7th International Conference on Modelling and Simulation in Nuclear Science and Engineering (7ICMSNSE)
Ottawa Marriott Hotel, Ottawa, Ontario, Canada, October 18-21, 2015

7ICMSNSE-054

OpenMC - TD, A New Module for Monte Carlo Time Dependent Simulations Used to Simulate a CANDU6 Cell LOCA Accident

M. Mahjoub[1] and J. Koclas[1]
[1] École Polytechnique de Montréal, Québec, Canada
mehdi.mahjoub@polymtl.ca, jean.koclas@polymtl.ca

Abstract

The solution of the time dependent Boltzmann equation remains quite a challenge. For transient problems, the two steps approach is still used, with the nuclear reactor data obtained using an adiabatic assumption, whereby perturbed cross-sections are calculated from static computations.

We developed a new method using the stochastic Monte Carlo approach to solve the time dependent transport equation. There are two reasons for this choice. First, at the cell level, we will be able to obtain time dependent homogenized cross sections for use in full core diffusion calculations. Second, the Monte Carlo methods are scalable to perform full core if and when appropriate computer resources become available.

The OpenMC code developed at MIT was used as the initial platform for our work, by adding time dependent modules. A particular attention was given to the treatment of delayed neutrons during the simulation.

Our results show that the adiabatic assumption for cross sectionsshould be reviewed and the time dependent Monte Carlo method is a significant alternative for the solution of time dependent neutron transport problems.

Keywords: Monte Carlo, Neutronic, Time dependent, CANDU6, LOCA, OpenMC.

1. Introduction

Predicting with efficiency the behavior of the reactor core during an accident is important. In neutronics, it is still difficult to simulate the entire reactor with its fine details. The approach is then to solve the Boltzmann equation for a small region of the reactor to obtain spatially homogenized and energy condensed properties of this region. These properties are subsequently used in a diffusion simulation to obtain the flux distribution over the entire reactor core.

However, software and hardware technologies are sufficiently developed to envisage solving the time dependent Boltzmann equation for the full reactor, perhaps in a few years from now. In the mean time, solving the time dependent Boltzmann equation for a unique cell, and using the results in diffusion calculation is very much attainable. In this regard, the Monte Carlo method is a good candidate to provide the time dependent parameters.

Time dependent Monte Carlo codes already exist, for example TART [1], G-Stork [2] and Serpent-2 [3]. The first one, TART, takes into account the delayed neutrons but does not allow any modification in the geometry definition. The two other codes do not involve delayed neutrons but allow some small changes in the cell properties such as temperature of materials.

We are developing a new time dependent Monte Carlo code, OpenMC-TD, that allows dynamic modifications of the cell properties and takes into account delayed neutrons both at the start, and during, the whole simulation process.

Using OpenMC-TD, we are able to perform a time dependent simulation for a CANDU6 cell for a small interval of time and obtain the time dependent homogenized and condensed cross sections. These properties can be compared to the static homogenized and condensed cross section used in a transient simulation with the adiabatic assumption.

2. Two Steps Scheme

The time dependent transport Boltzmann equation in standard notation is:

$$\begin{aligned}\frac{1}{V}\frac{\partial}{\partial t}\phi(\vec{r},E,\vec{\Omega},t) = & -\Sigma_t(\vec{r},E,t)\,\phi(\vec{r},E,\vec{\Omega},t) \\ & -\vec{\Omega}.\vec{\nabla}\phi(\vec{r},E,\vec{\Omega},t) \\ & +\int_0^\infty dE'\int_0^{4\pi}\Sigma_s\left(\vec{r},E'\to E,\vec{\Omega'}\to\vec{\Omega},t\right)\phi\left(\vec{r},E',\vec{\Omega'},t\right)d^2\Omega \\ & +(1-\beta)\chi_p(E)\int_0^\infty dE'\int_0^{4\pi}\nu(E')\,\Sigma_f(\vec{r},E',t)\phi(\vec{r},E',\vec{\Omega},t)d^2\Omega \\ & +\sum_{k=1}^{D}\lambda_k C_k(\vec{r},t)\chi_{dk}(E)\end{aligned} \tag{1}$$

where the concentration of precursors is obtained by:

$$\frac{\partial}{\partial t}C_k(\vec{r},t) = -\lambda_k C_k(\vec{r},t) + \beta_k\int_0^\infty dE'\int_0^{4\pi}\nu(E')\,\Sigma_f(\vec{r},E',t)\phi(\vec{r},E',\vec{\Omega},t)d^2\Omega \tag{2}$$

(k=1..D)

By removing the time dependency, condensing energy in groups, the steady state transport equation is obtained:

$$\begin{aligned}\Sigma_{t,g}(\vec{r})\,\phi_g(\vec{r},\vec{\Omega}) + \vec{\Omega}.\vec{\nabla}\phi_g(\vec{r},\vec{\Omega}) = & +\sum_{g'=1}^{G}\int_0^{4\pi}\Sigma_{s,g'\to g,\vec{\Omega'}\to\vec{\Omega}}(\vec{r})\phi_{g'}(\vec{r},\vec{\Omega'})d^2\Omega \\ & +\frac{1}{K_{eff}}\chi_g\sum_{g'=1}^{G}\nu_{g'}\Sigma_{f,g'}(\vec{r})\phi_{g'}(\vec{r})\end{aligned} \tag{3}$$

Numerous methods have been developed to resolve the steady state Boltzmann equation. They can be classified in two major groups, the deterministic and stochastic (Monte Carlo). DRAGON [4] is one of the deterministic codes. It uses collision probability method in 2D and 3D. It can also use the Method

Of Characteristics (MOC) or the **Sn** method to solve the transport equation. MCNP [5], Serpent [6], OpenMC [7] and TRIPOLI [8] are Monte Carlo codes that can solve the transport equation.

Results of such codes can be used in diffusion codes, such as DONJON [9], to estimate the behaviour of the reactor core. Diffusion codes solve the diffusion equation obtained from the transport equation using the Fick law:

$$\begin{aligned}\frac{1}{V_g}\frac{\partial}{\partial t}\phi_g(\vec{r},t) = \; & -\Sigma_{t,g}(\vec{r},t)\,\phi_g(\vec{r},t) + \vec{\nabla}.D_g(\vec{r},t)\vec{\nabla}\phi_g(\vec{r},t) \\ & + \sum_{g'=1}^{G} \Sigma_{s,g\rightarrow g'}(\vec{r},t)\phi_{g'}(\vec{r},t) \\ & + (1-\beta)\chi_{p,g}\sum_{g'=1}^{G} \nu_{g'}\Sigma_{f,g'}(\vec{r},t)\phi_{g'}(\vec{r},t) \\ & + \sum_{k=1}^{K}\lambda_k C_k(\vec{r},t)\chi_{dk.g}\end{aligned} \tag{4}$$

The two steps simulation starts by obtaining the reactor database after solving the steady-state transport equation for a reactor cell and then performing a diffusion calculation for the reactor core.

In transients, the adiabatic approximation for the cross sections is used. This approximation means that the reactor database is obtained by solving the steady state transport equation with parameters describing the instantaneous conditions of the reactor cell. The two steps standard scheme becomes a four steps scheme:

1. Creating libraries of cross sections for stationary undisturbed conditions using a transport code.

2. Performing an initial diffusion simulation.

3. Creating libraries of cross sections for perturbed conditions core using the same transport code.

4. Performing diffusion simulations using the new perturbed data libraries.

Calculating the database reactor using the steady state transport equation is an approximation that may introduce errors in the transient simulation. We present here a new method to obtain the reactor database by solving *the time dependent transport equation* (eq.1) using a stochastic approach.

3. Time Dependent Approach

3.1 Delayed neutrons

OpenMC is still under development and does not contain the isotopic evolution module yet. For delayed neutrons, it is important to sample the decay delay that each particle waits before appearing. The idea is so to use the decay constant groups. We can easily find the probability density function of precursors decay:

$$f(t) = \lambda_k e^{-\lambda_k t} \tag{6}$$

We can calculate the cumulative density function by using the inversion method for sampling. The decay delay D_T is thus obtained by:

$$D_T = \frac{-1}{\lambda_k}\log(r) \quad , r \; is \; a \; random \; number \; \in [0,1], \tag{7}$$

3.2 Tracking Particles

To introduce the time dependent to the procedure of tracking particle, the time variable for each neutron must be managed. It must have an appearing time T_0, a lifetime L_T and a disappearing time T_f. Using the assumption that reactions are instantaneous, the appearing time is equal to the disappearing time of the incident neutron. Each neutron can undergo multiple collisions from appearing and then disappearing by absorption. Some absorptions lead to a fission, (n,2n), (n,Xn), etc. and new neutrons are then created (figure 1).

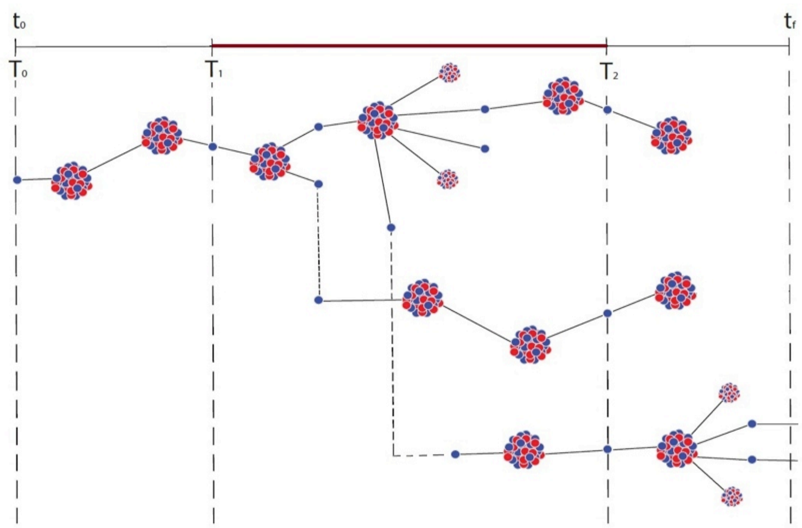

Figure 1 Particles tracking

The lifetime of each neutron can be obtained using the kinetic energy. Since the length of each track between collisions is known, the duration of each travel, t_i, is:

$$t_i = \frac{L_i}{\sqrt{\frac{2E_i}{m_n}}} \tag{8}$$

The total lifetime, L_T, is then obtained by summing all t_i:

$$L_T = \sum_i t_i = \sum_i \frac{L_i}{\sqrt{\frac{2E}{m_n}}} \tag{9}$$

Finally, The disappearing time is the sum of appearing time and duration lifetime. So, for prompt and delayed neutrons respectively we have:

$$T_{0,p}^{n+1} = T_0^n + {L_T}^n \tag{10}$$
$$T_{0,d}^{n+1} = T_0^n + {L_T}^n + D_T^{n+1}$$

where n is the index of the incident neutron, and p and d are symbols respectively for prompt and delayed neutrons.

3.3 Scoring Results

It is important to clarify the method used to score results. The approach used to track neutrons is completely independent of time intervals and neutrons are followed during their travel through the cell without any interruption. The only time that neutrons are stopped is the moment where geometry materials must be changed (such as in accident simulation). Figure 1 shows two different times, T_1 and T_2, where particles are stopped to update geometry properties.

However, the time dependent cross sections should not be a continuous function since they are to be used in a time dependent diffusion code usually based on an Euler solving method. The new approach does respect the continuous aspect of particles travel. The idea is to obtain a set of variables for each small interval of time Δt.

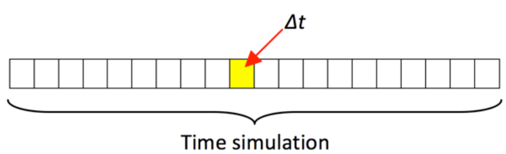

Figure 2 Interval decomposition of time simulation

There are two estimators available for tallies in OpenMC: the ***Analog*** and the ***Track-length*** estimator.

The ***Analog*** estimator counts the number of times that an event happens in each interval of time. The result of this estimator gives the reaction rate. For each interval of time Δt, the flux is obtained by dividing the total reaction rate by the total macroscopic cross section of each material in which the neutrons undergo collisions. To obtain the different homogenized cross sections, first the reaction rate is calculated and then divided by the obtained flux,

$$\phi_{\Delta t_i} = \frac{R_{t,\Delta t_i}}{\Sigma_{t,\Delta t_i}} = \sum_{collisions \in \Delta t_i} \frac{1}{\Sigma_{t,\Delta t_i}}$$

$$R_{x,\Delta t_i} = \Sigma_{x,\Delta t_i} \phi_{\Delta t_i} = \sum_{collisions \in \Delta t_i} \frac{\Sigma_{x,\Delta t_i}}{\Sigma_{t,\Delta t_i}}$$

$$\Sigma'_{x,\Delta t_i} = \frac{1}{\phi_{\Delta t_i}} \sum_{collisions \in \Delta t_i} \frac{\Sigma_{x,\Delta t_i}}{\Sigma_{t,\Delta t_i}}$$

The ***Track-length Estimator*** adds the distances traveled by neutrons in each interval of time. The total distance traveled by neutrons is equal to the integrated flux of particles. Note that a single track of a neutron can belong to two or more interval time of scoring (Figure 3).

By adding tracks, only the integrated flux can be obtained (eq. 12). However, tracks can be weighted by the macroscopic cross section of the material that is crossed by the neutron to obtain the reaction rate (eq 13).

$$V\emptyset = \int dr \int dE \int d\Omega \int_{t_i}^{t_{i+1}} dt\, \phi(\vec{r}, E, \vec{\Omega}, t) \tag{12}$$

$$V\emptyset = \int dr \int dE \int d\Omega \int_{t_i}^{t_{i+1}} dt\, v\, n(\vec{r}, E, \vec{\Omega}, t)$$

$$V\emptyset = \int dr \int dE \int_{l\,\in\,\Delta t} dl\;\, n(\vec{r}, E)$$

So, we obtain in a simplified form:

$$\phi_{\Delta t_i} = \sum_{l_i \in \Delta t_i} l_i$$

$$R_{x,\Delta t_i} = \sum_{l_i \in \Delta t_i} l_i\, \Sigma_{x,\Delta t_i}$$

$$\Sigma'_{x,\Delta t_i} = \frac{1}{\phi_{\Delta t_i}} \sum_{l_i \in \Delta t_i} l_i\, \Sigma_{x,\Delta t_i}$$

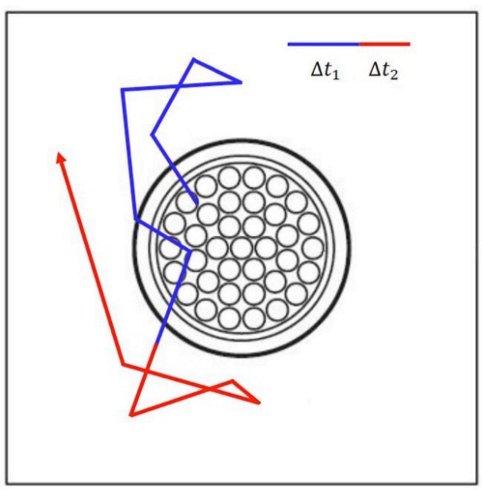

Figure 3 Particle travel throw a CANDU6 cell

5. Transient simulation

For a steady state simulation, where properties of the cell are not varying, the time dependent simulation results should also be constant and must converge to the steady state results. The importance of a time dependent simulation is observed in conditions where the reactor has to react to a reactivity perturbation. Such a condition happens in a LOCA simulation of a CANDU6 reactor where the void coefficient is positive [10].

Such static simulations were performed and results for a fresh-fueled CANDU6 cell were compared to the results of a voided cell using different transport static codes [11]. It was shown that OpenMC results are in a good agreement with Serpent results. A time dependent simulation for fresh fuel CANDU6 cell could not be performed since the effective multiplication factor is much greater than “1.0”. In this condition, the time dependent simulation will just jump to infinity in a few milliseconds.

We therefore adjusted the properties of the CANDU6 cell by adding boron in the moderator to obtain a K-effective closer to unity. These conditions are similar to the first time startup reactor conditions (figure 4).

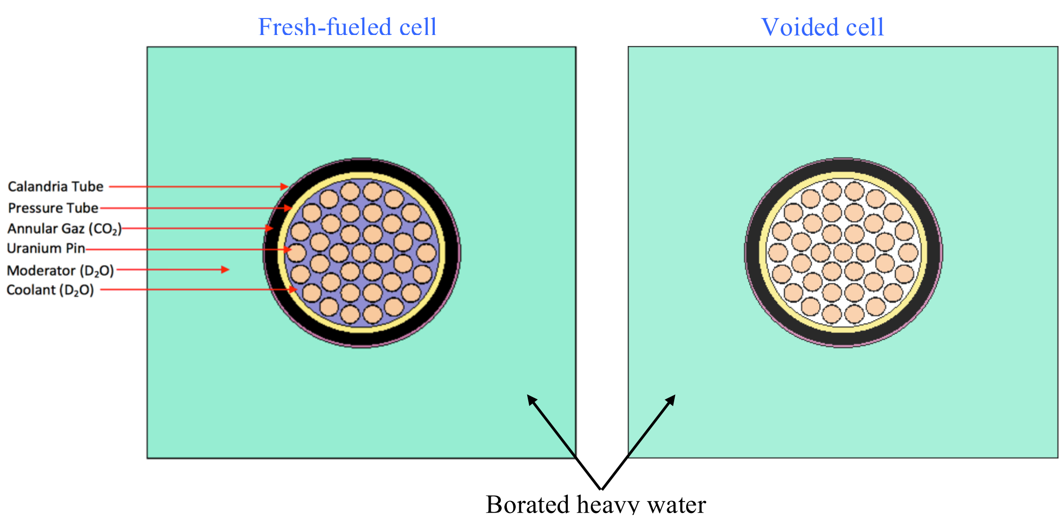

Figure 4 CANDU6 cell

6. Results and Analysis

The simulation conditions used to model a LOCA accident for a CANDU6 cell are:
- Boron 11 added to the moderator to obtain a $K_{eff} = 0.99$ in a static eigenvalue simulation
- Void introduced to replace the coolant at $t^{*}=0.1\ s$
- Total simulation time: $t = 0.2\ s$
- Scoring interval time $\Delta t = 10^{-4}\ s$

The total running time of the simulation was about $\approx 48h$ CPU time with these computer parameters: Processor 2.7 GHz i7, Memory 16 GB 1600MHz DDR3.

The simulation results of interest are neutron density, flux, total cross section, scattering cross section, fission cross section and are presented in figures 5 to 9.

The neutron density behavior depends on the reactivity of the cell. Initially the reactivity of the cell is negative since the K_{eff} is below critical. The cell then becomes super critical. Figure 5 shows that neutron density starts falling before the accident but increases exponentially after the introduction of void. The same behavior is observed for the neutron flux and also for the total reaction rate (figure 6).

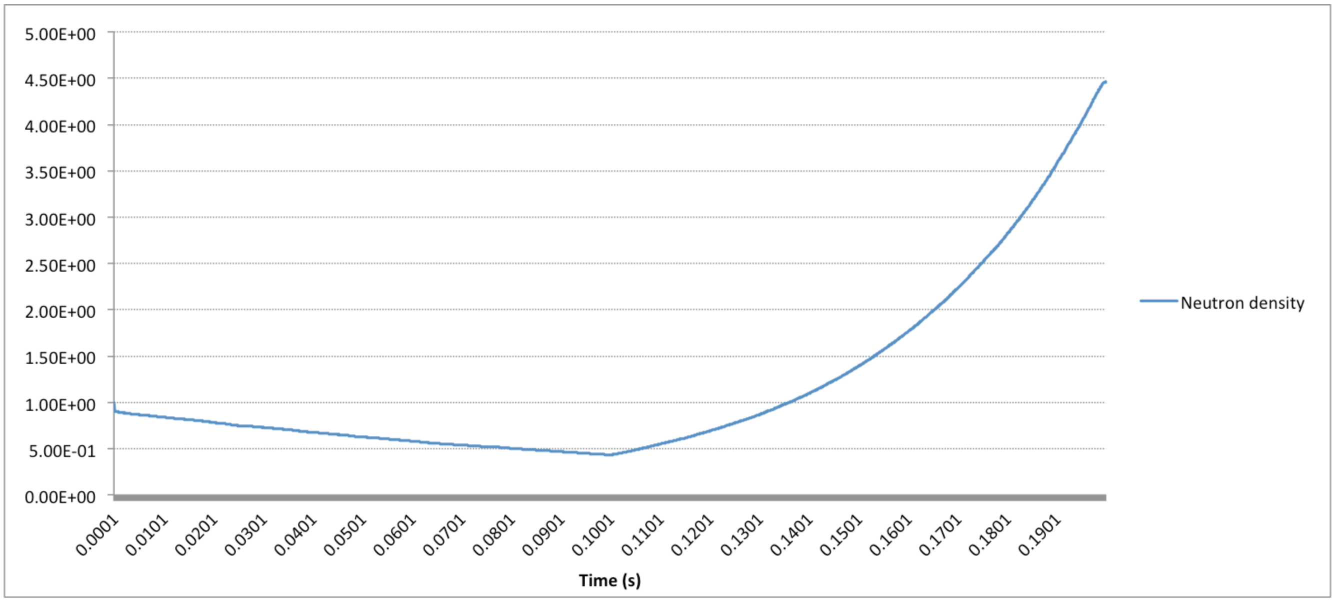

Figure 5 Normalized neutron density

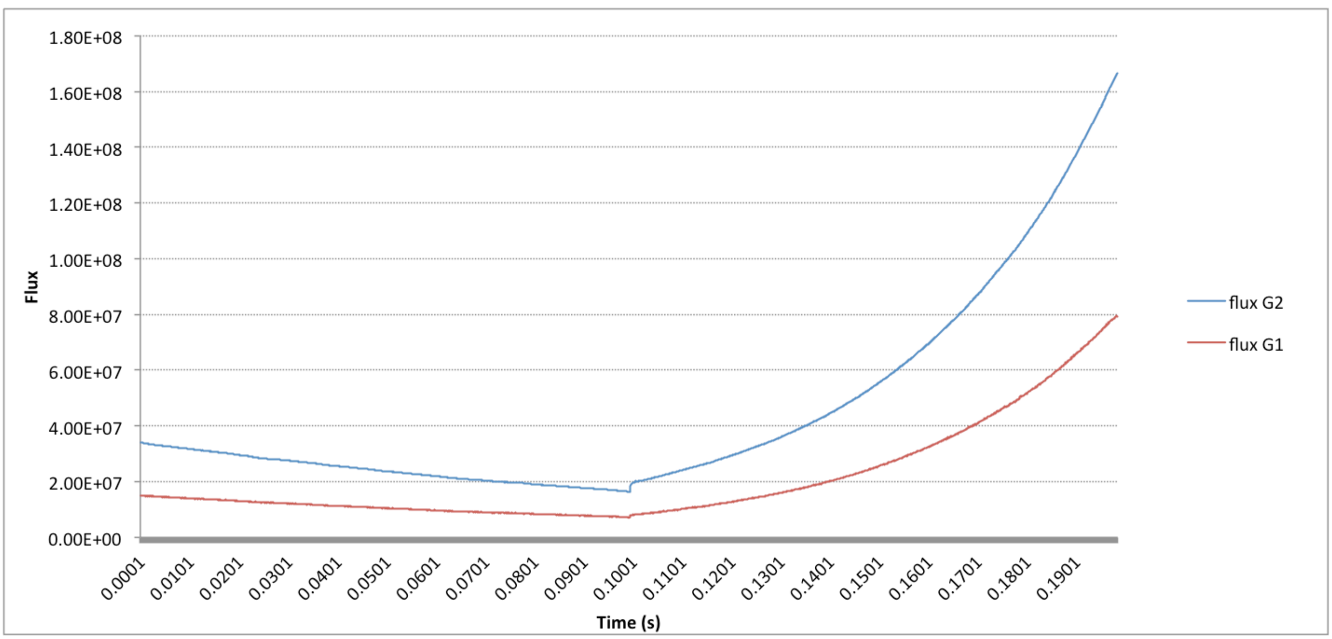

Figure 6 Flux for both fast (G1) and thermal (G2) groups

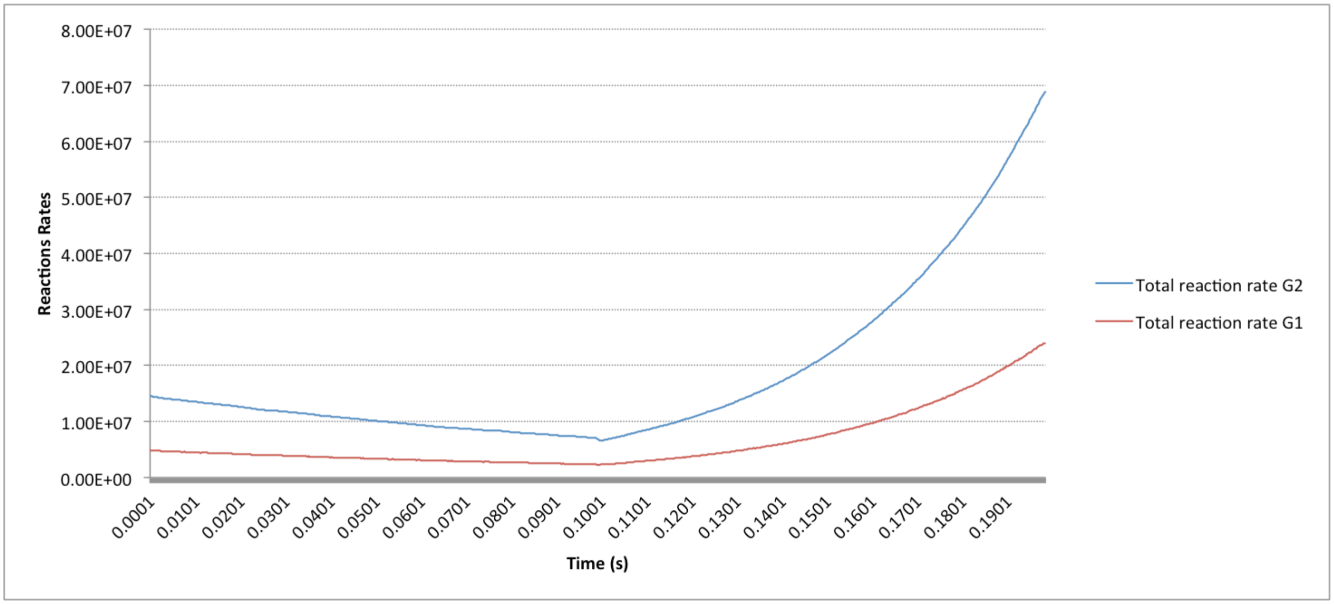

Figure 7 Total reaction rate for a CANDU6 cell using time dependent and static approches

Using the adiabatic assumption, only two different values of cross sections are obtained and used in the time dependent diffusion simulations. These two values present step change in the cross sections. This is represented by the interval function:

$$\begin{cases} \Sigma_x(t) = \Sigma_x^1 , \ t < t_{accident} \\ \Sigma_x(t) = \Sigma_x^2, \ t \geq t_{accident} \end{cases} \tag{14}$$

Dividing the reaction rates by the fluxes gives the corresponding cross sections (eq. 11). Figures 7 and 8 present the total and scattering cross sections obtained by the time dependent and the static simulations. *(XS is used in figures as symbol for "cross section")*

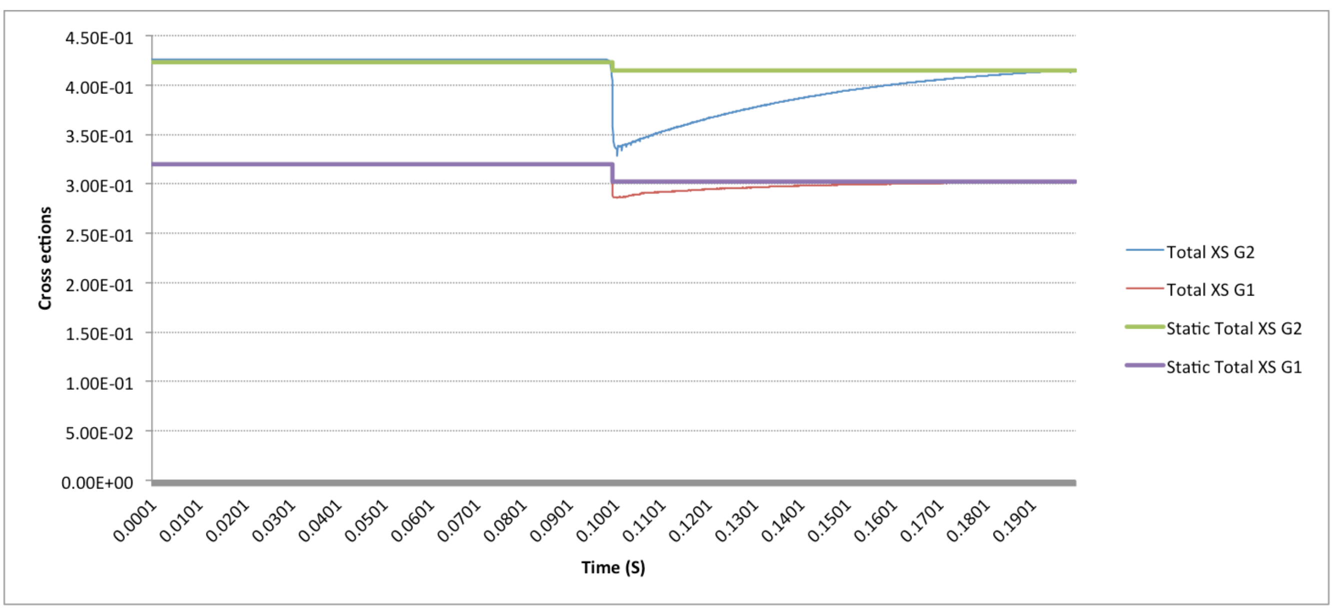

Figure 8 Total cross sections for a CANDU6 cell using time dependent and static approches

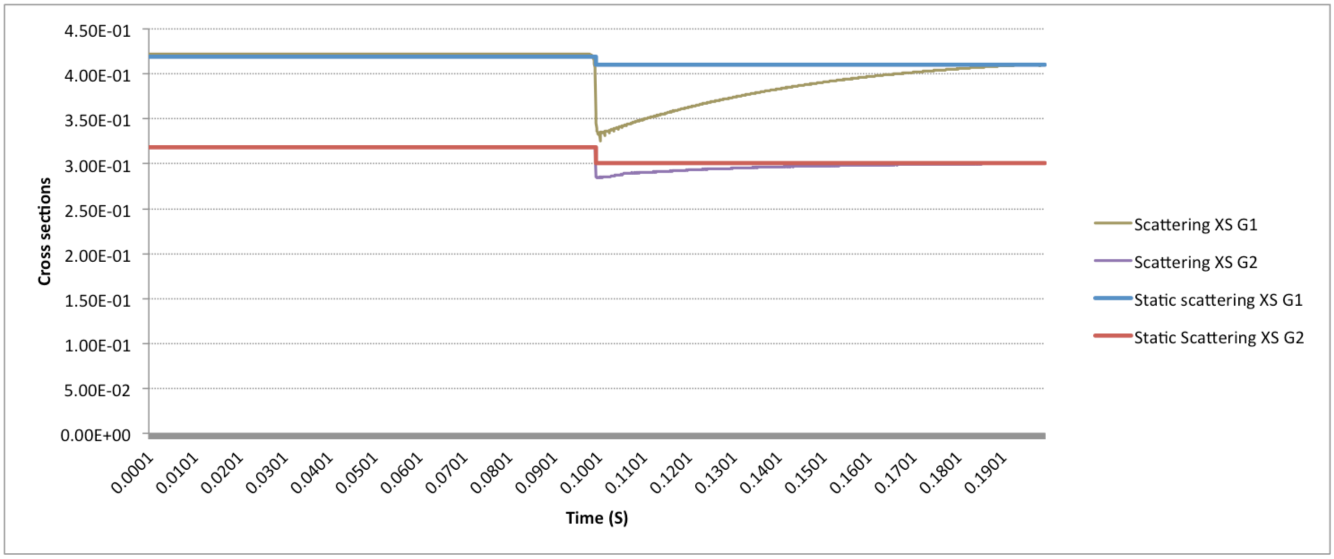

Figure 9 Sacttering cross sections for a CANDU6 cell using time dependent and static approches

Comparing the time dependent cross sections (TDCS) to the static cross sections (Figures 8-11), we observe that before the accident, they are in a good agreement, and fluctuations are very small, as shown in Table1.

	Before 0.1s Cooled cell		After 0.1s Voided cell	
	Fast Gr.	Thermal Gr.	Fast Gr.	Thermal Gr.
Static Cross Section	0.319646	0.423503	0.302310	0.414280
Mean value of TDCS	0.319711	0.423398	0.299488	0.406480
Standard Deviation of TDCS	0.000194	0.001172	0.004983	0.006027

Table 1 Total Cross Sections Analysis before and after accident

After the LOCA, an immediate drop is observed for both the static and time dependent cross sections. However, the TDCS show that they drop much more than the static cross sections before starting to increase slowly. After about 0.1s from the accident, the TDCS converge to the static values.

The fission TDCS shows an important result. It **decreases** first before increasing to eventually converge to the static value whereas the static fission cross section **increases** immediately after the LOCA as shown in Figure 10.

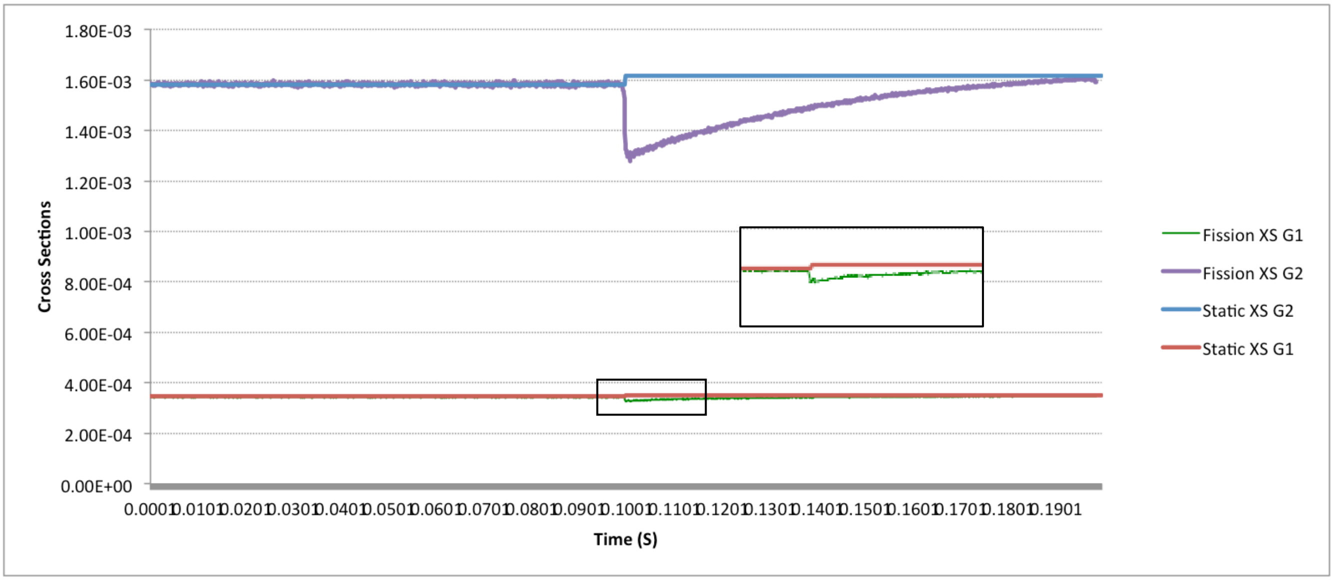

Figure 10 Fission cross sections for a CANDU6 cell using time dependent and static approches

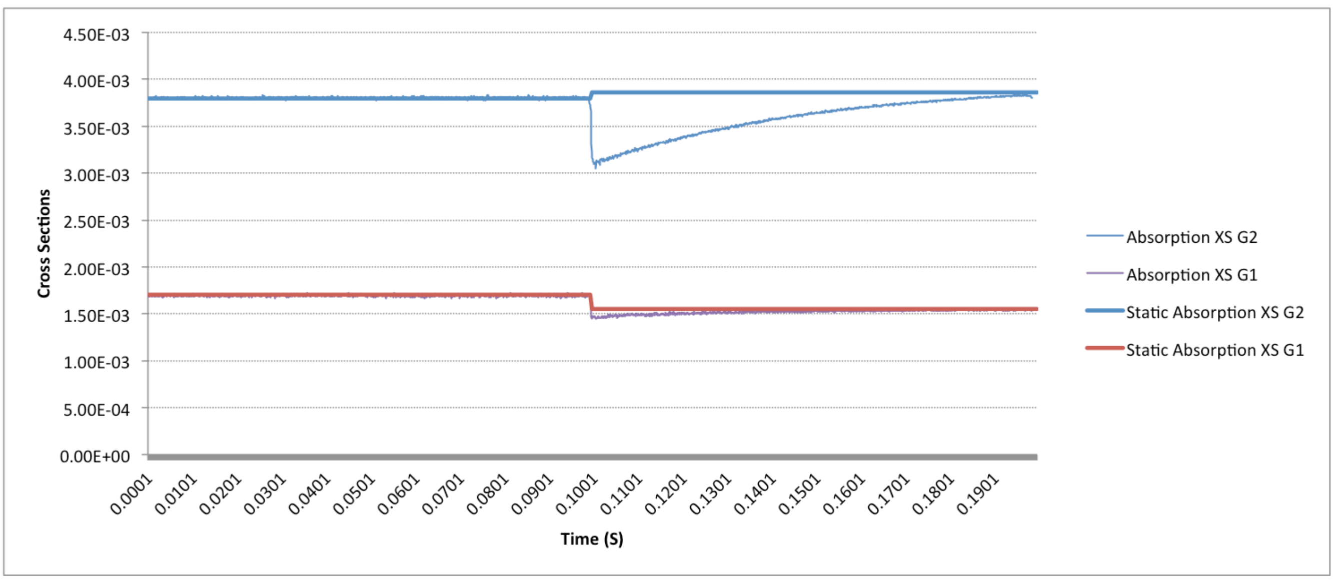

Figure 11 Absorption cross sections for a CANDU6 cell using time dependent and static approches

The introduction of void in a CANDU6 cell increases the absorption of fast neutrons by U238 causing more fast fissions. More of the neutrons thermalized in the moderator will reach the fuel to cause fissions of the U235. The result of this behavior is observed in in the static thermal fission cross section. However, neutrons need time to reach the moderator zone, to be thermalized, and then to come back to the fuel zone to cause the fission. So the increase in the fission cross section is not immediate as shown in the fission and absorption TDCS (figures 10-11).

By these results, we can suspect that the adiabatic assumption can introduce errors in transient simulations. The adiabatic method neglects all time variations of cross sections between the start of the LOCA and 0.1s later. For a full core space-time simulation, this interval of time could lead to new conclusions for CANDU6 simulations. This is currently under investigation.

7. Conclusion

Our results show that full core time dependent simulations need to be examined more closely. Also, the new time dependent model could be used to investigate the effects of reactivity devices such as control rods movement, injection of boron/Gadolinium, introduction of void, etc., for different types of nuclear reactors. Combining thermo-hydraulic codes with OpenMC-TD could improve the quality of simulations.

8. Acknowledgments

We gratefully acknowledge the collaboration of the Computational Reactor Physics Group (CRPG) at the MIT department of Nuclear Science and Engineering.

9. References

[1] D. Cullen, “Tart2005 - a coupled neutron-photon 3-d, combinatorial 31 geometry, time dependent monte carlo transport code”, Rapport technique SM-218009, UCRL, 2005.

[2] L. Russell, G. Jonkmans and A. Buijs, “A method for simulating real-time neutron populations, materials and geometries using the geant4 Monte Carlo toolkit”, 33rd Annual Conference of the Canadian Nuclear Society, Saskatoon, Saskatchewan, 2012.

[3] J. Leppanen, “Development of a Dynamic Simulation Mode in Serpent 2 Monte Carlo Code”, M&C 2013, Sun Valley, Idaho, 2013.

[4] G.Marleau,A.Hébert,andR.Roy,“A user guide for DRAGON version5, reportIGE-335”, École Polytechnique de Montréal, Canada, 2014.

[5] J. Briesmeister, “MCNP5−A General Monte Carlo N−Particle Transport Code”, Los Ala- mos National Laboratory, 2000.

[6] J. Leppänen, “PSG2 / Serpent - a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code”, Methodology - User’s Manual - Validation Report, 2009.

[7] P. K. Romano and B. Forget, “The OpenMC Monte Carlo Particle Transport Code”, Ann. Nucl. Energy, 51, pp.274–281, 2013.

[8] O. Petit, F. Hugot, Y. Lee and C. Jouanne, “TRIPOLI-4 version 4 user guide”, CEA-R- 6169, Commissariat à l’Énergie Atomique (CEA), 2008.

[9] A. Hébert, D. Sekki and R. Chambon, “A user guide for DONJON version5, report IGE-344”, École Polytechnique de Montréal, Canada, 2014.

[10] F. Doria, “Candu safety12: Large Loss Of Coolant Accident”, Rapport technique, Atomic Energy of Canada Limited, 2001.

[11] M. Mahjoub and J. Koclas, “Comparative Static Simulations of a CANDU6 Cell Using Different Transport Codes”, 35th Annual Conference of the Canadian Nuclear Society, New Brunswick, 2015.