TECHNICAL REPORT
IGE–379

AN OVERVIEW OF CSCT-D5C3

U. Le Tennier

Institut de génie nucléaire
Département de génie mécanique
École Polytechnique de Montréal
June 24, 2021

Copyright Notice for CSCT-D5C3

The development of CSCT-D5C3 was financially supported, directly or indirectly, by various organizations including Polytechnique Montréal, CNL, IMT Atlantique and the Natural Science and Engineering Research Council of Canada (NSERC).

Acknowledgments

The CSCT-D5C3 was developed in support of a master thesis prepared at Polytechnique Montréal by an exchange student from IMT Atlantique, France. The main author of this report would like to express his thanks to both Polytechnique Montréal and IMT Atlantique for their support along the years. I would also like to thank Guy Marleau, research director and nuclear engineering professor at Polytechnique Montréal, and Armando Nava-Dominguez, nuclear reactors thermal hydraulics researcher at CNL, for making this master thesis possible. Finally, none of this would never have happened without the financial support of the Natural Science and Engineering Research Council of Canada (NSERC).

Contents

List of Figures

1 Introduction

CSCT-D5C3 stands for Canadian SCWR Coupling Toolset - DONJON5 CATHENA3. The coupling between DONJON5 and CATHENA3 for the safety analysis of the Canadian SCWR was created during a 2019-2021 master thesis. It was developed with Python3 and uses DONJON5 (donjon5 2016, version 5.0.2 (Revision: 507)) and CATHENA3 (cat3_6_1_1-b01). The coupling programs are available at github.com/sesyul/CANDU-SCWR in *E-Coupling* directory, except CATHENA executable which access is restricted by the owner of the code. Besides, DONJON inputs (databases, equilibrium and coupled files) exceed file size limitation. To retrieve the archives, please contact the author Ulysse Le Tennier at letennier.u@gmail.com.

Are available the coupling functions and main program with the related input (in subdirectory *c-Coupling*), CATHENA pre-filled input generation functions (in *a-CathenaInputGen*), the DONJON5 structures handling functions (in *b-NeutronicInputsGen*, *c-Coupling* and *d-CouplingDataTreatmnt* for Windows and in *x-ASCIIButForLinux* for Linux), data treatment functions (in *d-CouplingDataTreatmnt*), the DONJON5 executable and the .bat file that performs DONJON5 execution.

This manual provides useful but non-technical information about the coupling process. Technical information are available in CSCT-D5C3UsersManual.pdf, where the python functions are described. CSCT-D5C3UsersManual.pdf is available in the *E-Coupling* directory. The master thesis is available in french at HYPERLIEN

Several Canadian SCWR design adjustments were proposed between 2015 and 2020. The toolset uses a 2032 MWth core, the fuel is axially enriched and contains Fuel Integrated Burnable Absorbers (FIBA). Overall, the main characteristics of the core are provided in [8]. To adapt the toolset to a different core would need a close understanding of how DONJON, CATHENA and DRAGON work. Such information is not provided in this document, it can be found respectively in [1], [5] and [2]. Instead, information about how operates the coupling is provided in the present document to give the user the possibility to run coupled calculations or adapt the algorithm to support new calculations. As of today, it is not possible to take into account the impact of reloading on nucleus decay, even if some explanation in the present manual mention this possibility.

If additional information or operational support is needed, feel free to contact the author at letennier.u@gmail.com.

2 Coupling algorithm

2.1 Environment setup

To use the coupling algorithm, a precise setup is required. Figure 1 displays the global architecture of directories. The root directory is named EXEC_DIRECTORY and can be placed anywhere. It is advised to stick closely to the architecture inside the root directory, displayed on figure 1, otherwise changes must be implemented in various scripts to cope with a different environment.

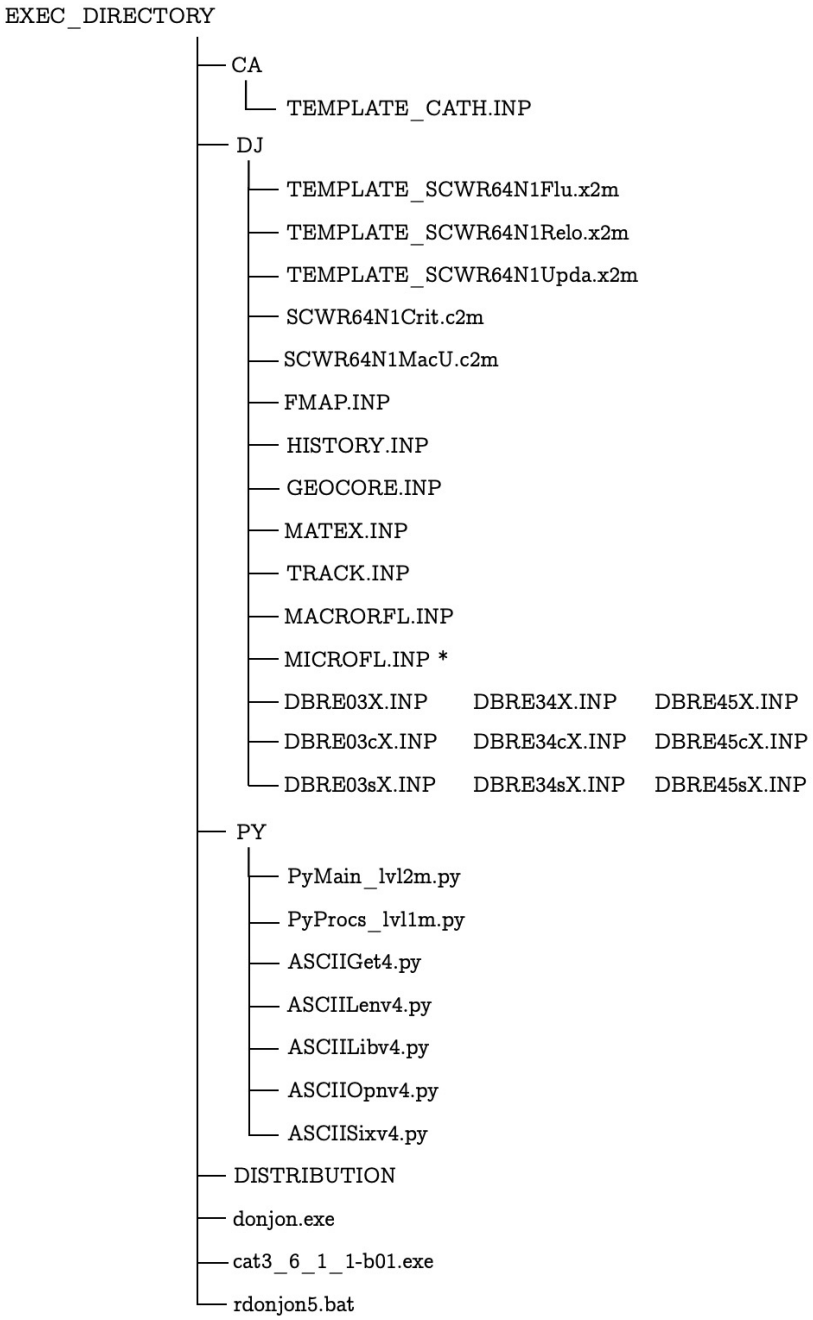

Figure 1: Directories architecture in support of coupling

The CATHENA template is stored in the CA directory. It can be generated thanks to functions described in section 4. It is a pre-filled file which has to be copied in the execution directory, then modified before executed. This file is discussed in section 4. All the inputs related to DONJON execution are contained in DJ directory. The .x2m files are main DONJON files, they perform flux calculation, reloading of the core and update thermodynamics parameters in **FMAP.INP**. The .c2m files are DONJON procedures, they are called by SCWR64N1Flu to find the boron quantity required to reach criticality and they interpolate the databases. Depending on the kind of simulation intended, two sets of SCWR64N1Flu, SCWR64N1Crit and SCWR64N1MacU are available. One set takes into account the impact of reloading on ^{233}Pa and ^{233}U inventories (which requires **MICROFL.INP**) and the other does not (**MICROFL.INP** not required). The user must provide a consistent set of inputs. The .INP DONJON files contain several information about the DONJON core model (geometry, tracking, mixes...) that cannot be updated. They are provided in additionnal .zip files unavailable online. To retrieve those, please contact the author of this manual at the address specified in introduction. The databases, which name begin by **DBRE**, are related to the batches model used. For instance, the assemblies at the bottom of the core (between 0 and 3m) are named **DBRE033** for the 3-batches and **DBRE034** for the 4-batches. Complementary .zip files provide **FMAP.INP** and **HISTORY.INP**. Those are the only files updated by the algorithm. **FMAP.INP** is the FuelMap, it contains detailed information about the core and the cycles. **HISTORY.INP** is a storage file containing useful information about the core and also crucial information for the execution of the coupling algorithm. More detailed insights about their role is given in section 2.2.

PY directory contains Python files : the coupling main program, the support functions called by the main (PyProcs) and the ASCII functions that handle DONJON data structures. The main imports all the functions it needs to perform coupling. Each call to the main must be consistent with the batch refueling chosen and the DONJON files provided. DISTRIBUTION directory is initially empty, then the algorithms fills it with different distributions, those which are not stored in **FMAP**, written in .txt files. At maximum, it contains data of one complete cycle. If the cycle is 18 steps long and if the 18 lasts steps modeled are stored in DISTRIBUTION, the new solution found will be stored after the oldest solution present in DISTRIBUTION is erased. For instance, if the solution of cycle 12 and step 16 is available, the algorithm will replace the solution for cycle 11 and step 16 with it. The coupling algorithm manages all of this operation. Finally, must be provided DONJON and CATHENA executables in addition to rdonjon5.bat. If the executables are provided with a different name, the execution calls in the Python main must be updated with the new names.

If an error occurs, only the execution report text file is created in the execution directory. Otherwise, files containing distributions of parameters are created in the execution directory. Final **FMAP** and **HISTORY** will be stored in the DJ directory.

Except for databases, it is possible to produce .INP files thanks to multicompos and inputs in *b-NeutronicInputsGen*. It is then possible to intend major changes in the reactor characteristics (total power, geometry, refueling etc.). Nevertheless, such operation requires a deep understanding of how DONJON works and a clear idea of the changes wanted. It is thus warmly advised to contact the author for operative support in such case.

2.2 Operation

2.2.1 Python algorithm

An overview of the Python coupling algorithm based on figures 2 and 3 is given in this section. The different data structures are not mentioned to keep the reading as simple as possible. DONJON uses databases to recover information about the fuel. Those databases include several thermalhydraulic dimensions that can be interpolated. The output of CATHENA provides DONJON interpolation coordinates. In return, DONJON provides CATHENA power distributions in the different channels. Besides, the execution of both DONJON and CATHENA involves the same routine : import a template, modify it and send it to the code for processing. Modifying a template consists in replacing a set of marks with coupling data.

Four loops are used in the algorithm, one loop corresponds to a grey frame in which the conditions are declared at the top-left corner. Two conditions use booleans while the other two use binaries. Binary conditions are used when criteria are read in **HISTORY** structure and updated by DONJON. Boolean conditions are only updated by Python. First, the user calls the coupling main program by giving the initial Step to consider and the last cycle to simulate in addition to complementary information. If the algorithm is interrupted at a certain Step, the user is able to restart the main from its ending point. To do so, the user needs to check in the **HISTORY** structure in DJ directory the last value of the Step variable and use it to restart the main program. If the user wants to start new cycles after performing several coupled ones, the same process is applied. The last Step is read in **HISTORY** and a new cycle target must be given to the main. If the cycle target is two but both **HISTORY** and **FMAP** contain eight coupled cycles, the algorithm will crash. For instance, to simulate two additional cycles after eight were performed, the cycle target must be ten. Finally, **FMAP** and **HISTORY** are tightly bound together, the information contained in **HISTORY** enable DONJON to handle properly the **FMAP** object. The algorithm will crash if non-related **FMAP** and **HISTORY** are given as inputs. In the same manner, if a **MICROLIB** object (not depicted on figures 2 and 3) is used, it has to be consistent with both **FMAP** and **HISTORY**.

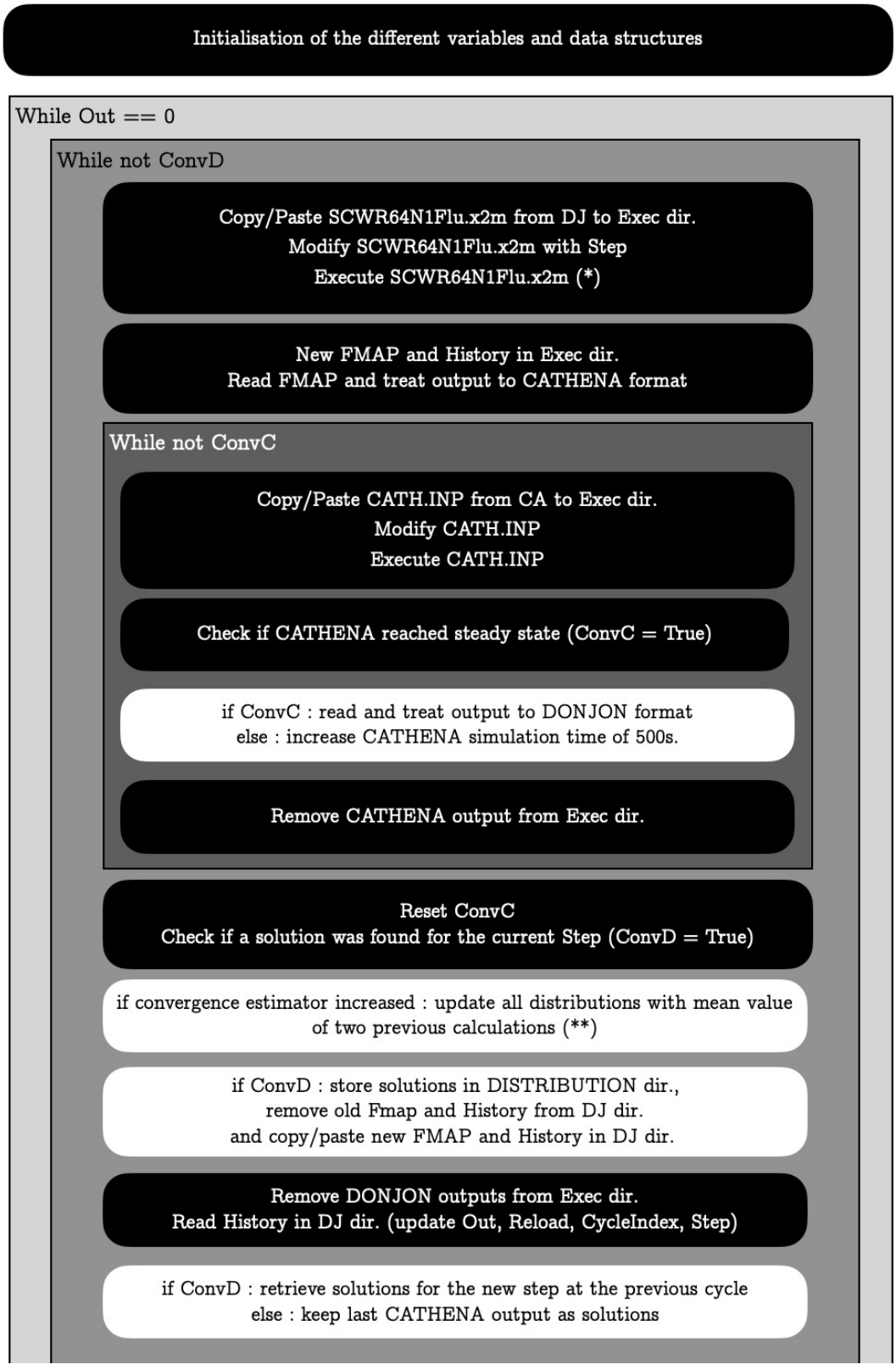

Figure 2: Python coupling algorithm, part 1

Figure 3: Python coupling algorithm, part 2

First, the main creates its internal variables and data structures. Two loops are started for the considered Step. It begins with the import of a DONJON flux calculation template in the execution directory. Then, it modifies it with Step and send it to DONJON for processing. Insights on internal algorithm of DONJON flux calculation are given in section 2.2.3 of this document. DONJON and CATHENA outputs are systematically written in the execution directory. Once the DONJON execution is over, power distributions are retrieved from the new **FMAP** located in the execution directory. The CATHENA loop is about to start.

Power distributions are passed to CATHENA after a CATHENA pre-filled input is imported from CA to execution directory. Once the execution is done, a test assesses whether CATHENA was able to reach steady state or not. If steady state is reached, the CATHENA outputs can be read and stored, otherwise the simulation time given as an input to CATHENA is increased by 500 seconds and the CATHENA execution is restarted. The tolerance criteria is discussed in section 2.2.5. By default, the CATHENA input is called with a simulation time of 1500s. CATHENA solves time-dependent equation, thus it requires a time target great enough to reach steady state. The higher the time target (or simulation time) is, the greater the calculation time is. The algorithm is designed to avoid these 500 seconds increase but the feature is kept just in case. Once the CATHENA steady state is reached, ConvC boolean turns True, the

thermalhydraulics solutions are stored in internal variables and the execution directory is cleaned.

Once out of the CATHENA loop, the ConvC boolean is reset and the latest set of solutions is compared to the set obtained in the previous ConvD loop for the current Step. If it is the first time the algorithm goes through this loop at Step, the set of solutions obtained is compared to a null set. If the convergence criteria is met, the solution for Step is considered found and ConvD turns True. Then, old **FMAP** and **HISTORY** (in the DJ directory) are replaced with new **FMAP** and **HISTORY**. If the convergence criteria is not met, ConvD remains False and the old **FMAP** and **HISTORY** are kept in DJ directory. Then, DONJON outputs are removed from execution directory and variables are read in **HISTORY** (DJ directory). If ConvD is True, a new Step is about to begin. To prepare it, the solutions found for the new CycleStep at the previous cycle are retrieved from text files in DISTRIBUTION directory. If such information is not available, the last solution found is kept in cache. If ConvD remains False, the last solution found is kept to serve a new flux calculation. Both for ConvD True or False, the solution kept in cache is implemented in **FMAP** thanks to a call to SCWRN1Upda.x2m. The updated **FMAP** replaces the former one in DJ directory. The process is repeated until a solution for Step is found.

Once out of the ConvD loop, ConvD is reset and the solutions for the considered Step are stored in internal strings. Those will only be exported when the algorithm exits the fundamental loop. At some point, if the algorithm crashes or if the computer is shut down, those strings are definitely lost. If Reload is triggered while reading **HISTORY**, the reloading loop begins. The reloaded **FMAP** replaces the former one in the DJ directory. Finally, if the latest value of the Out variable read in **HISTORY** equals one, the execution is over. The algorithm exits the fundamental loop and exports the thermalhydraulic solutions in text files that are created in execution directory.

2.2.2 Neutronic model overview

The neutronic core model is part of the toolset. To bring changes, it requires a close understanding of how DONJON works and especially how the CLE-2000 language operates [1–3]. Instead of giving exhaustive insights about the DONJON inputs, a brief overview of the geometry is given to provide comprehensive information related to the thermalhydraulic simulation.

The core is composed of 25 cm cubic cells, each vertical channel being composed of 20 cells. Contrarily to CATHENA, the full core is declared despite a $1/8^{th}$ symmetry. The fuel is axially graded, which results in three different zones depicted on the left part of figure 4. Besides, corner and side cells are particularized, they use specific databases. For a single neutronic calculation, a total of nine databases (multicompos) are used, three per axial zone. Two sets of DONJON inputs are available, one for the 3-batches loading plan (405 days long cycles) and the other for the 4-batches loading plan (325 days long cycles). The loading plans are depicted on the right of figure 4. The properties of each cell is stored in **FMAP**, the center of each cell corresponds to a node in the CATHENA model introduced in section 4.1. The reflector is composed of free of boron heavy water, one meter thick on the radial direction and 75 cm thick at the top and bottom. The inlet and outlet nozzles, which are at the top of the core, are not modeled. In addition to the corner and side cells, two subreflectors compose the reflector, one at the edge (25 cm thick) of the core and the other elsewhere. The two subreflectors have the same composition, their properties are retrieved from the same multi-assemblies lattice calculation. Their impact on the simulation is negligible but their use does not penalize the calculation time.

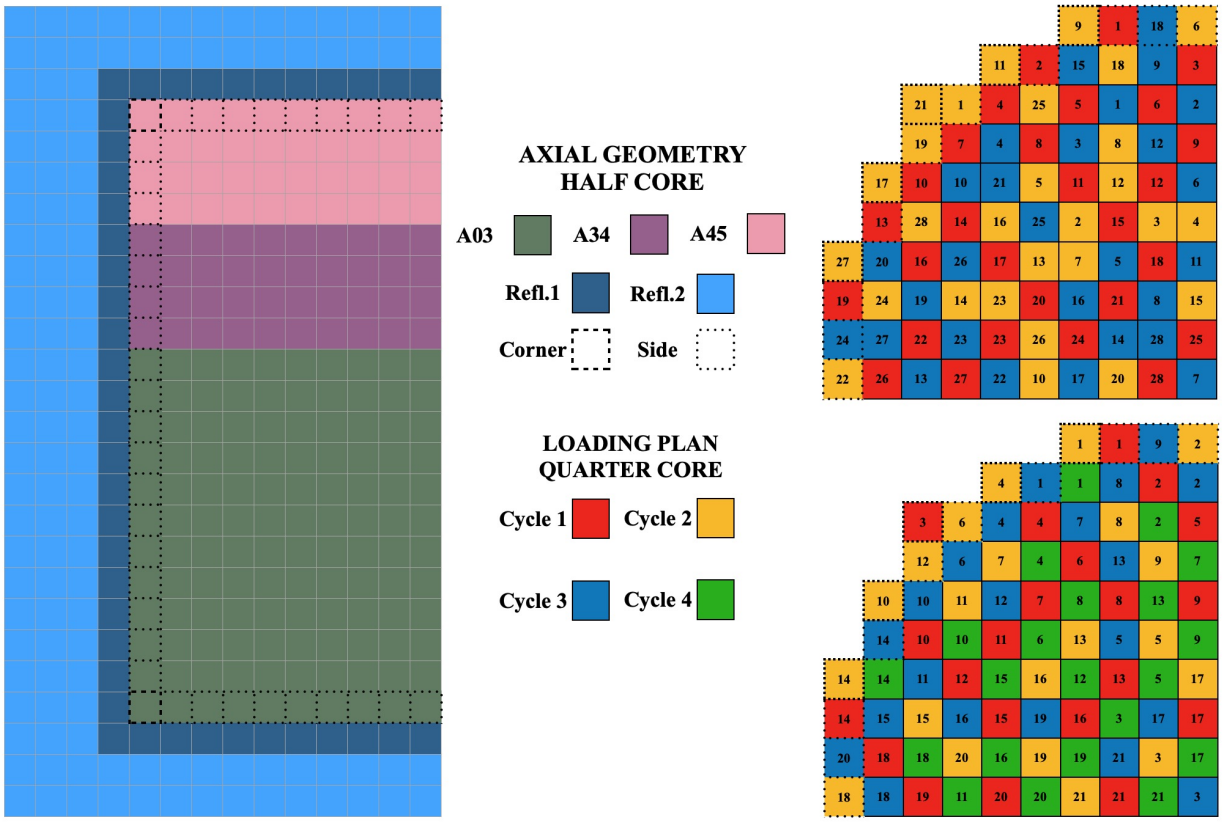

Figure 4: Neutronic geometry declared in DONJON (right) and loading plans (left)

To perform coupling, a preparation work is necessary. When a **FMAP** object is created, before being the support of any flux calculation, it is given an arbitrary burnup distribution. This burnup distribution has no physical reality. It is then necessary to simulate consecutive cycles, only with DONJON, to reach the neutronic equilibrium. Such equilibrium means that if additional cycles are simulated, they do not give different burnup distribution along the cycle. It is up to the user to define his own criteria to assess whether the neutronic equilibrium is reached or not. One relevant criteria is the greatest difference of critical boron between a cycle and the other. The critical boron is measured in ppm and stored in the equilibrium history structure. After a cycle is modeled, the critical boron of each step can be compared to the critical boron of the same step but at the previous cycle. Then, the greatest difference from a cycle to the other can be compared to the criteria. Because of the settings in DONJON procedures, which aim to find critical boron with a precision of 10^{-3} ppm, a criteria of 10^{-2} ppm proved to be a good one. With such criteria, it takes less than 20 and less than 30 cycles to reach neutronic equilibrium with respectively 3-batches and 4-batches loading plans when the impact of reloading on ^{233}Pa and ^{233}U is not taken into account. The Python file PyEquilibrium and the DONJON inputs associated (.zip), available in *b-NeutronicInputsGen* subdirectory in E-Coupling at github.com/sesyul/CANDU-SCWR, enable to create inputs set which reached the neutronic equilibrium. The Equi_3c and Equi_4c archives contain the DONJON main and its procedures stored in a new DJ directory. The user must copy the databases in the DJ directory, the DONJON executable and its rdonjon5.bat in addition to PyEquilibrium in the execution directory.

2.2.3 DONJON algorithm

On figure 2, the execution of DONJON with SCWR64N1Flu.x2m as input is marked with a (*). This mark means that this operation needs to be highlighted. As discussed in the previous section, an

important part of the internal variables are never directly updated by Python. It is the case for Out, which controls the end of the main call, Reload, which triggers reloading and Step, which is the most important variable. Those are directly updated by DONJON and stored in **HISTORY** data structure. Python only reads **HISTORY** structure, it never changes the value of one of those variables. This way of proceeding enables to keep the number of information to write in SCWR64N1Flu.x2m before executing it as low as possible. Besides, it enables to have a copy of every useful variables that cannot be lost if the main crashes. That is the reason why restarting the main after a crash is very user-friendly because only one information has to be manually retrieved.

The figure 5 introduces the internal DONJON algorithm. When executed, DONJON reads the input instructions in SCWR64N1Flu.x2m. Among those, Step will be used to retrieve time and cycle variables in **HISTORY**. Such information will be used to burn the fuel after the flux is calculated. Then, it will be used to update variables such as Reload and Out that will be written in **HISTORY**. Finally, the code exports **FMAP** and **HISTORY**. Several time steps scheme are available. This information has to be called in the python main, it is advised to use "CANDU5" which uses an initial 5 days step and then constant 20 days steps. The other options are described in commentaries of the SCWR64N1Flu.x2m template. The two loading plans can be called, the 3-batches is referred as "3c" and the 4-batches reloading is referred as "4c". The duration of a cycle can not be changed with a call to the python main. To change it, it is necessary to update the TimeEnd value in the SCWR64N1Flu.x2m template.

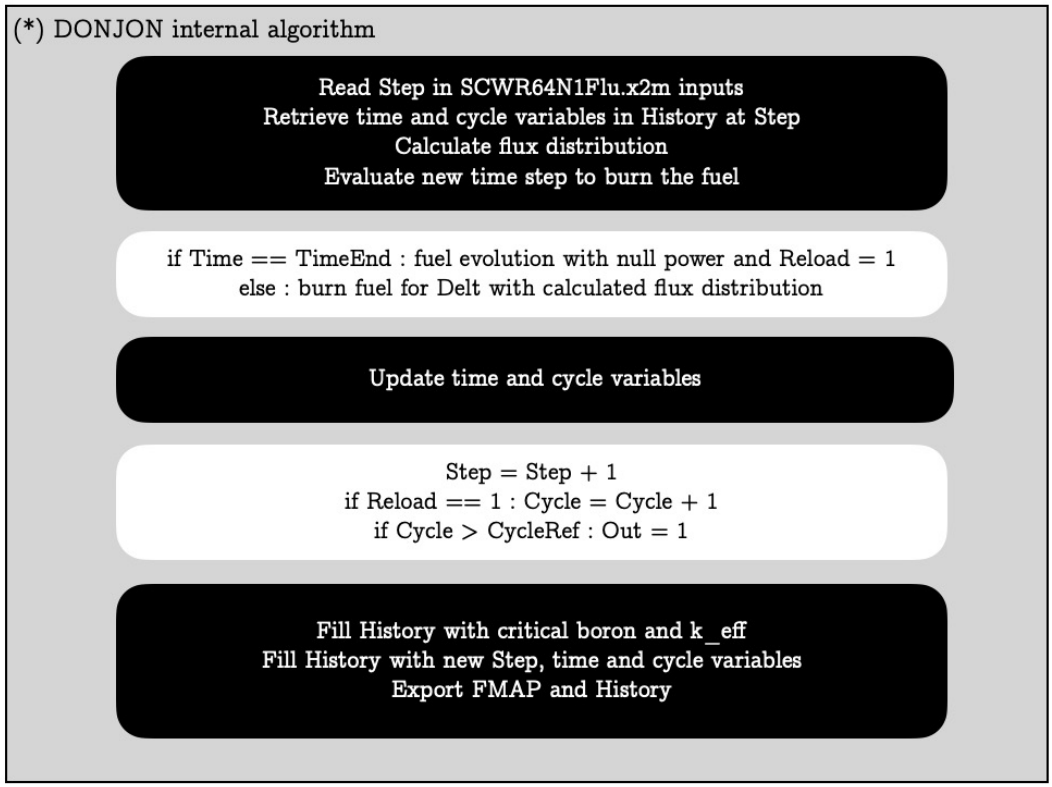

Figure 5: DONJON algorithm

2.2.4 Tolerance and convergence criteria

For the CATHENA convergence loop (ConvC), a steady state is reached when the mass flow in each channel is constant within a tolerance criteria. The mass flow is evaluated at 6 elevations (nodes) in both flow tube and fuel channel of each pressure tube. If the difference of mass flow between two consecutive nodes is greater than the tolerance criteria, it is assumed the CATHENA simulation did not reach a steady state. The value of the criteria is given as an input to the python main. By default, a criteria of 0.1 kg.s^{-1} can be used when the average mass flow in a channel is ≈ 3 kg.s^{-1}.

For the Step convergence loop (ConvD), the estimator used is the square root of the mean squared error on power distributions of each channel. The new power distribution is compared to the previous one. One estimator per channel is available but only the largest one is compared with the convergence criteria, which is also given to the main as an input. It is experimentally determined that a 5 kW criteria corresponds approximately to a precision of 3 K on the upward coolant temperature, 0.2 K for the downward coolant temperature and 15 K on the fuel temperature.

2.2.5 Convergence helper

When the algorithm assesses whether a solution for Step is found or not, it compares one estimator to the convergence criteria. If the estimator is greater than the criteria, the value of the estimator is stored in a dedicated array. If not, the array is reset. Therefore, the algorithms is able to determine if the estimator decreased or not when convergence is not met. It happens often that the estimator increases, which is possible because the core is composed of 336 channels. From an iteration to the other, the algorithm can come closer to finding a solution, the power deviation decreased for most of the 336 channels and still have a larger estimator, a minimum of four (due to symmetries) channels have increased their own estimator over the previous one. To accelerate the convergence, if the estimator increases, a special procedure is triggered. For each node and each distribution, the mean value between the latest solution and the one before is calculated. This set of mean distributions replaces the latest solution and is then implemented to use in the next DONJON calculation.

This trick relies on one main observation : the mean distribution of two solutions remains a solution. To clarify this assertion, it is necessary to focus on CATHENA inputs. The code solves the equations for the thermalhydraulic network while using a constraint on outlet temperature. A valve component, a proportional integral controller, is located at the inlet of each pressure tube. It adjusts the inlet admission surface to ensure the outlet temperature remains at 625°C within a certain tolerance. Thus, every CATHENA solution provides, for every channel, the same coolant temperature at both inlet and outlet, respectively 350°C and 625°C. Besides, along the coolant's course, its temperature only increases. A mean coolant temperature distribution has the same characteristics : coolant inlet and outlet temperatures are 350°C and 625°C while coolant's mean temperature only increases along the coolant's path. For other parameters like power or fuel temperature, the same mechanism does not apply. However, the mean power distributions keeps the total power of the core. Finally, the solutions are mainly driven by coolant density/temperature and power distributions thus, calculated mean distributions help convergence without deteriorating solutions.

Figure 6 illustrates that a mean coolant temperature distribution remains a valid solution for a channel. Besides, it shows how the mean distribution is able to tackle the oscillation that can take place around the mean value.

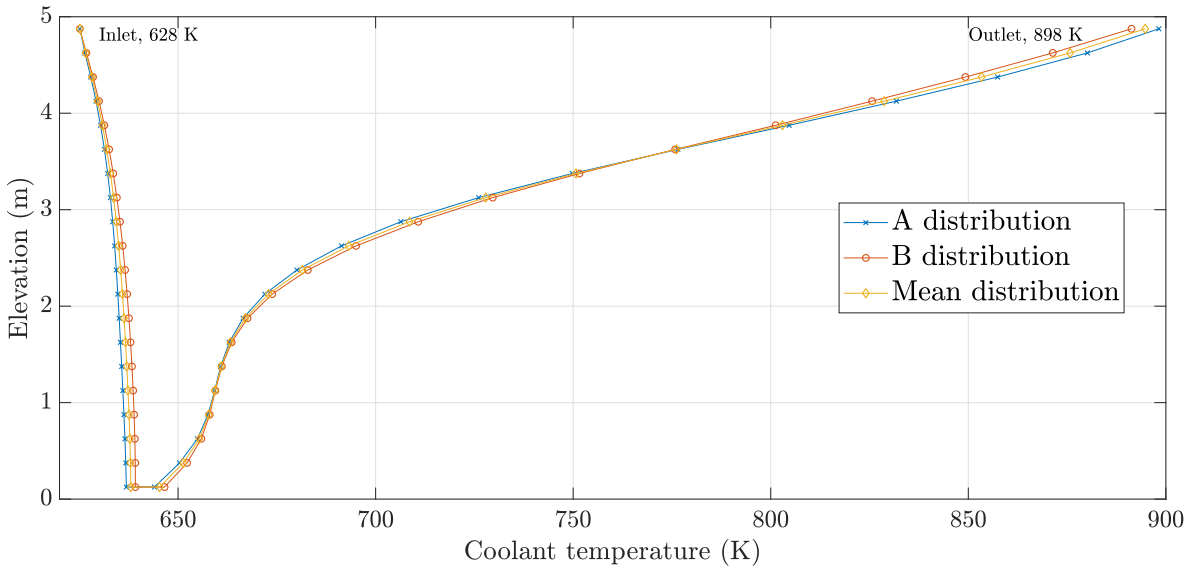

Figure 6: Mean distribution of coolant temperature compared to previous calculation solutions

3 Data treatment

In support to the coupling algorithm, data treatment functions are provided. They transform the outputs in a format that is consistent with the real geometry of the core. They are stored in *d-CouplingDataTreatmnt* directory in *E-Coupling* at github.com/sesyul/CANDU-SCWR. They use the DONJON structures handling functions (ASCII) to retrieve the information from **FMAP**. Two sets of ASCII functions exist. The first one is able to navigate in DONJON structures created or updated in a Windows environment. The second one only works with DONJON structures that are updated or created in a Linux environment and never updated in a Windows environment. A Windows environment is mandatory to execute CATHENA, therefore the user will most likely need the first set of ASCII functions. Nevertheless, the Linux version of the ASCII functions is available in *x-ASCIIButForLinux* directory in *E-Coupling*.

The data treatment functions are available in TreatMapMFlw.py and TreatMaps.py files. It is possible to directly use the functions with the coupling outputs. In the first file are provided functions to retrieve and plot quarter core mass flows. When retrieved, the data are stored in a matrix (array) after the geometry is unfolded. Then the matrix can be used for a plot. Such a matrix has two dimensions because each channel has only one mass flow, measured in $kg.s^{-1}$. CATHENA simulates a quarter core which accounts for a full core. Therefore, to have a relevant mass flow for a channel, each value provided by CATHENA must be divided by four. This operation is already implemented in the mass flow retrieval function. More insights about how CATHENA works are given in section 4.1. TreatMaps contains the function that retrieves power distributions in **FMAP** and the function that retrieves the thermalhydraulic parameters distributions from coupling outputs. A plot function is also provided. Contrarily to the mass flow distribution, the power or thermalhydraulic parameters distributions are contained in a three dimension matrix. Once again, the unfolding of the raw data is implemented in the retrieval functions to provide matrix which accounts for the real geometry of the core. The temperatures are given in kelvins (K) and the coolant densities in tons per cubic meter ($t.m^{-3}$). Technical information about the functions is available in the CSCT-D5C3 User's Manual, available at github.com/sesyul/CANDU-SCWR. Besides, TreatMapMFlw.py, TreatMaps.py and CSCT-D5C3UsersManual.pdf provide references where data presentation (colormaps, fonts etc.) are discussed.

4 CATHENA Canadian SCWR inputs generation

The CATHENA pre-filled template is based on an original work of D. Hummel [4]. It is composed of several groups which account for thousands of lines. Each group serves an individual purpose. For instance, one defines the physical components (pipes, nozzles, tanks etc.), another one the junctions between those components while a third one defines the initial conditions. In this section are provided insights about CATHENA Canadian SCWR inputs generation. Contrarily to DONJON templates, the information that the coupling algorithm has to write in CATHENA input file is quite sophisticated. To fill or write the CATHENA inputs consists in replacing numerous marks by coupling data. Each channel has to be provided a total amount of power produced by inner rods and outer rods, in addition with the relative axial distribution of it. Even minor changes to CATHENA inputs can require several modifications in the coupling functions. Therefore, CATHENA Canadian SCWR inputs generation python functions are available, to enable the user to modify CATHENA pre-filled input without breaking the coupling algorithm. After a brief description of the thermalhydraulic network, complementary information are given about the functions. More comprehensive literature about how to unleash CATHENA's potential is available in the different user's manuals [5, 6].

4.1 Thermalhydraulic network

CATHENA generates a quarter core physical geometry thanks to components and junctions. Then, thermal models and system models are implanted on the physical geometry. The Canadian SCWR is composed of 336 vertical pressure tubes soaking into a heavy water moderator pool. Exhaustive information about the core layout is available in the literature [7, 8]. The physical geometry of the reactor is introduced on figure 7. The components related to the turbine are not part of the geometry. Therefore, the system possesses two boundaries with fixed conditions : inbound and outbound. The 84 channels modeled are connected to both the inlet and outlet plenums, only one is represented on figure 7. If the position of a channel in the core has an impact on the neutronics, it does not have any on the thermalhydraulics. The moderator acts as a heat sink that prevents heat transfers from a channel to another. The only difference between the channels is the heat they produce which does not prevent them from being declared in the same manner. If the plenums are unique, a total of four pumps and four out pipes are used. The out pipes represent the connection to turbines.

Each channel begins with an inlet nozzle connected to a valve component. The valve is the support of the mass flow control system discussed in the next paragraph. Then, the coolant goes through the flow tube, reverses its course in rvrsevol, flows through the fuel channel and exits through the outlet nozzle. Because the outlet plenum stands above the inlet plenum, a pipe named riser enables the coolant to go from the outlet nozzle to the outlet plenum. The fuel channel and flow tube are decomposed in 20 axial nodes. Those correspond with the center of each assembly declared in DONJON. The rods are not declared as physical components, instead their impact is simulated thanks to the hydraulic resistance implemented in the fuel channel component.

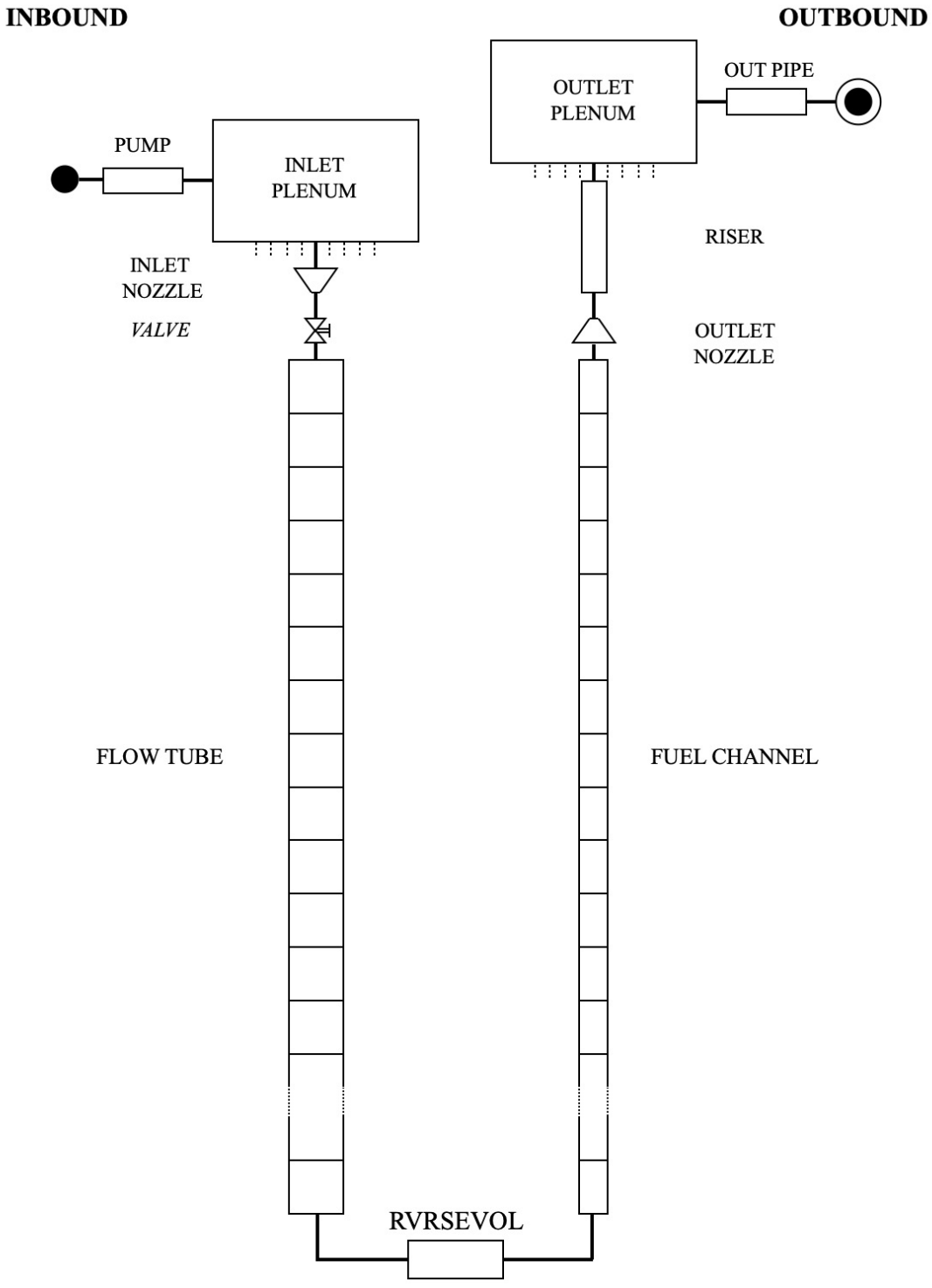

Figure 7: Hydraulic network declared in CATHENA

Figure 8 presents the thermalhydraulic network. Five elements are noticeable. First, the control system which is a proportional integral controller that modify the admission surface, thus the mass flow. It is set to ensure the outlet temperature remains at 625°C within a certain tolerance margin. The four other elements are the heat models : the tubewall, the inner or outer circle rods and the pressure tube. The tubewall represents the heat transfers which occur between the fuel channel and the flow tube. The power calculated by the neutronic simulation is shared between the inner and outer circle rods. The environment around the rods is exclusively the coolant in the fuel channel. Finally, the pressure tube conducts heats from the coolant in the fuel channel to the moderator. Thus, the only heat losses pass through the pressure tube heat model.

The thermalhydraulic model faces strong limitations. The most important one is that hydraulic resistance have a weak impact on the pressure drop profile. If the total pressure drop between inbound and outbound is 0.8 MPa, more than 0.7 MPa are lost before the coolant enters the channel : the valve component absorbs most of it. Consequently, the total pressure drop drives the pressure profile and inhibits the impact of the rods on the flow. Hence, the model is insensitive to variations of the hydraulic resistance in the fuel channel. The impact of the turbulences caused by the rods is not finely taken into account, the simulation lacks of precision about both pressure drops and rods cooling phenomenon. The last noticeable information is the amount of heat lost. When 2032 MW are transferred from the rods to the coolant, between 56 and 66 MW can be lost to the moderator depending on the loading plan and the cycle step. This represents losses between 2.75 to 3.25 %.

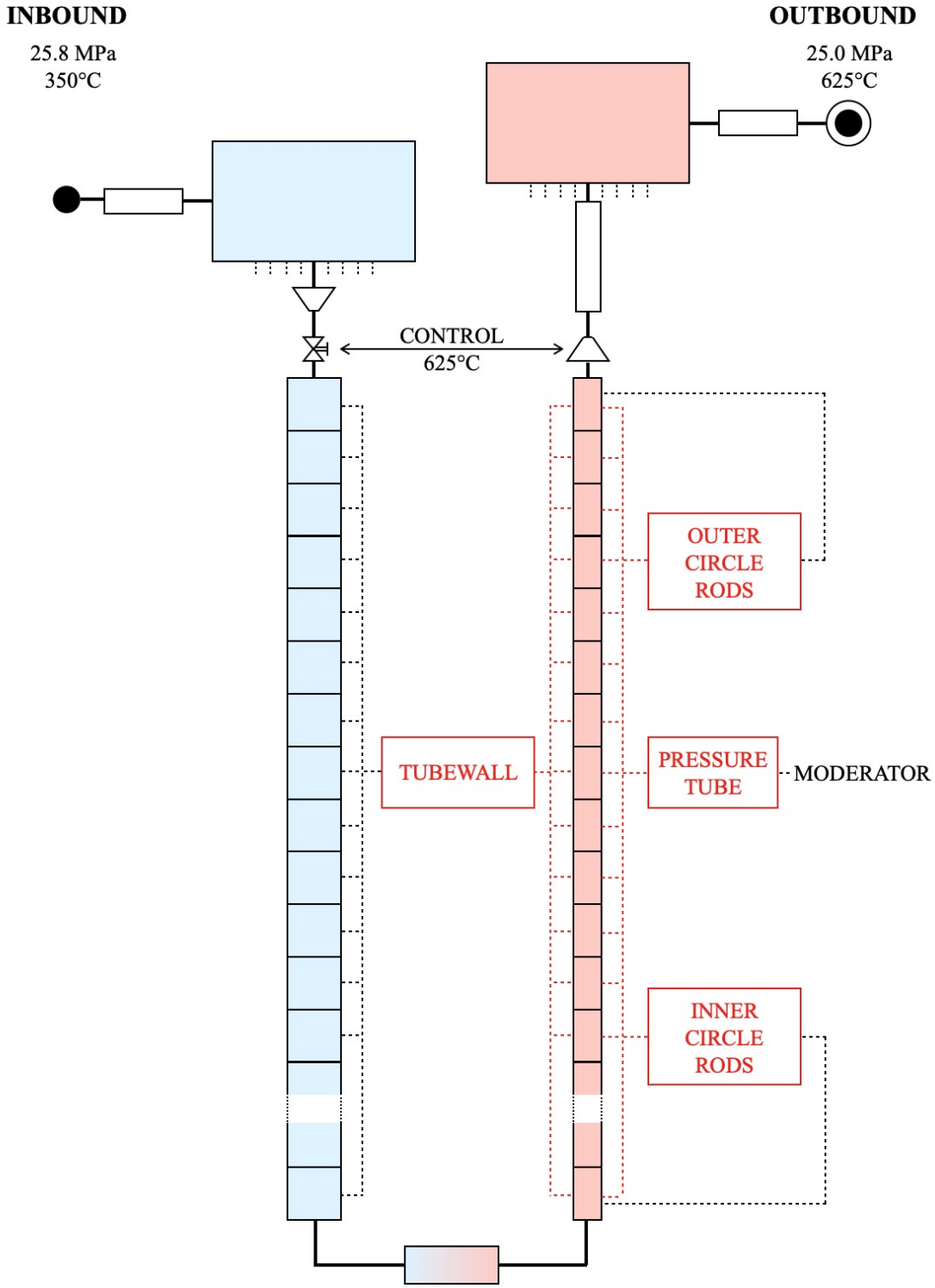

Figure 8: Thermalhydraulic network declared in CATHENA

4.2 Functions

A total of seven functions is provided in the WriteScriptM.py file, available at github.com/sesyul/CANDU-SCWR in *E-Coupling* directory. Six of them generate a part of the CATHENA input and the last function merges all of the different parts into a sole .INP file. The different functions are designed to create a quarter-core input (84 channels). Even if the aggregating function requires a number of channel as an input, it can only generate a valid file if 84 are given as an entry. To generate valid inputs with a different number of channels, several changes to the functions must be considered. They are not trivial and the user must have a precise comprehension of both the Canadian SCWR geometry and how to write CATHENA inputs to proceed. A description of the six input generating functions follows. Technical knowledge is provided in the CSCT-D5C3 User's Manual. All the different marks that are mentioned are replaced by writeN1CATH() (PyProcs, coupling support functions).

The first function is named writeBase(). It generates the CONTROL GROUP where are set the numeric simulation options and the restart file. The numeric options include the command to use supercritical properties, the simulation time and the frequency of outputs. The first mark is named *HERE00*, the coupling algorithm replaces it with the simulation time which is of 1500 seconds, by default. Because of this default time, the outputs are printed each 500 seconds (PRINT CONTROL). The output files generated follow the same logic but are set by another group. After a standard run, the outputs files contain information for four different times : 0s, 500s, 1000s and 1500s.

The second function, writeCompo(), declares the physical components (group named COMPONENTS). The 84 channels must be declared independently, thus each subcomponent is numbered. The other components such as the plenums or the bounds of the network are unique. The connections for those are declared with CONNECTIONS group generated by writeConnec(). This function is quite similar to writeCompo() in its definition.

The fourth function is writeInit() and declares the initial conditions (INITIAL CONDITIONS group). The initial conditions are given for each component and each junction. A new set of marks is declared. It is used by the coupling algorithm to give a mass flow distribution as an input. To ensure CATHENA provides a steady solution with only 1500s simulated, the mass flow distributions found previously are given as initial conditions. For the first calculation of the coupling algorithm, a default distribution is used. For the others, depending on which stage the algorithm processes, the last result found or the solution found for the current Step at the previous cycle can be used.

The fifth function is writeTherMod() and declares the thermal models (HEAT TRANSFER PACKAGE group). As depicted on figure 8, four different thermal models are declared for each channel : the flowtube, the pressure tube, the inner rods and the outer rods. Those models declare the nodes and the properties of the materials involved. The properties were retrieved from AECL. The rod elements use the last set of marks which enable to provide the power distribution. The power distribution consists in three information, the total power produced by the element, the radial distribution in each rod and the axial distribution. Except for the radial distribution, which is fixed, the total power and the axial distribution are calculated by treatN1paramDC() (PyProcs, coupling support functions). The total power stands for four channels (symmetry consideration), it is shared between the rods circles, 51.9% for the outer rods and the rest for the inner rods. In the writeTherMod() definition, two groups of four lines are written as commentaries in the rods definition. If these comments are is removed and the line just above them is erased, the input generated uses a mean power distribution and prevent the coupling from being performed (the power distribution cannot be updated). Such an input can be used to perform various tests without the coupling scheme.

The last function is writeSysCont() and declares several groups. The first group has the explicit name of BOUNDARY CONDITIONS. It declares the pressure and temperature conditions at inbound and outbound. The second group, SYSTEM MODELS, declares the valve components geometry for each channel. The last group is SYSTEM CONTROL. It declares the outputs and the proportional-integral controller of valve component. The settings are a gain of 10^{-5} and an integral period of 30s. The outputs are given a precise name that the coupling algorithm is able to read. They are numbered because

each can only contain information for up to 500 nodes. The first output is CONV.RES, it is used by checkN1convcath() (PyProcs, coupling support functions) to check if the simulation reached steady state or not. CONV.RES aggregates six measures of the mass flow for 83 channels. It is assumed that, if 83 reached steady state, the channel 84 is deemed to have too. The second output aggregates the mass flow of each channel, for a total of 84 information. Because of this format, it is also read by checkN1convcath(). The other outputs use one measure for each node, therefore the number of information is of 84x20, it exceeds the limit of 500. Then, four numbered outputs of 420 data are generated for each quantity. The data are sorted by elevation : an output file contains information for only five consecutive elevation. For each thermalhydraulic parameter, the output numbered one contains data of the nodes at the top of the core, the output numbered four contains data of the nodes at the bottom of the core. The same logic applies for each file, data of the nodes which are the closest to the top are written before data of the nodes the closest to the bottom. The outputs files are named after the quantity involved with an authentication number between one and four. The names are recognized by the coupling algorithm, especially by readN1CATH() (PyProcs, coupling support functions) and by the main program.

References

[1] A. HÉBERT, D. SEKKI and R. CHAMBON, "A User Guide for DONJON version5," Report IGE-344, Polytechnique Montréal, Institut de Génie Nucléaire (2019).

[2] G. MARLEAU, A. HÉBERT and R. ROY, "A User Guide for DRAGON version5," Report IGE-335, Polytechnique Montréal, Institut de Génie Nucléaire (2019).

[3] R. ROY, "The CLE-2000 tool-box," Report IGE-163, Polytechnique Montréal, Institut de Génie Nucléaire (1999).

[4] D. W. HUMMEL, "Coupled neutronic-thermalhydraulic transient behaviour of a PT- SCWR reactor," PhD. Thesis, McMaster University (2015).

[5] T. BEUTHE, A. VASIC and D. F. WANG, "CATHENA 3.5.5.1 input reference, user's manual," *C. N. Laboratories, édit.*, Canadian Nuclear Laboratories, (2016).

[6] T. BEUTHE, A. VASIC and D. F. WANG, "CATHENA 3.5.5.1 GENHTP input reference, user's manual," *C. N. Laboratories, édit.*, Canadian Nuclear Laboratories, (2016).

[7] M. GAUDET, M. YETISIR and A. SARTIPI, "Conceptual Plant Layout of the Canadian Generation IV Supercritical Water-Cooled Reactor," *CNL Nuclear Review* (2016)

[8] F. SALAUN et D. R. NOVOG, "Optimization of the Canadian SCWR Core Using Coupled Three-Dimensional Reactor Physics and Thermal-Hydraulics Calculations," *Journal of Nuclear Engineering and Radiation Science*, Vol. 4, No. 2, (2018).