of SALOME in the coupling calculations. ICOCO holds in the external coupling the same function than the subroutines developed in the internal coupling between COBAYA3 and COBRA-TF.

3.1.5 Integration of previous versions of COBAYA

COBAYA code has been continuously updated since its first version developed in eighties as part of SEANAP system. It was initially developed to perform 2D transport corrected calculations with two energy groups in Cartesian geometry.

As can be seen in Figure 3.4, COBAYA was expanded during NURESIM project to perform 3D neutron diffusion calculations using a multigroup discretization in energy and becoming COBAYA3. At this moment, also a *black-box* integration of the code was performed using yet a tight coupling with thermal-hydraulic codes COBRA-III (Jackson and Todreas 1981) (homogenous two-phase model) and COBRA-TF. Therefore, at this moment COBAYA was not yet able to be coupled with other TH codes integrated into the platform.

Next step, NURISP project, was the integration of the code in a full way allowing at this moment the coupling with other TH codes in the platform. This first full integration and coupling was performed with CEA TH code FLICA-4 (Toumi et al. 2000). But at this moment COBAYA was not still able to perform PbP analysis within SALOME due to the fact that the mesh generator necessary to create the PbP mesh library was not yet developed. Nevertheless a coupled nodal analysis of a Boron dilution transient was performed and compared with the results of other codes integrated in the platform (G. Jimenez et al. 2015).

Historic developments of COBAYA code are included in Figure 3.4.

Figure 3.4 Evolution of COBAYA

SALOME 6.6.0 includes the capabilities to perform a post-proc analysis of the calculations. The information of the fields exchange is stored in the meshes. The meshes are in files called Medfiles. SALOME

Figure 3.5 SALOME 6.6.0 post-proc

3.2 Modifications in COBAYA integration: COBAYA4

Integration COBAYA4 into NURESIM platform using Salome 6, the version of Salome officially adopted during NURESAFE project, required significant changes in the source of the last version of COBAYA3. At this moment COBAYA3 was coupled through a tight internal coupling with two thermal-hydraulic codes, COBRA-III and COBRA-TF, and with a legacy Med Library (previous to MedCoupling library) with FLICA4.

The first step to fully integrate COBAYA into NURESIM platform was to find and isolate the subroutines of the neutronic solver from the coupling and thermal-hydraulic solver subroutines. This was necessary because the previous internal coupling performed was made with the objective to save coding and memory and not to work in a computing platform like SALOME.

Figure 3.6 Structure of the internal coupling developed between COBAYA3 and the thermal-hydraulic codes

Also, it was necessary to rewrite some of the original coupling subroutines to be smaller, simpler and faster and to be able to interact with the calls to the MedCoupling library.

Any hard wire link with the TH code has been removed allowing the TH code to have different definitions for:

- Nodalization level (assemblies vs cells).
- Methodology of the parallelization (domain decomposition vs only one domain).
- Input files (being independent from each other)
- TH or NK can be updated to a new version without modify the other code.
- TH or NK can be coupled with any other code if implement the ICoCo interface.

It was necessary:

- Remove the neutronic and thermal-hydraulic interfaces
- Different clearly the thermal-hydraulic and neutronic solvers.

Figure 3.7 MedCoupling between the neutronic and any thermal-hydraulic code integrated into NURESIM

The redefinition of the integration of COBAYA4 into NURESIM platform implicated deep modifications in the source. The code was recoded to a more modular structure and updated completely to FORTRAN90. Input reading subroutines were redesigned to improve checking of inconsistencies due to user effect and readability. Also, output

writing subroutines were redesigned in a more modular and readable way in order to allow future modification or inclusion of variables.

The main subroutine where logical control of the different steps of code calculation is established has been redesigned to be adapted to the new ICoCo standard.

New external utilities have been also developed in order to improve COBAYA4 development process and capabilities: a non-regression tool, post-processing subroutines and some processing tools for the nuclear cross section libraries.

The integration allows to change the temporal coupling model implemented between COBAYA4 and any other of the thermal-hydraulic codes integrates into NURESIM platform, as COBRA-TF, FLICA4 and SUBCHANFLOW, and carry out calculations employing each code different space meshes, axial and radial, between codes.

The coupling schemes described in section 6.3 can be employed and changed easily each other due to the strong integration carried out with COBAYA4 and COBRA-TF into NURESIM platform.

The modification implemented in COBAYA4 were:

- Rewrite the code to Fortran 90
- Rewrite the code into a modular way
- Split COBAYA3 and the thermal-hydraulic codes, COBRA-TF, COBRA3, and SUBCHANFLOW.

Once all the changes were implemented, it was possible to carry out:

- Steady state calculations, nodal and at PbP level
- Transient calculations and nodal and at PbP level

Figure 3.8 depicts the integration of COBAYA4 into NURESIM platform, and the capability to couple with each one of the three thermal-hydraulic codes integrated into NURESIM platform through SALOME.

Figure 3.8 Neutronic thermal-hydraulic coupling in NURESIM platform through SALOME

The execution of the code is controlled by a Python script. The script calls to different functions, as initialization of the code, solver to be employed, etc. A software is written into a modular way once it is possible to call all these functions in isolation and employed just the functions requires for the calculation. For example, in a steady state calculation, all the functions require to carry out transient calculations are not required, and they do not have to be included in the script.

Chapter 4 Developments in the integration and coupling of COBAYA for steady-state calculations

4.1 Introduction

Several steps had to be carried out to perform steady-state analysis using COBAYA4/COBRA-TF with in NURESIM. The initial one was the development of a Python script to control the execution of the COBAYA4 code and the management of data. The script was later modified to include the coupling with the COBRA-TF code and the transference of fields between them. Some modifications had to be performed also in COBRA-TF to enable this coupling. Finally, some non-regression, verification and validations analysis was performed to assure a good performance of the system.

4.2 Initial stand-alone execution

As already explained, in order to execute an analysis with COBAYA4 within SALOME, it is necessary to develop a Python script including the calculation scheme. The Python script oversees the execution of the codes.

The script must include, as first part, some calls to load the SALOME environment and those components of the platform that are going to be used in the analysis. Once the SALOME environment is loaded, it is possible to call any function defined in the API of the code to be used.

Figure 4.1 Diagram stand-alone calculation

To assure that the integration was performed correctly, a set of non-regression and verification cases were calculated and compared by mean of the tool that is part of the COBAYA software (Sabater 2014).

4.3 Coupling with COBRA-TF

In order to carry out coupled calculations, the Python script must be expanded to include the coupling scheme. This is one of the most important capabilities of the integration in SALOME. If the integration of the codes is strong and they fulfil the ICoCo standard, the definition of different types of coupling schemes is possible without modifications in the code itself.

As part of the work to be performed by UPM in NURESAFE project, the transient analysis of a MSLB scenario was included. Therefore, a TH code from the platform had to be selected for that task. Due to the long experience of the research group with the COBRA-TF code, this was the one selected.

The integration of COBRA-TF into NURESIM platform was in charge of the GRS (Périn 2016). This was performed also fulfilling ICoCo standard. One of the difficulties in the

integration and coupling of COBRA-TF was the necessary definition of two meshes for the exchange of data; the fluid mesh and the fuel mesh. This is due to the fact that COBRA-TF is solving not only the thermal-hydraulic problem but also the thermal-mechanic. And the spatial discretization is not the same. This is especially important when two different discretizations were implemented for COBRA-TF by GRS. One was the most common approach for core thermal-hydraulic codes; the subchannel discretization. And the other that is also used but it is not the main objective for using the code, the channel averaged approach.

COBRA-TF includes in the source some functions prepared to be employed in a coupling calculation. One of these functions generated the average fuel temperature of all the pellet. The average temperature can be employed in the coupling, but COBAYA requires the Doppler temperature. In order to read the Doppler temperature three functions were created in the file COBRA-TF_Coupling_Interface.F90:

- Get the number of radial regions in which the fuel pellet is divided
- Get the temperature of each region
- Get the Doppler temperature

These functions were created with the idea to allow flexibility on the definition of this value in the future.

Modifications relative to the effective fuel temperature where implemented in the functions in charge to transfer the fuel temperature from COBRA-TF to the fuel rod mesh. COBRA-TF includes a function to transfer as effective fuel temperature the average fuel temperature. COBAYA4 requires the Doppler fuel temperature instead the average fuel temperature.

The Doppler effective fuel temperature is generated as a weight of the fuel centerline temperature and the surface fuel temperature:

$$\begin{aligned} \text{Doppler Temperature} \\ = 0.7 \times (\text{Surface Temperature}) + 0.3 \times (\text{Centerline temperature}) \end{aligned}$$

Three functions where included in the coupling subroutine of COBRA-TF:

1. Obtain the number of radial nodes in which the fuel pellet is divides
2. Obtain the temperature of each region
3. Calculates the Doppler temperature

With these modifications it is possible to perform neutronic/thermal-hydraulic coupling calculations.

The script to carry out a steady-state calculation using COBAYA4/COBRA-TF system includes ...

The coupling algorithm used is graphed in Figure 4.2.

Figure 4.2 Diagram COBAAYA4 and COBRA-TF

4.4 Spatial coupling using MedCoupling

With the integration proposed, the exchange of information does not require extra subroutines as in a tight coupling and the Medcoupling library is in charge to interpolate the fields among the different meshes. In the case analyzed in this thesis, the interpolation is performed between the neutronic mesh of COBAYA4 and the two meshes used by COBRA-TF, the fluid mesh and the fuel mesh.

Due to the capabilities of MedCoupling is now not necessary that the meshes have exact overlapping including one, two or more cells of one mesh into one cell of the other or the way around (Figure 4.3).

Figure 4.3 Interpolation between two axial meshes with exact overlap

In Figure 4.3, two axial regions of the green mesh match with one axial region of the red mesh. The variable store in one axial region of the red mesh is the average value of the two green meshes. In the opposite direction, the variable in each one of the green axial regions within the red region are the value of the red region.

The library is able to interpolate as explained in section 3.1.3 and supply the right values from one mesh to the other in the case showed in Figure 4.4, where there is not exact overlapping. This is very convenient in the axial direction because allows flexible nodalizations in neutronics and thermal-hydraulics. But also, in the radial plane, because this enable different radial nodalization for the codes or even codes with different resolution in the radial plane.

Figure 4.4 Interpolation between two axial meshes without exact overlap

The interpolation performed in the Figure 4.3 is performed in the radial direction too. The interpolation is carried out in the nodal-lumped channel, PbP-lumped channel and the PbP-subchannel calculations. The interpolation follows the same rule in radial direction than in axial direction. The percent of the variable interpolated from one mesh to another depended of the area shared by the meshes and the type of variable.

The radial interpolation in a subchannel-PbP calculation follows the same rule. Figure 4.5 represents two of the three meshes involve in the neutronic/thermal-hydraulic calculations the variables store in the blue mesh are the fluid temperature and the fluid density. The variable store in the red mesh are the power. The power exchange from the red meshes to the blues mesh in Figure 4.5 is 25% of the power of the four red meshes.

Figure 4.5 Radial Thermal-hydraulic mesh, blue, radial neutronic mesh, red at PbP sub-channel calculation

Figure 4.6. represents the radial fluid mesh of COBRA-TF (red square) and the radial power mesh of COBAYA4 (green square). The power generated by the pin-cell is divided into four channels.

Figure 4.6 Thermal-hydraulic sub-channel and neutronic pin node

The three meshes generated for the coupling between COBAYA4 and COBRA-TF are represented in Figure 4.7. The mesh on the left is the neutronic one used by COBAYA4 in a PbP calculation. The mesh stores the power generated by each pin. The mesh on the center is the fluid mesh. This mesh stores the fluid temperature and density. Fluid mesh does not match in the PbP calculations with the power mesh. The mesh on the right is the rod mesh that stores the Doppler fuel temperature.

Figure 4.7 Neutronic mesh, fluid thermal-hydraulic mesh, and rod thermo-mechanic mesh

Between the fluid and the neutronic mesh, MedCoupling interpolates the moderator temperature and density between the

Moreover, the meshes of Figure 4.7 are the meshes of PbP subchannel calculations. But it is possible to carry out nodal lumped calculation or PbP and lumped channel calculation. Medcoupling manages the interpolation between the different meshes with different size. The effect of the combination of different meshes will be described in Chapter 5.

The thermal-hydraulic calculations require two meshes to store the values of temperatures and densities. One mesh for the fluid temperature and density, and another mesh for the effective fuel temperature. A subchannel full core calculation would require uses a lot of time; the calculation would take a few days.

To simplify the calculation, COBRA-TF is able to perform lumped channel calculations (channel calculation). The lumped channel is defined by several fuel rods and sub-channels, usually the number of fuel rods and sub-channel of one fuel assembly(Cuervo 2007).

The channel is defined by three parameters, the wetted perimeter, the heated perimeter, and the flow area. In a channel calculation, these parameters are obtained as the sum of the sub-channel value. The moderator temperature and density are the same in all the sub-channels.

The total power is divided among the total number of fuel rods, and the thermomechanic calculation is performed employing the average power. One termomechanic calculation is performed for all the fuel assembly instead of as many calculations as fuel rod.

4.5 Validation of the coupling: MCNP6/CTF system

Along NURES SAFE project COBAYA3 was updated to COBAYA4 and integrated with all its capabilities to NURESIM platform (García-Herranz et al. 2017). It was necessary to validate all the modifications implemented into COBAYA4 and the new external coupling developed inside NURESIM platform with the thermal-hydraulic code COBRA-TF. To validate COBAYA4/COBRA-TF coupling the coupled system MCNP6/CTF developed by NCSU was selected (Bennett, Avramova, and Ivanov 2016). Details of the codes and coupling are given in next sections.

4.5.1 MCNP6

MCNP6 (Goorley et al. 2013) is a MonteCarlo code developed to solve the transport equation in a precise three-dimensional geometry, and continuous energy (Bennett, Avramova, and Ivanov 2016). The geometry in MCNP6 must be described accurately, unlike COBAYA4 that does not require the geometrical description beyond that the pin-cell. MCNP6 and COBAYA4 calculate the power distribution that will be the input of the thermal-hydraulic codes.

The cross sections libraries for MCNP coupled calculations must be generated in the range of temperatures of the model for all the isotopes. NJOY is used to generate the broadened cross sections. The number of material and the range of temperatures of the calculations would require large computing resources.

The cross-section files created at specific temperatures can be replaced by expansional function that represent the cross sections as a function of the energy and the temperature. These new cross sections, the OTF (On-the-Fly), replicate the Doppler-Broadened cross sections at the temperature of the neutron (W. R. Martin et al. 2013). The OTF cross sections are created before to initiate the calculations employing the program *fit_otf*.

The same problem than the Doppler broadened must be solved with the thermal scattering. The cross sections of thermal scattering must be created for the range of temperature of the problem. The code developed to create the libraries is *makxsf*.

4.5.2 CTF

CTF is the CASL and RDFMG upgraded version of COBRA-TF. Although there are some differences between CTF and COBRA-TF (NURES SAFE version), the description in section 2.5.2 about COBRA-TF can be applied to CTF.

MCNP6 and CTF are coupled through an internal coupling (see section 2.3). In this type of coupling, the interpolation functions must be created. These functions are created in subroutines and compiled at the same time than MCNP6 and CTF. These subroutines pass the fields to exchange in the coupling directly from one code to another. The subroutines reshape the information to adapt it to the new mesh. These kinds of couplings are developed for specific codes and geometry. To change one of the codes or modify the nodalization implies to modify the subroutines.

4.6 Description of case analyzed

The case selected for validation was a fuel assembly derived from the Exercise I-3 of OECD/NEA Benchmarks for Uncertainty Analysis in Modelling (UAM) for the Design, Operation And Safety Analysis of LWRs (UAM-LWR Benchmark) (Ivanov et al. 2013)

4.6.1 Fuel assembly specifications

The case is a 15x15 fuel assembly (assembly pitch = 21.6405 cm) with three types of pin-cells: 208 UO₂ pins, 16 guide tubes, and one instrumentation tube. The dimensions of each type of fuel rod are included in Table 4.1. Geometry dimensions and compositions were the same than the benchmark except for the boron concentration. It was set to 0 ppm.

Table 4.1 Geometry description of the fuel assembly

UO ₂ pins		Guide tube		Inst. tube	
Parameter	Dimension (cm)	Parameter	Dimension (cm)	Parameter	Dimension (cm)
R _{fuel}	0.46955				
R _{gap}	0.4791	R _{int}	0.63245	R _{int}	0.56005
R _{clad}	0.5464	R _{ext}	0.67310	R _{ext}	0.62610
Pin pitch	1.4427	Pin pitch	1.4427	Pin pitch	1.4427

Figure 4.8 Radial view of the fuel assembly

4.6.2 Generation of cross sections

COBAYA4 requires multigroup cross-section libraries to run the criticality analysis. The cross-section libraries for this case were created employing SCALE.

SCALE (Standardized Computer Analysis for Licensing Evaluation) is a modeling platform developed by Oak Ridge Nuclear Laboratory (ORNL) and widely used by the Universities, laboratories, and the Companies to develop many kinds of studies (B. Rearden and Jessee 2017):

- Critically safety
- Sensitivity Uncertainty analysis
- Reactor and lattice physics
- Sensitivity and uncertainty analysis
- Radiation shielding
- Spent fuel and radioactive source term characterization

Among the capabilities of SCALE system, reactor and lattice physics modules have been employed to calculate the macroscopic cross sections at PbP. The geometrical scheme of the fuel assembly is in Figure 4.8 . The cross sections libraries are calculated just once before running the calculations. The libraries include the value of different cross sections for the material of the problem at different values of Doppler temperature, moderator temperature, moderator density, and boron concentration. Figure 4.9 shows an overview of the problem:

1. SCALE calculates the cross section libraries just once
2. COBAYA4 and COBRA-TF iterate to achieve the solution

Figure 4.9 Neutronic thermal-hydraulic iteration including first cross-sections libraries calculations

MCNP6 does not need a previous step to create the cross-section libraries to perform a calculation because it runs a criticality calculation with continuous energy cross sections. The continuous energy cross sections are updated during the calculation to fit with the temperature of the materials. Two codes are employed to create cross sections at the temperature of the materials: *fit_otf* and *makxsf* (Bennett, Avramova, and Ivanov 2016).

The function of the *fit_otf* program is to create the on-the-fly cross sections as a function of the energy and temperature for each one of the isotopes of the problem to reproduce the Doppler broadening. *Makxsf* generates thermal-scattering libraries at various temperatures (Vazquez et al. 2012).

4.6.3 Differences in the mesh of COBRA-TF

Regarding the thermal-hydraulic model, the fluid mesh used by both coupled systems is not the same. The one from CTF is rod centered while the one of COBRA-TF is channel centered (Figure 4.10)

Figure 4.10 COBRA-TF channel centered cell and CTF rod centered cell

The different fluid mesh of the two models is due to the fact that CTF mesh was modified to fully overlap neutronic mesh of MCNP6. This fact can introduce differences in the solutions. A sensitivity analysis was performed with the two models. The results do not show significant differences (Bennett, Avramova, and Ivanov 2016).

4.7 Coupling algorithms used for steady-state calculations

COBAYA4/COBRA-TF and MCNP6/CTF include different algorithms to achieve the convergence in a steady state calculation. COBAYA4/COBRA-TF initialize the steady state calculation with the neutronic solver. COBAYA4 carries out the calculation with the moderator temperature, moderator density, Doppler temperature and boron concentrations specified in the input (Figure 4.2).

MCNP6/CTF works differently and initializes the steady state calculation with a thermal-hydraulic calculation (Figure 4.11). Once COBRA-TF converges, MCNP6 read the temperature and runs a neutronic calculation. Then, MCNP6 sets the temperatures and repeats the sequence until the temperature converges. In one case, the convergence is reviewed in the neutronic variable while, in the other case using thermal-hydraulic variables.

Figure 4.11 Diagram MCNP6-COBRA-TF

It is important to keep in mind that the differences in meshes, neutronic solvers and coupling algorithms will introduce differences in the solutions.

4.8 Results and comparison

The results will be expounded in different steps. Firstly, the Cold Zero Power (CZP) standalone calculations in 2D comparing the results from COBAYA4 and MCNP6. Secondly, the HZP 3D standalone calculation and finally, the comparison of the two solutions at HFP. The objective is to separate the differences originated by the neutronic calculation from the ones coming from the coupling.

4.8.1 2D standalone calculations at CZP

These calculations were carried out to estimate the differences between MCNP and COBAYA4 caused just by the neutronic model.

The calculation was performed in 2D. In this case, the neutronic boundary condition applied to all the interfaces, axial and radial, is reflective so it reproduces an infinite lattice.

The first calculation performed with MCNP was at Cold Zero Power (CZP) where the moderator temperature and the fuel temperature are 293K.

Table 4.2 Neutronic stand-alone 2D calculations with different codes

Code	K-eff	Diff with MCNP
NEWT(56EG)	1.509704	23.4
NEWT(252EG)	1.509475	0.5
COBAYA4(PbP)	1.509705	23.5
COBAYA4(Nodal)	1.509703	23.5
MCNP	1.50947	Reference

4.8.2 3D standalone calculations at CZP

Next part is the 3D standalone calculation. In the 3D model, the axial boundary condition is void, and the radial boundary condition continues as reflective. These differences may produce impact in the results.

In Table 4.3, the value of 3D calculations from COBAYA4 and MCNP6 are shown. It can be seen that the differences between COBAYA4 and MCNP increase becoming 79 pcm compared with the 23.4 pcm obtained in the 2D analysis. The differences are due to the way the codes model the void boundary condition at the top and bottom of the assembly.

Table 4.3 Solution from the 3D stand-alone neutronic calculation, MCNP6, and COBAYA4

	COBAYA4	MCNP6
KEFF	1.50519	1.50440
Diff(pcm)		79

4.8.3 3D coupled calculation at HFP

This analysis allows to study the neutronic and thermal-hydraulic coupling and validate the new coupling developed along NURESAFE project between COBAYA4 and COBRA-TF. MCNP6/COBRA-TF will be used as a reference solution.

The variables compared are:

- Effective multiplication constant (k-eff)
- Axial and radial power distribution
- Thermal-hydraulic variables exchange through the coupling; i.e. moderator temperature and density, and effective fuel temperature

If the comparison is done just with k-eff the two solutions appear to be very different (see Table 4.4). K effective is an integral variable and it is important to compare with other variables as the radial and axial power distribution or temperatures at the rod.

When performing the work explained here, a minor change had to be implemented in the coupled system MCNP6/CTF. Initially, the fuel temperature supplied to the neutronic code in the coupling was the average fuel temperature. But in system COBAYA4/COBRA-TF the variable exchanged is not the average fuel temperature but the Doppler temperature. Therefore, some modifications had to be done to the MCNP6/CTF to be able to exchange the Doppler temperature. The large differences obtained initially with different models for the fuel temperature were reduced implementing the model for Doppler temperature in both. The results can be seen in Table 4.4.

Table 4.4 Comparison of the K-EFF between COBAYA4/COBRA-TF and MCNP6/CTF employing two different effective fuel temperatures

	COBAYA4-COBRA-TF	MCNP-CTF	MCNP-CTF-Doppler
KEFF	1.42873	1.42272	1.42529
Diff(pcm)		601	344
			257

The differences in the k-eff value are the expected when such different models are used to solve the neutronic problem if the same model is used for the fuel temperature. Other variables can be analyzed from these calculations.

One of these variables is the axial power profile. The Figure 4.12 shows the axial power distribution.

Figure 4.12 Average axial power profile from COBAYA4/COBRA-TF and MCNP6/CTF with Doppler temperature and MCNP6/CTF with average fuel temperature as the effective fuel temperature

Figure 4.12 shows the axial power profile for the three calculations, i.e. one from COBAYA4/COBRA-TF, other with MCNP6/CTF using the volume average fuel temperature as the effective fuel temperature in the feedback and the last one MCNP6/CTF using for this the Doppler temperature.

These three curves have a very similar shape. The three curves are slightly bottom peaked as it is expected in an HFP calculation with fresh fuel. The main differences can be appreciated around the node of maximum power. The highest value of F_z corresponds to COBAYA4/COBRA-TF. Value for F_z from MCNP6/CTF using the Doppler temperature is very close to the value from COBAYA4/COBRA-TF. This can be explained because the Doppler temperature is lower than the average fuel temperature and increases the power in the central region of the fuel assembly.

Maximum F_z values for each case in Figure 4.12 are:

Table 4.5 F_z from the three calculations using the value of COBAYA4/COBRA-TF as a reference solution

	Max Fz	Relative difference (%)
MCNP6/CTF average fuel temperature	1.512	1.12
MCNP6/CTF Doppler fuel temperature	1.526	0.20
COBAYA4/COBRA-TF	1.529	

When the Doppler temperature is used in the coupling, the axial power profile is more pronounced than when the coupling uses the average fuel temperature.

The axial distributions of the average thermal-hydraulics variables exchanged through the coupling are given at Figure 4.13, Figure 4.14, Figure 4.15.

As both systems can supply the values of rod temperatures in the domain, it is interesting to analyze the impact that different models can have in its values.

Figure 4.13 Axial distribution of the average fuel temperature

The evolution of the average moderator temperature and density are very similar. As the axial position changes, the moderator temperature and density diverge. These

differences are more remarkable at the top of the fuel assembly than the bottom of the fuel assembly.

The values of temperature of the fuel is very different depending on the use of the Doppler temperature or of the average fuel temperature as pellet temperature for exchange. The Doppler temperature is defined as 70% of the fuel temperature and 30% centerline temperature. The weight of the fuel surface temperature is important, and the value of the Doppler fuel temperature is nearer to the value of the fuel surface than average fuel temperature. This difference increases the reactivity when the effective fuel temperature is the Doppler temperature. It is the reason why the F_z from COBAYA4/COBRA-TF and MCNP6/CTF with the Doppler temperature is higher than the cases where the average fuel temperature is used.

Even though, the differences in the moderator temperature and density are less than 1K. So, the evolutions of the moderator temperature and density are very similar.

Figure 4.14 Axial distribution of the average fluid temperature

Figure 4.15 Average axial density distribution

Another important variable to analyze is the radial power distribution. Figure 4.16 to Figure 4.20 show the value of the radial power distribution for both codes and the difference between COBAYA4/COBRA-TF and MCNP6/CTF with different fuel temperature model. The relative error has been calculated using MCNP6 as a reference solution.

Figure 4.16 2D Radial Power distribution COBAYA4-COBRA-TF

Figure 4.17 2D radial power distribution from MCNP6-CTF employing the Doppler fuel temperature as an effective fuel temperature

Figure 4.18 2D radial power distribution from MCNP6-CTF employing the Average fuel temperature as an effective fuel temperature

Figure 4.19 Relative difference between COBAYA4/COBRA-TF and MCNP6/CTF using the average fuel temperature as effective fuel temperature, employing the solution from MCNP6 as a reference solution

Figure 4.20 Relative error between COBAYA4/COBRA-TF and MCNP6/CTF using the Doppler fuel temperature as effective fuel temperature, employing the solution from MCNP6 as a reference solution

The two radial power distributions from MCNP6/CTF, Figure 4.17 and Figure 4.18, show differences in the maximum values. The maximum power factors of the fuel rods when the coupling employed the Doppler temperature as effective fuel temperature are higher than when the effective fuel temperature is the average fuel temperature.

The relative differences between the 2 D radial power map employing in both calculations, MCNP6 and COBAYA4, the Doppler fuel temperature effective fuel temperature show minimum values, around 0.0 % Figure 4.20. The differences between COBAYA4 and MCNP6 when the effective fuel temperature of MCNP6/CTF is the average fuel temperature are more significant, Figure 4.19.

The effect of the Doppler coefficient is not so high when the radial power profile is analyzed. It just reduces the level of the reactivity, but it does not exert significant effect on the values of the radial power distribution.

Taking into account the differences in both systems of codes; i.e. the use of two different neutronic models, the radial nodalization of the thermal-hydraulic code and the coupling scheme, the values obtained for the thermal-hydraulic variables in both cases can be considered as very similar. The results obtained in both cases are very close when the same effective fuel temperature is employed. The axial distribution of the average values of the thermal-hydraulic fields and the radial power distribution are also very much alike.

Chapter 5 Spatial discretization analysis

5.1 Introduction

One of the advantages of integrating COBAYA in SALOME computing platform, as was explained in Chapter 2, is the versatility to combine different meshes with different accuracy level at each one of the physics implied into the problem. It is possible to use any type of discretization of the domain for each of the codes that are coupled so they do not need to have the same discretization or even those types of discretization where one cell on first code contains an integer value of cells of the other code domain.

SALOME is in charge of interpolating the field to exchange among the meshes so this is transparent for the user once the meshes have been defined correctly in space.

In this chapter, the goal is to analyze the effect exerted by the combination of meshes with different radial nodalization on the results of steady state and transient calculations.

The nuclear transients, as the MSLB, generates a strong gradient of neutron flux within the fuel assemblies. The use of a combination of meshes for the coupled codes with different accuracy level to simulate the accident could exerts an effect on the solution.

This is a typical approach when a system thermal-hydraulics code is used for plant analysis. Usually the NK calculation is performed at assembly level using a diffusion code and the TH one is performed lumping several assemblies in one channel to reduce the number of channels in the core.

Therefore, it is interesting to analyze the effect that the lumped approach can have in the results and this analysis can be performed with the capabilities available in the NURESIM platform due to the use of SALOME.

5.2 Fuel assembly analysis

As first step to evaluate the impact of the radial nodalization in the coupled system results, two small problems were set up. Both problems analyze a fuel assembly at Hot Full Power (HFP) but using two different boundary conditions (BC) at the neutronic variables:

- Reflective radial BC at the four interfaces
- Reflective radial BC at three interfaces and vacuum BC at one interface

The reason why the second problem is using vacuum condition in one side is to force to have a strong flux gradient within the assembly. The neutron flux increases from the face with the vacuum BC to the opposite face with reflective BC. On the other side, the reflective BC applied at the four interfaces generate flat profile for the flux inside the fuel assembly.

5.2.1 Problem specification

The fuel assembly selected to carry out the fuel assembly analysis for the test problem comes from OECD/NEA PWR MOX/UO₂ core transient benchmark (Kozłowski and Downar 2006)

As section 2.5.1 explains, COBAYA4 includes two different solvers to analyze a coarse or fine mesh problem. The impact of the different resolutions in NK and TH physics on the 3D fuel assembly steady-state calculations were analyzed using the combinations of meshes shown in Figure 5.1:

- a) Nodal and channel
- b) PbP and channel
- c) PbP and sub-channel

As previously explained, the assembly was analyzed employing two different radial neutronic boundary conditions in order to analyze the effect of having a strong flux gradient.

The combination of both, boundary conditions and different meshes allows to determine the effect of the interpolation among the different meshes and physical states.

Firstly, the calculation was performed using reflective radial boundary condition at the four interfaces. In the case of the TH solver, radial adiabatic boundary conditions were

used. At the thermal-hydraulic problem, both are solve using adiabatic channels without friction at the periphery. Axially steady-state flow conditions for inlet mass flow and enthalpy and outlet pressure were imposed extracted from the original benchmark. In this case, a very flat power distribution is obtained when using a pin-by-pin resolution.

Figure 5.1 Different meshes used for the three combinations for NK-TH resolutions

The second calculation with the fuel assembly was designed to generate a strong gradient in the radial power distribution within the assembly. Vacuum boundary conditions for NK were assigned at the south side of the assembly. Reflective conditions were still attached to the rest of the sides. TH conditions are the same than in previous case.

5.2.2 Results and conclusions

In order to analyze the results, the PbP/sub-channel results are taken as a reference solution because they are the most accurate calculations.

Considering the multiplication factor (Table 5.1), when using reflective boundary conditions, the solution for PbP/channel and PbP/sub-channel calculations are very close each other, being the difference 1 pcm. In the case of the nodal/channel, the difference with the reference solution is higher becoming 74 pcm.

When using vacuum boundary condition on the south side, the solutions of the nodal/channel and PbP/channel are quite different compared to PbP/sub-channel solution. In the case of PbP/channel, this difference becomes 630 pcm. The difference increases in nodal/channel calculation are reaching 675 pcm.

If an analysis of $F\Delta H$ is carried out (Table 5.2), the same tendency than in multiplication factor can be observed. In this case, nodal/channel radial peaking factor does not make sense as the power distribution within the assembly is flat. When using reflective boundary conditions, the difference between PbP/channel and PbP/subchannel solutions is -0.1% indicating that power distribution is more pronounced in the reference solution.

In the same case but with vacuum boundary condition $F\Delta H$ is 2.9% higher in PbP/channel than in PbP/subchannel indicating higher gradients in power in the average channel calculation. This may be explained by the different feedback that the average value of coolant temperature and density and Doppler temperature that the NK solver reads from the TH calculation, producing an increase in the power where power is higher and a decrease where is lower and, therefore, magnifying the difference.

For F_z (Table 5.3), the tendency is what expected and the opposite of F_dH , being always higher for the reference solution. When the nodal/channel approximation is used, and boundary conditions include vacuum in one side, F_z becomes 4.6% lower than the reference solution. In the case of running a PbP-channel calculation, the factor is 2.7% lower than reference. This may be explained by differences in the feedback. At all the axial levels, the channel average coolant temperature and the channel average fuel temperature are lower than the coolant and fuel temperatures of the hottest fuel rod of the assembly. The channel average coolant density is higher than the coolant density in subchannels around the hottest fuel rod. Those differences between the average values and the subchannel values are higher at the top of the fuel assembly than at the bottom. The higher temperatures of coolant and fuel and lower densities of the coolant when performing the PbP/subchannel calculation reduce the reactivity at the top of the

fuel assembly, also reducing the power at these axial levels and therefore increasing the peaking factor F_z .

Table 5.1 Multiplication Factor

	Nodal NK – channel TH	Pin-by-pin NK – channel TH	Pin-by-Pin NK – subchannel TH
Reflective BC at all interfaces	1.21903 (+74 pcm)	1.21830 (1 pcm)	1.21829 (ref)
Void BC on the south side and reflective on the rest sides	0.97616 (+675 pcm)	0.97571 (+630 pcm)	0.96942 (ref)

Table 5.2 Radial power peaking factors $F_{\Delta H}$ (axially integrated)

	Nodal NK – channel TH	Pin-by-pin NK – channel TH	Pin-by-pin NK – subchannel TH
Reflective BC at all interfaces	1	1.017 (-0.1%)	1.019
Void BC on the south side and reflective on the rest sides	1	1.447 (2.9%)	1.406

Table 5.3 Axial power peaking Factor, F_z

	Nodal NK – channel TH	Pin-by-pin NK – channel TH	Pin-by-pin NK – subchannel TH
Reflective BC at all interfaces	1.539 (-1.1%)	1.555 (-0.1%)	1.556
Void BC on the south side and reflective on the rest sides	2.119 (-4.6%)	2.161 (-2.7%)	2.220

Figure 5.2 Radial power distribution with three different radial nodalization and combinations

The radial power factors axially integrated for the three configurations using vacuum boundary conditions are shown in Figure 5.2. In the nodal/channel case, power profile is plane since it is a nodal calculation. In the other two cases, there is a strong gradient power profile in the fuel assembly. The difference in the power distribution can be observed at the north side of the assembly showing a darker red color in this area for the PbP/channel calculation in Figure 5.2b.

5.3 Full core analysis

Previous section analyzed the effect of combining meshes with different accuracy level at each one of the physics in the neutronic - thermal-hydraulic problem using just a fuel assembly. The differences among the three combinations are more considerable when there is a strong asymmetry in the neutronic radial flux distribution. It is important to take into account the effect of the radial combination of meshes. The combination of PbP/channel calculations underestimates the thermal-hydraulic feedback and increase the asymmetry in the neutron distribution (Figure 5.2b).

All the pin-cell inside the average thermal-hydraulic channel reads the same Doppler temperature, moderator temperature, boron concentrations, and moderator density. This means that the Doppler temperature of the average channel calculations is lower than the Doppler temperature of the sub-channel calculations in the regions that generate more power. The neutronic flux will be higher in the PbP/channel than in the PbP/subchannel because the negative reactivity introduced in these regions are lower.

In order to understand the effect exerted on the power evolution by the different combinations of meshes with different accuracy level at each one of the physics, a full core calculation was carried out employing the PbP solver for the neutronic calculations. COBAYA4 solved the problem at PbP level with the cross sections created at pin level while COBRA-TF solved the problem at an average channel level.

With this work, the term of the UPM in NURESAFE demonstrates the capabilities of its code, COBAYA4, to carry out a full core calculation at pin level.

Although COBRA-TF is able to carry out full core calculations at sub-channel level, with the version included in the NURESIM platform it is still prohibited in computational time for the cases to be analyzed for this work. Therefore, in this case, the thermal-hydraulic calculation was performed employing one lumped channel per fuel assembly.

5.3.1 Overview of the NURESAFE benchmark

Within NURESAFE project, a MSLB transient was defined and solved by the codes integrated into NURESIM platform and COBAYA/COBRA-TF coupled system participated in the analysis.

The reference reactor used was a four loop Westinghouse Zion PWR. Two types of fuel assemblies were included made from MOX and UOX, each one with two different initial

enrichments. The cross-sections libraries were created according to the fuel assembly geometry at End Of Cycle (EOC) and included axial burnup distribution. The EOC cross-sections maximize the overcooling effect because the boron concentration at EOC is zero. The transient is initiated at Hot Zero Power (HZP) from a subcritical state corresponding to a 1% of shutting down margin (SDM), all rods inserted with the highest worth control rod stuck in the overcooled area (Kliem et al. 2017).

Helmholtz-Center Dresden-Rossendorf (HZDR) provided time-dependent thermal-hydraulic boundary conditions using ATHLET system code to take into account the plant response. The thermal-hydraulic boundary conditions were set as an input in COBRA-TF.

The reactor experiments a power excursion once the reactivity achieves specific value, close to 400 pcm. More details about the transient can be found in (Chanaron et al. 2013)

Two transient calculations were performed. The first one was a nodal-channel analysis and the second one, a PbP-channel calculation. The PbP/channel calculations require the capabilities of the MedCoupling library included in SALOME platform to interpolate the values between the meshes with different size.

5.3.2 Steady-state analysis

A detailed analysis of the steady state calculation from the benchmark NURESAFE was performed in (Sánchez-Cervera 2017).

5.3.3 Transient calculations

The evolution of global core parameters during the MSLB transient is shown in Figure 5.3. Pin-by-pin and nodal simulations provide comparable results, with re-criticality and sudden power increase further followed by a smooth power rise. Comparisons of relevant parameters characterizing the main events during the transient are in Table 5.4.

The PbP solution showed an initial power excursion delayed with respect to the nodal solution by around 1 s. The maximum level of core power was 240.3 MW, 1.7% higher than for the nodal solution, in both cases ~89s after the MSLB event (occurring at 0 s). Regarding the average fuel temperature, slight differences can be appreciated due to the different power feedback in both simulations (García-Herranz et al. 2017).

Figure 5.3 Power evolution nodal and pin by pin calculation

Figure 5.4 Zoom of the power evolution in the power excursion region

The power evolution of the two transients shows differences generated by the neutronic radial nodalization as can be seen in Figure 5.3 and Figure 5.4. The steady state is the same in both calculations, but the radial nodalization introduces changes in the transient evolution. The differences come from the nodalization because the evolution of the

thermal-hydraulic variables is the same before the power excursion as Figure 5.5 shows. Power excursion takes place earlier in the nodal calculation than in the PbP one. The peak power magnitude is bigger in the PbP calculation than in the nodal calculation in 15 MW approximately.

Figure 5.5 Average moderator temperature evolution nodal and pin by pin

As in case analyzed in section 5.2.2, the PbP/channel calculation underestimates the feedback and increases the F_z factor. In the transient calculation, something similar happens and the peak power is higher with this configuration.

Table 5.4: Comparison of parameters corresponding to the main events in the transient

	COBAYA4/PbP		COBAYA4/Nodal	
	Time (s)	Value	Time (s)	Value
Core criticality	28.5	-	28	-
Maximum reactivity	41	387.6 pcm	40.5	403.0 pcm
Maximum core power	89	240.3 MW	88.5	236.6 MW
Return to subcriticality	106.5	-	111	-

Table 5.4 shows the time when the most important events occur from both calculations. Peaking factors values at the moment of maximum core power are illustrated in Table 5.5 and temporal evolution can be seen in figures Figure 5.6, Figure 5.8 and Figure 5.8.

Pin power peaking factors (F_{xy} and F_q) are around 14% higher than the corresponding averaged-assembly power peaks, indicating that the gradients in the hot assembly are similar to the steady-state condition. It should be taken into account that in the pin-by-pin calculation, sub-channel effects are not considered, so the local heterogeneity in thermal-hydraulics model inside the assembly is lost, and an average of the TH fields are used during the pin-wise cross sections feedback. As in section 5.2.2, the thermal hydraulic code returns to the neutronic one the same Doppler temperature and moderator temperature for all the channels and fuel rods inside the fuel assembly. This increases the power of the hottest fuel rod and increases the power peaking factors. As a result, the power distribution within the fuel assemblies could be less pronounced than if a subchannel calculation was used because COBAYA4 reads an average value of the fuel assembly instead the value corresponding to the sub-channel and the fuel rod.

Table 5.5: Peaking factors at the moment of maximum core power and differences in percent with respect to the pin-by-pin solution

	COBAYA4/PbP	COBAYA4/Nodal	Diff (%)
Axial power peaking factor (F_z)	1.70	1.68	-1.1%
Radial assembly peaking factor ($F_{xy_assembly}$)	7.89	8.18	3.6%
Radial pin peaking factor (F_{xy_pin})	9.00	-	-
Hot spot power peaking factor ($F_{q_assembly}$)	12.93	13.27	2.6%
Hot spot power peaking factor (F_{q_pin})	14.72	-	-

Figure 5.6 Comparison F_q nodal and PbP integrated to assembly calculation

Figure 5.7 Comparison F_{xy} nodal and PbP integrated to assembly calculation

Figure 5.8: Evolution of the radial (axially-integrated) peaking factor F_{xy} , F_z (radially-integrated) peaking factor and F_q along the transient for pin-by-pin and nodal calculations

Figure 5.10 shows the radial power distribution of the nodal calculation. In this case, each fuel

Figure 5.9 show the relative power factor of each fuel assembly for the nodal calculations, although the neutronic calculations were performed at the pin level.

Figure 5.9 Radial power distribution at 95 seconds of transient obtained by the PbP simulation

Figure 5.10 Radial power distribution at 95 seconds of transient obtained by the nodal simulation

A detailed comparison of the normalized radial assembly-power profile is shown in Figure 5.11. The main differences take place around the thermal-hydraulic perturbation, as figure shows. The pin-by-pin simulation predicts an assembly-averaged power distribution slightly more flattened than the nodal simulation.

Figure 5.11 Relative difference taking the maximum value of the nodal solution as a reference solution

In the region where the thermal-hydraulic perturbation takes place, the relative differences taking as reference solution the nodal solution, is around -3.4 %.

The negative value shows that the relative power is greater in the nodal-channel calculation than in the PbP-channel one. The same behavior is appreciated in section 5.2.2. The transient pin-by-pin calculation took about 50 hours on a single Intel Core i7 at 3.07 GHz, while the transient nodal calculation took about 40 min. on the same core.

Figure 5.12 shows the radial power density distribution obtained with SALOME6.6 visualization tool, where the resolution can be observed for both types of analysis.

Figure 5.12: Axially-integrated radial power distribution for the pin-by-pin (at the pin level) and the nodal calculation

Chapter 6 Transient analysis in NURESIM Platform

6.1 Introduction

As part of the safety evaluation of the nuclear power plant, a detailed analysis of the plant response to postulated disturbances of variables or equipment is necessary. (R. Martin and Frepoli 2018). This results in different types of transients that have to be analyzed by mean of the simulation codes.

In order to analyze the impact that the type of temporal coupling scheme may have in the results obtained, two transients were selected that are classified as a Reactivity Insertion Accidents (RIA). These types of transients involve an important increase of fission rates and consequently of reactor power that may damage the reactor. On this kind of transients, the behavior of the coupled neutronic and thermal-hydraulic codes is very important due to the significant impact of the physics feedbacks on the transient evolution.

The first one consists in the ejection of a control rod and it can be classified as a neutronic transient. Therefore, the transient is initiated by a change in the cross sections without initial variation of the thermal-hydraulic parameters. The change of the composition of the core introduces the reactivity and initiates the transient. The transient is led by COBAYA4.

The second transient selected is a MSLB. In this case the reactivity insertion is due to the decrease of the moderator temperature that follows to the break of the main steam line in a PWR and the subsequent depressurization on this line. Due to this fact the primary yields high amount of energy to secondary at the steam generator reducing the temperature of the coolant/moderator at the core entrance.

The MSLB can be classified as a thermal-hydraulic transient. The thermal-hydraulic transients are those initiated by variations in the thermal-hydraulic variables, as fuel temperature, moderator temperature that generates variations in the power. In this case the transient is led by COBRA-TF.

This chapter analyzed the effect exerted by the coupling scheme implemented between COBAYA4 and COBRA-TF on the transient evolutions and search which coupling shows

better performance according to the type of transient. An accident of each group, neutronic and thermal-hydraulic transients, will be analyzed in this document.

6.2 Temporal coupling with COBRA-TF

Along NURESAFE project COBAYA4 was integrated into SALOME platform with all its capabilities. From that moment, it was possible to couple COBAYA4 with any of the other thermal-hydraulic code integrated into NURESIM platform, i.e. COBRA-TF, SUBCHANFLOW, FLICA4.

The coupling developed between COBAYA4 and COBRA-TF is an external coupling. The external coupling needs fewer changes into the source of the codes and simplifies the switch to any other code integrated into NURESIM platform. The fields exchanged in the coupling, as power, temperature, and density, are stored in different meshes created by each code (Bennett, Avramova, and Ivanov 2016). COBAYA4 creates one mesh to save the power and COBRA-TF creates two meshes with a different number of nodes: one channel center to store the moderator temperature, and density and another rod centered for Doppler temperature. COBRA-TF solves two problems, the thermo-hydraulic problem to obtain the moderator properties, and the thermo-mechanic one to calculate the fuel and clad temperature distribution. For lumped channel calculations, the same mesh is considered for rods and channels.

In order to perform transient analysis with the coupled system COBAYA/COBRA-TF a temporal evolution algorithm had to be developed that will be explained in detail in section 6.3.

But these developments required also some modifications and adjustment of the integration of COBRA-TF into the NURESIM platform.

The first one was related with the subroutine in charge of calculating the time step.

Python script establishes time step for coupling. COBRA-TF chooses internal the time step size employing the Courant number and other internal or input parameters. During the coupling time step several COBRA-TF steps are performed as can be seen in Figure 6.1. Last COBRA-TF time step do not usually match coupling time step. A modification had to be done at COBRA-TF source code to set the time step size of the last iteration to match with the end of the coupling time step length. The width of the last thermal-hydraulic iteration must be appropriate to match with the endpoint of the neutronic iteration.

Figure 6.1 Neutronic thermal-hydraulic iteration

These requirements relative to the size of the last thermal-hydraulic iteration had to be addressed with several modifications in the COBRA-TF source:

- The subroutine developed to calculate the time step of the next thermal-hydraulic calculation was modified to include a modification of the final TH time-step. The modification implicated a definition of a flag to select if COBAYA4/COBRA-TF were employing a staggered temporal mesh or not, and the capabilities to define the length of the last thermal-hydraulic iteration match with the end of the time step.
- The input was modified to include a new variable (flag) to indicate the neutronic and thermal-hydraulic time step and another to indicate if the coupling employs a staggered temporal mesh.

Other modifications were implemented that will be explained in section 6.3.2 to carry out calculations with different temporal meshes.

6.3 Coupling schemes implemented

The coupling developed between COBAYA4 and COBRA-TF is an external coupling through the SALOME platform. A python script guides the execution of COBAYA4 and COBRA-TF.

The external coupling allows to study different coupling methods and analyze the effect exerted by them on the transient evolution and in the safety analysis of the thermal-hydraulic variables as the DNBR and the fuel centerline temperature.

The coupling schemes analyzed in this work will be the Operator Splitting and the enhanced method from the operator splitting

The most widely employed method to solve multi-physics problems using an external coupling is the Operator-Splitting. In this case, the problem is split into sub-problems, as many as physical problems are solved together.

6.3.1 Operator splitting

Operator Splitting (OS) is the broader and easier method employed in the multi-physics problem to implement a coupling technique. The physical fields to be solved are represented by the two codes coupled, COBAYA4 and COBRA-TF.

The problem is divided into so many differential equations as the physics in which the problem can be divided. The solution of one physic is the input of the other. The translation of the coupling scheme with the OS to the equations is represented by Eq. (6-1) and (6-2) with two pair of differential coupled equations (Omar Zerkak, Kozlowski, and Gajev 2015):

$$\frac{\partial u}{\partial t} = g(u, v, t) \quad (6-1)$$

$$\frac{\partial v}{\partial t} = f(u, v, t) \quad (6-2)$$

Eq. (6-1) and (6-2) are coupled, and they can be discretized by:

$$\frac{u^{n+1} - u^n}{\Delta t} = g(u^n, v^n) \quad (6-3)$$

$$\frac{v^{n+1} - v^n}{\Delta t} = f(u^{n+1}, v^n) \quad (6-4)$$

Two semi-implicit equations describe the coupling problem. Eq. (6-3) is solved in the first place as an explicit equation. Eq. (6-4) is an implicit equation because the function f requires the value of the variable u at the time $n+1$, u^{n+1} .

To integrate the equations (6-3) and (6-4) into the coupling script is necessary to discretize the functions f and g of the right side of both equations. The functions f and g represent the derivative.

Three techniques can be applied to obtain the value of the derivative, Eq (6-5), (6-6) and (6-7).

$$f'(x_0) = \frac{F(x_0 + h) - F(x_0)}{h} \quad (6-5)$$

$$f'(x_0) = \frac{F(x_0) - F(x_0 - h)}{h} \quad (6-6)$$

$$f'(x_0) = \frac{F(x_0 + h) - F(x_0 - h)}{2h} \quad (6-7)$$

The Eq. (6-5) and (6-6) represent the forward and backward derivative. Both are first order methods. Eq. (6-7) is middle point derivative and, it is a second order method.

Each one of the methods introduces a different value for the derivative and can modify the coupling evolution. Figure 6.2 represents the three methods for the function derivatives:

Figure 6.2 Numerical derivative methods

The three discretization methods can be employed to obtain the derivative showed in the Eq. 2a and 2b. Moreover, it is important to be sure which discretization is employed to apply a first or second order coupling method.

Moreover, it is important to understand at Eq.2b and 2a that the right side represents the values transferred through the coupling, and the function g has as argument the power transferred through the coupling and the thermal-hydraulic variable from the previous calculation.

Thus, Eq. (6-8) can be rewritten with the terminology of the problem as:

$$\frac{TH_{coupling}^{n+1} - TH_{coupling}^n}{\Delta t} = g(POW_{coupling}^n, TH_{calculation}^n) \quad (6-8)$$

The thermal-hydraulic code is the first one to perform a new calculation once the steady state has been achieved, employing the power and the thermal-hydraulic variables at the end of the steady state (iteration 0). Writing Eq. (6-8) with the scripts of the first iteration:

$$\frac{TH_{coupling}^1 - TH_{coupling}^0}{\Delta t} = g(POW_{coupling}^0, TH_{calculation}^0) \quad (6-9)$$

The values of the thermal-hydraulic variables through the coupling at the first iteration ($t = 0$) are the thermal-hydraulic variables calculated at the steady state ($TH_{coupling}^0 = TH_{calculation}^0 = TH_{steady}$). From the steady state, with the power and starting the transient calculations from the thermal-hydraulic code, COBRA-TF runs a new calculation to obtain the thermal-hydraulic variables at time $n=1$. Applying the forward derivation to the Eq. (6-9) to discretize the function g , the values of the thermal-hydraulic parameters exchanged through the coupling in the first iteration can be set.

$$\begin{aligned} \frac{TH_{coupling}^1 - TH_{coupling}^0}{\Delta t} &= g(POW_{coupling}^0, TH_{calculation}^0) \\ \frac{TH_{coupling}^1 - TH_{coupling}^0}{\Delta t} &= \frac{TH_{calculated}^1 - TH_{calculated}^0}{\Delta t} \end{aligned} \quad (6-10)$$

The starting point $TH_{calculated}^0$ of Eq. (6-10) is the steady state. Moreover, the value transferred once the thermal-hydraulic calculations finishes is the value calculated, or in other words

$$TH_{coupling}^0 = TH_{calculated}^0 \quad (6-11)$$

Introducing Eq. (6-11) into Eq. (6-10) the result of the OS is Eq. (6-12):

$$TH_{coupling}^1 = TH_{calculated}^1 \quad (6-12)$$

The values of thermal-hydraulic variables transferred to COBAYA4 at the first iteration are the calculated by COBRA-TF at the first iteration.

This analysis can be performed too with the neutronic code employing the forward derivative achieving the same conclusion. With the operation splitting, the value transferred through the coupling is the value calculated at the last iteration because the value calculated and transferred at the steady state is the same.

$$\frac{POW_{coupling}^1 - POW_{coupling}^0}{\Delta t} = f(POW_{coupling}^0, TH_{coupling}^1) \quad (6-13)$$

$$\frac{POW_{coupling}^1 - POW_{coupling}^0}{\Delta t} = \frac{POW_{calculated}^1 - POW_{calculated}^0}{\Delta t}$$

$$POW_{coupling}^1 = POW_{calculated}^1 \quad (6-14)$$

Eq. (6-14) and (6-12) describe the first iteration, but they can be employed to explain all the iterations. The values transferred and calculated at the previous iterations (n) in the equations are the same, so the coupling equations are:

$$TH_{coupling}^{n+1} = TH_{calculated}^{n+1} \quad (6-15)$$

$$POW_{coupling}^{n+1} = POW_{calculated}^{n+1} \quad (6-16)$$

The Eq. (6-15) and (6-16) represent the coupling between COBAYA4 and COBRA-TF employing the operator splitting.

The sequence of the performance of the coupling is described as follows (Figure 6.3):

- COBRA-TF updates the power with the results of the neutronic calculation (1)
- COBRA-TF carries out a new calculation with the power of the previous time-step solution or the power of the steady state for the first step in time (2)
- Temperatures and densities go through the coupling to COBAYA4 (3)
- COBAYA4 updates the cross sections with the new temperatures and densities calculated by COBRA-TF and carries out a new calculation (4)
- Once COBAYA4 achieves the end of the time-step transfers the power to COBRA-TF before the next iteration (5)

Figure 6.3 Operator Splitting scheme with COBAYA4 and COBRA-TF

The discretization employed to calculate the derivative is the forward derivative. The backward derivative cannot be employed, at least at the first iteration of the transient calculation, because the previous point required to calculate does not exist. The same problem appears with the central derivation.

These two derivation methods, backward and central, can be employed after the first iteration. Eq. (6-17) and (6-18) after the first iteration employing the backward derivation:

$$\frac{POW^{n+1} - POW^n}{\Delta t} = f(POW^n, TH^{n+1}) =$$

$$\frac{F(x_0) - F(x_0 - h)}{h} = \frac{POW^n - POW^{n-1}}{\Delta t}$$

$$POW^{n+1} = 2POW^n - POW^{n-1} \quad (6-17)$$

$$TH^{n+1} = 2TH^n - TH^{n-1} \quad (6-18)$$

The same problem but employing the central derivation in power:

$$\frac{POW^{n+1} - POW^n}{\Delta t} = f(POW^n, TH^{n+1})$$

$$\frac{F(x_0 + h) - F(x_0 - h)}{2h} = \frac{POW^{n+1} - POW^{n-1}}{2\Delta t}$$

$$\frac{POW^{n+1} - POW^n}{\Delta t} = \frac{POW^{n+1} - POW^{n-1}}{2\Delta t}$$

$$POW^{n+1} = POW^n + \frac{POW^{n+1} - POW^{n-1}}{2}$$

$$POW^{n+1} = 2POW^n - POW^{n-1} \quad (6-19)$$

$$TH^{n+1} = 2TH^n - TH^{n-1} \quad (6-20)$$

The result employing the backward and the central derivation is similar to the technique employed in an implicit Euler problem to predict the term of the next iteration.

The predictor-corrector method is employed to calculate the implicit functions as Eq.(6-21)

$$y_{k+1} = y_k + hf(y_{k+1}, t_{k+1}) \quad (6-21)$$

To solve Eq. (6-21), it is necessary to know the point y_{k+1} previously to solve the equations. One technique to solve this equation is to employ a predictor-corrector method. Predictor estimates the value of y_{k+1}^p within the function (Eq. (2-21)); and the corrector calculates the value of the variable y_{k+1} (Eq. (6-22)).

$$y_{k+1} = y_k + hf(y_{k+1}^p, t_{k+1}) \quad (6-22)$$

The predictor values are calculated employing a Tylor series, and it could be employed in the problem of the neutronic-thermal-hydraulic coupling (Eq. (6-24) and (6-25)).

$$f(x + \Delta x) = f(x) + \Delta x f'(x)$$

$$f(x + \Delta x) = f(x) + \Delta x \frac{f(x) - f(x - \Delta x)}{\Delta x}$$

$$f(x + \Delta x) = 2f(x) - f(x - \Delta x)$$

$$y_{k+1}^p = 2y_k - y_{k-1} \quad (6-23)$$

$$POW^{n+2} = 2POW^{n+1} - POW^n \quad (6-24)$$

$$TH^{n+1} = 2TH^n - TH^{n-1} \quad (6-25)$$

The graphical view of the Eq. 8 is represented in Figure 6.4. The predicted value is a combination of the two previous calculations.

Figure 6.4 Graphical representation of the predictor technique

The coupling model employing the backward and the central derivation generates a coupling based on the predictor equations. This means that the value transferred is a

prediction instead to be the value calculated at the last iteration. These two derivatives methods could be employed but introduce an error, as Figure 6.4 shows.

The best derivative method to employ in the case of the OS is the forward method.

As the Eq. (6-15) and (6-16) show, the field exchanged with the Operator Splitting is the value calculated by each code at the last iteration.

The diagram of the Operator Splitting method employed between COBAYA4 and COBRA-TF is represented in Figure 6.5.

Figure 6.5 Diagram of the OS

6.3.2 Enhanced methods

The OS is the easiest and widely used coupling method (TRACE-PARCS??). OS is the coupling method implemented initially between COBAAY4 and COBRA-TF.

It is possible to enhance the OS employing staggered temporal meshes. The staggered meshes allow estimating the field exchange among codes at a different temporal point.

The enhanced methods are two; both of them use staggered time mesh. The first one is the staggered mesh and the second one is a staggered mesh extrapolated or extrapolated method (EM).

In order to implement those methods new modifications had to be performed in COBRA-TF. As both methods require that the first thermal-hydraulic calculation for the coupling employ half time step instead of full time step, this had to be included in the subroutine that controls the COBRA-TF temporal discretization.

6.3.2.1 Staggered mesh method

In the staggered mesh method, the start and endpoint of the iteration of the two codes do not match in the time. The endpoint of the iteration of one code is the middle point of the iteration of the other code. The value exchanged from one to another code is the interpolated at the middle point of the iteration.

Staggered method performance is described in figure2.

Figure 6.6 Staggered mesh scheme performance

1. A thermal-hydraulic calculation is carried out between $n-1/2$ and $n+1/2$
2. COBAA4 updates the cross sections libraries and carries out a new calculation between n and $n+1$.

The two meshes do not match in time, but implementation on different coupling methods are possible.

Figure 6.6 could be represented by a set of two coupled PDE:

$$\frac{u^{n+\frac{1}{2}} - u^{n-\frac{1}{2}}}{\Delta t} = \theta_1 f(u^{n+1}, v^n) + (1 - \theta_1) f(u^n, v^n) \quad (6-26)$$

$$\frac{v^{n+1} - v^n}{\Delta t} = \theta_2 g\left(u^{n+\frac{1}{2}}, v^{n-\frac{1}{2}}\right) + (1 - \theta_2) g\left(u^{n+\frac{1}{2}}, v^{n-\frac{1}{2}}\right) \quad (6-27)$$

The equations (6-26) and (6-27) represent the semi-implicit problem, where the implicit and explicit functions are employed to solve the coupling problem. Different configurations can be generated changing the value of the parameters θ_1 and θ_2 . In this work, the value assigned to the two parameters are the same, $\theta_1 = \theta_2 = 1/2$. The value of the two parameters, θ , corresponds with the Crank-Nicolson scheme (CN). CN is a second order method and one of the advantages of the second staggered meshes is to keep the accuracy order of the CN method employing the central derivative discretization.

The Eq. (6-26) and (6-27) can be rewritten to the problem analyzed here by Eq. (6-28) and (6-29).

$$\frac{POW^{n+\frac{1}{2}} - POW^{n-\frac{1}{2}}}{\Delta t} = \frac{1}{2} (f(POW^{n+1}, TH^n) + f(POW^n, TH^n)) \quad (6-28)$$

$$\frac{TH^{n+1} - TH^n}{\Delta t} = \frac{1}{2} \left(g\left(POW^{n+\frac{1}{2}}, TH^{n-\frac{1}{2}}\right) + g\left(POW^{n+\frac{1}{2}}, TH^{n-\frac{1}{2}}\right) \right) \quad (6-29)$$

Eq. (6-28) represents the power, the calculations are performed between the point n and $n+1$, but the left side of the equation is referred to the time mesh of the thermal-hydraulic calculation. The same happens in the thermal-hydraulic equation Eq. (6-29). This structure of the coupling indicates that the value calculated is not the value

exchanged, like the OS, but the values exchanged are a combination of the previous calculations.

Crank-Nicolson is a second order method, so the discretized method employed for the two functions g and f could keep or degrade the accuracy order. Application of different discretization methods with different accuracy orders will allow studying the effect of the order in the transient evolution accuracy.

The first discretization of the functions f and g will be done with a first-order method employing the backward discretization.

$$\begin{aligned}\frac{POW^{n+\frac{1}{2}} - POW^{n-\frac{1}{2}}}{\Delta t} &= \frac{1}{2}(f(POW^{n+1}, TH^n) + f(POW^n, TH^n)) \\ \frac{POW^{n+\frac{1}{2}} - POW^{n-\frac{1}{2}}}{\Delta t} &= \frac{1}{2}\left(\frac{POW^{n+1} - POW^n}{\Delta t} + \frac{POW^n - POW^{n-1}}{\Delta t}\right) \\ POW^{n+\frac{1}{2}} &= POW^{n-\frac{1}{2}} + \frac{1}{2}(POW^{n+1} - POW^{n-1})\end{aligned}\tag{6-30}$$

The value exchanged in the coupling is the value exchanged in the previous iteration and the average value transferred in current and two previous iterations.

Applying the same scheme to the thermal-hydraulic calculation, describes de thermal-hydraulic coupling:

$$TH^n = TH^{n-1} + \frac{1}{2}\left(TH^{n+\frac{1}{2}} - TH^{n-\frac{3}{2}}\right)\tag{6-31}$$

The Eq. (6-31) and (6-31) represents the coupling with a Crank-Nicolson scheme degrades to first order.

Eq. (6-31) and (6-32) have to be employed in the second iteration, because there is not effect on the TH fields because the fields are interpolated employing the term $n - 3/2$ in the thermal-hydraulic.

To keep the second-order accuracy of the Crank-Nicolson methods, it is necessary to employ a second order method for the discretization of the functions f and g . The discretization of second order is implemented the middle point discretization.

$$\frac{POW^{n+\frac{1}{2}} - POW^{n-\frac{1}{2}}}{\Delta t} = \frac{1}{2} \left(\frac{POW^{n+2} - POW^n}{2\Delta t} + \frac{POW^{n+1} - POW^{n-1}}{2\Delta t} \right) \quad (6-32)$$

The middle derivative presents a problem. It requires a point that has not been calculated yet, i.e. the point POW^{n+2} must be predicted. The predictor method has to be a second order to keep the second order of the method.

The value of the power at the iteration $n+2$ is predicted by a Taylor polynomial as follow:

$$f(x) = f(x_0) + (x - x_0)f'(x_0)$$

$$f(x)_{pre} = f(x_0) + \frac{\Delta x(f(x_1) - f(x_0))}{\Delta x}$$

$$f(x)_{pre} = f(x_0) + \frac{\Delta x(f(x_0) - f(x_{-1}))}{\Delta x}$$

$$POW^{n+2} = 2POW^{n+1} - POW^n + O(\Delta t^2) \quad (6-33)$$

Introducing the predictor, Eq (6-33), into Eq. (6-32):

$$\frac{POW^{n+\frac{1}{2}} - POW^{n-\frac{1}{2}}}{\Delta t} = \frac{1}{2} \left(\frac{2POW^{n+1} - 2POW^n}{2\Delta t} + \frac{POW^{n+1} - POW^{n-1}}{2\Delta t} \right)$$

$$POW^{n+\frac{1}{2}} - POW^{n-\frac{1}{2}} = \frac{1}{4} (2POW^{n+1} - 2POW^n + POW^{n+1} - POW^{n-1})$$

$$POW^{n+\frac{1}{2}} - POW^{n-\frac{1}{2}} = \frac{1}{4} (3POW^{n+1} - 2POW^n - POW^{n-1})$$

$$POW^{n+\frac{1}{2}} = POW^{n-\frac{1}{2}} + \frac{1}{4}(3POW^{n+1} - 2POW^n - POW^{n-1}) \quad (6-34)$$

The equation 14 represents the second order staggered mesh coupling scheme in power. The same equation but applied to the thermal-hydraulic is the next one:

$$TH^{n+1} = TH^n + \frac{1}{4}\left(3TH^{n+\frac{1}{2}} - 2TH^{n-\frac{1}{2}} - TH^{n-\frac{3}{2}}\right) \quad (6-35)$$

Both Eq., (6-32) and (6-33), represent the coupling equations keeping the second accuracy order.

As a summary of this part, two pairs of coupling equations have been developed in this part of the paper; one pair is first order, Eq. (6-30) and (6-31), meanwhile the other pair of equations is a second order, Eq. (6-34) and (6-35). The two pairs of equations allow studying the effect exerted by the accuracy order method on the transient evolution.

Moreover, Eq. (6-34) and (6-35).are more complex than Eq. (6-30) and (6-31). This will affect the executing time and will be interesting is the improvement are more important than the weight of the time when a second order method is employed.

6.3.2.2 Extrapolated method

The staggered mesh method derives from the Crank-Nicolson method. This one is obtained when the θ problem of Eq.9a and b is solved with $\theta_1 = \theta_2 = 1,5$.

The value employed with the value of $\theta_1 = \theta_2 = 0.5$ represents an interpolation. The values transfer in the feedback is an interpolation between the current and the previous calculations. In the stagger mesh, the thermal-hydraulic code initiates the transient and calculate with the middle point of the other code grid.

In the EM, the values transferred by the thermal-hydraulic are referred to the neutronic time grid, but instead of interpolating as the CN of the stagger mesh, the value of the thermal-hydraulic variable is extrapolated to the endpoint of the neutronic calculation.

Figure 6.7 represents the performance of this coupling method.

Figure 6.7 Extrapolated method performance

The temperatures and densities transferred through the feedback correspond with the endpoint of the neutronic iteration, point $n+1$. The cross sections are updated with the extrapolated value of the temperatures and densities instead of their values at the middle point if the iteration as the Crank-Nicolson method. The power changes are faster than the changes in the thermal-hydraulic variables, and it is not necessary to apply the extrapolation (Merino 1993).

In order to extrapolate the feedback θ should be equal to 1.5. With this value, the feedback is extrapolated middle time-step until the endpoint of the next iteration is achieved as represent Figure 6.7.

Applying $\theta = 1.5$ to the system of equations, the coupling equations are:

$$\frac{u^{n+\frac{3}{2}} - u^{n+\frac{1}{2}}}{\Delta t} = 1.5f(u^{n+1}, v^{n+1}) - 0.5f(u^n, v^{n+1}) \quad (6-36)$$

$$\frac{v^{n+1} - v^n}{\Delta t} = 1.5g\left(u^{n+\frac{1}{2}}, v^{n+\frac{1}{2}}\right) - 0.5g\left(u^{n+\frac{1}{2}}, v^{n-\frac{1}{2}}\right) \quad (6-37)$$

The Eq. (6-36) and (6-37) define a general problem. The adaptation to the neutronic and thermal-hydraulic problem is as follow.

$$\frac{POW^{n+\frac{3}{2}} - POW^{n+\frac{1}{2}}}{\Delta t} = 1.5f(POW^{n+1}, TH^{n+1}) - 0.5f(POW^n, TH^{n+1}) \quad (6-38)$$

$$\frac{TH^{n+1} - TH^n}{\Delta t} = 1.5g\left(POW^{n+\frac{1}{2}}, TH^{n+\frac{1}{2}}\right) - 0.5g\left(POW^{n+\frac{1}{2}}, TH^{n-\frac{1}{2}}\right) \quad (6-39)$$

Both equations represent the coupling between the two codes. The two functions, f and g can be again translated to NK and TH fields:

$$\begin{aligned} \frac{POW^{n+\frac{3}{2}} - POW^{n+\frac{1}{2}}}{\Delta t} &= 1.5f(POW^{n+1}, TH^{n+1}) - 0.5f(POW^n, TH^{n+1}) \\ \frac{POW^{n+\frac{3}{2}} - POW^{n+\frac{1}{2}}}{\Delta t} &= 1.5 \frac{\left(POW^{n+\frac{3}{2}} - POW^{n+1}\right)}{\frac{\Delta t}{2}} - 0.5 \frac{\left(POW^{n+\frac{1}{2}} - POW^n\right)}{\frac{\Delta t}{2}} \\ POW^{n+\frac{3}{2}} - POW^{n+\frac{1}{2}} &= 3 \left(POW^{n+\frac{3}{2}} - POW^{n+1}\right) - POW^{n+\frac{1}{2}} - POW^n \\ 2POW^{n+\frac{3}{2}} &= 3POW^{n+1} - POW^n \\ POW^{n+\frac{3}{2}} &= \frac{3}{2}POW^{n+1} - \frac{1}{2}POW^n \\ TH^{n+1} &= \frac{3}{2}TH^{n+\frac{1}{2}} - \frac{1}{2}TH^{n-\frac{1}{2}} \end{aligned} \quad (6-40)$$

$$(6-41)$$

It could be possible to think that the OS has the same performance than the EM, or very similar. The thermal-hydraulic values employed to update the cross sections libraries with the OS corresponds with the endpoint of the neutronic calculation, as the EM.

Two characteristics make the extrapolated coupling method interesting versus the others described previously. The first one is that the power read by COBRA-TF to update the temperatures and densities does not correspond with the end point of the thermal-hydraulic iteration, but the power corresponds with the middle point of the next

thermal-hydraulic iteration. The second is that the extrapolation in the thermal-hydraulic is higher when the time-step is larger.

The extrapolation and the staggered mesh reduce the sensitivity to the time-step size allowing the use of higher time-step size is reducing the calculation time.

6.4 Rod ejection transient

As explained in section 6.1, two reactivity transients were simulated to evaluate the performance of the coupling algorithms implemented. The first one is a neutronic transient consistent in a rod ejection event. The event consists of a rod ejection followed by a power excursion and was defined as Part IV of MOX PWR MOX/UO₂ Core Transient Benchmark (Kozłowski and Downar 2006). The transient takes just 1 second. The rod

ejection begins at 0.0 second of the transient when the steady state is achieved. The fuel rod is fully withdrawn in 0.1 seconds.

The rod ejection has the next technical features:

- The start point is a Hot Zero Power (HZP) scenario with $k\text{-eff}=1.00000$ (J. Jimenez 2010).
- A Westinghouse PWR core with four loops is considered. 3565 MWt and 193 fuel assemblies of UO₂ and MOX.
- The calculations are performed with one node per fuel assembly by COBAYA4, and one lumped channel per fuel assembly by COBRA-TF.

A more detailed description of the benchmark can be read in the above-mentioned reference.

The coupling schemes described in previous sections are used for the analysis in order compare the performance. Next section analyzes the transient with the four coupling schemes.

6.4.1 Reactivity transient with the Operator Splitting

Operator Splitting is the simplest coupling scheme that can be implemented between COBAYA4 and COBRA-TF.

The evolution of the transient is described in Figure 6.8 using different time steps.

Figure 6.8 Power and Doppler temperature evolution in the rod ejection transient using OS coupling method

Figure 6.8 shows how transient evolution has a strong sensitivity to the time step size represented by the different curves. Power and Doppler temperature change when the time-step changes too. A reduction in the time-step size reduces the power, particularly at the peak.

Table 6.1 Magnitude of the peak power

Time-step (s)	0.01	0.005	0.0025
Peak Power (MW)	7097	6663	6332

The dependence of the transient evolution with the time-step is translated into a variation in the thermo-hydraulic variables. The Doppler temperature rises faster with a larger time-step. The power evolution of the transient shows the same tendency than the Doppler temperature.

Figure 6.9 describes temporal coupling with the OS method and explains the reason why this method is so sensitive to the time-step.

Two calculations with the OS method and different time-step are performed. The first one finalizes the thermal-hydraulic calculation at temporal point $n+1$.

$$TH(n + 1) = CTF[TH(n), POW(n), t] \quad (6-42)$$

$$POW(n + 1) = COB4[POW(n), TH(n + 1), t] \quad (6-43)$$

If the time step is reduced to half, there is a new neutronic/thermal-hydraulic iteration between n and $n+1$. The values of the thermal-hydraulic variables and the power at point $n+1$ are as follow

$$TH(n + 1) = CTF \left[TH \left(n + \frac{1}{2} \right), POW \left(n + \frac{1}{2} \right), t \right] \quad (6-44)$$

$$POW(n + 1) = COB4 \left[POW \left(n + \frac{1}{2} \right), TH(n + 1), t \right] \quad (6-45)$$

If there is a power excursion, as the rod ejection before the peak power:

$$POW \left(n + \frac{1}{2} \right) > pow(n) \text{ and } TH(n + 1)_{\Delta t=x} < TH(n + 1)_{\Delta t=\frac{x}{2}} \quad (6-46)$$

Lower temperatures introduce more reactivity so the peak power increases when increasing the time step.

The system of equations probes that OS is a semi-implicit coupling scheme. COBAYA4 reads the thermal-hydraulic variables of the end point of the iteration instead of at the beginning. COBRA-TF reads the power at the beginning of the iterations, therefore the thermal-hydraulic calculation represents an explicit equation.

Figure 6.9 Effect exerted by the time-step size on the transient evolution with OS

The variations in the power due to the time step affect the safety parameters. The Doppler temperature plotted in Figure 6.10 is the average value for the core, and the maximum value is lower than 585 K (312 °C).

Figure 6.10 depicts the axial profile of the fuel centerline temperature in the hottest fuel assembly (assembly number 53). The maximum value of the temperature changes 11 °C just changing the time step.

Figure 6.10 Effect exerted by time-step size on the fuel centerline temperature of the hottest fuel assembly employing OS method

The analysis has been focused on the Doppler temperature and in the total power evolution. Rod ejection transient is a short transient. It takes just one second to happen, and the moderator does not feel the power excursion (in the local and the average variables). Figure 6.11 shows the evolutions of the average moderator temperature and density. The average temperature changes 2K. The changes are not significant. The Doppler coefficient reduces the power before than the moderator coefficient.

Figure 6.11 Evolution of the average moderator temperature and density with the OS

The analysis performed for the next coupling schemes will not include the moderator temperature and density analysis.

OS method shows a significant sensitivity to the time step. If the time step would be larger than 0.01 second, the peak power will be higher.

Next sections will analyze the other three proposed methods in other to study advantages and disadvantages compared with the OS method.

6.4.2 Reactivity transient with the first order Crank-Nicolson method

CN first-order method is the next coupling scheme analyzed. The Figure 6.12 is the total power and average Doppler evolution employing different time step.

Figure 6.12 Power and Doppler temperature evolution using first order Crank-Nicolson method

The power and average Doppler temperature show a keen sensitivity to the time-step size. The power evolution of the CN1 method is different than the OS one; the maximum value of the total power is larger in the CN1 than in the OS.

When time step is 0.01 seconds, the peak power is around 10GW. The peak power with a time step of 0.0025 seconds is 6.4 GW. The range is wide.

Table 6.2 Comparison of the evolution of the peak power with the time step of CN1 and OS

Time-step (s)	0.01	0.005	0.0025
Peak Power (MW), OS	7097	6663	6332
Peak Power (MW), CN1	10174	7885	6620

Table 6.2 shows the value of the peak power with the OS and CN1 for three coupling schemes. The peak power magnitude and the sensitivity to the time-step change with the coupling scheme. The size of the peak power is greater with the CN1 than the OS. Both coupling schemes requires to work with small time step.

The differences at the peak power in CN1 scheme have the same origin than at the OS, and it is explained in Figure 6.13.

Figure 6.13 Effect exerted by the time-step size on the CN1

The same analysis done in section 6.4.1 with the OS method can be done with the CN1. Figure 6.13 represent the same time interval with two different time steps. A time step and half the value. The equations involved in the coupling when the codes iterate from n to $n+1$ in a stagger time grid are:

$$TH\left(n + \frac{1}{2}\right) = CTF\left[TH\left(n - \frac{1}{2}\right), POW\left(n - \frac{1}{2}\right), t\right] \quad (6-47)$$

$$POW(n + 1) = COB4[POW(n), TH(n), t] \quad (6-48)$$

Both equations are explicit. The fields read by each code match with the beginning of the iteration. The coupling interpolates these values to make them match with the starting point of the iteration.

Employing half time step the coupling equations to achieve the point in the time grid $n+1$ are:

$$TH\left(n + \frac{3}{4}\right) = CTF\left[TH\left(n + \frac{1}{4}\right), POW\left(n + \frac{1}{4}\right), t\right] \quad (6-49)$$

$$POW(n + 1) = COB4\left[POW\left(n + \frac{1}{2}\right), TH\left(n + \frac{1}{2}\right), t\right] \quad (6-50)$$

If the power is rising:

$$TH\left(n + \frac{1}{2}\right) > TH(n) \quad (6-51)$$

As with the OS, increasing the time step the reactivity will increase too. This produces a more pronounced peak power.

CN1 is an explicit method. This makes the method less stable when it employs long time steps also emphasizing the peak power.

Table 6.3 depicts the average values (the same performance appears in the values of each fuel assembly) of the thermal-hydraulic variables calculated by COBRA-TF and transferred through the coupling employing a time step of 0.01 seconds.

Table 6.3 Differences between the values exchanged by the coupling and the values calculated by each code

Time COBRA-TF	Avg. Doppler temp calculated by COBRA-TF	Time Coupling	Value transfer
0.295	561.309	0.290	561.045
0.305	562.162	0.300	561.736
0.315	563.480	0.30	562.821

The values of the Doppler temperature transferred through the coupling are lower than the Doppler temperature calculated at this iteration by COBRA-TF. The value transferred is interpolated to the middle point of the thermal-hydraulic iteration.

Table 6.3 shows another characteristic of the CN1. Most works reviewed discretize the value transferred as the average value between the earlier and the current iterations (Tyliszczak 2014), what can be written as:

$$T_{Transfer}^{n+1/2} \approx \frac{T^{n+1} + T^n}{2} \quad (6-52)$$

Eq. 1.1 can be applied just to the average values. If the values of each fuel assembly are analyzed in an isolate way, Eq. 1.1 does not agree with the results.

Doppler temperature transfer at Table 6.3 at 0.3 seconds correspond with the average value between the Doppler temperature calculated at 0.295 and 0.305 seconds.

The performance of the coupling is reflected in the values of the temperatures achieved by the materials in the core. The fuel centerline temperature is different according to which time-step is employed. The differences, in this case, are more significant than with the OS method, reaching around 30K.

In less than one second, the fuel centerline temperature at the hottest fuel assembly (assembly43) achieves the 850K, in the case with 0.01 seconds time-step.

The CN1 shows a high sensitivity to the time step, and the response of the fuel changes significantly with the time-step.

The differences among the temperatures in the hottest axial nodes are about 30 K. These differences are more significant than the OS.

Figure 6.14 shows the axial fuel temperature distribution of the hottest fuel assembly.

Figure 6.14 Effect exerted by time-step size on the fuel centerline temperature of the hottest fuel assembly employing CN1 method

6.4.3 Reactivity transient with the second order Crank-Nicolson method

CN2 method is similar to CN1 one but employs a second order method to discretize the derivative. The CN2 is more complex to implement within the coupling scheme than the CN1 one.

CN2 shows the same sensitivity of the peak power to the coupling scheme than CN1. The changes in the peak power are less pronounced than the CN1.

Figure 6.15 Power and Doppler temperature evolution using second order Crank-Nicolson method

Figure 6.15 shows the power and Doppler temperature. The power shows a significant dependence of the time step, more than in the OS and less than in the CN1.

Table 6.4 Comparison of the peak power values

Time-step (s)	0.01	0.005	0.0025
Peak Power (MW), OS	7097	6663	6332
Peak Power (MW), CN1	10174	7885	6620
Peak Power (MW), CN2	8383	7214	6345

Table 6.4 shows the peak power of the two previous coupling schemes. The CN2 method converges with the OS when the time step is low. The second order is appreciated in the value of the peak power. It is lower than the peak power of the CN1.

The comparison of the two methods with the same time step is represented in Figure 6.16. The Power with the CN1 is higher than the power with CN2. To Change the coupling scheme from CN1 to CN2 reduces the power.

Figure 6.16 Comparison of the power and Doppler temperature evolution in the transient with the same time step, 0.005 seconds

The performance of CN2 against the time step size is the same than the explained in sections 6.4.2. The difference is the second order discretization. A second order accuracy method reduces the sensitivity of the coupling method to the time step in comparison with CN1.

The fuel centerline temperature of the hottest fuel rod with three different time-steps employing CN2 is drawn in Figure 6.17.

Figure 6.17 Effect exerted by time-step size on the fuel centerline temperature of the hottest fuel assembly employing CN2

Using CN2 method, the total power is lower compared to the total power of CN1 and hence, the fuel temperature is lower too. The differences among the three fuel centerlines temperatures are not so significant than in Figure 6.14.

Differences between the calculated values and the transferred values of the Doppler temperature are shown in Table 6.5. The differences between the two values are lower than the differences of the same values when a first-order method is applied. Although Crank-Nicolson has a large sensitivity to the time-step, to increase the accuracy order reduces the sensitivity of the maximum power to the coupling.

Table 6.5 Differences between Doppler temperatures calculate and transfer through the feedback in a CN2 coupling

Time COBRA-TF	Avg Doppler temp calculated by COBRA-TF	Time Coupling	Value transferred Avg Doppler temp
0.295	561.448	0.290	561.305
0.305	562.358	0.300	562.131
0.315	563.712	0.310	563.373

The difference of the average Doppler temperature calculated and exchanged through the coupling is 0.227 K while the difference at the same time between the calculated and exchanged value employing CN1 is 0.399K. The difference between the value calculated and the value exchange is lower in the CN1.

Comparison of the reactivity and the Doppler temperature values along the transient simulation using the three methods analyzed, OS, CN1, and CN2, are shown in Figure 6.18. The neutronic calculation at CN1 and CN2 obtains lower Doppler temperature than the value calculated by the last thermal-hydraulic calculation, and the maximum power is consequently higher.

Figure 6.18 Reactivity and the Doppler temperature values along the transient using CN2, CN1 and OS methods

6.4.4 Reactivity transient with the Extrapolated Method

The coupling schemes analyzed in the earlier sections show the high sensitivity of the power evolution to the coupling scheme and the time-step size. The fuel centerline temperature changes with these two variables in the cases analyzed previously.

As Figure 6.8, Figure 6.12 and Figure 6.15 show, large time-step sizes increase the power peak. The transient calculations with OS, CN1 and CN2 methods require low time steps. For cases employing an accurate radial grid, as PbP and subchannel calculations, the time step reduction increases the computational requirement. In order to decrease the computational effort, it would be interesting to implement a coupling scheme with low sensitivity to the time step.

The last coupling scheme implemented between COBAYA4 and COBRA-TF is the extrapolated method (EM), described in section 6.3.2.2. The extrapolation is just performed in the thermal-hydraulic variables. The changes in the neutronic flux are fast enough and do not require the extrapolation.

The EM scheme employees the current and the earlier calculation of the thermal-hydraulic code to exchange an extrapolated value of the thermal-hydraulic parameters half time step until the endpoint of the neutronic calculation (Figure 6.21). The neutronic code updates the cross sections libraries with the thermal-hydraulic variables referred to the end point of the neutronic iterations.

The EM works as a predictor-corrector method and estimates the temperatures and densities at the endpoint and correct the power and reactivity evolution. As larger the time-step, larger will be the extrapolation diming the sensitivity of the coupling method to the time-step size. With the staggered mesh, the thermal-hydraulic code reads the power referred to the middle point of the thermal-hydraulic iteration, instead the power referred to the starting point of the iteration.

Three calculations have been carried out employing the same time step than in the earlier cases, 0.01, 0.005, and 0.0025 seconds. Figure 6.19 shows the evolution of the total power for the simulations. The three graphics are similar. The value of the peak power in the three curves is close to 6000MW. In this coupling, reducing the time step does not mean a reduction of the power peak as it is the case with the time step 0.005 seconds. In the previous implementation of the extrapolated method, the same performance was appreciated (J. Jimenez 2010)

Figure 6.19 Power evolution with the extrapolated method

The three curves of Figure 6.19 are similar and sensitivity to time-step size is despicable. The effect exerted by the time step is appreciated at the moment when the peak power takes place. Reduction of the time step moves forward in time the peak power.

As in the previous cases, the value of the power peaks obtained with the three time-steps are included in Table 6.6. The extrapolated method is more stable allowing to work with larger time steps than the three other cases. The value of the total power is lower regardless the time step employed in the calculations.

Table 6.6 Comparison of the power peak values

Time-step (s)	0.01	0.005	0.0025
Peak Power (MW), OS	7097	6663	6332
Peak Power (MW), CN1	10174	7885	6620
Peak Power (MW), CN2	8383	7214	6345
Peak Power (MW) EM	6084	6157	5860

Figure 6.21 shows a zoom of the region of the peak power of Figure 6.19. The figure shows the three curves with the power evolution and six more with the value of the Doppler temperature (three with the value calculated and three with the value extrapolated). The differences between the calculated and the extrapolated Doppler temperatures are more significant with a large time-step. Reduced the length of the time step reduces the difference between the two values and made useless the extrapolation. This coupled scheme was developed to run with large time-steps

Figure 6.20 Power and average Doppler temperature evolution with EM zoom in the central region

The more significant deviations between the extrapolated values and the calculated ones come out for a larger time step, as the case of 0.01 seconds in Figure 6.20.

Figure 6.21 describes the performance of the coupling and the reason why the extrapolation will be larger with longer time-steps.

Figure 6.21 Extrapolation with a different time step

Figure 6.21 can be explained as in the previous sections as follows. The thermal-hydraulic fields are extrapolated half time step, i.e. COBRA-TF calculates the value of the temperatures and densities at point $n+1/2$, $TH_{\Delta t=0.01s}(n+1/2)$, but COBAYA4 reads $TH_{\Delta t=0.01s}(n+1)$.

$$TH\left(n+\frac{1}{2}\right)=CTF\left[TH\left(n-\frac{1}{2}\right),POW(n),t\right] \quad (6-53)$$

$$POW(n+1)=COB4[POW(n),TH_{\Delta t=0.01s}(n),t] \quad (6-54)$$

If the time step is reduced to the half, there is a new iteration and the middle point (Figure 6.21). Halving the time lapse the last iteration can be described as:

$$TH\left(n + \frac{3}{4}\right) = CTF\left[TH\left(n + \frac{1}{4}\right), POW\left(n + \frac{1}{2}\right), t\right] \quad (6-55)$$

$$POW(n + 1) = COB4\left[POW\left(n + \frac{1}{2}\right), TH_{\Delta t=0.005s}(n + 1), t\right] \quad (6-56)$$

As the extrapolations are more significant with long time steps:

$$TH_{\Delta t=0.005s}(n + 1) < TH_{\Delta t=0.01s}(n) \quad (6-57)$$

COBAYA4 update the cross sections with higher temperature when the time step is 0.01 seconds than 0.005 seconds. The higher temperature is translated to lower reactivity and lower value of the peak power. The power excursion is damped allowing to employ a longer time step.

Figure 6.22 shows the fuel centerline temperature of the hottest fuel assembly (assembly 53). The three curves are similar, the differences appear at the central region of the fuel assembly, but these differences are lower than 5°C. The response of the thermal-hydraulic variables is comparable to the response of the total power.

Figure 6.22 Effect exerted by time-step size on fuel centerline temperature on the hottest fuel assembly by the EM

6.4.5 Comparison of results

One of the main differences between the OS method and the other three coupling schemes is the number of thermal-hydraulic iterations calculated. COBRA-TF carries out just one iteration when it is coupled with the OS while, the same calculations with CN1, CN2, and EM take more than one iteration. The staggered mesh of the CN1, CN2, and EM made COBRA-TF performs more calculations, increasing the time to calculate the transient.

With the same time step, each of the couplings scheme generate different evolution of the transient in the case of the rod ejection. Figure 6.23 shows the evolutions of the transient with the same time step.

Figure 6.23 Comparison of the Four coupling schemes with time step 0.005 seconds

The EM does not show any of the problems that the other three coupling exhibit when it is employed to study transients with larges time steps. EM does not reveal sensitivity to the time step. The solutions with 0.01 and 0.00025 seconds are similar in the case of the rod ejection event.

The EM is the best choice to run a neutronic transient with a large time step. Figure 6.24 shows the time consumption of the four coupling schemes. CN2 is the slowest coupling

scheme while OS is the faster. In this case. The best choice is the EM because it allows using long time step.

Figure 6.24 Time to run a rod ejection transient with the time step of 0.01 seconds

6.5 MSLB transient

The MSLB transient described in section 5.3.1 has been used to evaluate the performance of the coupling schemes implemented. Just to recall, the reference reactor is a four loop Westinghouse Zion PWR using two types of fuel assemblies, MOX and UOX, each one with two different initial enrichment. Initial state is a Hot Zero Power (HZP) from a subcritical state corresponding to a 1% of shutting down margin (SDM) with all rods inserted and the highest worth control rod stuck in the overcooled area (Kliem et al. 2017). As a result of the reactivity increase, a power excursion is observed that will be analyzed using the different coupling schemes.

6.5.1 Comparison of results obtained with the different coupling schemes

The same coupling models employed in section 6.2 have been tested in this case. There are many differences between the MSLB and the Rod Ejection event. But one of the most important one is the temporal scale of the transient. The rod ejection takes just one second meanwhile the MSLB takes about 60 seconds to occur.

The changes in the neutronic and thermal-hydraulic variables happen in a more extensive time scale. These features about the length of the transient and the length of the time-step have a significant effect on the transient evolution for the four coupling schemes. It is important to remark that the thermal-hydraulic boundary conditions are set in the COBRA-TF input; i.e. the inlet temperature, the mass flow and the outlet pressure. The values are established for each second.

The calculations have been performed employing the same time steps than in the rod ejection transient; 0.01, 0.005, and 0,0025 seconds. As Figure 6.25 shows, there are not significant differences among the four coupling schemes in the transient. The evolution of the power and the Doppler temperature do not show differences among the four coupling schemes.

Figure 6.25 Power and average Doppler temperature with the four coupling models implemented employing the same time step

Figure 6.26 shows the sensitivity of the total power and reactivity to the time step using the OS and the EM schemes. The CN methods (CN1 and CN2) have been eliminated because the time consumption is considerably greater and does not show any improvement over the OS and EM (Figure 6.24).

The evolution of the power and the reactivity does not show appreciable differences between both coupling schemes and using two different time-steps. The reduction of the time-step delays the power excursion temporal point but just in less than one second.

Figure 6.26 Power and reactivity with the Extrapolated method and the Operator Splitting

The differences in the power evolution with the time-step length at Figure 6.26 generate variations in the axial fuel centerline temperature as in the case of the rod ejection transient. These differences are not so meaningful than in the previously analyzed transient.

Figure.6.27 shows the axial distribution of the fuel centerline temperature. The differences observed using the same coupling scheme but with different time step is around 5K. The performance of the fuel centerline evolutions is quite different in the MSLB transient than in the rod ejection event.

Figure.6.27 Fuel centerline temperature hottest fuel assembly at the end of the transient with OS and Extrapolated model

The evolution of the MSLB transient shows less sensitivity to the time step and the coupling scheme than the first case analyzed in section 6.2. The differences among the four cases are not significant.

The explanation to this behavior can be found in the thermal-hydraulic BC. COBRA-TF employs a forcing function to introduce the BC evolution to the transient calculations. The forcing function employed changes the inlet temperature and the inlet mass for each fuel assembly and the outlet pressure for the whole core each second. COBRA-TF interpolates the value of the BC between each interval of second (Avramova and Salko 2016a).

As the variation of the BC is slow, the interpolation performed does not have significant effect when the time-step size is several orders of magnitude lower than the time required to obtain a real variation in the boundary conditions.

The evolution of the power is totally different. Figure 6.28 shows the differences between the extrapolated values of the moderator temperature. Both curves are the same.

Figure 6.28 Evolution of the average moderator temperature calculated by CTF and the extrapolate moderator temperature extrapolated

The extrapolation does not exert any effect if the thermal-hydraulic variables do not change significantly on each iteration.

These analyses of the fuel centerline temperature depict how the temperature is lower with the Extrapolated Method than with the OS although the maximum differences between the two methods with different time-step are the same in both cases, 5 K, and the EM does not reduce the sensitivity to the time-step length.

Although the difference obtained is not large, the performance of the two models is the same. The analysis of the thermal-hydraulic transient shows how the EM method does not exert any effect on the transient evolution because the temperatures and densities change slowly, and the interpolation or extrapolation does not exert any effect.

In this case, the best coupling scheme to employ is the OS due to the fact that it is easier and faster than the enhanced methods, as Figure 6.24 shows.

6.6 Numerical convergence Rates

A useful analysis that can be performed also with the coupling schemes implemented between COBAYA4 and COBRA-TF within NURESIM European Platform is the order of accuracy obtained with each of them. The four coupling methods implemented in this work have different accuracy order. The first order methods are:

- Operator splitting
- Crank-Nicolson 1
- Extrapolated

While the second order method is:

- Crank-Nicolson 2

It is important to analyze if the coupling scheme keeps the accuracy order that they have. Some works reviewed include also this type of analysis as (Mahadevan and Ragusa 2006) (Mousseau, Zhang, and Zhao 2008).

The error between the value calculated, $\widetilde{u}_h$, and the real value, u , can be represented as a function of the temporal grid side. If the numerical method is of order p the accuracy order is the value of the exponent p . There is a value of C that:

$$E = |\widetilde{u}_h - u| \leq Ch^p \quad (6-58)$$

The error is error depends of the value of h . An increase in h will increase the error. The accuracy order is the value of the exponent p (Ferziger and Perić 2002).

Taking logarithms in the equations allows to convert the exponent into a multiplication:

$$\begin{aligned} E &= |\widetilde{u}_h - u| \leq Ch^p \\ \log(E) &= \log(|\widetilde{u}_h - u|) \leq \log(Ch^p) \\ \log(|\widetilde{u}_h - u|) &= p \times \log(h) + \log|C| + O(h^{n+1}) \end{aligned} \quad (6-59)$$

The reference value, u , to calculate the error should be a real value of the experiment. The analysis of the accuracy order will be performed with the rod ejection accident. There are not experimental data of the transients. Other choice can be computing a transient with small h and employ the results as a reference solution. A small h will be the solution used to calculate the order of accuracy.

The widest method to estimate the order of accuracy is to plot a sequences or errors for different values of h . The slope of the curve is the accuracy order of the numerical method (Runborg 2012). Once they are calculated, there are a pairs of errors and h : $((E_1 = |\widetilde{u}_{h_1} - u|, h_1); (E_2 = |\widetilde{u}_{h_2} - u|, h_2); \dots; (E_n = |\widetilde{u}_{h_n} - u|, h_n))$.

The error will be estimated as:

$$E = |X_h - X_{0.0025}| \quad (6-60)$$

The reference solution for each coupling scheme will be the solution with $h = 0.0025s$. The analysis in the case of the rod ejection will be performed for two variables, the total power and the average Doppler temperature.

In section 6.2, the best coupling method is the extrapolated. The extrapolation supplies stability because convert the coupling in an implicit system. CN1 and CN2 does not improve the solution of the transient.

Figure 6.29 Order of accuracy extrapolated and OS coupling schemes of the total power

Figure 6.29 stands for the order of accuracy of the extrapolate method and the operator splitting method. For a small tie steps, the slope, and the order of accuracy, is more than 1. Once the time step is more than 0.01 seconds the slope changes to 1. This has sense because, as was observed in section 6.4.1, OS shows a strong sensitivity to time step. OS works better with smalls time steps.

Extrapolate method shows a slope closer to 2, being even more similar when using a larger time step. The EM is more stable with larger time-steps, as expected. It allows to carry out calculations employing larger time-steps.

The same analysis has been performed with the two Crank-Nicolson schemes implemented.

Figure 6.30 Order of accuracy of the CN first and second order schemes

Figure 6.30 shows the accuracy order of the two Crank-Nicolson schemes implemented. The figure shows how the performance of the two methods is better for a small time-steps. When time-step is greater than 0.01 second the coupling schemes have problems to keep the accuracy order.

A zoom in Figure 6.30 shows how the CN2 scheme matches better with the line with slope 2 and the CN1 with the slope 1. The curves of the two couplings are between the two lines with slope 1 and 2.

The CN scheme requires a small time-step to keep the accuracy order. Although, in the CN2 the second order has been kept, both CN are semi-implicit. They transfer a value at the beginning of the iterations.

Figure 6.31 Zoom order of accuracy Crank-Nicolson

The EM is the most stable coupling scheme from the four compares here. EM keeps its order of accuracy even for considerable large time-steps. EM is, therefore, a semi-implicit method. OS is a semi-implicit method too in the neutronic solver but not in the thermal-hydraulic one. CN1 and CN2 are explicit methods and they show a strong sensitivity to time step.

EM can work even using large time-steps. Figure 6.32 displays the evolution of the MSLB employing such a time-step. Not all the coupling schemes are able to achieve the end of the transient as is the case for CN1.

Figure 6.32 shows the evolution of the MSLB employing large time-steps with the four coupling schemes. At peak power that appears at the beginning of the power excursion. The value of the peak power is a function of the coupling scheme. The peak power of the EM is the smallest one, as in the cases analyzed into section 6.2.

Figure 6.32 MSLB employing a long-time step

6.7 Results

Among the four coupling schemes implemented and analyzed, the OS and the extrapolated methods are the more interesting. The extrapolated scheme has less sensitivity to the time step allowing to perform a calculation with a larger time step. The OS is the easiest coupling scheme to implement and faster.

The two Crank-Nicolson schemes show a strong sensitivity to the time step and, to carry out a calculation with these two coupling schemes requires to employ small time-steps. Moreover, Figure 6.24 shows how these two methods are slower than the OS and EM.

The analysis of the rod ejection shows how the EM is the better choice as a coupling scheme. EM allows to employ a large time step without significant modifications in the transient evolution. At the MSLB, the better coupling scheme is the OS. The EM does not exert any effect on the transient evolution and requires more time to finish the calculation.

Chapter 7 Conclusions

The work described in this document has been developed along the NURESAFE European Project but also as a continuation of the project applying the capabilities of the codes and platform for further studies.

During NURESAFE project COBAYA3 was updated to COBAYA4 and integrated with all its capabilities into NURESIM platform. Update and full integration of COBAYA was a challenging work performed by the NURESAFE team of the Universidad Politécnica de Madrid that supposed a notable change in the model of coupling. COBAYA3 was previously coupled with COBRA-TF and COBRA-III through a tight coupling. The new external coupling developed through SALOME supplied flexibility for the definition of spatial and temporal meshes.

In order to verify and validate the code, several comparisons with reference solutions for different cases were carried out although not included in the documents as they are not considered of interest. But one interesting comparison was the one performed with the coupled system MCNP6/CTF for a fuel assembly considering detailed definition of the fuel pins for the MonteCarlo analysis and Pin-by-Pin and nodal resolution for COBAYA4. A systematic study of the results obtained at 2D, 3D HZP conditions and 3D HFP ones was performed.

The results obtained for this analysis demonstrated, first, that at Cold Zero Power the results obtained with both systems are very close each other with differences of less than 24 pcm that are originated by the different approximation for the neutronic solver. Secondly, that when using a correct model for Doppler fuel temperature (see section 4.8.3) the differences become 344 pcm, difference that can be explained by the feedback effects that the thermal-hydraulic fields impose over different solvers. Nevertheless, it has been shown that the differences in the fuel temperature and axial power profile are almost negligible using both systems it is of importance from the safety point of view.

Regarding spatial discretization, the different codes integrated into NURESIM platform employ different geometrical nodalization. SALOME, the computing platform that is the base for NURESIM platform, is able to interpolate the fields to be exchanged among the meshes created by the codes. This capability allows to carry out calculations employing different accuracy level in the geometrical discretization.

COBAYA4 and COBRA-TF are able to perform calculation with a fine radial mesh, PbP for the NK and subchannels for the thermal-hydraulic. The interpolation of SALOME enables to combine different accuracy levels in each of the physics. It is

possible to perform a PbP neutronic calculation using lumped thermal-hydraulic channels. The effect exerted by the combination of different resolution in the radial grid has been analyzed.

The combination of different meshes can be done considering the computational requirements, but it is important to take into account the effect exerted by the interpolation among meshes on the solution.

As first step, a fuel assembly was simulated to test the capabilities of the system to run full power analysis but considering small domains. Although initially reflective boundary conditions were imposed to the assembly, new calculations were performed introducing vacuum neutronic condition at one side of the assembly with the intention of generate strong gradients in the cases analyzed.

If there is a perturbation in the core, as a rod ejection, the combination of PbP/lumped-channel meshes could lead to overstating the thermal-hydraulic feedback. COBAYA4 reads the average value of the temperatures and densities. These values are lower than the sub-channel values in the hottest region and vice-versa in the coldest sector. This fact leads, in the case of using PbP/lumped channel modelization to more step power profiles than the PbP/subchannel calculations.

In order to analyze the effects of the radial coupling, a coupled full core PbP/lumped channel calculation was performed. The goal of the calculations is to prove the capability of COBAYA4 to conduct a calculation with an accurate radial grid. The result was compared with the same calculation but employing one node per fuel assembly.

Regarding temporal coupling, once the integration is done, the implementation of different coupling schemes is easier than using an internal coupling although is limited to some explicit or semi-implicit algorithms.

Another characteristic of this coupling through the SALOME platform is that the control of the codes is performed employing a Python script. In order to integrate COBAYA4 into NURESIM platform, it had to be rewritten into a modular way. Then the Python script is able to call the different functions of the codes to initialize, perform steady-state time step, perform transient time step, write data, etc. The script includes the functions that exchange the information from one mesh to another. The information can be modified before being stored into the target mesh. Examples are the possibility to employ damping to improve the stability in the steady state calculations or capacity to apply different coupling algorithm as the ones described into Chapter 6.

Four coupling algorithms have been implemented in the Python script between COBAYA4 and COBRA-TF. The four coupling models have been tested with two transients, a rod ejection event and a MSLB scenario. The differences between the two transients allow to study the performance of the different schemes regarding each transient. The results state how the effect exerted by the coupling schemes are different according to the transient.

Section 6.4 shows the analysis of the rod ejection with the four coupling schemes. Among the four coupling schemes, the Extrapolated Method is the most stable. The Extrapolate Method show less sensitivity to the time step size. The stability can be appreciated in the fuel centerline temperature. The transient changes considerable employing the three other coupling schemes. The CN schemes and the OS requires smalls time steps to reproduce an accurate transient.

Section 6.5 includes the analysis with the four coupling schemes in a thermal-hydraulic transient. The effect exerted by the coupling in a MSLB is different than the rod ejection. The results show the same evolution of the transient in the four cases. The impact of the coupling scheme on the transient evolution is negligible.

Considering the result of the two transients, several recommendations can be done. The time consumed by the calculation increases with the degree of complexity of the coupling scheme. More complex coupling scheme does not mean better performance of the calculations. Taking into account the last two observations:

- In a rod ejection event (NK transient), the best coupling scheme among the four analyzed in this thesis is the extrapolated method. It is the most stable and allows to work with large time steps.
- In a MSLB scenario (TH transient) the better choice is OS. The four couplings reproduce similar transient. OS is the easiest coupling scheme to implement and the one that fewer resources consumed.

The accuracy order and stability play an important role in transient calculations. Increase of the time step reduces the memory requirements but could lead to convergence problems. The memory requirements are more significant in a PbP/subchannel calculation. The study of the accuracy order of the four methods displays how the accuracy order of the CN1 and CN2 degrade for a large time step. Extrapolate has a slope of two matching with the second order and keeps the accuracy order with larger time step than CN2. OS shows better performance to larger time steps than the CN1 and CN2.

The better coupling model for a fast transient is the Extrapolated Method, and for a thermal-hydraulic the Operator Splitting Method.

Chapter 8 Future Work

8.1 Effective fuel temperature

The thermal-hydraulic codes as COBRA-TF solve the temperature profile within the fuel pellets. The neutron diffusion codes as COBAYA4 does not have well define geometry. In a PbP calculation the nodalization includes the fuel rod and the sub-channels around it, Figure 2.10. The difference in the fuel rod nodalization between the two codes force to find a solution.

The solution is to calculate from the thermo-mechanic calculation one value of the fuel temperature for the neutronic diffusion code. This value is the Effective fuel temperature. The effective fuel temperature tries to represent the most significant neutronic phenomenon in the fuel pellet.

One of the most significance nuclear trend within the fuel pellet is the self-shielding. The energetic self-shielding of the uranium 238 produces the spatial self-shielding within the fuel pellet. 90% of the fuel temperature coefficient worth is due to the $^{238}\text{U}(n, \gamma)$ broadening. This effect reduces the neutron flux in the inner part of the pellet.

The Doppler effect comes mostly from the outer regions of the fuel pellet. 60% of the Doppler effect comes from the first 800 μm of the pellet. Figure 8.1 shows the distribution of the captures within the fuel pellet.

It is important to establish a good effective fuel temperature in the feedback to reproduce the effect inside the pellet with just one value.

The value of the effective fuel temperature has to take into account the effect of the fuel reactivity temperature coefficient. (Santamarina and Bernard 2015).

Figure 8.1 Neutron capture within the fuel pellet (Santamarina and Bernard 2015)

There are different ways to obtain the effective fuel temperature (Kruijf 1994):

1. The effective fuel temperature as a weight of the fuel temperature of each region of the fuel pellet, $T_{F,EFF} = \sum_i w_i T_i$.
2. The effective fuel temperature as volume average fuel temperature. In this case, the weight of each temperature is obtained as a relation between the volume of the ring and the volume of the pellet.
3. The effective fuel temperature as a relation between the centerline fuel temperature and the surface fuel temperature, $T_{F,EFF} = T_s + 0.3(T_c - T_s)$.

Among the three models, the most widely used in the neutronic/thermal-hydraulic calculations are the effective third. The effective fuel temperature is computed as relation between the fuel centerline temperature and the outer surface temperature (Grandi et al. 2010):

$$T_{Doppler} = 0.7 \times T_{surface} + 0.3 \times T_{centerline}$$

The different ways to obtain the Doppler temperature depend on the energy range. The volume average effective fuel temperature is valid for high energy range. At these energy levels, the resonances are low, instead of the fuel

temperature as a weight of the temperature of each region valid for the low energy range. (Kruijf 1994)

To use the effective fuel temperature as combinations of the surface fuel temperature and the centerline fuel temperature is very spread, but this formulation may lead to significant errors (Kruijf 1994).

It would be interesting to study the effect of other models of the effective fuel temperature of recode COBRA-TF to deliver different effective fuel temperature according to the energy range of the neutron.

8.2 Time step calculation

The transient calculations carried out in Chapter 5 haven been performed with a fixed time-step. In section 6.6, convergence and accuracy order analyses were performed. The accuracy order was obtained with the equations (6-58).

The length of the time step is h , the accuracy order is the variable in the exponent, p , in E_p . (6-58). The error can be estimated employing a calculation with small time step as a reference solution.

If a new calculation if performed with the same coupling scheme or another, it is possible to set up the right time step to keep the error. This would implicate modifications in the

8.3 Uncertainty propagation at each coupling scheme

4.5 depicts the validation of the new coupling of COBAYA4/COBRA-TF comparing against MCNP6/CTF. COBRA-TF and CTF are different versions of the same code, the basis of the codes is the same. MCNP6 and COBAYA4 are different codes, the method employed by each code to solve the neutronic is different.

Table 4.2 shows the comparison among COBAYA4, KENO and MCNP in a 2D stand-alone calculations. There are differences, but these differences in the case of MCNP and COBAYA are 23 ppm.

Table 4.3 shows the differences between MCNP6 and CTF in a FA stand-alone calculation. The deference's in the K-EFF is 79 pcm.

Moreover, the two versions of the thermal-hydraulic codes, both employ different radial in the FA problem. COBRA-TF employs a channel center fluid mesh. Meanwhile, CTF employs a rod center fluid meshes. Their differences in the fluids meshes do not exert any effect of the thermal-hydraulic calculations (Bennett, Avramova, and Ivanov 2016).

The differences in the k-eff between MCNP6/CTF and COBAYA4/CTF are more significant than the cases at HZP, 2D and 3D. Table 4.4 shows the difference in the K-EFF in a FA HFP calculations. The differences are 344 pcm.

If the differences in the thermal-hydraulic code does not exert important effect on the solution, and the neutronic stand-alone calculations show differences, but around 100 pcm. The coupling could produce the difference of 344 pcm.

The calculations of 4.5 are steady state. In this condition the coupling is the OS. In a transient calculation the coupling scheme could be modify employing the models in 2.5.2 or another.

An uncertainty study in a transient calculation employing different couplings would be interesting, analyzing the uncertainty propagation and, analyzing the part of the “noise” of the solution comes from the coupling.

The solution would be more difficult to analyze because, instead a value, there would be a band on the transient evolution.

8.4 Effect exerted by the

Into section 5, it is possible to see the effect exerted by the coupling scheme into the fuel centerline temperature of the hottest fuel assembly. The Figure 6.14, show how the coupling scheme modifies the fuel centerline temperature.

Another phenomenon that concerns the nuclear safety is DNB. DNB appears in some places on the surface of the fuel rod when the heat transfer regime moves from nucleate boiling to film boiling. The water does not touch the surface of the fuel rod, and its temperature increases, causing damage to the clad.

The DNB is analyzed with a variable DNBR. DNBR is the ratio between the heat flux and the critical heat flux (equation). COBRA-TF employs empirical correlations to calculate the CHF. The thermal-hydraulic code includes two correlations to obtain

the CHF Bias correlation and W-3 correlation. When the DNBR drops below than 1, DNB occurs.

Previous analyses have been performed to study the effect of the CHF correlation. The studies have been carried out simulating a RIA (Kucukboyaci, Cao, and Sung 2014). Each CHF correlation predicts different heat transfer regime, and DNB appears under different conditions.

This work has proved the effect exerted by the coupling scheme on the transient evolution. Keeping on with this idea will be interesting to analyze the impact on the DNB with the two correlations of COBRA-TF.

8.5 Convergence algorithm Neutronic-Thermohydraulic

The section 4.5 describes the validations performed by the team of the Universidad Politécnica de Madrid of OCBAYA4/COBRA-TF against the coupled system MCNP6/CTF from NCSU.

The section describes the algorithm of the coupling from both, COBAYA4/COBRA-TF and MCNP6/COBRA-TF. MCNP6/CTF achieves the convergence once the relative difference of the temperature achieves a value

$$\frac{T_{n+1} - T_n}{T_n} < \epsilon \quad (8-1)$$

In the case of COBAYA4/COBRA-TF the variable analyze is the power. The convergence in the steady state is achieved when

$$\frac{P_{n+1} - P_n}{P} < \epsilon \quad (8-2)$$

An interesting analysis would be the study of the steady-state sensitivity to the convergence criteria. Analyze with COBAYA4/COBRA-TF witch method reduces the number of iterations.

8.6 Implementation of Approximated Block Newton Methods

The ABN method is described in section 2.2.3. This coupling method keeps the solvers as a black box (Mylonakis, Varvayanni, and Catsaros 2017) and could be implemented into the coupling between COBAYA4 and CTF developed through SALOME platform (Chanaron et al. 2013).

There are several types of ABN methods. The most common is ABN-J. The differences between the ABN and ABN-J is that the ABN starts from block Jacobi iteration and the ABN-J starts with the Gauss-Seidel iteration.

Section 2.2.3 explains the ABN coupling scheme. In the case of the OS, the value will be obtained directly from one of the codes. But with the ABN, an iterative process will be employed to calculate the next value of the other code at the end of the iteration. The iterative process employs a Krylov method.

To implement the ABN-J between COBAYA4 and COBRA-TF would require some modifications into the two codes and in the coupling.

The performance will require a nested loops structure. The outer loop in the current that controls the execution from both codes. The inner loop is the Krylov solver (Yeckel, Lun, and Derby 2009). This loop is employed to solve the neutronic calculation to perturb and calculates the delta. This inner loop will be in the python script and will require calls to the two codes to solve the matrix-vector.

Once the inner loop achieves the convergence, the neutronic solution is updated and calculates the thermal-hydraulic solution.

The other modifications would have to be implemented into the codes. The two codes have been developed to executed transient calculations. The current versions run in an OS scheme. They are ready to carry out one iteration, stop and wait until the other code finishes. The ABN method requires several executions to solve the Krylov system. ONCE COBAYA4 and COBRA-TF initiate the transient calculations the time advance. If the transient is a thermal-hydraulic transient like a MSLB, the boundary conditions would be modified at each iteration. In the neutronic calculation, the advance of the time changed the precursors concentrations and could change the transient evolution. The variables as temperatures, velocities and the neutron flux must be stored, before the inner loop. Once the inner loop achieves the convergence, the variables have to be updated to the previous to their values according to the time.

Chapter 9 Bibliography

A. Abarca. 2017. “Desarrollo y Verificación de una Plataforma Multifísica de Altas Prestaciones para Análisis de Seguridad en Ingeniería Nuclear,” 286.

Aragonés, José M., Carol Ahnert, Oscar Cabellos, Nuria García-Herranz, and Vanessa Aragonés-Ahnert. 2004. “Methods and Results for the MSLB NEA Benchmark Using SIMTRAN and RELAP-5.” *Nuclear Technology* 146 (1): 29–40.

Avramova, M. 2007. “Development of an Innovative Spacer Grid Model Utilizing Computational Fluid Dynamics Within a Subchannel Analysis Tool.” PhD Thesis, Pennsylvania State University.

Avramova, M., and R. K. Salko. 2016a. “CTF User Manual.”

Avramova, M., and R.K. Salko. 2016b. “CTF Theory Manual.” <https://doi.org/10.2172/1340446>.

Bennett, A., M. Avramova, and K. Ivanov. 2016. “Coupled MCNP6/CTF Code: Development, Testing, and Application.” *Annals of Nuclear Energy* 96 (October): 1–11. <https://doi.org/10.1016/j.anucene.2016.05.008>.

Castro, E. 2018. “Methodologies for Sensitivity/Uncertainty Analysis Using Reactor Core Simulators with Application to Pressurized Water Reactors.” Universidad Politécnica de Madrid.

Chanaron, B. 2017. “Overview of the NURES SAFE European Project.” *Nuclear Engineering and Design* 321 (September): 1–7. <https://doi.org/10.1016/j.nucengdes.2017.09.001>.

Chanaron, B., C. Ahnert, N. Crouzet, V. Sanchez, N. Kolev, O. Marchand, S. Kliem, and A. Papukchiev. 2015. “Advanced Multi-Physics Simulation for Reactor Safety in the Framework of the NURES SAFE Project.” *Annals of Nuclear Energy* 84 (October): 166–77. <https://doi.org/10.1016/j.anucene.2014.12.013>.

Chanaron, B., C. Ahnert, D. Bestion, M. Zimmermann, D. Cacuci, and N. Crouzet. 2013. “Project Final Report.”

Crouzet, N. 2014. “D41.21_Report on the Update of the Mesh and Field Structures in the Coupled Codes.” NURES SAFE D4.1.2.1. CEA.

———. 2015. “Report on Code Integration in the Platform.” D.4.1.1.1. CEA.

Cuervo, D. 2007. “Análisis Termohidráulico de Núcleo PWR Con Modelización de Flujo Bifásico Para Acoplamiento Con La Neutrónica.” Thesis PhD, Universidad Politécnica de Madrid. http://oa.upm.es/418/1/DIANA_CUERVO_GOMEZ.pdf.

Daeubler, M., N. Trost, J. Jimenez, V. Sanchez, R. Stieglitz, and R. Macian-Juan. 2015. “Static and Transient Pin-by-Pin Simulations of a Full PWR Core with the Extended Coupled Code System DYN SUB.” *Annals of Nuclear Energy*, Multi-Physics Modelling of LWR Static and Transient Behaviour, 84 (October): 31–44. <https://doi.org/10.1016/j.anucene.2014.09.057>.

Deville, E., and F. Perdu. 2012. “Documentation of the Interface for Code Coupling : ICoCo.” STMF/LMES/RT/12-029/A. CEA.

Ferziger, Joel H., and M. Perić. 2002. *Computational Methods for Fluid Dynamics*. 3rd, rev. ed ed. Berlin ; New York: Springer.

García-Herranz, N., D. Cuervo, A. Sabater, G. Rucabado, S. Sánchez-Cervera, and E. Castro. 2017. “Multiscale Neutronics/Thermal-Hydraulics Coupling with COBAYA4 Code for Pin-by-Pin PWR Transient Analysis.” *Nuclear Engineering and Design*, March. <https://doi.org/10.1016/j.nucengdes.2017.03.017>.

Gaston, Derek, Chris Newman, Glen Hansen, and Damien Lebrun-Grandié. 2009. “MOOSE: A Parallel Computational Framework for Coupled Systems of Nonlinear Equations.” *Nuclear Engineering and Design* 239 (10): 1768–78. <https://doi.org/10.1016/j.nucengdes.2009.05.021>.

Gaston, Derek R., Cody J. Permann, John W. Peterson, Andrew E. Slaughter, David Andr?, Yaqi Wang, Michael P. Short, et al. 2015. “Physics-Based Multiscale Coupling for Full Core Nuclear Reactor Simulation.” *Annals of Nuclear Energy* 84 (October): 45–54. <https://doi.org/10.1016/j.anucene.2014.09.060>.

Gill, D F, and Y Y Azmy. 2009. “A JACOBIAN-FREE NEWTON-KRYLOV ITERATIVE SCHEME FOR CRITICALITY CALCULATIONS BASED ON THE NEUTRON DIFFUSION EQUATION,” 15.

Gómez Torres, A.M., Puente Espel, F, del Valle Gallegos, E, François, J.L., Martin-del-Campo, C, and Espinosa-Paredes, G. 2015. “AZTLAN: Mexican Platform for Analysis and Design of Nuclear Reactors.” In .

Gomez-Torres, Armando Miguel, Victor Hugo Sanchez-Espinoza, Kostadin Ivanov, and Rafael Macian-Juan. 2012. “DYN SUB: A High Fidelity Coupled Code System for the Evaluation of Local Safety Parameters – Part II: Comparison of Different Temporal Schemes.” *Annals of Nuclear Energy* 48 (October): 123–29. <https://doi.org/10.1016/j.anucene.2012.05.033>.

- Goorley, T., M. James, T. Booth, and J. Bull. 2013. "Features of MCNP6." In . Paris, France. https://laws.lanl.gov/vhosts/mcnp.lanl.gov/pdf_files/la-ur-13-28114.pdf.
- Grandi, G., K. Smith, Z. Xu, and J. Rhodes. 2010. "Effect of CASMO-5 Cross-Section Data and Doppler Temperature Definition on LWR Reactivity Initiated Accidents," 13.
- Hales, J D, S R Novascone, R L Williamson, D R Gaston, and M R Tonks. n.d. "Solving Nonlinear Solid Mechanics Problems with the Jacobian-Free Newton Krylov Method," 32.
- Herrero, J.J. 2012. "Métodos Avanzados Multiescala Núcleo-Celdas En Geometrías Tridimensionales y Multigrupos Para El Cálculo de Reactores de Agua Ligera." Universidad Politécnica de Madrid.
- Hidalga, Patricio, Agustín Abarca, Rafael Miró, Abdelkrim Sekrhi, and Gumersindo Verdú. 2019. "A Multi-Scale and Multi-Physics Simulation Methodology with the State-of-the-Art Tools for Safety Analysis in Light Water Reactors Applied to a Turbine Trip Scenario (PART I)." *Nuclear Engineering and Design* 350 (August): 195–204. <https://doi.org/10.1016/j.nucengdes.2019.05.008>.
- Ivanov, K., M. Avramova, S. Kamerow, I. Kodeli, E. Sartori, E. Ivanov, and O. Cabellos. 2013. "Benchmarks for Uncertainty Analysis in Modelling (UAM) for the Design, Operation and Safety Analysis of LWRs Volume I: Specification and Support Data for Neutronics Cases (Phase I)." OECD Nuclear Energy Agency.
- Ivanov, K, T Valentine, J-P Hudelot, M Avramova, and U Rohatgi. 2016. "Definition of the Scope of Multi-Physics Applications."
- Jackson, J.W., and N.E. Todreas. 1981. "COBRA-IIIc MIT-2: A Digital Computer Program for Steady State Analysis of Rod Bundle Nuclear Fuel Elements." MIT.
- Jimenez, G., J.J. Herrero, A. Gommlich, S. Kliem, D. Cuervo, and J. Jimenez. 2015. "Boron Dilution Transient Simulation Analyses in a PWR with Neutronics/Thermal-Hydraulics Coupled Codes in the NURISP Project." *Annals of Nuclear Energy* 84 (October): 86–97. <https://doi.org/10.1016/j.anucene.2014.11.002>.
- Jimenez, J. 2010. "Desarrollo e Implementación de La Descomposición En Subdominios Del Acoplamiento Neutrónico-Termohidráulico Mediante Disecciones Alternadas Con Un Sistema de Cálculo Multiescala." PhD Thesis, Universidad Politécnica de Madrid.
- Kane Yee. 1966. "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media." *IEEE Transactions on Antennas and Propagation* 14 (3): 302–7. <https://doi.org/10.1109/TAP.1966.1138693>.

Kliem, S., Y. Kozmenkov, J. Hadek, Y. Perin, F. Fouquet, F. Bernard, A. Sargeni, et al. 2017. "Testing the NURESIM Platform on a PWR Main Steam Line Break Benchmark." *Nuclear Engineering and Design* 321 (September): 8–25. <https://doi.org/10.1016/j.nucengdes.2017.05.028>.

Knoll, D.A., and D.E. Keyes. 2004. "Jacobian-Free Newton–Krylov Methods: A Survey of Approaches and Applications." *Journal of Computational Physics* 193 (2): 357–97. <https://doi.org/10.1016/j.jcp.2003.08.010>.

Kozlowski, T., and T. Downar. 2006. *Pressurised Water Reactor MOX/UO₂ Core Transient Benchmark: Final Report*. Edited by OECD Nuclear Energy Agency. Issy-les-Moulineaux, France: OECD Nuclear Energy Agency.

Kozmenkov, Y., and S. Kliem. 2014. "D12.11 – PWR MSLB Specification Report (NURESAFE Project)."

Kruijf, W. M. J. de. 1994. *Reactor Physics Analysis of the Pin-Cell Doppler Effect in a Thermal Nuclear Reactor*. ECN. http://www.iaea.org/inis/collection/NCLCollectionStore/_Public/27/011/27011189.pdf.

Kucukboyaci, V.N., L. Cao, and Y. Sung. 2014. "COBRA-TF Simulation of Fuel Thermal Response During Reactivity Initiated Accidents Using the NSRR Pulse Irradiation Experiments." In *Volume 6B: Energy*, V06BT07A027. Montreal, Quebec, Canada: ASME. <https://doi.org/10.1115/IMECE2014-38513>.

Lozano, J.A. 2009. "Desarrollo de Un Código Nodal Analítico de Difusión Neutrónica En Multigrupos Para Reactores En Geometría 3D Rectangulares y Hexagonales." Universidad Politécnica de Madrid.

Lozano, J.A., J. Jiménez, N. García-Herranz, and J.M. Aragonés. 2010. "Extension of the Analytic Nodal Diffusion Solver ANDES to Triangular-Z Geometry and Coupling with COBRA-IIIc for Hexagonal Core Analysis." *Annals of Nuclear Energy* 37 (3): 380–88. <https://doi.org/10.1016/j.anucene.2009.12.001>.

Mahadevan, Vijay S., and Jean C. Ragusa. 2006. "Consistent Accelerated Schemes for Nonlinear Coupling Problems in Reactor Analysis." In *Proceedings of the American Nuclear Society's Topical Meeting on Reactor Physics (PHYSOR)*, 10–14.

Martin, R., and C. Frepoli. 2018. *Design-Basis Accident Analysis Methods for Light-Water Nuclear Power Plants*. Vol. Volume 3. Modern Nuclear Energy Analysis Methods, Volume 3. WORLD SCIENTIFIC. <https://doi.org/10.1142/11139>.

Martin, W.R., S. Wilderman, F.B. Brown, and G. Yesilyurt. 2013. "Implementation of On-the-Fly Doppler Broadening in MCNP," 14.

Merino, F. 1993. “Desarrollo de Un Simulador Dinámico Tridimensional Para El Análisis de Transitorios En Núcleos de Reactores de Agua a Presión.” Universidad Politécnica de Madrid.

Mousseau, Vincent A., Hongbin Zhang, and Haihua Zhao. 2008. “Coupled High Fidelity Thermal Hydraulics and Neutronics for Reactor Safety Simulations.” Idaho National Laboratory (INL).

Mylonakis, Antonios G., M. Varvayanni, and N. Catsaros. 2017. “A Newton-Based Jacobian-Free Approach for Neutronic-Monte Carlo/Thermal-Hydraulic Static Coupled Analysis.” *Annals of Nuclear Energy* 110 (December): 709–25. <https://doi.org/10.1016/j.anucene.2017.07.014>.

Ochoa, R., J.J. Herrero, and N. Garcia-Herranz. 2011. “On the Extension of the Analytic Nodal Diffusion Solver Andes to Sodium Fast Reactors.” *Transport Theory and Statistical Physics* 40 (4): 225–42. <https://doi.org/10.1080/00411450.2011.610205>.

Périn, Y. 2016. “Development of a Multi-Physics, Multi-Scale Simulation Tool for LWR Safety Analysis.” Technical University of Munich.

Périn, Y., and K. Velkov. 2017. “CTF/DYN3D Multi-Scale Coupled Simulation of a Rod Ejection Transient on the NURESIM Platform.” *Nuclear Engineering and Technology* 49 (6): 1339–45. <https://doi.org/10.1016/j.net.2017.07.010>.

R. Salko, S Palmtag, and B. Collins. 2016. “CTF Parallel Performance Improvements.” *Transactions of the American Nuclear Society* 115: 4.

Ragusa, Jean C., and Vijay S. Mahadevan. 2009. “Consistent and Accurate Schemes for Coupled Neutronics Thermal-Hydraulics Reactor Analysis.” *Nuclear Engineering and Design* 239 (3): 566–79. <https://doi.org/10.1016/j.nucengdes.2008.11.006>.

Rearden, B., and M. Jessee. 2017. “SCALE Code System (Version 6.2).” ORNL/TM-2005/39,. Oak Ridge National Laboratory. https://www.ornl.gov/sites/default/files/SCALE_6.2.3.pdf.

Rearden, Bradley T., Robert A. Lefebvre, Brandon R. Langley, Adam B. Thompson, and Jordan P. Lefebvre. 2018. “NEAMS Workbench 1.0 Beta.” ORNL/TM--2018/752, 1427614. <https://doi.org/10.2172/1427614>.

Runborg, O. 2012. “Numerical Solution of Differential Equations.” KTH.

Sabater, A. 2014. “Definición y Análisis de Casos Para Uso de Una Herramienta de Análisis de No Regresión En Los Desarrollos de Código Neutrónico COBAYA4.” Master Thesis, UPM.

Sabater, A., D. Cuervo, and S. Sánchez-Cervera. 2016. “Comparison of Subchannel and Averaged Channel Thermal-Hydraulic Descriptions on Coupled Pin-by-Pin Neutronic Calculations.” *Transactions of the American Nuclear Society* 115: 4.

Salome Platform. 2015. “The Open Source Integration Platform for Numerical Simulation.” 2015. <http://www.salomeplatform.org/>.

Sánchez-Cervera, S. 2017. “Generación de Librerías Optimizadas Con Cuantificación de Incertidumbres Para Cálculos Realistas (BEPU) de Reactores Nucleares de Agua Ligera.” PhD Thesis, Universidad Politécnica de Madrid. <https://doi.org/10.20868/UPM.thesis.47154>.

Santamarina, and Bernard. 2015. “Biases and Uncertainties in Doppler Reactivity Worth Calculations.” UAM-9 Workshop, May 20.

Spasov, I., N. Kolev, and D. Cuervo. 2012. “VVER MSLB Benchmark Summary Results.” NURISP D3.1.3.4.

Spasov, I., S. Mitkov, N.P. Kolev, S. Sanchez-Cervera, N. Garcia-Herranz, A. Sabater, D. Cuervo, J. Jimenez, V.H. Sanchez, and L. Vyskocil. 2017. “Best-Estimate Simulation of a VVER MSLB Core Transient Using the NURESIM Platform Codes.” *Nuclear Engineering and Design* 321 (September): 26–37. <https://doi.org/10.1016/j.nucengdes.2017.03.032>.

Toptan, A., and M. Avramova. 2016. “Advances in Fuel Rod Modeling in Subchannel Thermal-Hydraulic Codes.” In *Transactions of the American Nuclear Society*. [zotero://attachment/35/](https://doi.org/10.1016/j.tam.2016.03.035).

Toumi, I., A. Bergeron, D. Gallo, E. Royer, and D. Caruge. 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): 139–55. [https://doi.org/10.1016/S0029-5493\(99\)00332-5](https://doi.org/10.1016/S0029-5493(99)00332-5).

Turinsky, Paul J., and Douglas B. Kothe. 2016. “Modeling and Simulation Challenges Pursued by the Consortium for Advanced Simulation of Light Water Reactors (CASL).” *Journal of Computational Physics* 313 (May): 367–76. <https://doi.org/10.1016/j.jcp.2016.02.043>.

Turner, John A., Kevin Clarno, Matt Sieger, Roscoe Bartlett, Benjamin Collins, Roger Pawlowski, Rodney Schmidt, and Randall Summers. 2016. “The Virtual Environment for Reactor Applications (VERA): Design and Architecture.” *Journal of Computational Physics* 326 (December): 544–68. <https://doi.org/10.1016/j.jcp.2016.09.003>.

Tyliszczak, Artur. 2014. “A High-Order Compact Difference Algorithm for Half-Staggered Grids for Laminar and Turbulent Incompressible Flows.” *Journal of*

Computational Physics 276 (November): 438–67.
<https://doi.org/10.1016/j.jcp.2014.07.043>.

Vazquez, M., H. Tsige-Tamirat, L. Ammirabile, and F. Martin-Fuertes. 2012. “Coupled Neutronics Thermal-Hydraulics Analysis Using Monte Carlo and Sub-Channel Codes.” *Nuclear Engineering and Design* 250 (September): 403–11.
<https://doi.org/10.1016/j.nucengdes.2012.06.007>.

Watson, Justin K., and Kostadin N. Ivanov. 2012. “Demonstration of Implicit Coupling of TRACE/PARCS Using Simplified One-Dimensional Problems.” *Nuclear Technology* 180 (2): 174–90. <https://doi.org/10.13182/NT12-A14632>.

Yeckel, Andrew, Lisa Lun, and Jeffrey J. Derby. 2009. “An Approximate Block Newton Method for Coupled Iterations of Nonlinear Solvers: Theory and Conjugate Heat Transfer Applications.” *Journal of Computational Physics* 228 (23): 8566–88.
<https://doi.org/10.1016/j.jcp.2009.08.003>.

Yilmaz, M. O. 2014. “Development of Burnup Dependent Fuel Rod Model in COBRA-TF.”

Yilmaz, Mine O., Maria N. Avramova, and Jens G.M. Andersen. 2016. “Development, Verification, and Validation of a Fuel Thermal Conductivity Degradation Model in CTF.” *Annals of Nuclear Energy* 97 (November): 246–61.
<https://doi.org/10.1016/j.anucene.2016.07.020>.

Zerkak, O, P Coddington, N Crouzet, E Royer, J Jimenez, and D Cuervo. 2007. “LWR Multi-Physics Developments and Applications Within the Framework of the NURESIM European Project,” 22.

Zerkak, Omar, Tomasz Kozlowski, and Ivan Gajev. 2015. “Review of Multi-Physics Temporal Coupling Methods for Analysis of Nuclear Reactors.” *Annals of Nuclear Energy* 84 (October): 225–33. <https://doi.org/10.1016/j.anucene.2015.01.019>.

Annex A

The calculations are managed with a python script instead than executable. The current annex shows a script employed to launch a transient calculation.

```
import sys
from os import getenv

if (len(sys.argv) != 4):
    print ("ERROR: mismatch of argument [usage: python XXX.py
input_cobaya geom]");
    exit(1);

inputFile_ctf = "deck.inp"
inputFile_cob = sys.argv[1];
geom          = int(sys.argv[2]);
dt_NK         = float(sys.argv[3]);

print ("FILENAME: "+ inputFile_cob)
print ("geom: "+ str(geom))

import sys
import math
import numpy
from time import clock
import matplotlib
#matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
from os.path import basename
from MEDCouplingRemapper import*
from MEDLoader import MEDLoader
import Cobaya4SWIG
COB4=Cobaya4SWIG.Cobaya4()
COB4.setDataFile(inputFile_cob);

import CTFSWIG
CTF=CTFSWIG.CTF()
CTF.initialize();
CTF.genMeshCTF(geom, 193);

#####
#####
##### FUNCTIONS #####
#####
def TH_Solver(CTF):
    conv = False;
    iterCTF = 1;
```



```

minIter = 100;
maxIter = 300;
while iterCTF <= minIter or (iterCTF <= maxIter and conv == False):
    print " [STEADY] Calculating TH iteration: %04d"%(iterCTF),
    iterCTF += 1;
    dt = CTF.computeTimeStep();

    CTF.initTimeStep(dt)
    success=CTF.solveTimeStep();
    if success:
        CTF.validateTimeStep();
        print "OK"
    else:
        CTF.abortTimeStep();
        print "ERROR"

    if iterCTF > minIter:
        conv=CTF.isStationary();

def renormalize_meshes():
    #Normalize the meshes
    cob_size = COB4.getSize();
    ctf_size = [cob_size[0]/100, cob_size[1]/100, cob_size[2]/100]
    factor = [cob_size[0] / ctf_size[0], cob_size[1] / ctf_size[1],
cob_size[2] / ctf_size[2]];
    coord = mesh_ctf_fluid.getCoords()
    coord[:,0] *= factor[0];
    coord[:,1] *= factor[1];
    coord[:,2] *= factor[2];
    mesh_ctf_fluid.setCoords(coord);
    coord = mesh_ctf_power.getCoords()
    coord[:,0] *= factor[0];
    coord[:,1] *= factor[1];
    coord[:,2] *= factor[2];
    mesh_ctf_power.setCoords(coord);
    print "size_cob:", cob_size
    print "size_ctf:", ctf_size
    print "factor:", factor

    cob_center = [364.14-21.42, 21.42, 365.76/2] ;
    ctf_center = [0, 0, 365.76/2 - 18.288/2]
    mesh_cob.translate([-cob_center[0],          -cob_center[1],          -
cob_center[2]]);
    mesh_ctf_fluid.translate([-ctf_center[0],          -ctf_center[1],          -
ctf_center[2]]);
    mesh_ctf_power.translate([-ctf_center[0],          -ctf_center[1],          -
ctf_center[2]]);
    print "center_cob:", cob_center
    print "center_ctf:", ctf_center

    center = [0, 0, 0]
    plane = [0, 0, 1]
    angle = math.pi / 2.0
    mesh_ctf_fluid.rotate(center, plane, angle)
    mesh_ctf_power.rotate(center, plane, angle)
    print "centeb:", center
    print "plane:", plane
    print "angle:", angle

```

```
def NK_to_TH(i, power_pre):
    print " EXCHANGE: %s ->%s"%(cob_ids_out[0], ctf_ids_in[0])
    field_POWER_cob = COB4.getOutputMEDField(cob_ids_out[0]);
    field_POWER_cob.setMesh(mesh_cob);
    if(i > 0):
        power_now = field_POWER_cob.getArray();
        power_now = 0.7 * power_now;
        power_pre = 0.3 * power_pre;
        power = power_now + power_pre
        field_POWER_cob.setArray(power);
    else:
        power = field_POWER_cob.getArray();
    field_POWER_ctf = remapNK_to_TH.transferField(field_POWER_cob, -
55);
    field_POWER_ctf.setName(ctf_ids_in[0])
    field_POWER_ctf.setMesh(mesh_ctf_power);
    print "updating POWER in CTF"
    CTF.setInputMEDField(ctf_ids_in[0], field_POWER_ctf);

    print "before saving the NK fields"
    MEDLoader.WriteFieldUsingAlreadyWrittenMesh(COB4_fileName,
field_POWER_cob)
    MEDLoader.WriteFieldUsingAlreadyWrittenMesh(CTFr_fileName,
field_POWER_ctf)
    return power

def NK_to_TH_tran():
    print " EXCHANGE: %s ->%s"%(cob_ids_out[0], ctf_ids_in[0])
    field_POWER_cob = COB4.getOutputMEDField(cob_ids_out[0]);
    field_POWER_cob.setMesh(mesh_cob);
    field_POWER_ctf = remapNK_to_TH.transferField(field_POWER_cob, -
55);
    field_POWER_ctf.setName(ctf_ids_in[0])
    field_POWER_ctf.setMesh(mesh_ctf_power);
    print "updating POWER in CTF"
    CTF.setInputMEDField(ctf_ids_in[0], field_POWER_ctf);

    print "before saving the NK fields"
    MEDLoader.WriteFieldUsingAlreadyWrittenMesh(COB4_fileName,
field_POWER_cob)
    MEDLoader.WriteFieldUsingAlreadyWrittenMesh(CTFr_fileName,
field_POWER_ctf)
    return

def TH_to_NK():
# Exchange field (tfuel,...)
    for th_id in cob_ids_in:
        print " EXCHANGE: %s ->%s"%(th_id, TRANSLATE[th_id])
        field_TH_COB = None
        field_TH_CTF = CTF.getOutputMEDField(TRANSLATE[th_id]);
        average = field_TH_CTF.getAverageValue();
        print "averageeeee", average
        if (th_id == "rho"):
            values = field_TH_CTF.getArray();
            values /= 1000;
            rho = values;
```

```

        field_TH_CTF.setArray(values);
        field_TH_COB =
remapTH_to_NK2.transferField(field_TH_CTF, average);

MEDLoader.WriteFieldUsingAlreadyWrittenMesh(CTFf_fileName,field_TH_CTF
)

        elif (th_id == "tfuel"):
            tfuel = field_TH_CTF.getArray();
            field_TH_COB =
remapTH_to_NK1.transferField(field_TH_CTF, average);
            print "before saving the TH fields"

MEDLoader.WriteFieldUsingAlreadyWrittenMesh(CTFr_fileName,field_TH_CTF
)

        else:
            tmod = field_TH_CTF.getArray();
            field_TH_COB =
remapTH_to_NK2.transferField(field_TH_CTF, average);
            print "before saving the TH fields"

MEDLoader.WriteFieldUsingAlreadyWrittenMesh(CTFf_fileName,field_TH_CTF
)

        COB4.setInputMEDField(th_id, field_TH_COB)

        MEDLoader.WriteFieldUsingAlreadyWrittenMesh(COB4_fileName,field_
TH_COB)

def TH_to_NK_tran(actual_time_NK, dt_NK):
# Exchange field (tfuel,...)
    for th_id in cob_ids_in:
        print " EXCHANGE: %s ->%s"%(th_id, TRANSLATE[th_id])
        field_TH_COB = None
        field_TH_CTF = CTF.getOutputMEDField(TRANSLATE[th_id]);
        average = field_TH_CTF.getAverageValue();

        if (th_id == "rho"):
            values = field_TH_CTF.getArray();
            values /= 1000;
            avg_rho = field_TH_CTF.getAverageValue();
#            min_rho = min(values);
#            max_rho = max(values);
            print 'average rho:',avg_rho;
            #print 'minimun density:', min_rho;
            #print 'maximun density:', max_rho;
            field_TH_CTF.setArray(values);
            field_TH_COB =
remapTH_to_NK2.transferField(field_TH_CTF, average);

MEDLoader.WriteFieldUsingAlreadyWrittenMesh(CTFf_fileName,field_TH_CTF
)

        elif (th_id == "tfuel"):

```

```

        field_TH_COB =
remapTH_to_NK1.transferField(field_TH_CTF, average);
        values = field_TH_CTF.getArray();
#       min_tfuel = min(values);
#       max_tfuel = max(values);
        avg_tfuel = field_TH_CTF.getAverageValue();
        print actual_time_NK+ dt_NK , 'average tfuel:', avg_tfuel
        #print 'minimun tfuel:', min_tfuel;
        #print 'maximun tfuel:', max_tfuel;
        #print "before saving the TH fields"

MEDLoader.WriteFieldUsingAlreadyWrittenMesh(CTFr_fileName,field_TH_CTF
)

        else:
            field_TH_COB =
remapTH_to_NK2.transferField(field_TH_CTF, average);
            values = field_TH_CTF.getArray();
#           min_tmod = min(values);
#           max_tmod = max(values);
            avg_tmod = field_TH_CTF.getAverageValue();
            print actual_time_NK+ dt_NK , 'average tmod:', avg_tmod;
            #print 'maximun tmod:', max_tfuel;
            #print "before saving the TH fields"

MEDLoader.WriteFieldUsingAlreadyWrittenMesh(CTFf_fileName,field_TH_CTF
)

        print "updating TH field in COBAYA4"
        COB4.setInputMEDField(th_id, field_TH_COB)

MEDLoader.WriteFieldUsingAlreadyWrittenMesh(COB4_fileName,field_TH_COB
)

#####
#####
#####      MAIN      #####
#####
#####
start = clock()
print "Test Program of Cobaya4 component"

# Get the meshes
mesh_cob      = COB4.getMEDMesh();
mesh_ctf_fluid = CTF.getFluidMeshCTF();
mesh_ctf_power = CTF.getRodMeshCTF();
if (geom == 0) :
    mesh_ctf_inlet = CTF.getInletMeshCTF();
    mesh_ctf_outlet= CTF.getOutletMeshCTF();

COB4_fileName = basename(inputFile_cob) + "_COB4_mesh.med"
CTFr_fileName = basename(inputFile_cob) + "_CTFr_mesh.med"
CTFf_fileName = basename(inputFile_cob) + "_CTFf_mesh.med"
if (geom == 0) :
    CTFinlet_fileName = basename(inputFile_cob) + "_CTFinlet_mesh.med"

```

```

        CTFoutlet_fileName      =      basename(inputFile_cob)      +
        "_CTFoutlet_mesh.med"

print "center: ", COB4.getCenter();
print "size: ", COB4.getSize();

#
renormalize_mesheres()

# Saving the meshes
print ("Saving Meshes...");
MEDLoader.WriteUMesh(COB4_fileName, mesh_cob, True);
MEDLoader.WriteUMesh(CTFr_fileName, mesh_ctf_power, True);
MEDLoader.WriteUMesh(CTFf_fileName, mesh_ctf_fluid, True);
if (geom == 0) :
    MEDLoader.WriteUMesh(CTFinlet_fileName, mesh_ctf_inlet, True);
    MEDLoader.WriteUMesh(CTFoutlet_fileName, mesh_ctf_outlet, True);
print ("Saving Meshes... OK");

# Interpolating the meshes
print ("Doing remapper...");
remapNK_to_TH = MEDCouplingRemapper();
remapNK_to_TH.prepare(mesh_cob, mesh_ctf_power, "P0P0");
remapTH_to_NK1 = MEDCouplingRemapper();
remapTH_to_NK1.prepare(mesh_ctf_power, mesh_cob, "P0P0");
remapTH_to_NK2 = MEDCouplingRemapper();
remapTH_to_NK2.prepare(mesh_ctf_fluid, mesh_cob, "P0P0");
print ("Doing remapper... OK");

# Matching the variables names (NK TH)
#ctf_ids_out = CTF.getOutputFieldsNames();
##ctf_ids_in  = CTF.getInputFieldsNames();
#cob_ids_out  = COB4.getOutputFieldsNames();
#cob_ids_in   = COB4.getInputFieldsNames();
ctf_ids_out   =      ["fuel_temperature",      "moderator_temperature",
"moderator_density"];
ctf_ids_in    = ["power"];
#cob_ids_out   = ["power"];
cob_ids_out   = ["power", "volume"];
cob_ids_in    = ["tfuel", "tmod", "rho"];
TRANSLATE = dict(zip(cob_ids_in, ctf_ids_out));

#ctf_unit_in   = ["kW"];
#ctf_unit_out  = ["", " ", " "];

```

```
#####
#####          STEADY CALCULATION          #####
#####
i = 0
power_pre = [] #COB4.getOutputMEDField(cob_ids_out[0]);
res = COB4.initialize_Steady()
# NK SOLVER
res = COB4.solve_Steady()

# Transfer the power to TH
NK_to_TH(i, power_pre)

#field_VOLUME_cob = COB4.getOutputMEDField(cob_ids_out[1]); # the volume
field is writen inside medfile
#MEDLoader.WriteFieldUsingAlreadyWrittenMesh(COB4_fileName,
field_VOLUME_cob)

while (not COB4.isStationary()):
    # TH SOLVER
    TH_Solver(CTF);

    # Transfer TH to NK
    TH_to_NK()

    # NK SOLVER
    res = COB4.solve_Steady()

    # Transfer the power to TH
    power_pre = NK_to_TH(i, power_pre)
#    values = field_POWER_cobpre.getArray()
    print " [STEADY] ##### Iteration: %d" % (i)
    i +=1

#    if (i == 2):
#        exit(1);
else:
    print " [STEADY] Steady state achieved"
pass
res = COB4.terminate_Steady()

#exit(0);

#####
#####          TRANSIENT CALCULATION          #####
#####
res = COB4.initialize_Trans() # Inicializar a partir del STEADY STATE
CTF.transientMode()
actual_time = 0.
end_time    = 1.0

#plt.ion()
#plt.xlim(0, end_time)
#plt.ylim(-1, 18000 )
```

```

#plt.ylabel('power(MW)')
#plt.xlabel('time(s)')

#time = []
#power = []
while actual_time <= end_time:
    actual_time_NK = actual_time

    # Solve the TH
    while abs(actual_time_NK + dt_NK-actual_time) > 0.000001:
#     while actual_time < actual_time_NK + dt_NK:
        time_graphic = actual_time_NK
        # Get the TH time step
        dt_TH = CTF.computeTimeStep()
        dt_TH = min(dt_TH, actual_time_NK + dt_NK - actual_time)

        print " [TRANSIENT] Calculating TH iteration: %f + %f
==>%f"%(actual_time, dt_TH, actual_time_NK + dt_NK),
CTF.initTimeStep(dt_TH) # Set the new delta T
CTF.initTimeStep(dt_TH) # Set the new delta T
res = CTF.solveTimeStep()
if (res == False):
    print "Error"
    break
else:
    CTF.validateTimeStep()
    print "OK"
    # Calculate new time
    actual_time = actual_time + min(dt_TH, actual_time_NK +
dt_NK - actual_time)

# Transfer the TH to NK
TH_to_NK_tran(actual_time_NK, dt_NK)

# Solve the NK
res = COB4.setNextTimeStep(dt_NK) # Set the new delta T
res = COB4.solveTimeStep_Trans()
res = COB4.validateTimeStep();

# Transfer NK to TH
NK_to_TH_tran()
#power.extend = NK_to_TH_tran()

    print " [TRANSIENT] #####"
#####
    print " [TRANSIENT] #####"
#####
    print " [TRANSIENT] ##### Actual time (py - TH - NK):
%f - %f - %f" % (actual_time, CTF.presentTime(), COB4.presentTime())
    print " [TRANSIENT] #####"
#####
    print " [TRANSIENT] #####"
#####

else:
    print " [TRANSIENT] End of the simulation"
pass
print "time clock", (clock()-start)
res = COB4.terminate_Trans()

```

```
raw_input() # If you comment this line and discomment the exit(0), when
the execution finishes the graphic will disappear.
           #if you use the command raw_input when the execution finishes
the graphic will not disappear.To finish the execution push ENTER
#exit(0);
```


Curriculum Vitae

Education

- **Naval Engineer.**
Universidad Politécnica de Madrid, 2013.
- **Master in Nuclear Science and Technology.**
Universidad Politécnica de Madrid, 2014.

Courses

- **Frédéric Joliot/Otto Hahn Summer School on Nuclear Reactors: Physics, Fuels and Systems.**
Karlsruhe Institute of Technology. 2015.
- **Workshop on research-article writing.**

Universidad Politécnica de Madrid. 2017.
- **Radioactive Waste Management.**
Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas. 2013.

Work experience

- **Enusa Industrias Avanzadas, December 2017**

Internship abroad

- **North Carolina State University, Raleigh, USA.**

Duration, November,8 2016-March 10, 2017.

Supervisor: Maria Avramova

Journal articles

- N. García-Herranz, D. Cuervo, A. Sabater, G. Rucabado, S. Sánchez-Cervera, and E. Castro (2017). "Multiscale neutronics/thermal-hydraulics coupling with COBAYA4 code for pin-by-pin PWR transient analysis". In: *Nuclear Engineering and Design* 321, pp. 38–47. DOI: 10.1016/j.nucengdes.2017.03.017.
- Sabater, Adrián, Diana Cuervo, and Santiago Sánchez-Cervera. 2016. "Comparison of Subchannel and Averaged Channel Thermal-Hydraulic Descriptions on Coupled Pin-by-Pin Neutronic Calculations." *Transactions of the American Nuclear Society* 115: 4.
- Spasov, I., S. Mitkov, N.P. Kolev, S. Sanchez-Cervera, N. Garcia-Herranz, A. Sabater, D. Cuervo, J. Jimenez, V.H. Sanchez, and L. Vyskocil. 2017. "Best-Estimate Simulation of a VVER MSLB Core Transient Using the NURESIM Platform Codes." *Nuclear Engineering and Design* 321 (September): 26–37. <https://doi.org/10.1016/j.nucengdes.2017.03.032>.
- Kliem, S., Y. Kozmenkov, J. Hadek, Y. Perin, F. Fouquet, F. Bernard, A. Sargeni, et al. 2017. "Testing the NURESIM Platform on a PWR Main Steam Line Break Benchmark." *Nuclear Engineering and Design* 321 (September): 8–25. <https://doi.org/10.1016/j.nucengdes.2017.05.028>.

POLITÉCNICA

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES
UNIVERSIDAD POLITÉCNICA DE MADRID

José Gutiérrez Abascal, 2. 28006 Madrid
Tel.: 91 336 3060
info.industriales@upm.es

www.industriales.upm.es