International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011)
Rio de Janeiro, RJ, Brazil, May 8-12, 2011, on CD-ROM, Latin American Section (LAS) / American Nuclear Society (ANS)
ISBN 978-85-63688-00-2

A FLEXIBLE COUPLING SCHEME FOR MONTE CARLO AND THERMAL-HYDRAULICS CODES

J. Eduard Hoogenboom*
Delft University of Technology
Mekelweg 15, 2629 JB Delft, The Netherlands
J.E.Hoogenboom@tudelft.nl

Aleksandar Ivanov and **Victor Sanchez**
Karlsruhe Institute of Technology, Institute of Neutron Physics and Reactor Technology
Herman-vom-Helmholtz-Platz-1, 76344 Eggenstein-Leopoldshafen
Aleksandar.Ivanov@kit.edu, Victor.Sanchez@kit.edu

Cheikh Diop
Commissariat à l'Énergie Atomique
F-91191 Gif-sur-Yvette, France
Cheikh.Diop@cea.fr

ABSTRACT

A coupling scheme between a Monte Carlo code and a thermal-hydraulics code is being developed within the European NURISP project for comprehensive and validated reactor analysis. The scheme is flexible as it allows different Monte Carlo codes and different thermal-hydraulics code to be used. At present the MCNP and TRIPOLI4 Monte Carlo codes can be used and the FLICA4 and SubChanFlow thermal-hydraulics codes. For all these codes only an original executable is necessary. A Python script drives the iterations between Monte Carlo and thermal-hydraulics calculations. It also calls a conversion program to merge a master input file for the Monte Carlo code with the appropriate temperature and coolant density data from the thermal-hydraulics calculation. Likewise it calls another conversion program to merge a master input file for the thermal-hydraulics code with the power distribution data from the Monte Carlo calculation. Special attention is given to the neutron cross section data for the various required temperatures in the Monte Carlo calculation. Results are shown for an infinite lattice of PWR fuel pin cells and a 3x3 fuel BWR pin cell cluster. Various possibilities for further improvement and optimization of the coupling system are discussed.

Key Words: Monte Carlo, Thermal-hydraulics, MCNP, TRIPOLI, SubChanFlow, FLICA.

1. INTRODUCTION

High fidelity numerical reactor simulations including different physics and spatial scales are gaining increasing importance in the nuclear community. In Europe an advanced platform for reactor simulations -called NURESIM- is being developed within the collaborative EU NURISP Project. It includes the coupling of neutronics and thermal hydraulics at nodal and pin cell level as well as the coupling among different thermal hydraulic solutions (meso-, component and

* Corresponding author\

system scale). Regarding the coupling of stochastic neutron physics methods with thermal hydraulic codes, important work has been started in the frame of the NURISP project for the development of coupling approaches between Monte Carlo and thermal hydraulic codes.
Monte Carlo codes are applied for reactor analysis over a long period of time to estimate the effective multiplication factor of a reactor core and more detailed information like the power distribution. Since about a decade Monte Carlo codes have been coupled to depletion modules to calculate the fuel composition after a certain burnup. It is not surprising that in the last few years attempts have been made to couple Monte Carlo (MC) codes to thermal-hydraulics (TH) modules [1-3] to include the feedback of temperature and coolant density to the neutronics calculation. This will provide the option to perform reactor core analysis with thermal-hydraulics feedback while retaining the exact neutronics calculation with all geometrical and continuous-energy cross section detail in the Monte Carlo calculation. As Monte Carlo calculations are generally CPU time intensive it is not anticipated that coupled MC-TH will be used for design purposes, but they will be suitable for verification of traditional deterministic reactor core calculations in which many approximation will be used like multi-group calculations with probably a very limited number of energy groups, homogenization of fuel pin cells or even complete fuel assemblies. Moreover, the MC-TH coupling bypasses the multi-group cross section generation stage of deterministic calculations in which only a very limited part of the reactor geometry is taken into account in 2D approximation, possibly with an axial buckling. In the current stage it is only possible to include axially varying composition, e.g. due to varying enrichment, and the effect of varying temperature and coolant density or void fraction using branches for the different parameters to vary one by one. All these approximations make it valuable to have a tool for verification.
Although integration of the thermal-hydraulic feedback into the Monte Carlo code [2] may have some advantages, external coupling of a Monte Carlo code and a thermal-hydraulics code will have the advantage of greater flexibility. In the research presented in this paper we will discuss the details of a coupling scheme in which there is a choice in both the Monte Carlo code and the thermal-hydraulics code to be used. Moreover, both the Monte Carlo code and the thermal-hydraulics code can be used in their original version without any modification. Hence, if only a working executable is available that will be sufficient.
One option for the Monte Carlo code in our system is the general purpose code MCNP5 [4]. The other option is the TRIPOLI4 code [5], which is the Monte Carlo code in the European NURESIM platform [6]. For the thermal-hydraulics codes one can choose the FLICA4 code [7], developed at the Commissariat à l'Énergie Atomique at Saclay, France, or the SubChanFlow code [8], developed at the Karlsruhe Institute of Technology, Karlsruhe, Germany.
In this paper emphasis is on the details of the coupling and especially how to address in the Monte Carlo codes the cross sections at the temperatures resulting from the thermal-hydraulics calculation.

2. COUPLING SCHEME

2.1. Coupling generalities

For a typical nuclear reactor with separate fuel rods and coolant channels a scheme for coupling a Monte Carlo and a thermal-hydraulics calculation in its basic form is shown in Fig. 1. The

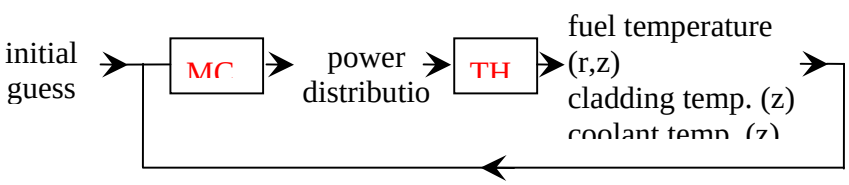


Figure 1. General MC-TH coupling scheme

Monte Carlo code should calculate the axial power distribution for all fuel rods. This output is fed into the thermal-hydraulics code to calculate the fuel temperature in each fuel rod, as well as the cladding temperature and the coolant temperature and coolant density of each coolant channel. All these quantities should be obtained as a function the axial coordinate *z*, which requires both in the Monte Carlo and the thermal-hydraulics code a subdivision in axial zones. The temperatures and densities calculated by the thermal-hydraulics code are fed back to the Monte Carlo code in such a way that the proper cross sections will be used at the relevant temperatures in fuel, cladding and coolant. When starting this process the temperatures and densities must be given an initial (guessed) value. After iterations this process should end in converged values for the power and temperature distributions.

2.2. Codes used in coupling

The coupling scheme is kept flexible and allows different Monte Carlo codes and thermal-hydraulics codes to be coupled. As each code has it specific input and output structure, the conversion programs have to deal with that. The current scheme allows the MCNP and TRIPOLI codes for the Monte Carlo calculation and the SubChanFlow and FLICA codes for the thermal-hydraulic calculations.

2.2.1. The Monte Carlo codes used

The allowed Monte Carlo codes are MCNP and TRIPOLI. After a short description of these codes we will concentrate on the relevant input and output items for the coupling scheme. Both codes are general purpose Monte Carlo codes and well suited for nuclear reactor calculations. They can both be run in parallel, which is very helpful for otherwise relatively long Monte Carlo calculations.
Although there are differences in the way to specify the geometry and a material composition, both codes are similar in the basic technique for the neutron history simulation. However, there are important differences in cross section libraries and in the way one has to specify the temperature of a medium.

a) The MCNP code

The MCNP code [4] is a general purpose Monte Carlo code developed at the Los Alamos National Laboratory (LANL), USA. We used the MCNP5 code version 1.51, but as there are no specific input options used for the power estimation, it should be possible to substitute the MCNPX code as well without any modification to the MCNP input file or the coupling scheme. MCNP uses a special cross section library to provide the relevant nuclides in any medium in the system. A standard library is delivered with the code and contains cross section data at room

temperature. For a certain nuclide there are libraries available based on different evaluations of basic cross section like ENDF/B-VI and ENDF/B-VII. There are also available cross section libraries based on JEF2.2 and JEF3.1. The last one contains cross section data for many nuclides at a set of different temperatures ranging from 300 K to 1000 K in steps of 100 K plus at 1200 and 1800 K. The MCNP library format are called ACE libraries and can be produced by the NJOY code [9] using the ACER module (and first the RECONR and BROADR modules to produces cross sections at the required temperature) starting from a basic evaluated cross section file in ENDF format. All major evaluated cross section libraries like ENDF/B, JEFF and JENDL are produced in ENDF format. In the ACE library based on the JEFF3.1 evaluation for each available nuclide cross sections are present at the above mentioned temperatures.
In the MCNP input a volume containing a certain material is defined as a cell. The cell card in the input specifies the mass density or the total nuclide density and refers to a material number for the specification of the nuclide composition. In the material definition all nuclides present in the medium should be listed together with their relative nuclide fraction or mass fraction. Nuclides are defined by their alpha-numeric nuclide identification number in the for ZZAAA.XXc with ZZ a one or two-digits identification for the atom number (e.g. 92 for Uranium) and a three-digits identification for the mass number (e.g. 235 for ^{235}U). The two-digits extension number XX indicates the specific evaluation of the cross section, including its temperature. The last character ‘c’ indicates continuous energy representation of the cross sections. It depends on the available library what can be chosen for the extension number ‘XX’. The correspondence between nuclide extension number and temperature for the JEFF3.1 based libraries is shown in Table I.

Table I. Correspondence between nuclide extension number and cross section temperature for the JEF3.1 based cross section library for MCNP

Nuclide extension number	Temperature of cross section data (K)
31	300
32	400
33	500
34	600
35	700
36	800
37	900
38	1000
39	1200
15	1500
40	1800

As the thermal scattering of some materials like water depend on the chemical binding of the molecule, a separate library is needed to specify the probabilities for angular and energy changes

when a neutron scatters with the dominant scattering nuclide in the molecule (H in H_2O). If a medium contains water the use of a thermal scattering cross section library must be indicate on a MT card for the medium. On the MT card the scattering nuclide must be referenced (for light water using the JEFF3.1 evaluated cross section data) as lwtrXX.YYt with XX and YY extension numbers and 't' indicating that it concerns thermal scattering. The two-digit extension number XX is related to the temperature at which the thermal scattering is evaluated. For the JEFF3.1 library it is specified in Table II. The extension number YY is set to 31 to indicate the basis of the evaluated data (JEFF3.1).

Table II. Correspondence between nuclide extension number for thermal scattering in light water and cross section temperature for the JEF3.1 based cross section library for MCNP

Nuclide extension number	Temperature of cross section data (K)
01	293.6
02	323.6
03	373.6
04	423.6
05	473.6
06	523.6
07	573.6
08	623.6
09	647.6
10	-
11	1000

As nuclide cross sections are not available at all temperatures it is necessary to take specific actions in the input to MCNP to represent the cross section at the desired temperature as closely as possible. This is achieved by using for each nuclide present in a medium two references to a nuclide in the input file specification for a medium but with different extension numbers and different nuclide densities. The nuclide extension numbers are chosen such that they refer to the next lower and next higher temperature available in the library. The corresponding nuclide densities can be calculated in two ways, using a linear combination and a square root combination as follows

$$N(T_1) = \frac{T_2 - T}{T_2 - T_1} N \tag{1}$$

with T_1 and T_2 the nearest lower and higher temperatures with respect to T for which cross section data are available for the nuclide, $N(T_1)$ the nuclide density to be specified for the nuclide with extension number corresponding to temperature T_1 and N the actual density of the nuclide at temperature T. The density $N(T_2)$ for the nuclide with extension referring to temperature T_2 is obtained from N-$N(T_1)$. In some cases one gets more accurate results when a square root interpolation is used according to

$$N(T_1) = \frac{\sqrt{T_2} - \sqrt{T}}{\sqrt{T_2} - \sqrt{T_1}} N \quad (2)$$

To facilitate the preparation of an MCNP input file during the iterative coupling process, a master input file must be prepared in advance containing all specifications for the problem including the desired number of axial zones in each fuel pin, cladding and coolant medium, but with basically arbitrary coolant densities and nuclide extension numbers for all fuel, cladding and coolant material definitions. Hence, a master file is a complete input file that can be run by MCNP, except that all quantities that are to be replaced by appropriate values according to the desired temperatures are preceded by a ~ sign. This sign must be added in order to be able to detect in the conversion program where specific numbers to be replaced are located.
Currently there is no such interpolation method implemented for the thermal scattering of bound nuclides like H in H_2O. Therefore, the extension n umber for the nearest available temperature is used. As such a method is more complicated than for the standard nuclide identification, such an interpolation method is foreseen in future work (see Sect. 5).
To obtain the axial power distribution from MCNP the FMESH tally is used with a modification to obtain the fission energy deposition. The FMESH tally uses a track-length estimator for the average flux in a mesh volume and multiplies in our case the score by the product of the fission cross section of a fissile nuclide, its nuclide density and the deposited fission energy for that nuclide. These products are summed over all fissile nuclides of the medium. The fission energies per nuclide are hard coded values in the MCNP source code, which makes it difficult to assess their correctness. The MCNP5 manual states that they are the recoverable fission energy. The total energy released in a fission is composed of several component carried by various particles: the fission products, the neutrons that are released in a fission (including a small fraction of delayed neutrons), photons (prompt and delayed), beta particles and other decay products (delayed) and neutrinos. The recoverable energy is the total fission energy minus the energy of the neutrinos, which escape from the reactor anyway. In MCNP the recoverable energy is assumed to be deposited at the fission site. However, the energy of the neutrons is partly deposited in the coolant due to moderation, while for capture of non-thermal neutrons their kinetic energy is deposited at the position of the capture event, which can be anywhere in the reactor. Also the energy of the photons can be deposited away from the fission site. Although most of the recoverable fission energy is taken up by the fission fragments and deposited very near the fission site, as well as the energy of the beta particles from decay of the fission products, the assumption made when using the modified FMESH tally in MCNP that it gives the actual power production is an approximation. Moreover, at neutron capture not only the kinetic energy of the neutron is released (which is negligible for thermal neutrons), but also binding energy is released by photons and may be deposited elsewhere. See Ref. 10 for more details and a more accurate calculation of the actual power production. Using the FMESH tally in MCNP the results are listed in a separate output file from which the results are easily accessible.

b) The TRIPOLI4 code

The TRIPOLI4 code [5] is a general purpose Monte Carlo code developed at the Commissariat `a l'Energie Atomique (CEA) in Saclay, France. We have access to a development version 4.6.1 from which the option to produce cross section data at any requested temperature in the calculation prior to simulating the neutron histories is important.
TRIPOLI4 uses cross section data in ENDF format with separate files for each nuclide. Partly the original ENDF file for a nuclide is used to retrieve the angular scattering data and the energy

spectra for fission, (*n*,2*n*) reactions, etc. Partly a so-called PENDF file with a format like ENDF files, containing the cross section data evaluated at a certain temperature is used. For use with TRIPOLI4 the PENDF files are written in binary mode, normally in the general XDR format often used in a C++ environment. The temperature is included in the file name as the integer value of the temperature in Kelvin.
In the TRIPOLI4 input media are defined by their geometrical form together with a material assignment. The material composition is specified either by the nuclides composing the material with their (absolute) densities, or by the mass density of the material and the nuclides with their relative mass fraction. In either case the temperature of the medium is specified explicitly as an integer value representing the temperature in Kelvin. This makes it possible to specify for each zone in the fuel, cladding or coolant temperature explicitly. TRIPOLI4 will look for a PENDF file for all nuclides in a medium for the specified temperature. If the relevant PENDF file is not found and the –j option is specified at the command line when running TRIPOLI4 the NJOY code [9] is started to produce the cross section file at the specified temperature using the basic ENDF file for the nuclide as well as a PENDF file at 0 K. Hence, for all nuclides in a problem the PENDF files at 0 K must be generated before the calculation can be done. As this need to be done only once a Python script was written to produce these files for all nuclides for which basic ENDF files are present in the TRIPOLI4 cross section library directory. With the -j option specified one need not to care anymore about the presence of the relevant cross section files, nor apply any form of interpolation between data at different temperatures. Although the generation of cross sections by NJOY adds to the computer time used, especially for nuclides with many energy points in the resonance region, the time used by NJOY normally remains only a small fraction of the simulation time used by TRIPOLI4 and is therefore acceptable.
Disadvantageous are that TRIPOLI4 cannot be run in parallel when using the –j option and that during the iteration process with thermal-hydraulic feedback many cross section files may be generated at slightly different temperatures (with at least a difference of 1 K). After many iterations or running different cases with (partly) the same nuclides, one may end up with sets of cross section file at every 1 K temperature in certain temperature ranges often occurring in reactors.
The –j option does not provide a cross section file for the thermal scattering data of bound nuclides as they cannot be generated by NJOY in the same easy way as the cross section data for the total energy range. For the thermal scattering data ENDF files are present in the TRIPOLI4 library directory at the same temperatures as for the MCNP libraries and specified in Table II. For TRIPOLI4 these data files are identified by their file name containing .therm. as well as the corresponding temperature, rounded off to the nearest integer value. Hence, if a medium is specified at a temperature T TRIPOLI4 will not only look for a PENDF file at this temperature, but also for a thermal scattering data file at that temperature (or in a small temperature range around T), provided in the medium a bound nuclide as H in H_2O is specified. This file will normally not be present and cannot be generated on the spot. Hence, it is necessary to create in advance of the TRIPOLI4 calculation a soft link to the thermal scattering file at the nearest available temperature.
Other potential options to deal with the availability of cross section data at a certain temperature that might be implemented in the future will be discussed in Sect. 4.
As in the case when using MCNP a master input file for TRIPOLI4 must be prepared in advance of the calculation containing all specifications for the problem. In the input file to TRIPOLI4 media are identified by an alphanumeric character string which may be on the same input line as

the temperature and the density of the medium. This makes it easier to find the right place in the input file where modifications have to be made. If media names start with FUEL, CLAD or COOL followed by a number indicating its radial and axial position, it is simply possible by scanning the master input file for these names in the relevant section of the input file to find the position where to substitute the required temperature or the density in case of coolant material from the thermal-hydraulics output. As the cross section data at the requested temperature will be looked for and if necessary generated, there is no need to modify or manipulate the various nuclide densities.
To obtain the axial power distribution from TIPOLI4 a response function must be selected defining the type of quantity to be estimated together with a score definition specifying for which separate volumes the requested quantity must be estimated. TRIPOLI4 provides a deposited energy response function for this goal. In the score definition all the fuel volumes must be specified. The energy range should cover all energies in the Monte Carlo simulation. It should be noted that this response function is not exactly the same as the fission energy deposition tally in MCNP. TRIPOLI4 scores all the energy deposited in the requested volumes, which not only covers the locally deposited fission energy (the energy carried by the fission products and their decay energy by beta particles), but also the kinetic energy loss of neutrons in collisions and the binding energy of neutrons when captured. What happens to the photon energies (prompt and delayed??? . This is physically more correct, but requires for a complete picture a coupled neutron-photon calculation to determine the amount and position of energy deposited by the photons.
It should be noted that a Monte Carlo calculation cannot determine the absolute value of the power density in a reactor. It calculates in fact deposited energy per source neutron. Hence, the results must always be scaled to the total power which must be specified from other sources. This also happens in the thermal-hydraulic calculation where the total reactor power is entered in the input. Thus only the relative distribution of the power or deposited energy has to be calculated. Therefore differences in the total fission energy considered, e.g. neglecting the fission energy carried by prompt and delayed photons, in a Monte Carlo calculation are less relevant for the thermal-hydraulic calculation.
As the deposited energy tally in TRIPOLI4 is physically more correct it opens the possibility to take advantage of that by also considering the deposited energy in the cladding and the coolant. This possibility have not yet been used as it complicates the input to the thermal-hydraulic code and makes the comparison with MCNP more difficult. It remains, however, an option for future extensions as correct physical modeling is the primary aim of all Monte Carlo calculations.
The output of the deposited energy tally in TRIPOLI4 can be found in standard output file in which also all other information is recorded. Moreover, the tally output is not that simple structured, which makes it more difficult to retrieve the required results for the various fuel volumes by scanning the output file.

2.2.2. The thermal-hydraulics codes used

The allowed thermal-hydraulics codes are SubChanFlow and FLICA. After a short description of these codes we will concentrate on the relevant input and output items for the coupling scheme.

a) The SubChanFlow code

The SubChanFlow code [8] is a quasi 3D subchannel code developed at the Karlsruhe Institute of Technology (KIT), Germany. It simulates the thermal hydraulic conditions of fuel assemblies and cores of light water reactors (LWR) and liquid metal fast reactors (FR) solving the mass, momentum and energy equations based on the three-equations approach. Both hexagonal and square fuel assembly geometries can be treated. It is written in the Fortran-95 language in a fully modular way. Global data structure as well as fluid and material properties are stored in separate modules. Executables for Linux and Windows are available. Its input is made very flexible. Its output is also simplified with some output files directly suitable for coupling with a Monte Carlo code as they list in a simple format the axial temperature distributions in fuel, cladding and coolant, as well as the axial density distributions in the coolant.
For the coupling scheme the input of the axial power distribution is of importance. SubChanFlow uses a single input file in which the relevant input quantities are entered with an alpha-numeric identifier or a specific header if a table is to be entered. This is the case for entering the axial power distribution from the Monte Carlo calculation. Only one axial power distribution can be entered, which is applied to all fuel rods in a problem.
As with the Monte Carlo input a master file must be composed with all required input, including an arbitrary axial power distribution. When the master input file is canned for the header of the power distribution to be entered, it can easily be found and the proper power distribution be substituted. The power distribution is normalized to the total reactor power, which must be specified in the master input file.

b) The FLICA4 code

The FLICA4 code [6] is a 3D two-phase flow code developed at the Commissariat `a l'Energie Atomique (CEA) in Saclay, France, for the simulation of steady and transient two-phase flow phenomena in rod bundles and cores of nuclear reactors. The two-phase mixture is modeled by a set of four balance equations (mass, momentum, and energy of mixture, and mass of vapor). The mechanical non-equilibrium is taken into account by a drift flux correlation. The heat conduction in solids is solved by a one dimensional model. Different constitutive relations are available to close the system of equations. FLICA numerics are based on the finite volume method including the Roe approximate of the Riemann solver. The time integration is based on a fully implicit scheme together with a Newton iterative solution method. FLICA allows a flexible geometry definition and includes different working fluid such as water and gases (hydrogen, helium).
FLICA4 is one of the corner blocks for reactor analysis in the European NURESIM platform [7] for numerical reactor calculations. Integrated in the Salomé platform it can be coupled with various deterministic neutronics codes. The Salomé platform provides easy tools for coupling codes and visualization of results. However, it turned out that the integrated version of FLICA4 could not be used for our purpose of coupling with a Monte Carlo code. Hence, an executable of the stand-alone version was obtained from CEA. It can be run by calling the script f4 delivered with the code, using a number of arguments on the command line, mainly for addressing the directories where input and output files can be found. At a later stage it will be investigated whether the Salomé platform with FLICA4 can actually be used for our purpose in order to take advantage of various tools available with Salomé.
The input to FLICA4 is a file in the GIBIANE language, which provides a flexible way to define necessary parameters and functions. Although all input can be collected in a single input file, it is useful and customary to divide the input for various data parts over a few input files. From a

header file the other input files can be referenced. The structure of input files used in our coupling scheme is shown in Table III.

Table III. Input files used for FLICA4

File name	Contents
header	General input parameters Include directives for other input files Possible post-processing commands
geom	Geometry definition of fuel rods and coolant channels
bndcond	Boundary conditions for inlet temperature and flow and outlet pressure
power	Power distributions

Together these files provide a complete specification of the problem to be calculated. For our purpose only the contents of the power input file needs to be modified. As this file is directed only to the specification of the axial power distributions it can easily be read and the actual distributions be substituted.
The required output of FLICA4 can be determined via the header input file as well as to include the temperature calculations in the fuel rods. This makes it possible to request separate output files for the axial temperature distributions radially averaged over the fuel rod or the cladding, of the coolant in the coolant channels as well as the axial density distribution in the coolant channels. Then this output can easily be read for further processing by the conversion program in our coupling scheme.

2.3. The specific coupling scheme

When the Python script is started the choice of the Monte Carlo code and the thermal-hydraulics code should be indicated on the command line, using default values if they are not specified on the command line. In the Python script some preliminary actions are taken depending on this choices, the number of fuel rods and axial zones is determined from the thermal-hydraulics input files and an iteration loop is started. In each iteration step the thermal-hydraulics code is called for with a system call to execute a child process. After completion of the child process executing the thermal-hydraulics code the conversion program TH2MC is called via a child process indicating on its command line the thermal-hydraulics code used and the Monte Carlo code to be used thereafter.
The conversion program TH2MC is written in Fortran-95 and retrieves the choice of the thermal-hydraulics and Monte Carlo code from the command line arguments when starting the program. Internally it calls a subroutine for reading a master input file to the thermal-hydraulics code,

which has a file name depending on the choice of the thermal-hydraulics code. The master input file is a complete input file for running the thermal-hydraulics code including the specification of the geometry, the material composition, the boundary conditions like total power, inlet temperature, inlet flow, exit pressure, etc., and arbitrary axial power distribution. If the thermal-hydraulics code facilitates the option that the input data is divided over two or more separate files which are referenced in the main input file (as is usually the case for the FLICA4 code), the master file is only the input file with the axial power distribution, as the input in all other input files remain unchanged during the whole iteration process.
For an actual coupling scheme one has to consider how the axial power distribution output from the Monte Carlo calculation has to be merged with the input to the thermal-hydraulics code and how the temperatures and coolant densities output from the thermal-hydraulics code has to be incorporated in the Monte Carlo input. To that end two conversion programs were developed MC2TH and TH2MC. This leads to the coupling scheme of Fig. 2. A Python script was developed to drive the whole process. Although the system was run under Linux, the Python script can also be run under Windows provided that executables of the Monte Carlo codes and thermal-hydraulics codes are also available for Windows.

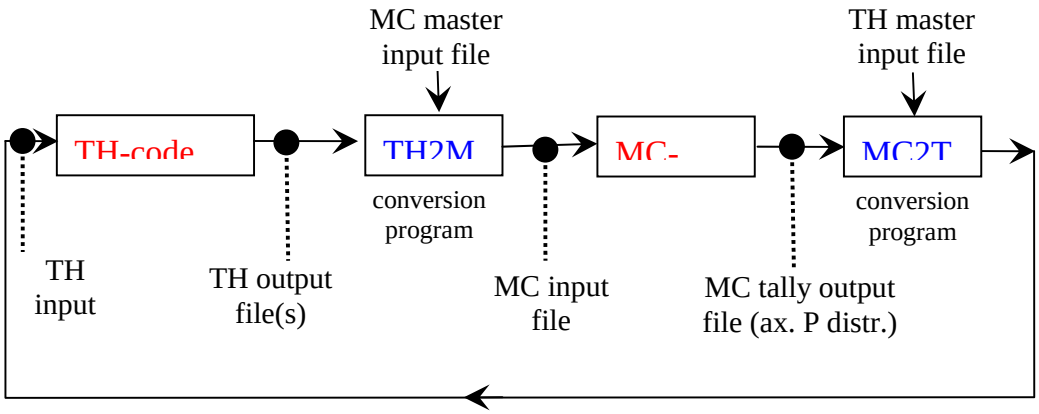


Figure 2. The flexible coupling scheme actually applied.

The Python script contains a loop for successive iterations, calls successively the programs to be run, including the conversion programs and does some input/output file handling. To start the iterations first the thermal-hydraulics code is run with a guessed initial power distribution. As the Monte Carlo runs will take much more time it is advantageous to start with a thermal-hydraulics calculation to obtain a reasonable estimate of the temperature distributions for the first Monte Carlo calculation, although the input power distribution will not be correct.
The TH2MC conversion program reads from the command line issued by the Python script the name of both the Monte Carlo code and the thermal-hydraulics code, as its action will depend on which codes are selected. It reads the necessary thermal-hydraulics output files with fuel, cladding and coolant temperature distributions as well as the coolant density distributions from either FLICA4 or SubChanFlow and stores these data internally. Then it scans the MC master input file for either MCNP or TRIPOLI4 to find all places where the temperature and coolant density data must be replaced. For the input to MCNP additional nuclide cards are added for all

materials for which the temperature is variable during the iteration process and the nuclide densities are calculated using the above described procedure to force cross section interpolation between available cross section data. The adapted master input file forms the input file to the next Monte Carlo calculation.
The Python script adds to all input and output files an extension with the current iteration number to retain these files for possible later analysis. As the names of the files read or written by the conversion program need to be fixed, the Python script copies temporarily the necessary files during the iterations to a unique name. Figure 3 shows the necessary files and file names used by the MC2TH conversion program, which depend on the actual Monte Carlo and thermal-hydraulics code used. The upper part of the figure shows the file names used with FLICA4, the lower part for SubChanFlow. However, the output files from the Monte Carlo calculation with the axial power distributions (file TRIPOLI4.out from TRIPOLI4 or meshtal from MCNP) can be combined with either of the thermal-hydraulics codes. The MC2TH program produces as output an input file to the selected thermal-hydraulics code (power.upd for FLICA4 or Subch.inp for SubChanFlow).

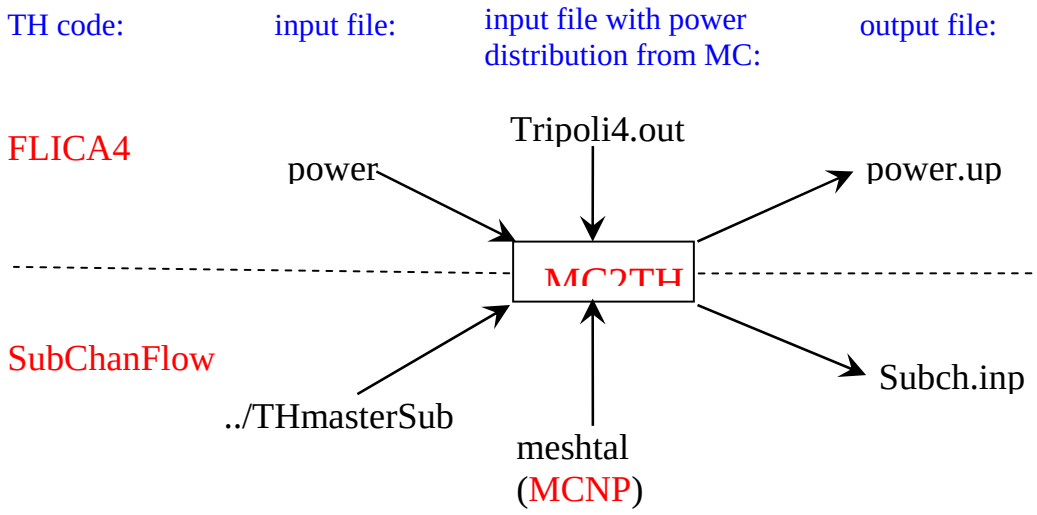


Figure 3. Input and output files for MC2TH conversion

The Python script checks for a normal end of the TH2MC program looking for a specific line in the standard output file of TH2MC. When it ends normally, the Python script renames files to include the iteration number and starts the selected Monte Carlo code with input and output file referenced on the command line and other necessary parameters depending on the code called. For MCNP one can use a file which contains the source distribution from the previous iteration to speed up the convergence of the fission source distribution for the new case. For TRIPLOI4 this is not yet possible. After completion of the Monte Carlo run the Python script checks here also for a normal end of execution.
Then it calls the conversion program MC2TH to prepare the input file for the next thermal-hydraulic calculation. As with the TH2MC program the call to MC2TH specifies on the

command line the selected Monte Carlo and thermal-hydraulics code. Fig. 4 shows the input and output file names used by MC2TH.

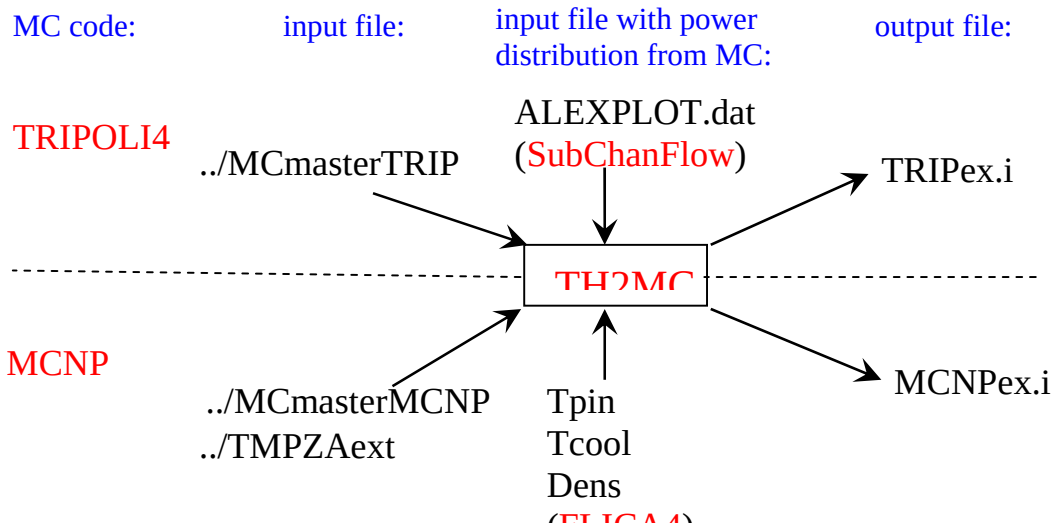


Figure 4. Input and output files for TH2MC conversion

3. DEMONSTRATION RESULTS

As the development of the flexible coupling scheme is still in an early stage, we selected two relatively simple cases for demonstration of the scheme. First the elementary case of a lattice of identical fuel pin cells and second a cluster of 3x3 pin cells in which the difference between the geometries considered in the Monte Carlo and thermal-hydraulics code become apparent, as well as some other problems in coupling Monte Carlo and thermal-hydraulics codes. The thermal-hydraulic regimes (whether there is boiling) are automatically detected by the thermal-hydraulics codes.

3.1. An infinite lattice of PWR fuel pin cells

As a first demonstration case we used an infinite lattice of PWR fuel pin cells, which means a single squared fuel pin cell with reflective boundary conditions. This is the geometry used in the Monte Carlo calculation and shown in Fig. 5 at the left. However, the thermal-hydraulics code will consider the coolant channel as the main component and its geometry is shown in Fig. 5 at the right. In this case the correspondence between the geometries is still simple.
The fuel rod, composed of UO_2 with a fixed density of 10.25 g/cm^3, has a radius of 0.41 cm. In the Monte Carlo calculation no gap is included. The outer radius of the cladding is 0.475 cm and the density of the cladding (pure Zr for simplicity) is reduced to 5.77 g/cm^3 to account for smearing out the gap between the fuel and the cladding. The lattice pitch is 1.26 cm. For the thermal-hydraulic calculation the modeling of the gap is important for the fuel temperature and a

gap of 0.01 cm is taken into account. The length of the fuel elements is 366 cm. Below and above the fuel there are homogenized regions from coolant and structural materials taken into account in the Monte Carlo calculations, below the fuel a zone of 46 cm height and above the fuel of 40 cm with different compositions.
For all regions (fuel, cladding and coolant) 20 axial zones are considered of 18.3 cm each, both in the Monte Carlo and thermal-hydraulics calculation. The fuel has a varying enrichment over the axial zones to force a clearly axial asymmetry and to demonstrate the capability of the coupled calculation. It is not intended as a realistic physical representation of an actual reactor design. For a deterministic neutronics calculation it is not straight-forward to take that into account in a proper way. For the Monte Carlo calculation it is no problem.
For the thermal-hydraulics calculation the pressure at the outlet was fixed at 15.8 MPa. The coolant inlet temperature is set to 564 K (291 ˚C) and the inlet mass flow rate at 0.326 kg/s. The total power of one fuel rod is set to 66.5 kW. The first thermal-hydraulic calculation runs with a cosine axial power/heat flux distribution.

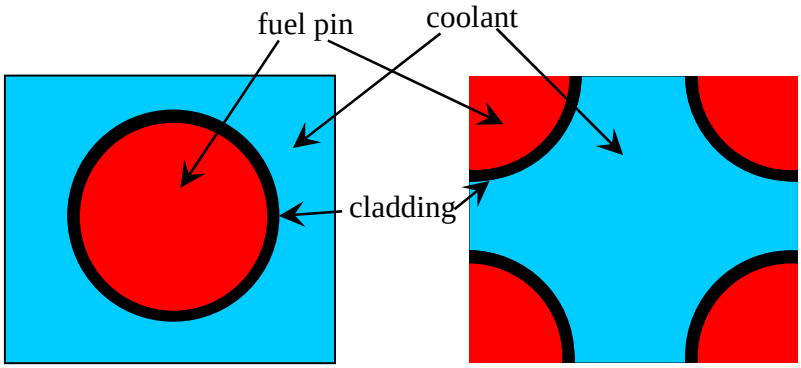


Figure 5. Pin cell geometry for MC (left) and for TH

This problem has been calculated with all four possible combinations of the TRIPOLI4 and MCNP5 Monte Carlo codes and FLICA4 and SubChanFlow thermal-hydraulics codes. Fig. 6 shows the axial power distribution estimated by the TRIPOLI4 Monte Carlo code using FLICA4 for the thermal-hydraulics after various iterations. The initial cosine distribution is not calculated by TRIPOLI4 and only used as input to the first FLICA4 run. From this figure it can be seen that there is a slight oscillation in the axial power distribution during the iterations and that it is already almost converged after 3 iteration steps. The axial variation of the fuel enrichment is also shown.
Fig. 7 shows the converged axial power distribution for the 4 different combinations of Monte Carlo and thermal-hydraulics codes. Fig. 8 shows the radially averaged fuel temperature distribution, Fig. 9 the coolant temperature and Fig. 10 the coolant density. The temperature and density distributions of the different code combinations agree well. However, there is a spread in the various axial power density distributions, which also has some influence on the fuel temperature distributions. This spread has to be investigated further.

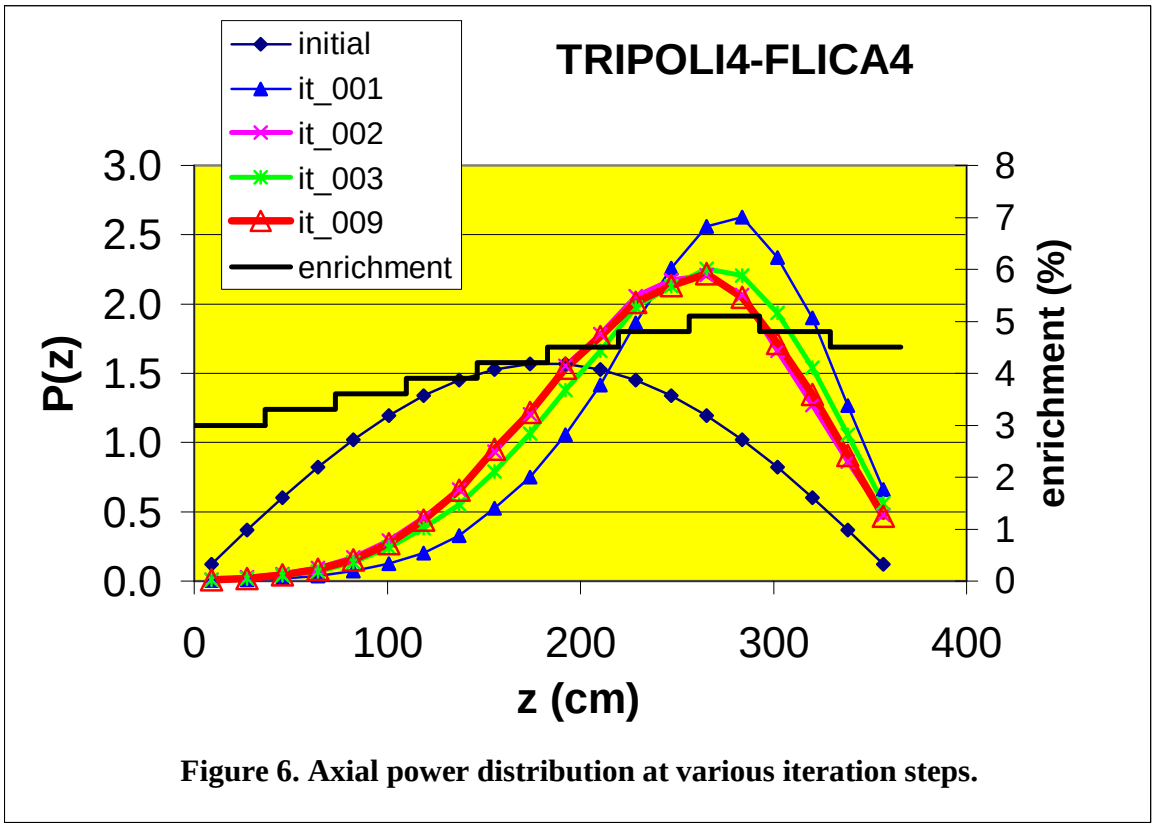


Figure 6. Axial power distribution at various iteration steps.

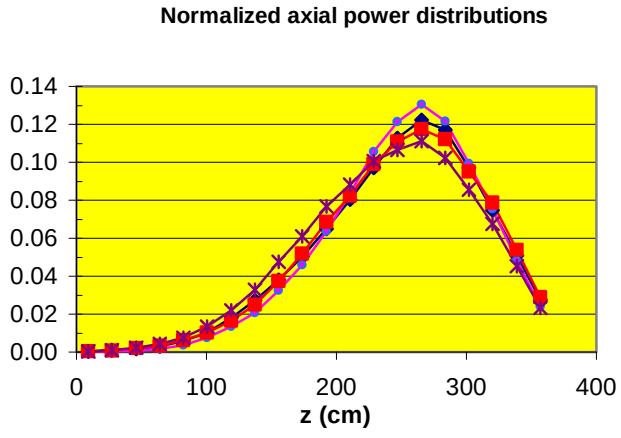


Figure 7. Axial power distribution for temperature distribution

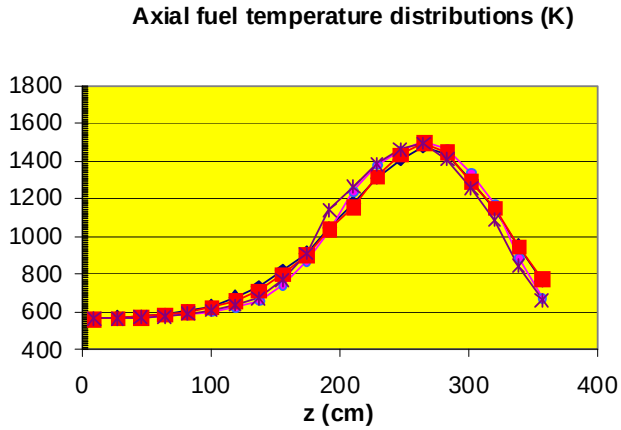
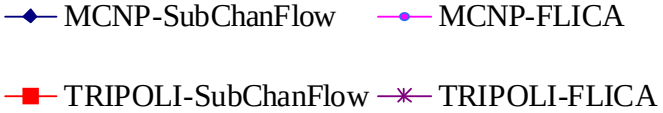


Figure 8. Axial fuel



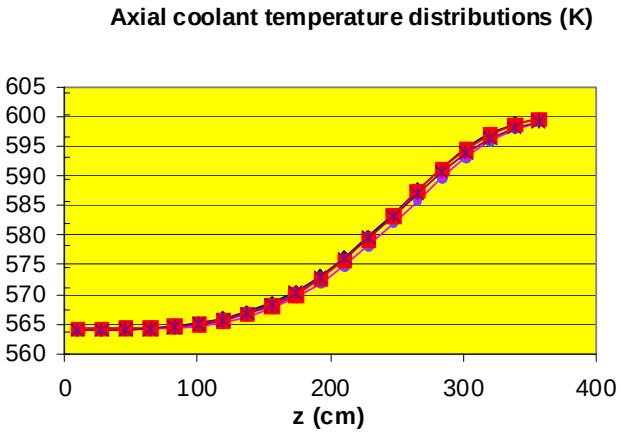


Figure 9. Coolant temperature distribution distribution for

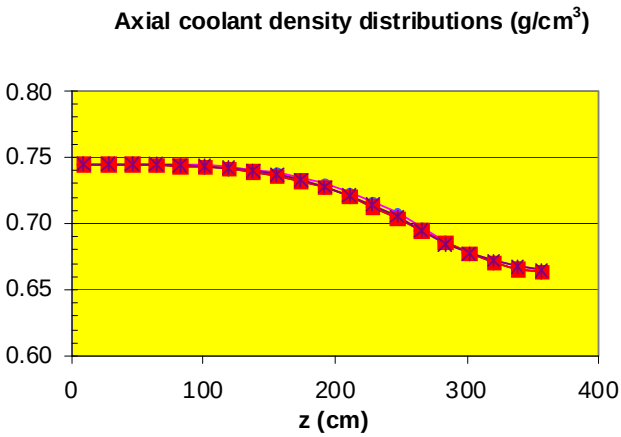
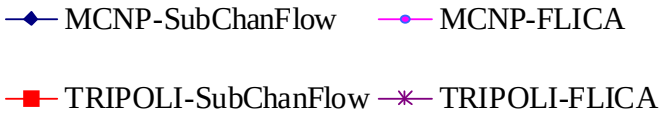


Figure 10. Coolant density



In our calculations both Monte Carlo codes use cross section data based on the JEFF3.1 evaluated nuclear data file. Nonetheless there are differences in the actual cross section data used in the calculations due differences in cross section data processing, different methodology for getting cross sections at the required temperature and possibly different temperatures due to different thermal-hydraulics codes and random fluctuations in the estimated axial power densities. Nonetheless, there is a good agreement between the different combinations of Monte Carlo and thermal-hydraulics codes.

3.2. A 3x3 BWR pin cell cluster

For a more complex situation with different fuel pins and coolant channel we considered a cluster of 3x3 BWR fuel pin cells. Table IV summarizes the most important data. Due to the strongly varying coolant density and void fraction in a BWR 37 axial zones were chosen. The

Table IV. Physical data for 3x3 BWR pin cluster

Total fuel length	3.71 m
Number of axial nodes	37
Inlet coolant temperature Mass flow rate Total power Exit pressure	278.78 ˚C 1.1205 kg/s 0.551 MW 7.06 MPa
Fuel pin radius	0.5375 cm
Pellet radius	0.4555 cm
Pitch of fuel pins	1.43 cm
Fuel enrichment	4.2 %

geometries are shown in Fig. 11.

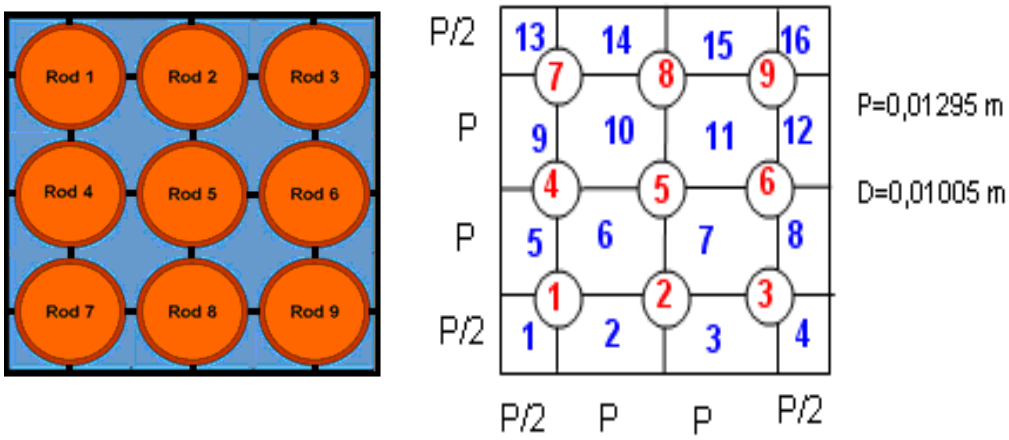


Figure 11. Geometries of 3x3 BWR pin cluster; left: Monte Carlo input; right: thermal-hydraulics input.

Calculations were performed using the TRIPOLI4 and SubChanFlow codes. Fig. 12 shows the final axial power distribution averaged over the 9 fuel pins. Fig. 13 shows the averaged fuel temperature distribution.

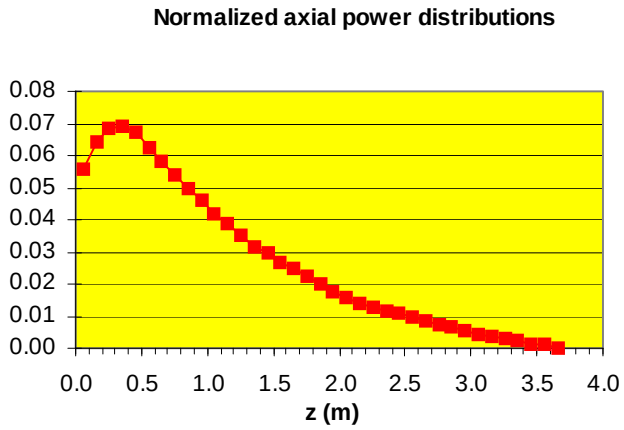


Figure 12. Axial power distribution averaged over all fuel pins

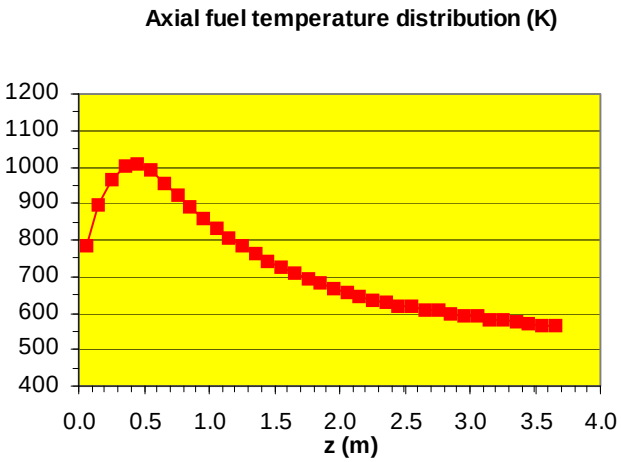


Figure 13. Averaged axial fuel temperature distribution

A more detailed account of this problem and application of the MCNP-SubChanFlow coupling to a steam-cooled fast reactor is given in Ref. 11.

4. FUTURE IMPROVEMENTS

The Monte Carlo-thermal-hydraulics scheme is still under development and a number of improvements and new features may be considered. We will here discuss a number of items that are of interest in that respect.

Of paramount importance in the coupling scheme is the correct representation of cross sections at the requested temperatures. This item was tackled in the MCNP input using an interpolation of cross sections at two available temperatures by defining for each nuclide a pair of nuclide identifiers with appropriate nuclide densities. Although more detailed investigations are necessary, it seems that this method is satisfactory, provided that cross sections are available for a set of temperatures not to far away from each other. For the MCNP data based on the JEFF3.1 evaluation it seems better to have the temperatures at steps of 50 K up to 1000 K and with steps of 100 K above 1000 K up to 2000 K.

For TRIPOLI4 the option to generate the cross section at any requested temperature is very convenient, but at the long run not satisfactory. Hence, the method of interpolation may also be introduced here, although it is more complicated than for MCNP. Instead of introducing pairs of nuclide names for each nuclide with different identification numbers, it will be necessary to introduce for each nuclide a pseudo name and include those in the TRIPOLI4 dictionary file together with a soft link to the basic ENDF file for the nuclide and appropriate soft links to the PENDF file at 0 K and other standard files. Then, if cross sections are actually needed at temperature T, TRIPOLI4 will look for PENDF files for the actual nuclide at temperature T as well as for the pseudo nuclide. Now if T_1 and T_2 are the nearest lower and higher temperatures with respect to T for which cross section data are available, we need to establish before starting a TRIPOLI4 run a temporary soft link for a PENDF file of the actual nuclide with file name including the temperature T to the existing PENDF file at T_1. Likewise, a soft link for the pseudo nuclide with file name including temperature T to the existing PENDF file at T_2. In the input file we need to establish proper nuclide densities according to Eqs. (1) or (2). The TRIPOLI4 code will take a proper average of the cross sections at T_1 and T_2 to represent the cross section for the actual nuclide at T.

From recent literature there is another possibility to obtain a cross section in the resonance region at the required temperature by on-the-fly calculating the cross section resonance broadening with a very fast algorithm [11]. This may also be considered. It requires, however, modifications in the Monte Carlo code itself.

Another problem is the cross section data for the thermal scattering, both in MCNP and in TRIPOLI, which is now taken at the nearest available temperature, which may be up to 25K wrong. Hence, here also the introduction of a pseudo nuclide for each bound thermal scattering will be needed to get an interpolated value. This is possible both in TRIPOLI and in MCNP.

If we do not any longer use the –j option in TRIPOLI we are also able to run both Monte Carlo codes in parallel. This requires some additions in the system call to run either Monte Carlo code. Parallel processing in the Monte Carlo run is of paramount importance in handling larger systems like complete fuel assemblies.

Other items to consider are the iteration scheme. To make the iteration scheme more stable it will be useful to introduce a relaxation factor. A more rigorous method for optimizing the iteration scheme is stochastic optimization [13], which also considers optimum values for the number of histories in successive iterations. It will guarantee stability and find the optimum road to a converged solution.

The convergence test is another issue. Currently the fuel temperature values in a pin from two successive iterations are tested whether they are within a certain range of deviation. It needs further study whether the coolant temperature and coolant density should also be included in a convergence test.
In the input to the Monte Carlo code the number of inactive cycles must be specified. This is now set to a fixed value. It would be nice to have in these code the option that it detects internally when convergence of the source distribution is reached, for instance on basis of the calculated entropy for each cycle or batch. For TRIPOLI it will be much more efficient to be able to use the converged source distribution from the previous iteration as a start in a new iteration. Now in each iteration the same initial source distribution over the length of the fuel pins is used. This takes every time many more batches to reach a converged source distribution. These modifications, however, require a modification of the Monte Carlo code itself. A temporary partial solution can be obtained if the initial axial source distribution specification is updated at every iteration by the axial power distribution from the previous TRIPOLI4 run.
For the 3x3 pin cluster problem producing an input file for either the TRIPOLI and MCNP Monte Carlo codes becomes already quite laborious due to the required separate material definitions for all axial zones of all fuel cells to allow different temperatures for each material. Due to the different specifications there is no advantage in using the repeated structures facilities in either code. Hence, it will be very useful to have a program that generates the input for these Monte Carlo codes in a more automatic way. For TRIPOLI it may be possible to use the ROOT geometry package option of the code to facilitate generating the input geometry.
Finally, we like to mention the possibility of physically better modeling the energy deposition as is possible with the TRIPOLI4 code. Then also energy deposition tallies in the coolant (and possibly in the cladding) need to be included. This only makes sense if the thermal-hydraulics code can take into account a heat source in the coolant and cladding. Another refinement is to use different radial power tally volumes in each fuel rod. One must realize that smaller tally volumes will require more neutron histories to get the same relative standard deviation in estimated power and hence more CPU time.

5. CONCLUSIONS AND DISCUSSION

A flexible scheme for coupling of Monte Carlo and thermal-hydraulics codes have been developed that allow for the substitution of different Monte Carlo codes (at present MCNP and TRIPOLI4) and different thermal-hydraulics codes (at present FLICA4 and SubChanFlow). A Python script drives the iteration process until convergence by successively calling the execution of the thermal-hydraulic and Monte Carlo code. Conversion programs update the input to these codes with results from the last run of the Monte Carlo and thermal-hydraulics code, respectively. Special attention is paid to the representation of the neutron cross section data at the requested temperature in each zone in the geometry of the Monte Carlo calculation.
Results are shown for two demonstration problems of different complexity. The infinite lattice problem with axial varying fuel enrichment is calculated by all four possible combinations of Monte Carlo and thermal-hydraulics codes. The results show fair agreement between the various combinations of codes for the axial power distribution as well as the axial fuel and coolant temperature and coolant density distributions. A few details have to be investigated further.
The second demonstration case consists of a 3x3 BWR fuel pin cluster. For this problem 16 coolant channels have to be considered. The boiling conditions in each channel are automatically

taken care of by the thermal-hydraulics code. The results show that problems of this size can well be tackled by the coupling scheme, especially if the Monte Carlo code can be run in parallel.
As the coupling scheme is still under development there are several issues that should be addressed for further improvement. These items are discussed in Sect. 4. From those should be mentioned here the improvements in representation of cross section data at all relevant temperatures during the Monte Carlo calculations. To this end several options are mentioned. Moreover, to take full advantage of the parallel processing capabilities of the Monte Carlo codes is also a very important issue.
As a coupled Monte Carlo thermal-hydraulics calculation is primarily intended for verification and validation of approximate deterministic calculations, it is desirable to provide the best physical modeling in the Monte Carlo calculation. Improvements are possible here with regard to the estimation of the deposited energy, which is not only in the fuel, but also in the cladding and coolant. It is therefore concluded that a coupling scheme for Monte Carlo and thermal-hydraulics calculations as presented in this paper promises to be a very useful tool for validation of such calculations.
For larger systems with many fuel pins, like a 17x17 pin PWR fuel assembly or clusters of assemblies and certainly a full core (which will be out of our practical possibilities for quite some time) further optimization is necessary. For such systems other ways for assigning material specifications and temperatures to the many zones in the geometry of the input file to the Monte Carlo code have to be developed. This may require a considerable redesign of the Monte Carlo codes.

ACKNOWLEDGMENT

This work is partially funded by the Integrated Project NURISP (contract n° 232124) in the 7th Euratom Framework Programme of the European Union.

REFERENCES

1. V. Sanchez and A. Al-Hamry, “Development of a Coupling Scheme between MCNP and COBRA-TF for the Prediction of the Pin Power of a PWR Fuel Assembly,” *Proceedings of the M&C2009 Conference*, Saratoga Springs, New York, USA, May 3-7, 2009, on CD_ROM, American Nuclear Society, Lagrange Park, IL, USA (2009).
2. D.P. Griesheimer *et al*., “An Integrated Thermal Hydraulic Feedback Method for Monte Carlo Reactor Calculations, “*Proceedings of the Physor2008 Conference*, Interlaken, Switzerland, September 14-19, 2008, on CD_ROM (2008).
3. D. Kotlyar, E. Fridman and E. Shwageraus, “Coupled Neutronic Thermo-Hydraulic Analysis of Full PWR Core with BGCore System,” *Proceedings of the Annual Meeting of Nuclear Technology*, Dresden, Germany (2009).
4. X-5 Monte Carlo team, “MCNP - A General Monte Carlo N-Particle Transport Code, Version 5,” Technical Report LA-UR-03-1987, Los Alamos National Laboratory (2008).
5. TRIPOLI4 version 6 team, “TRIPOLI4 version 6 User Guide,” Report SERMA/LTSD/RT/08-4557/A, Commissariat à l’Énergie Atomique, Saclay, France (2009).

6. S. Aniel et al., “FLICA4: Status of numerical and physical models and overview of Applications,” *Proceedings 11th International Topical Meeting on Nuclear Thermal-Hydraulics NURETH-11*, Avignon, France, October 2-6, 2005.
7. B. Chanaron, ”The European Project NURISP for Nuclear Reactor Simulation,” *Proceedings of the ANS Conference*, Washington DC?, USA (2010).
8. V. Sanchez, U. Imke, A. Ivanov, R. Gomez; SUBCHANFLOW: A Thermal-Hydraulic Sub-Channel Program to Analyse Fuel Rod Bundles and Reactor Cores, *Proceedings of the 17th Pacific Basin Nuclear Conference*, Cancún, Q.R., México, October 24-30 (2010).
9. R.E. MacFarlane and D. Muir, “The NJOY Nuclear Data Processing System Version 91,” report LA-12740-M, Los Alamos National Laboratory, Los Alamos, NM, USA (1994).
10. F. Brown, W.R. Martin, R.D. Mosteller, “Monte Carlo – Advances and Challenges”, Workshop at PHYSOR-2008, Interlaken, Switzerland, 14-19 September 2008, Report LA-UR-08-05891, Los Alamos National Laboratory; http://mcnp-green.lanl.gov/publication/mcnp_publications.html (2009).
11. A. Ivanov; U. Imke and V. Sanchez, “Development of a Coupling Scheme Between MNCP5 and Subchanflow for the Pin- And Fuel Assembbly-Wise Simulation of LWR and Innovative Reactors,” this conference (2011).
12. G. Yesilyurt and W. R. Martin, “On-the-fly Doppler Broadening for Monte Carlo Codes,” *Proceedings of the M&C2009 Conference*, Saratoga Springs, New York, May 3-7, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2009).
13. D. van Veen and J.E. Hoogenboom, “Efficiency Improvement of Local Power Estimation in the General Purpose Monte Carlo Code MCNP,” *Proceedings of the SNA+MC2010 Conference*, Tokyo, Japan, October 17-21, 2010.
14. J. Dufek and W. Gudowski, “Stochastic Approximation for Monte Carlo Calculations of Steady-State Conditions in Thermal Reactors,” *Nuclear Science and Engineering* **152**, pp. 274-283 (2006).