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

IMPLEMENTATION OF THE DYNAMIC MONTE CARLO METHOD FOR TRANSIENT ANALYSIS IN THE GENERAL PURPOSE CODE TRIPOLI

Bart L. Sjenitzer and J. Eduard Hoogenboom
Delft University of Technology
Mekelweg 15, 2629JB Delft, the Netherlands
B.L.Sjenitzer@TUDelft.nl; J.E.Hoogenboom@TUDelft.nl

ABSTRACT

A new Dynamic Monte Carlo method is implemented in the general purpose Monte Carlo code Tripoli4.6.1. With this new method incorporated, a general purpose code can be used for safety transient analysis, such as the movement of a control rod or in an accident scenario. To make the Tripoli code ready for calculating on dynamic systems, the Tripoli scheme had to be altered to incorporate time steps, to include the simulation of delayed neutron precursors and to simulate prompt neutron chains.
The modified Tripoli code is tested on two sample cases, a steady-state system and a sub-critical system and the resulting neutron fluxes behave just as expected. The steady-state calculation has a constant neutron flux over time and this result shows the stability of the calculation. The neutron flux stays constant with acceptable variance. This also shows that the starting conditions are determined correctly. The sub-critical case shows that the code can also handle dynamic systems with a varying neutron flux.

Key Words: Tripoli, Dynamic, Transient, Monte Carlo, Neutron transport, Precursor

1. INTRODUCTION

When calculating the time-dependent behavior of a nuclear system, there are two main approaches. The first approach is to turn the problem into an eigenvalue problem. The eigenvalues can be calculated using deterministic methods, but a more accurate method is to use Monte Carlo to calculate the eigenvalues of the system. With the largest eigenvalue, the time evolution of the flux can be approximated, using point-kinetic equations[1][2]. Another approach for calculating the time dependent behavior is to do a direct calculation. This is done using deterministic methods and in this case a mesh is applied not only in space, but also in time. Then the neutron flux is calculated using an implicit or explicit calculation[3].

A direct calculation is more accurate then an eigenvalue calculation combined with point-kinetics equations, but a direct calculation is at the moment only possible by means of a deterministic method and therefore approximations are still needed. The accuracy of the calculation in this case depends on, for example, the mesh size or the homogenization. With these kind of approximations it is difficult to estimate the error in the result and therefore the reliability of the solution.

A new Dynamic Monte Carlo method to calculate the dynamic behavior of a nuclear reactor has been proposed [4][5]. This method does not need approximations and ultimately, a method like this can be used as a validation tool in safety analysis. This would be much cheaper than doing validation experiments. The method is able to simulate both precursors and prompt neutrons in one calculation. This is not straight forward because the time scales of neutrons and precursors have different orders of magnitude. In this work the Dynamic Monte Carlo method is being implemented in the general purpose Monte Carlo code Tripoli4.6.1[6] to show the general applicability of the scheme.

First the simulation of the precursors is discussed. The precursors have to be created at the initial state and new precursors are created during the simulation. Also the decay of the precursors, starting a new prompt neutron chain, is addressed. Next, the rearrangement of the simulation scheme is discussed, the batches that Tripoli uses are changed into time steps. This reduces the variance and can make the calculation dynamical. Finally a test case is calculated to show the first results achieved by applying the Dynamic Monte Carlo method in Tripoli4.6.1.

2. DYNAMIC MONTE CARLO SCHEME

2.1 Monte Carlo Precursors

Tripoli4.6.1 can already sample delayed neutrons, but it only uses this feature for determining the appropriate energy spectrum; the delayed neutrons are in fact not delayed in time. When considering a steady-state problem this is valid, but in a dynamic case the decay time is needed. To achieve this, a precursor particle is introduced in the Tripoli code.

This precursor particle has a few attributes; it can be created from a source distribution or from a fission reaction and it can produce a new prompt neutron after a certain amount of time. To calculate the correct decay behavior it also needs to know its decay constants and delayed fractions and its creation time. These properties are all stored with the particle.

A precursor particle can represent all precursor families in one particle or just one precursor family; combining all families into one particle will reduce variance. The number of precursors is denoted with C and the subscripts i or j are used to indicate precursor family i or j. When a combined precursor particle is created from fission, the fraction per precursor family is given by

$$\frac{C_i(0)}{C(0)} = \frac{\beta_i}{\beta} \tag{1}$$

However, after the creation of the precursor there is a time evolution in these precursor family fractions, because the different precursor families have different decay probabilities. This time evolution is described by

$$\frac{C_i(t)}{C(t)} = \frac{\frac{\beta_i}{\beta} e^{-\lambda_i t}}{\sum_j \frac{\beta_j}{\beta} e^{-\lambda_j t}} \tag{2}$$

The time at which the precursor will have a decay can be sampled according to the combined decay probability:

$$p(t) = \sum_i \lambda_i \frac{\beta_i}{\beta} e^{-\lambda_i t} \tag{3}$$

2.1.1 Precursor production

The precursor particle can be produced in two different settings. The first precursors are created at the start of the calculation. To sample the starting particles a source distribution is needed. A fraction of these source particles are precursors and a fraction are prompt neutrons. The fraction of prompt neutrons in a steady-state situation can be calculated using

$$\text{fraction of neutrons} = \frac{n_0}{n_0 + C_0} = \frac{1}{1 + \sum_i \frac{\beta_i}{\lambda_i} \mathrm{v}\nu\Sigma_f} \tag{4}$$

Here n_0 is the number of neutrons at steady state and C_0 is the number of precursors at steady state.

The second occasion where a precursor can be created is during a fission reaction. Tripoli can already determine if a neutron is a delayed neutron, the only difference is that in the dynamic case a precursor is created. The moment of creation is stored in the particle to be able to calculate the time of decay.

2.2 Matching Time Scales

There is a big difference in the time scale of the lifetime of a precursor and the lifetime of a prompt neutron chain. The lifetime of a precursor can go up to a 100 s, but the typical time scale for prompt neutron chains is $1/\beta$ times the prompt neutron life time, which is about 15 ms for a LWR and up to 150 ms for a HTR. In a steady-state case one prompt neutron chain will produce, on average, one precursor and the precursor will then start a new prompt neutron chain. There is, however, no tallying of power while waiting for the precursor to decay, as shown in Fig. 1. To reduce the variance in the calculation it is necessary to sample the precursor decay in such a way that the delayed neutrons can be sampled on a equal time scale.

To match these time scales, the problem has been divided into time steps. All power tallies are averaged over these time steps and all precursors are forced to produce a neutron in each time step. This ensures that there are enough prompt neutron chains in each time bin. To achieve this in Tripoli, the batches have been altered to become time steps: the first batch has time boundaries t_0 and t_1, the second batch has time boundaries t_1 and t_2, etc. When a neutron creates a new particle this particle is processed in the same batch/time step. Neutrons are only stored for the next time step when they cross the time boundary. Precursors are stored for the next time step after their forced decay.

To force a precursor to produce a neutron in every time step the weight of the resulting neutron has to be adjusted to ensure a fair game. The precursor is given a uniform probability to produce a neutron at time t in the time step.

$$\overline{p}(t) = \frac{1}{\Delta t} \tag{5}$$

The actual probability is given by

$$p(t) = \sum \frac{\beta_i}{\beta} \lambda_i e^{-\lambda_i (t - t_0)} \tag{6}$$

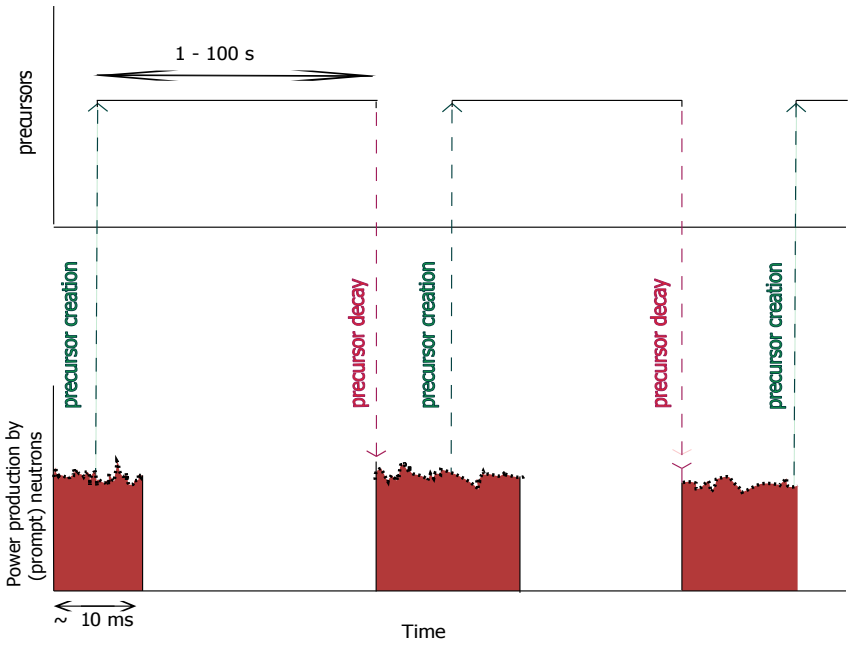


Figure 1: The difference in lifetime between precursors and prompt neutron chains creates variance in the power production.

here t_0 is the time where the precursor was created. The weight of the resulting neutron then becomes

$$w_n = \frac{p(t)}{\overline{p}(t)} = w_p \Delta t \sum \frac{\beta_i}{\beta} \lambda_i e^{-\lambda_i (t-t_0)} \tag{7}$$

Here w_p is the weight of the precursor. After this forced decay, the precursor is saved for the next time step.

This way there are always enough prompt neutron chains per time bin. Only these neutron chains can score tallies and enough scores per bin are needed to give low variance.

2.3 Weight Distributions

Typically the size of the time step is much smaller then the average lifetime of a precursor. Therefore the weight of a resulting delayed neutron, which is samples as described above, is much smaller than the weight of the precursor. This causes most of the created delayed neutrons to be killed by Russian roulette immediately.

To prevent this from happening, the weight of the precursors is increased. The weight of a precursor is set at the level where the resulting delayed neutron will have a weight which is comparable to the weight of the prompt neutrons. The average weight of a neutron resulting from forced decay in time interval i can be calculated using

$$w_{n,av} =< w_n >= \frac{1}{\Delta t} \int_{t_i}^{t_i + \Delta t} w_p \Delta t \sum \frac{\beta_j}{\beta} \lambda_j e^{-\lambda_j (t-t_0)} \mathrm{d}t \tag{8}$$

which is equal to

$$w_{n,av} = w_p \sum \frac{\beta_j}{\beta} e^{-\lambda_j(-t_0)}(e^{-\lambda_j t_i} - e^{-\lambda_j(t_i+\Delta t)}) \tag{9}$$

The weight of the precursors can be increased by reducing the probability of creating a precursor from fission. The weight of the prompt neutrons at the start of the time interval is also set to the same level using Russian roulette and splitting. The decaying precursor generates on average neutrons with the same weight as these prompt neutrons to ensure equal contribution from all particles.

3. CALCULATION SCHEME

For the implementation of the dynamic calculation scheme, Tripoli is adapted at a number of points. Not only the order of the simulation of particles needs the be rearranged, but also the source input, the handling of fission and the tallying needs to be altered. This is due to the fact that there are no cycles as with a criticality calculation and therefore there is no problem with source convergence. However the calculation is divided in steps, unlike a fixed-source calculation, because the simulation must be able to adapt to a changing system.

First a new mode is created, which is called *DYNAMIC*. This is mode is created next to the *CRITICALITY*, *SHIELDING* and *FIXED_SOURCES_CRITICALITY* modes. This mode has a few input variables which are specific for a dynamic calculation. First of all there must be a time mesh and the number of time bins must be equal to the number of batches. Also the particle type of simulated particles must include the particle type PRECURSOR and as source particle the PRECURSOR can also be chosen.

3.1 Particle Tracking

In shielding calculation a particle is followed until it is either killed, leaked or out of bounds in some other way and cannot create new fission neutrons. When doing a criticality calculation this is essentially the same, but the produced fission neutrons are stored for the next batch. In a fixed sources criticality calculation the particle can also produce a fission neutron, but now this neutron will be followed in the same batch. In the dynamic mode the simulation of the particles histories is very similar to the fixed sources criticality case, except that in the dynamic case the neutron simulation is stopped when crossing a time boundary and the neutron is then transferred to the next batch.

In the three conventional modes the variance is calculated from the batch average tally results. In a dynamic calculation the batches become time intervals. These intervals are strongly dependent on each other. Therefore they cannot be used any more to calculate the variance between them. The variance must be calculated between the particles themselves. This gives also an accurate estimation of the variance[7].

3.2 Sources

In a criticality calculation the source of a batch is given by the the fission production of the previous batch. The fission neutrons of the previous batch, are the initial state of the next batch. Only for the first batch a source distribution is given, but often this is only a flat source and a number of inactive batches is simulated to get a true source distribution. In both shielding and fixed-sources criticality modes the user is asked to input a source distribution. Each batch starts with particles sampled from this source distribution. In the fixed-sources criticality calculation fission neutrons can be produced, but they are not used the source for a next batch.

In the dynamic case only for the first time bin an external source is needed. This can be started with a source distribution input, similar to the shielding or the fixed-sources criticality case. Also a criticality calculation can be done to converge to the appropriate source distribution. In both cases Eq. 2 it is needed to determine the fraction of neutrons and the fraction of precursors. Also the fraction per precursor family must be calculated. At the moment a precursor is created, the fractions are simply like Eq. 1, but in steady state these become

$$\frac{\beta_i}{\lambda_i}\frac{\lambda}{\beta} \tag{10}$$

It is not so trivial to generate the precursor families in different fractions, since all precursors are combined into a single particle. The way to solve this is to sample a creation time of the precursor particle uniform between $-\infty$ and t_0. Now also the effective weight of the precursor particle is needed, which is equal to 1 minus the probability it already had an decay.

$$w_{p,0} = 1 - \int_{t_0}^{0} \sum \frac{\beta_i}{\beta} \lambda_i e^{-\lambda_i (t-t_0)} \mathrm{d}t = \sum \frac{\beta_i}{\beta} e^{\lambda_i t_0} \tag{11}$$

Here t_0 is before the starting time of the calculation. In the weighted case Eq. 4 becomes

$$\text{fraction of neutrons} = \frac{\sum w_n}{\sum w_n + \sum w_p} = \frac{1}{1 + \sum_i \frac{\beta_i}{\lambda_i} \mathrm{v}\nu\Sigma_f} \tag{12}$$

When the precursor is selected as source particle this division is made by the code and a source of prompt neutrons and precursors is created.

4. FIRST RESULTS

For the test cases of the new dynamic method in Tripoli4.6.1, the dynamic mode has been implemented, the batches have been transformed to time steps and the precursor particles have been introduced. The simulated particles are neutrons and precursors and a source composed out of both precursors and neutrons is generated in a cosine shape. The fractions of neutrons and precursors are calculated as explained above. The time mesh is in equal 100 ms steps over 70 s.

4.1 Steady State

The first test case for the new Dynamic Tripoli code is a simple system that is in a near critical state, $k_{\text{eff}} \approx 1$. It is a block made out of an artificial fissile material. The cross sections of this

material and some other properties are given in Table I. The size of the system is 10 cm by

Table I: Material properties that have been used in the test problem. The box is made out of a homogeneous material

Material properties
$\Sigma_t = 1$ cm^{-1}
$\Sigma_f = 0.25$ cm^{-1}
$\Sigma_s = 0.4118$ cm^{-1}
$\nu = 2.5$
$\beta = 0.00685$

12 cm by 24 cm, the system consists out of one energy group and all neutrons have a speed of 2.2×10^4 cm/s. The decay constants and delayed fractions are given in Table II. The k_{eff} is 0.99990 ± 0.00001 and this value is calculated using the unmodified Tripoli version 4.6.1.

Table II: The precursors are divided in six families, here the fractions and decay constants per precursor family i are given. Also the total delayed fraction and average decay constant are shown.

Group	λ (s^{-1})	β
1	0.0127	0.00026
2	0.0317	0.001459
3	0.1156	0.001288
4	0.311	0.002788
5	1.4	0.000877
6	3.87	0.000178
av/tot	0.0784	0.00685

Tripoli will calculate the time evolution of this system over the 70 s and to be able to calculate such a system, the scheme must be stable and it is a good test to see if the starting conditions are correct.

The calculated time evolution of the neutron flux is plotted in Fig. 2 and as clearly shown the neutron flux in this system is nearly constant from the start. Also the results are stable over time. There is of course some variance, but this does not influence the stability of the total simulation.

4.2 Sub-Critical Calculation

To show the response in a dynamic situation, the system has been made smaller, so $k_{eff} < 1$. The reactor is now 8 cm wide, 18 cm deep and 22 cm high. The k_{eff}, calculated with the regular Tripoli4.6.1, and is in this case equal to 0.97823 ± 0.00001. In the time evolution of this system you can first see the prompt jump and after this the power production of the system starts to decrease, Fig. 3(a). The calculation is compared to a point-kinetics calculation where the generation time $\Lambda = 7 \times 10^{-5}$ and the reactivity $\rho = -0.022252$ and the results seem to agree, but the exact statistics cannot be estimated at the present time, since Tripoli normally uses

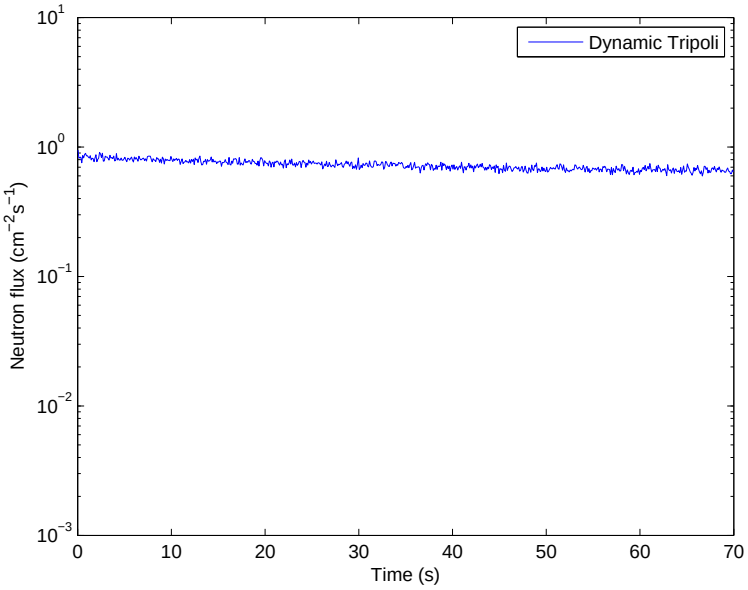
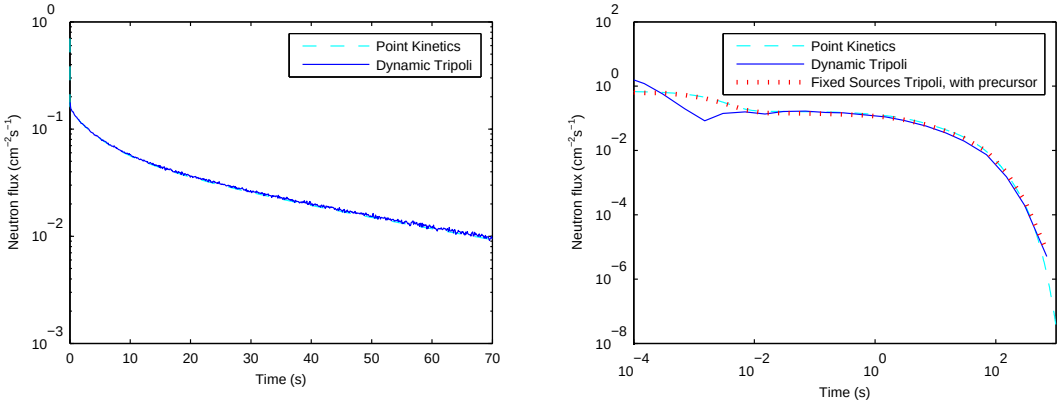


Figure 2: The time evolution of the total neutron flux in a stable system.

the variance between batches to estimate the variance of a result. This result shows that the simulation can also handle a calculation with a changing neutron flux.



(a) The neutron flux tallied in time steps of 100 ms (b) The neutron flux tallied in time steps that increase exponentially

Figure 3: The time evolution of the total neutron flux in a sub-critical system.

To investigate the behavior of the neutron flux during the prompt jump the same calculation has been done, but this time with time steps that increase in size exponentially. The results are plotted in Fig. 3(b) and they show that there is a deviation from the theoretical value for small time scales. Although the exact variance cannot be estimated, the deviation is larger then the variance observed. When the same calculation is done with Tripoli using the precursor particle, but using the regular fixed-sources criticality mode instead of the dynamic mode, the results do agree. This indicates that the neutron behavior is not simulated optimal at the time boundaries.

When the time step is large compared to a neutron lifetime this can be neglected, but for small time steps this is an issue that needs to be addressed.

5. CONCLUSIONS AND FURTHER WORK

The first steps have been taken towards full Monte Carlo Dynamic calculations. This paper illustrates the point where this research is currently at. It is shown that this kind of calculation is feasible, also with a general purpose code like Tripoli. The results have been compared to a point-kinetics calculation and the results agree nicely. Although the example problem is a simple system, for the Monte Carlo technique the complexity of a system is less relevant. The calculation of a more complex system will not become much more difficult.

The next development in the implementation of the dynamic scheme into Tripoli is to add the sampling of variance and then adding variance reduction methods for Dynamic Monte Carlo. At this time, not all variance reduction techniques that are known for Dynamic Monte Carlo are implemented. With these methods the calculation will become more efficient and therefore faster and more accurate. Also the neutron behavior needs to be simulated more accurately at the time boundary, to simulate accurately the short term behavior of the system.

The next development will be the addition of a dynamic system into the calculation. In this case the system can be altered at every time step to simulate for example the withdrawal of a control rod or an accident scenario. This way also feedback mechanisms can be implemented into the code.

When this is done and verified, the ultimate goal is to couple the dynamic Monte Carlo to a thermal-hydraulics code and to see if a coupled calculation can be done[8]. When Dynamic Tripoli is coupled with a thermal-hydraulics code it can be used to do realistic accident scenario simulations and it can be used as a validation and verification tool.

ACKNOWLEDGEMENTS

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

REFERENCES

1. S. Yun, J.W. Kim and N.Z. Cho, Monte Carlo Space-Time Reactor Kinetics Method and its Verification with Time-Dependent SN Method, *Proceedings of PHYSOR 2008*, Sep. 14-19, 2008, Interlaken, Switzerland, (2008).

2. M. Shayesteh, M. Shahriari, Calculation of time-dependent neutronic parameters using Monte Carlo method, *Annals of Nuclear Energy*, **36**, pp. 901-909 (2009).

3. S. Goluoglu, H.L. Dodds, A Time-Dependent, Three-Dimensional Neutron Transport Methodology, *Nuclear Science and Engeneering.*, **139**, pp. 248-261 (2001).

4. D. Legrady, J.E. Hoogenboom, Scouting the feasibility of Monte Carlo reactor dynamics simulations, *Proc. PHYSOR-2008*, Sep. 14-19, 2008, Interlaken, Switzerland, (2008).

5. B.L. Sjenitzer, J.E. Hoogenboom, A Monte Carlo Method for Calculation on the Dynamic Behaviour of Nuclear Reactors, *Proceedings of SNA+MC 2010*, Tokyo, Japan, Oct. 18-21, (2010).

6. O. Petit, F.X. Hugot, et al. Tripoli-4 Version 6 User Guide, *Technical Report CEA, SERMA/LTSD/RT/08-4557/A*, CEA Saclay, (2009)

7. J.E. Hoogenboom, Improved estimation if the variance in Monte Carlo criticality calculations, *Proc. PHYSOR-2008*, Sep. 14-19, 2008, Interlaken, Switzerland, (2008).

8. J.E. Hoogenboom et al., A Flexible Coupling Scheme For Monte Carlo and Thermal-Hydraulics Codes, *this conference*, (2011).