COMPARATIVE STUDY OF DRAGON AND TRIPOLI FOR A PWR ASSEMBLY

C. Camand and G. Marleau
École Polytechnique de Montréal, Québec, Canada

Abstract

Neutronic simulation codes are essential to understand the behaviour of nuclear reactors. These codes use deterministic or stochastic methods to solve the Boltzmann equation for the flux and effective multiplication constant. In this paper, we compare the results of the deterministic code DRAGON and the Monte Carlo code TRIPOLI for the simulation of a PWR assembly in two situations: when the control rods are out of the assembly and when they are used as a neutron absorber. We will compare the two methods and discuss the selection of adequate simulation parameters in order to obtain coherent models for the two codes. Here, we concentrate our work on the calculations of the effective multiplication factor and the scalar flux.

1. Introduction

Numerical simulation is an essential part of nuclear reactor analysis. It provides an understanding of phenomena that take place in a reactor as well as the behaviour of the population of neutrons in the core. It generally requires the solution to the Boltzmann transport equation, such solutions being obtained using either deterministic or Monte Carlo numerical simulation techniques. [1]

The deterministic methods solve directly the Boltzmann equation using some approximations, like the multigroup formalism for the energy treatment or a numerical differentiation procedure for the spatial dependence of the flux. As a consequence, the physical values calculated contain inherent errors. We can quote the collision probability method, the method of characteristic, or the spherical harmonics as some of the most used deterministic methods. [1]

The stochastic methods, usually called the Monte Carlo methods, are based on statistical theories. [2] They solve the transport equation by simulating a large number of neutron histories and recording their interactions with materials present in the reactor regions under investigation. They can use continuous energy or multigroup cross sections and treat complex geometries. They do not contain numerical errors, but the statistical sampling procedure contains an intrinsic statistical error. That is why results of the calculations are always given with their standard deviation. An increase in the number of neutrons simulated decreases the statistical error, and improves the accuracy of the computation. Because it generally takes a very large number of histories to simulate adequately transport problems, the computation time of Monte Carlo calculation is often larger than for deterministic computations. This is why it is usual to use Monte Carlo codes only for reference calculations.

In this paper we present simulations of a PWR assembly using these two methods. Section 2 introduces the DRAGON [3] and TRIPOLI [4] codes, which are used in this work, while in Section 3 we present the two configurations of the PWR [5] assembly studied. Then in Section 4 we discuss some simulation hypothesis and we present results from simulations of this assembly in Section 5. Finally, in Section 6 we conclude and discuss future work plans that would combine DRAGON and TRIPOLI simulations in order to achieve a fast and efficient calculation scheme.

2. The DRAGON and TRIPOLI codes

DRAGON 3.06 [3] is an open source lattice code developed at the Institut de Génie Nucléaire of École Polytechnique de Montréal. It can solve the transport equations for the flux in arbitrary 1-D, 2-D and 3-D geometries. It can also solve the Bateman equations to follow the isotopic contents of the fuel as it burns. This code is composed of several modules that are linked together by the GAN generalized driver [6]. The main components of this software of interest in this paper are:

- Tracking modules to analyse the geometry and generate files containing the integration lines required for collision probability (CP) evaluation or for a solution using the method of characteristics (MOC). The tracking module NXT: was used for our PWR lattice calculations [7],
- SHI: module for the resonance self-shielding of cross sections. It relies on the Stamm'ler method and allows for a Livolant-Jeanpierre normalisation [8],
- ASM: module to compute CP matrices using the tracking files,
- FLU: module to solve the Boltzmann equation using the collision probability method [9],
- MOCC: module to solve the transport equation using the MOC,
- EDI: module to homogenise and condense fluxes and reaction rates.

In this paper we only present simulations using the collision probability method in a stationary case which is the reason why the EVO: module is not mentioned explicitly.

TRIPOLI-4.6 [4] is a three-dimensional Monte Carlo code developed at Commissariat à l'Énergie Atomique (France). It can use both continuous-energy or multigroup cross sections and can treat four types of particles: neutrons, photons, electrons and positrons. The most advanced functionalities of this code are only available for neutron transport. This code is designed to solve two classes of problems: shielding simulations that involve particle propagation in a non multiplying media over long distances and with many orders of magnitude of flux attenuation; and neutronic problems related to the behaviour of particles in multiplying media, like the PWR assembly we deal with in this paper.

3. The PWR assembly

Our PWR assembly [5] is a lattice of 17×17 pin cells. Of the 289 cells in this assembly, 264 contain a fuel pin while the remaining 25 are empty and contain only a guide tube. In some circumstances, control rods can be inserted into these guide tubes. Only the central pin remains empty in this configuration. Figure 1 shows the geometries we are considering here.

The fuel is composed of uranium oxide (UO2) with 3.7% enrichment in Uranium 235. The moderator is light water containing 1100 ppm of natural boron. The control rods are made of natural silver, cadmium and indium, and are used to absorb neutrons [5].

Because we only have cross sections at 300 K for TRIPOLI, all temperatures are taken equal to 300 K. We use the ENDF/B-VI evaluation [10] for both our TRIPOLI and DRAGON simulations.

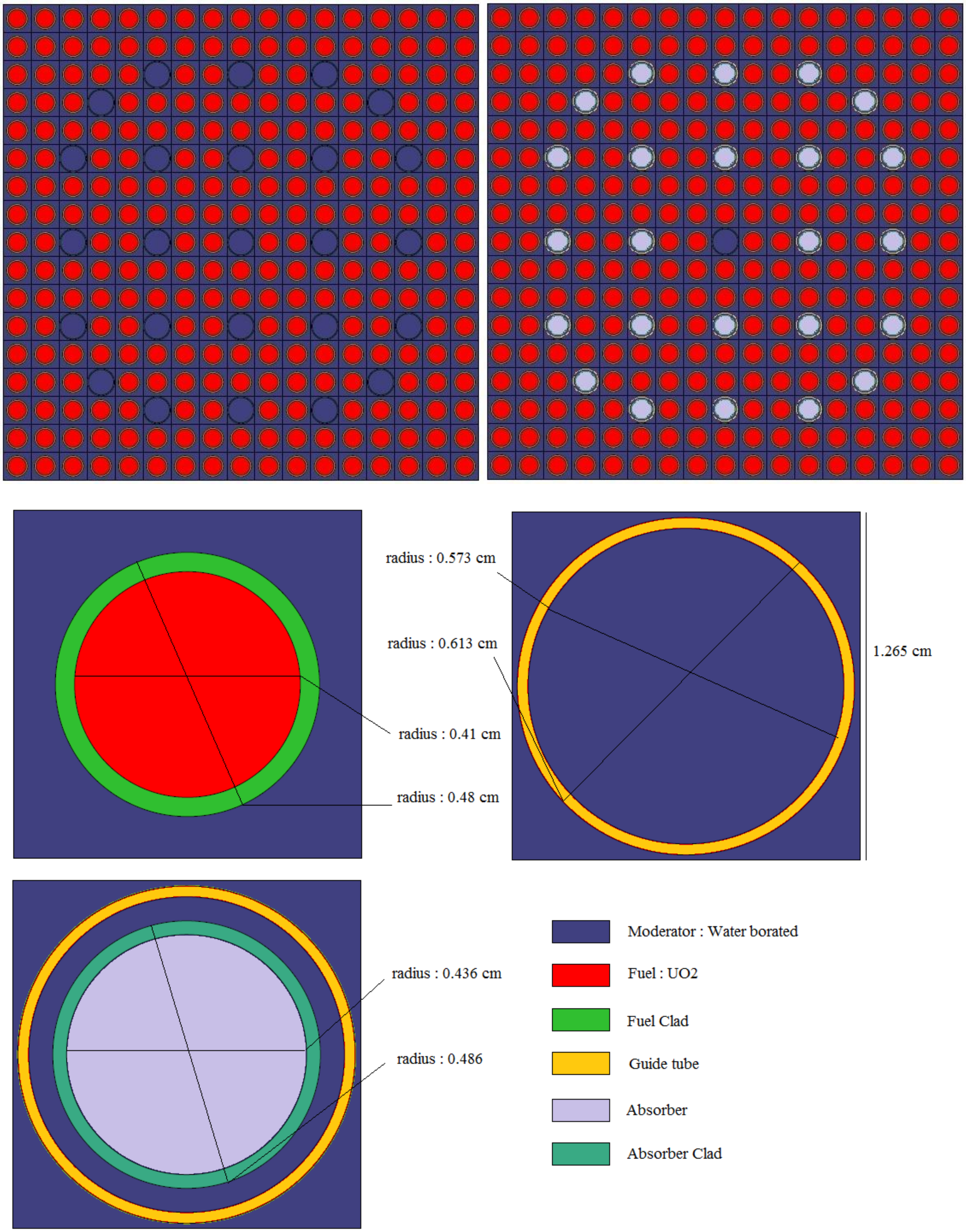

Figure 1 PWR assemblies (control rods out and inserted), fuel cell, empty water hole, and water hole with control rod inserted

4. Choices of simulation methodology

For our DRAGON calculations we choose to solve the Boltzmann equation with the collision probability method, using a Livolant-Jeanpierre normalisation during the self-shielding treatment. The NXT: module is chosen for the tracking because it allows us to define our assembly by blocks, each block corresponding to a cell (pin or guide tube, empty or with a control rod), and because the cell by cell homogenization for the flux, which is the most suitable homogenization option for data analysis, is only available with this module.

In order to be as accurate as possible, we choose to split cells annularly. We also consider a Cartesian mesh splitting, particularly in the empty cells filled with moderator, in order to better simulate scattering in light water. This choice increases substantially the number of regions and the calculation time should increase significantly. In order to have a reasonable computation time, an option is to use the two diagonal symmetries and the two median symmetries of the assembly geometry. We define our geometry as 1/8 of the original assembly with symmetric boundary conditions on these surfaces. With these special options, the tracking lines correspond to an unfolding of the geometry on the full assembly. However, the dimensions of the collision probability matrix are reduced substantially. The results will be exactly the same, but the eighth of assembly allows a gain of time. These options allow us to simplify and accelerate the calculation.

For TRIPOLI simulations the same simplifications are not available. Here, we generate the entire assembly and we simulate neutrons as they travel through it. As we saw in the previous Section, the number of neutrons simulated influences considerably the accuracy and computation time. One therefore has to be careful and find a good compromise between these two aspects. We will analyze in the next section the influence of the number of neutrons on the results. We also selected three different types of neutron source initialization: a punctual source in a corner, a punctual source located at the center, and a factorized source in each fuel pin. We will comment latter on the impact of the initial source on final results.

As we stated before both codes use cross sections taken from the ENDF/B-VI evaluation. With DRAGON, we used 69 and 172 groups cross sections libraries. We were able to analyse the impact of the number of groups on simulation results. With TRIPOLI we selected the continuous-energy cross sections, which are closer to the reality than multigroup cross sections, because the TRIPOLI model will be the reference in our future works.

5. Results

We present here effective multiplying coefficient (k_{eff}) and scalar flux calculations for the two PWR assembly selected.

5.1 Computation of the effective multiplying coefficient

We first analyse results on the PWR assembly with the control rods out. We use three TRIPOLI estimators for k_{eff} calculations [11]:

- the KSTEP estimator records the number of neutrons generated by fission:

$$K = \nu \tag{1}$$

- the KCOLL estimator calculates the mean number of fission neutrons produced at each interaction with a nucleus:

$$K = \nu(i) \frac{\sigma_f(i)}{\sigma_t(i)} \quad (2)$$

- the KTRACK estimator calculates for each path within a fissile mixture:

$$K = l \sum_i \nu(i) N_i \sigma_f(i) \quad (3)$$

For each simulation including a k_{eff} calculation, these three estimators are automatically and simultaneously used. Their standard deviations may not decrease at the same speed, but the computation time is the same for a given simulation.

Table 1 show the values of the k_{eff} for a simulation of 1 000 batches of 10 000 neutrons per batch for the three source initialization types. Standard deviations are given in pcm (10^{-5}).

Table 1: k_{eff} for different source initialization in TRIPOLI (10 000 neutrons per batch)

Estimator	Corner source		Center source		Distributed source	
	keff	Standard deviation (in pcm)	keff	Standard deviation (in pcm)	keff	Standard deviation (in pcm)
KSTEP	1.23953	37	1.23859	37	1.23999	38
KCOLL	1.23949	36	1.23856	36	1.23969	36
KTRACK	1.23949	49	1.23961	48	1.23856	49

One can notice that the KCOLL estimator converges faster than the others. We observe differences of around 100 pcm between two simulations even if the standard deviation remains below 50 pcm. The type of source initialization should not have, in principle, any impact on the final result when the problem is converged. We therefore have to refine these results.

By increasing the number of neutrons per batch (100 000 instead of 10 000), we obtain the results presented in Table 2.

Table 2: k_{eff} for different source initialization in TRIPOLI (100 000 neutrons per batch)

Estimator	Corner source		Center source		Distributed source	
	keff	Standard deviation (in pcm)	keff	Standard deviation (in pcm)	keff	Standard deviation (in pcm)
KSTEP	1.23897	12	1.23910	12	1.23903	12
KCOLL	1.23894	11	1.23909	11	1.23904	11
KTRACK	1.23893	15	1.23918	15	1.23886	15

The KCOLL estimator is still the fastest to converge, and the standard deviation is divided by $\sqrt{10}$, which corresponds to multiplying by 10 the number of histories sampled. We notice a maximum difference of 16 pcm between two simulations, which is similar to the standard deviation of the results. We can now neglect this difference as we have showed that the value of the k_{eff} does not depend on the initial distribution of the sources when a sufficient number of neutrons are simulated.

Let us now compare these results with the k_{eff} calculated by DRAGON. As the KCOLL estimator is the fastest to converge, we use the k_{eff} computed by this estimator in our comparison (see Table 3).

Table 3: Comparison of TRIPOLI and DRAGON k_{eff} (control rods out)

	Corner source		Center source		Distributed source	
	keff	Difference (pcm)	keff	Difference (pcm)	keff	Difference (pcm)
TRIPOLI (KCOLL)	1.23894	-	1.23909	-	1.23904	-
DRAGON (69 groups)	1.23770	-123.9	1.23770	-139.8	1.23770	-134.2
DRAGON (172 groups)	1.24066	172.2	1.24066	156.3	1.24066	161.9

A difference of around 150 pcm between the two codes is observed. With the 69 energy groups library DRAGON underestimates the k_{eff} with respect to TRIPOLI. This effect is inverted with the 172 groups library for a difference of 300 pcm between the 69 and 172 groups DRAGON k_{eff}. We also observe this difference with other cross section libraries (ENDF/B-VII and JEF 3.1). This seems to be a systematic effect. Although we would expect TRIPOLI to be closer to the DRAGON 172 group results, it falls nearly dead centre between our two sets of DRAGON results. This could indicate that error compensation play an important in the DRAGON simulation. On the other hand, the differences being relatively small, one can conclude that these results are acceptable and that the DRAGON and TRIPOLI models of the PWR assembly without the control rods are coherent in regard of the k_{eff} calculation.

We consider now the PWR assembly with the control rods inserted. Taking into account the previous conclusions, we directly consider the k_{eff} given by the KCOLL estimator for a simulation of 100 000 neutrons per batch. We also showed that the initial source had no impact on the final results, a calculation with only one source distribution is sufficient. For the same number of neutrons simulated, the punctual source located in the center gave us the best computation time, so we choose it as the initial source.

Table 4: Comparison of TRIPOLI and DRAGON k_{eff} (control rods in)

	keff	Difference (in pcm)
TRIPOLI (KCOLL)	0.92864	-
DRAGON (69 groups)	0.92536	-328
DRAGON (172 groups)	0.92809	-55

We observe a smaller difference than for the assembly with empty water holes between TRIPOLI and 172 group DRAGON calculations. As the 172 groups cross sections are the closest to continuous-energy cross sections, this low difference is a good point to notice. Our model with control rods inserted confirms the coherence of our k_{eff} calculations.

5.2 The flux

The scalar fluxes are homogenized by cell and condensed over two energy groups. The limit between the two groups is at 0.625 eV and we will analyze the fast and thermal fluxes in each cell of the assembly.

Before comparing DRAGON and TRIPOLI fluxes, we have to normalize the DRAGON flux with respect to TRIPOLI. To this end, we use:

$$\phi_{i,g}^{norm}(D) = \phi_{i,g}^{calc}(D) \times \frac{\sum_i \sum_g (V_i(T) \times \phi_{i,g}^{calc}(T))}{\sum_i \sum_g (V_i(D) \times \phi_{i,g}^{calc}(D))} \tag{4}$$

with:

- $\phi_{i,g}^{norm}(D)$ the DRAGON normalized flux
- $\phi_{i,g}^{calc}(D)$ the flux calculated by DRAGON
- $\phi_{i,g}^{calc}(T)$ the flux calculated by TRIPOLI
- $V_i(D)$ the volume of the region i considered in DRAGON
- $V_i(T)$ the volume of the region i considered in TRIPOLI

We then calculate the differences between normalized DRAGON and TRIPOLI fluxes:

$$DIFFERENCE\ (\%) = 100 \left(\frac{\phi_{i,g}^{norm}(D) - \phi_{i,g}^{calc}(T)}{\phi_{i,g}^{calc}(T)} \right) \tag{5}$$

First we can observe that the number of energy groups has a minor influence on the fluxes: the changes between 69 and 172 groups are tiny and have no significant impact on the difference between fluxes given by the two codes.

We display on figure 2 the flux distributions over the assembly and the difference between DRAGON and TRIPOLI fluxes (fluxes from DRAGON are presented for 172 energy group computations):

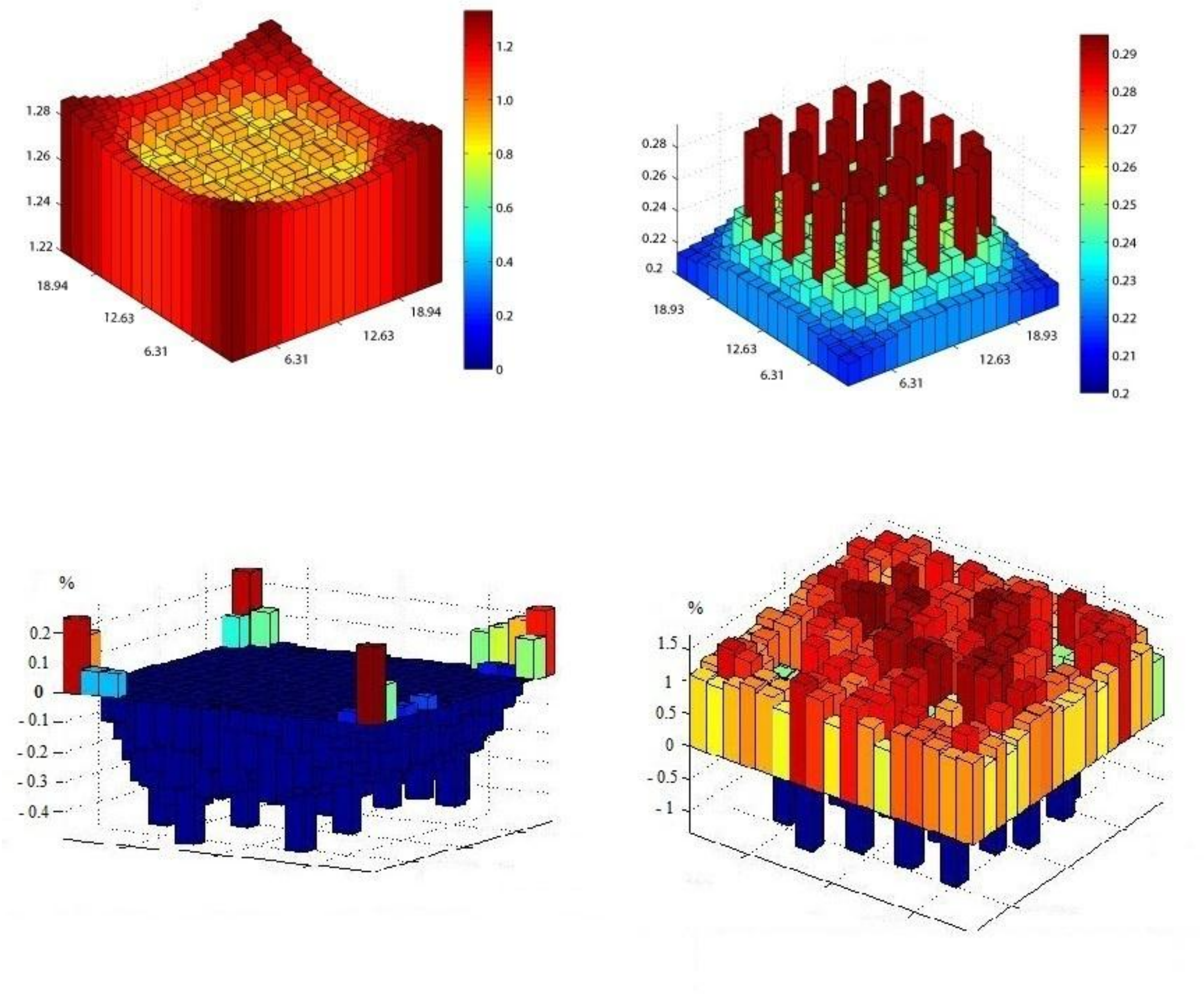

Figure 2 PWR Assembly with no control rods: DRAGON fast flux (top left), DRAGON thermal flux (top right), % difference between DRAGON and TRIPOLI for fast flux (bottom left) and % difference between DRAGON and TRIPOLI for thermal flux (bottom right)

We observe that the fast fluxes of DRAGON and TRIPOLI are very close, over the assembly. A maximum underestimation of TRIPOLI of 0.4% is observed, except close to the corners. The differences are more visible in the thermal domain reaching around 1.5% in the pin cells. This difference is not sensitive to the Cartesian splitting of the cells in DRAGON.

On the other hand, we observed a noticeable sensitivity on the splitting on the water holes. Without splitting, the difference is close to 1.7%, and decreases when we refine the Cartesian splitting. We can explain this behaviour by the fact that empty cells contain light water, and with the collision probability method we do not consider flux anisotropy. We therefore have to discretize more these regions in order to be more accurate on the estimation of the flux.

We now have found the good parameters in order to have coherent flux values between DRAGON and TRIPOLI for the PWR assembly without control rods. Now let us consider the assembly with control rods inserted and display the same distributions as above (see Figure 3).

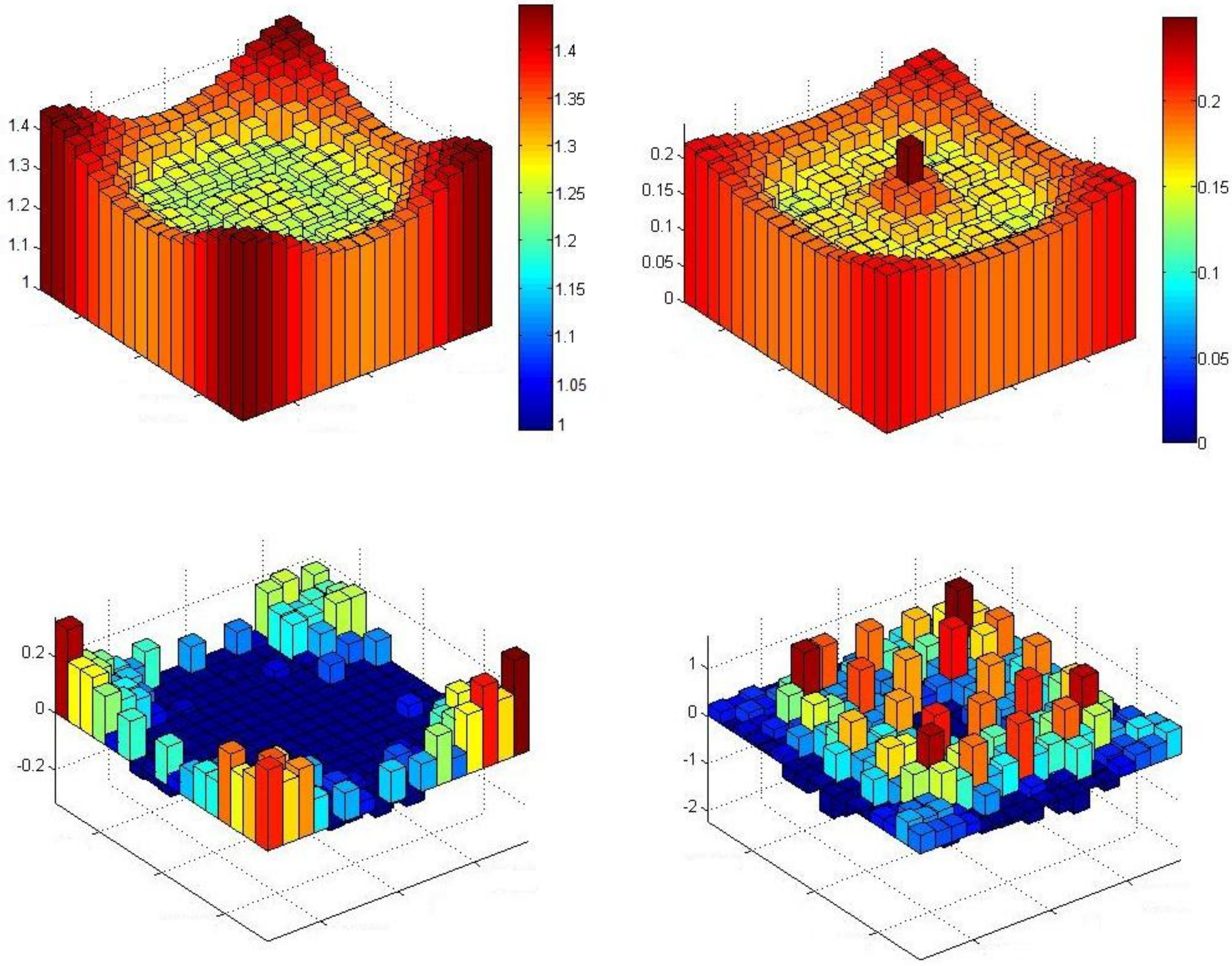

Figure 3 PWR Assembly with control rods inserted: DRAGON fast flux (top left), DRAGON thermal flux (top right), % difference between DRAGON and TRIPOLI for fast flux (bottom left) and % difference between DRAGON and TRIPOLI for thermal flux (bottom right)

We observe that TRIPOLI and DRAGON fast fluxes are again very close: a maximum of 0.4% of difference between the two codes is notable. For the thermal fluxes the main difference appears in the central cell: 2% between DRAGON (without splitting) and TRIPOLI. Cartesian splitting has the same effect as before on the water hole. However, mesh splitting does not have a significant impact on the neutron absorber rods, where the difference remains around 1.5%, or the fuel pins where the difference is even lower.

These differences are low enough to conclude that our DRAGON and TRIPOLI models of the assembly with control rods inserted are coherent.

6. Conclusion and future work

The purpose of this work was to compute a PWR assembly with the deterministic code DRAGON and with the Monte Carlo code TRIPOLI. We have focused our comparison on the effective multiplication factor k_{eff} and the scalar fluxes (fast and thermal).

First we studied the k_{eff} in order to select the number of particles to simulate in TRIPOLI in order to ensure that the problem convergence is independent of the choice of the initial source. We obtained coherent results between DRAGON and TRIPOLI. The differences observed seem to be caused, in part, by the difference of the formalism used for the energy treatment (multigroups for DRAGON, continuous for TRIPOLI).

Finally we studied the scalar fluxes. We were able to find coherent flux distributions between DRAGON and TRIPOLI. The differences observed remain below 1.5% and can be explained by the differences of flux calculation techniques. We improved the coherence of flux distributions in water holes using a Cartesian mesh splitting. This technique was less successful on control rods and fuel pins. We can therefore conclude that the PWR assembly models we used for DRAGON reproduce very well the predictions of TRIPOLI. The addition of control rods in the assembly made the DRAGON and TRIPOLI PWR assembly results even closer.

The next step of the project is to develop a coupling between DRAGON and TRIPOLI. The objective is to use DRAGON to generate a source file that could be used to initialize the source distribution for a TRIPOLI simulation. This would allow the first batches to be significant in the Monte Carlo calculation, and to improve the convergence of the stochastic simulation process. Having generated coherent DRAGON and TRIPOLI models ensures that the DRAGON source file will provide a realistic initial neutron source distribution for TRIPOLI.

7. Acknowledgments

This work was supported in part by Électricité de France (EDF) and the Natural Science and Engineering Research Council (NSERC) of Canada.

8. References

[1] A. Hébert, *Applied Reactor Physics*. Montréal, Canada: Presses internationales Polytechnique, (2009).

[2] I. Lux, L. Koblinger, *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*. CRC Press, Boston (1990).

[3] G. Marleau, A. Hébert, R. Roy, *A User Guide for DRAGON 3.06*, Rapport IGE-174R8, École Polytechnique de Montréal (2008).

[4] O. Petit, F.X. Hugot, Y.K. Lee, C. Jouanne, A. Mazzolo, *TRIPOLI-4 Version 6 User Guide*, Report CEA-R-4556, DSI, CEA/Saclay, France (2009).

[5] Anne Barreau, *Burn-up Credit Criticality Benchmark: Phase 2-D. PWR-UO2 Assembly. Study of Control Rod Effects on Spent Fuel Composition*, Nuclear Energy Agency report (2006). http://www.oecd-nea.org/science/nea6227-burnupIID.pdf

[6] R. Roy, A. Hébert, *The GAN Generalized Driver*, Report IGE-158, Institut de génie nucléaire, École Polytechnique de Montréal, Montréal, Québec (2000).

[7] G. Marleau, *New Geometries Processing in DRAGON: The NXT: Module*, Technical Report IGE-260, École Polytechnique de Montréal (2006).

[8] A. Hébert, G. Marleau, "Generalization of the Stamm'ler Method for the Self Shielding of Resonant Isotopes in Arbitrary Geometries_," *Nucl. Sci. Eng.*, 108, 230 (1991).

[9] G. Marleau, *DRAGON Theory Manual Part 1: Collision Probability Calculations*, Technical Report IGE-236 Rev. 1, École Polytechnique de Montréal (2001).

[10] *ENDF-201, ENDF/B-VI Summary Documentation, National Nuclear Data Center (NNDC) at Brookhaven National Laboratory (BNL)*. BNL/NNDC BNL-NCS-17541 (2000), available in http://www.nndc.bnl.gov.

[11] B. Roesslinger, *TRIPOLI-4: Code de transport de particules par la méthode Monte Carlo. Principes théoriques*, Report CEA-R-3770, CEA/Saclay, France (2005).