Implementation of an Approximate Zero-Variance Scheme in the TRIPOLI Monte Carlo Code

Stavros Christoforou^{*1} and J. Eduard Hoogenboom¹ Eric Dumonteil², Odile Petit² and Cheikh Diop² ¹Delft University of Technology, Mekelweg 15, 2629 JB Delft, The Netherlands ²Commissariat à l'Energie Atomique, Gif-sur-Yvette, France

Abstract

In an accompanying paper it is shown that theoretically a zero-variance Monte Carlo scheme can be devised for criticality calculations if the space, energy and direction dependent adjoint function is exactly known. This requires biasing of the transition and collision kernels with the appropriate adjoint function. In this paper it is discussed how an existing general purpose Monte Carlo code like TRIPOLI can be modified to approach the zero-variance scheme. This requires modifications for reading in the adjoint function obtained from a separate deterministic calculation for a number of space intervals, energy groups and discrete directions. Furthermore, a function has to be added to supply the direction dependent and the averaged adjoint function at a specific position in the system by interpolation. The initial particle weights of a certain batch must be set inversely proportional to the averaged adjoint function and proper normalization of the initial weights must be secured. The sampling of the biased transition kernel requires cumulative integrals of the biased kernel along the flight path until a certain value, depending on a selected random number is reached to determine a new collision site. The weight of the particle must be adapted accordingly. The sampling of the biased collision kernel (in a multigroup treatment) is much more like the normal sampling procedure.

A numerical example is given for a 3-group calculation with a simplified transport model (two-direction model), demonstrating that the zero-variance scheme can be approximated quite well for this simplified case.

KEYWORDS: Monte Carlo, zero variance, adjoint function biasing, TRIPOLI, discrete ordinates

1. Introduction

It is known for a long time that a zero-variance Monte Carlo scheme in a source-detector problem can be obtained when the transition kernel for determining a next collision site and the collision kernel for determining energy and direction after a scattering collision are biased by the appropriate adjoint functions. [1] In an accompanying paper it is shown that a zero-variance scheme can also be devised for a criticality calculation to calculate the effective multiplication factor k_{eff} using successive batches of neutrons to obtain the converged source eigenfunction distribution and calculating from each batch the source distribution for the next generation of

^{*} Corresponding author, Tel.. +31-15-2782163 E-mail: s.christoforou@tudelft.nl

neutrons. [2] When simulating the n^{th} batch of neutrons with a normalized source distribution $S_n(P)$ the estimate of k_{eff} after sufficient convergence is obtained from

$$k_{e_{ff}} = \int S_{n+1}(P)dP = \int \frac{\nu \Sigma_f(P)}{\Sigma_t(P)} \psi(P)dP$$
(1)

with $\psi(P)$ the neutron collision density at a phase space point $P=(r,E, \Omega)$, $v\Sigma_f$ the neutron production cross section and Σ_t the total cross section. For each batch or generation of neutrons, this is in fact running a source-detector problem with $v\Sigma_f\Sigma_t$ as the detector response function.

2. The TRIPOLI code

TRIPOLI is a versatile general purpose Monte Carlo code suitable for shielding problems, criticality calculations, core physics analysis and instrument studies. Besides neutron transport it can also handle photon, electron and positron transport in combined mode. The code is developed by Commissariat à l'Energie Atomique, Saclay, France. The latest version TRIPOLI-4.4 is publicly available from the OECD Nuclear Energy Agency Data Bank. New options are regularly added. The code is thoroughly benchmarked against experiments.

The code is mainly written in C++ and can be run on different computer platforms. It can be used with parallel processing. TRIPOLI can handle cross sections in continuous energy given in the ENDF/B format and group cross sections with any number of groups. For the resonance region probability tables may be used.

The geometry definition is flexible with options to use predefined bodies in combinatorial geometry or to define material volumes by surface equations. A rotating operator is available to create specific forms from basic bodies. Repeated structures can be defined in rectangular and hexagonal form. The description of sources is especially flexible. All kind of combinations of space, energy, direction and time dependent sources can be described, including the possibility to use analytical functions to represent the source dependence in any variable.

TRIPOLI provides a variety of estimators. Standard estimators are available to calculate particle fluxes in specified volumes, either with a collision or a track-length estimator, fluxes on surfaces or at a point, currents at a surface, but also dose rates, reaction rates using a specified response cross section, deposited energy or recoil energy. TRIPOLI also provides a perturbation option.

One of the most important features of TRIPOLI is its possibility to apply particle weighting schemes to reduce the variance of estimators. An importance function can be defined for the whole space, energy, direction and time domain, supplying a minimum number of input data for it. The importance function is used to control the weight of the particle by splitting or Russian roulette depending on its position, energy, direction and time. An importance function can also be used to bias the transport kernel to direct particles towards a predefined position.

3. Implementation of the approximate zero-variance biasing in TRIPOLI

3.1 Reading in the adjoint function

The adjoint function used for biasing the transition and collision kernels is supposed to be derived from a deterministic calculation, probably using an approximate geometry description. From a discrete ordinates code run in adjoint mode the adjoint function for particle leaving the

source or a collision is obtained in the form

$$\widetilde{\chi}_{i,j,k,g,m}^{*} = \widetilde{\chi}^{*}(x_{i}, y_{j}, z_{k}, g, \boldsymbol{\Omega}_{m}) \quad i = 0, N_{x}; \ j = 0, N_{y}; \ k = 0, N_{z}; \quad g = 1, N_{group}; \quad m = 1, N_{dir} \quad (2)$$

with N_x , N_y , and N_z , the number of mesh intervals in the *x*, *y*, and *z* direction, respectively. N_{group} is the number of energy groups and N_{dir} the number of discrete directions. The adjoint directional function is supposed to be given at special mesh boundaries. A discrete ordinates code uses a dummy direction with zero weight to start the flux solution. This dummy direction is excluded from the adjoint functions read in. Also the mesh sizes and the direction cosines and weights of the quadrature set have to be read in.

The adjoint function as defined above is stored in TRIPOLI as member of a C++ class object for defining multigroup cross sections. A class method is defined to calculate the adjoint function at a specific space point, energy group and direction by interpolation between the values at the neighboring mesh boundaries. The adjoint function averaged over direction is then also calculated.

3.2 Calculating the adjoint function for particles entering a collision

For biasing the transition kernel to select a new collision site, the zero-variance scheme requires the adjoint function ψ^* for particles entering a collision as the biasing function. In the general case this adjoint function is related to the adjoint function χ^* for particles starting a flight path as follows:

$$\psi^{*}(\boldsymbol{r}, \boldsymbol{E}, \boldsymbol{\Omega}) = \eta_{\psi}(\boldsymbol{r}, \boldsymbol{E}, \boldsymbol{\Omega}) + \int \int C(\boldsymbol{r}, \boldsymbol{E} \to \boldsymbol{E}', \boldsymbol{\Omega} \to \boldsymbol{\Omega}') \chi^{*}(\boldsymbol{r}, \boldsymbol{E}', \boldsymbol{\Omega}') d\boldsymbol{E}' d\Omega'$$
(3)

with η_{ψ} the detector response function with respect to the collision density and *C* the collision kernel. For the discretized situation this adjoint function becomes

$$\psi^{*}(x, y, z, g, \boldsymbol{\Omega}_{m}) = \eta(x, y, z, g, \boldsymbol{\Omega}_{m}) + \sum_{g'} \sum_{m'} \frac{\Sigma_{s}(x, y, z, g \rightarrow g', \boldsymbol{\Omega}_{m} \rightarrow \boldsymbol{\Omega}_{m'})}{\Sigma_{t}(x, y, z, g')} \chi^{*}(x, y, z, g', \boldsymbol{\Omega}_{m'})$$

$$(4)$$

This function is also stored in a C++ class object.

3.3 Batch initialization

The energy of neutrons generated by fission are selected from the fission spectrum biased by the direction averaged adjoint function χ^* . Their direction is selected from the isotropic distribution biased by the directional adjoint function. Then their weight must be set inversely proportional to the adjoint function for the selected energy group and direction. This requires appropriate normalization of the weights of all particles in a new batch. In a multigroup treatment the initial weight is set equal to

$$w_{init}(\boldsymbol{r}, g, \boldsymbol{\Omega}) = \frac{\sum_{g''} \chi_{fis, g''} \chi_{g''}^{*}(\boldsymbol{r})}{\chi_{g}^{*}(\boldsymbol{r}, \boldsymbol{\Omega}_{m})}$$
(5)

34 Biasing the transition kernel

When a particle starts a flight path the transition kernel must be sampled to select a next collision site. According to the theoretical zero-variance scheme this transition kernel must be biased by the adjoint function ψ^* for a particle entering a collision. Hence, a new collision site must be selected from the biased kernel

$$\overline{T}(\mathbf{r}' \to \mathbf{r}, E, \Omega) = \frac{T(\mathbf{r}' \to \mathbf{r}, E, \Omega)\psi^*(\mathbf{r}, E, \Omega)}{\int T(\mathbf{r}' \to \mathbf{r}'', E, \Omega)\psi^*(\mathbf{r}'', E, \Omega)dV''}$$
(6)

via integration along the flight direction. To determine the integral in the denominator for normalization, one has to track the geometry from the last collision site r into the direction Ω up to the outer boundary. Using a random number ξ the track length s to the next collision site is determined from the solution of

$$\xi = \frac{\int\limits_{s_{max}}^{s} T(\mathbf{r}' \to \mathbf{r}' + s'' \boldsymbol{\Omega}, E, \boldsymbol{\Omega}) \widetilde{\psi}^{*}(\mathbf{r}' + s'' \boldsymbol{\Omega}, g, \boldsymbol{\Omega}_{m}) ds''}{\int\limits_{0}^{s_{max}} T(\mathbf{r}' \to \mathbf{r}' + s'' \boldsymbol{\Omega}, E, \boldsymbol{\Omega}) \widetilde{\psi}^{*}(\mathbf{r}' + s'' \boldsymbol{\Omega}, g, \boldsymbol{\Omega}_{m}) ds''}$$
(7)

with $\tilde{\psi}^*$ the approximated adjoint function, g the energy group where the neutron energy E belongs to and Ω_m the discrete direction for which the adjoint function is given nearest to the actual direction of the neutron flight path. Eq.(7) requires retrieval of the direction and energy dependent adjoint functions on many positions along the maximum possible track length and the transition kernel biasing is therefore a computational expensive process.

A weight factor should be applied to the particle weight to compensate for selecting from the biased transition kernel equal to

$$W_{T} = \frac{\widetilde{\psi}^{*}(r, g, \Omega_{m})}{\int_{0}^{S_{\max}} T(r' \to r' + s''\Omega, E, \Omega)\widetilde{\psi}^{*}(r' + s''\Omega, g, \Omega_{m})ds''}$$
(8)

with z the number of mean free path between r and r at energy E. The denominator is already calculated when selecting the track length according to Eq.(7).

35 Biasing the collision kernel

The collision kernel should be biased by the (approximated) adjoint function $\tilde{\chi}^*$ for a particle entering a new flight. In a multigroup treatment the energy group *g* after collision is selected first with probability

$$\overline{p}(g \mid \boldsymbol{r}, g') = \frac{\sum_{s} (\boldsymbol{r}, g' \to g) \widetilde{\chi}^{*}(\boldsymbol{r}, g)}{\sum_{g''} \sum_{s} (\boldsymbol{r}, g' \to g'') \widetilde{\chi}^{*}(\boldsymbol{r}, g'')}$$
(9)

After selection of the new energy group, the direction is selected from the probability

$$\overline{p}(\boldsymbol{\Omega} | \boldsymbol{r}, g', g) = \frac{\sum_{s} (\boldsymbol{r}, \boldsymbol{\Omega}' \to \boldsymbol{\Omega} | g', g) \widetilde{\chi}^{*}(\boldsymbol{r}, g, \boldsymbol{\Omega}_{m})}{\int \sum_{s} (\boldsymbol{r}, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}'' | g', g) \widetilde{\chi}^{*}(\boldsymbol{r}, g, \boldsymbol{\Omega}_{m}) d\Omega''}$$
(10)

with again *m* indicating the discrete direction nearest to the direction $\boldsymbol{\Omega}$.

A weight factor should be applied to the particle weight to compensate for selecting from the biased collision kernel equal to

$$W_{\overline{c}} = \frac{\sum_{g''} \Sigma_s(\boldsymbol{r}, g' \to g'', \boldsymbol{\Omega}' \to \boldsymbol{\Omega}) \widetilde{\chi}^*(\boldsymbol{r}, g'', \boldsymbol{\Omega}_m)}{\widetilde{\chi}^*(\boldsymbol{r}, g, \boldsymbol{\Omega}_m)}$$
(11)

4. Numerical example

To test the theory for approximate zero-variance biasing as discussed above we implemented the method in a local version of the TRIPOLI-4.4 code for a simplified transport model, namely the two-direction model, which allows the particles to move only in the +x or -x directions.[2] This model comes very near the normal transport model as far as its simulation in Monte Carlo is concerned. Moreover, it can be described by a diffusion-type differential equation, which allows an analytical solution for not too complicated systems in order to compare numerical results with analytical ones.

The system used consists of a homogeneous slab of fissionable material. The energy dependence is treated in a 3-group approximation, which allows for testing most of the theoretical formulas given in Sect. 3. The total cross section for each group was chosen equal in order to have the same (converged) spatial distribution for each group, which allows an analytical calculation of k_{eff} . The 3-group scattering matrix allowed only downscattering.

The adjoint function χ^* was obtained from the one-dimensial S_N code XSDRN.[3] A cross section library for this code was prepared with the same cross sections as for the Monte Carlo runs, except that the material was not fissionable. Unfortunately the adjoint calculation with XSDRN did not work out as expected, as the (transposed) scattering matrix was renormalized by the code. Therefore, a forward calculation was performed with a cross section set in which the scattering matrix was already transposed. As this implies that the scattering cross section for a group as the sum of the in-group and out-of-the-group scattering cross sections is no longer equal to the original total scattering cross section for that group, there is no balance between the source plus in-scatter and the absorption plus out-scatter of particle per group. Nonetheless, the correct adjoint function is obtained in this way.

A reference run with TRIPOLI was done without biasing, starting with a flat source distribution. For this run 50 batches of neutrons were simulated in order to have sufficient batches with a converged source distribution from which the averaged effective multiplication factor can be estimated. Next a run was done with biasing by the adjoint functions. Here too, the run was started with a flat spatial distribution. As the convergence of the source distribution is faster, only 30 successive batches were done.

Table 1 shows the results for k_{eff} together with the obtained relative standard deviation. The values for the multiplication factor are in good agreement with the analytical value of 1.12865.

case	k _{eff}	σ	particles	number of
			per batch	batches used
unbiased	1.12808	0.00052	100000	50
biased	1.12841	0.00023	10000	30

Table 1: Comparison of an unbiased and a biased criticality calculation.

As expected, the standard deviation in the biased run is much lower than that of the unbiased run, especially if we take into account that the total number of particles in the latter is much bigger. Statistics were calculated over a smaller number of batches in the biased case. This is done because in the full transport model, it is expected that the biased case will have a higher CPU cost, and therefore it should be able to deliver an improvement over a smaller number of batches relative to an unbiased case.

Fig. 1 shows the convergence rate as far as k_{eff} is concerned. One can see that in the biased case convergence is reached already after 10 batches, and we can start sampling the statistics. Again, note that in order to have comparable results, the unbiased calculation is simulating a much larger total number of particles (hence the similar convergence), and still it shows fluctuation during the simulation, which is not the case with the biased case. By using a number comparable to the biased calculation, the convergence is much slower in terms of batches simulated.



Figure 1: Convergence of k_{eff} over the course of the simulation

5. Conclusions and discussion

The results from the numerical example show that the theory underlying the zero-variancebased scheme for biasing the Monte Carlo criticality calculation and the application to a multigroup case is valid and results in lower standard deviation. As a byproduct, the convergence to the eigenfunction distribution is accelerated, which also saves CPU time. However, no CPU times and figures of merit (*FOM*) are given for our numerical case as the CPU time for the biased calculation strongly depends on the implementation details, which were in no way optimized in our case. From other tests it was already concluded that also in terms of the *FOM* a big advantage can be obtained in the biased calculation with proper optimization.

The implementation used so far is still rather limited with the two-direction model. Generalization requires the solution of a few practical problems for proper implementation, which will be tackled in the near future.

Acknowledgement

This work was supported by the European Commision under the NURESIM Integrated Project within the 6th Framework Program, project code #516560. [4]

References

- 1) S. Christoforou and J.E. Hoogenboom, "A Zero-Variance-Based Scheme for Variance Reduction in Monte Carlo Criticality", this conference.
- J. E. Hoogenboom, "Monte Carlo Zero-Variance Schemes: Theory, Demonstration and Practical Consequences," Proceedings M&C2003 conference, Gatlinburg, Tenessee, USA, April 6-10, 2003, CD-ROM (2003).
- 3) N. M. Greene and C.W. Craven Jr, "XSDRN, A discrete ordinates spectral averaging code," ORNL-TM 2500 (1969).
- 4) D. G. Cacuci et al., "NURESIM: A European platform for nuclear reactor simulation", Proceedings FISA 2006, Conference on EU Research and Training in Reactor Systems, Luxembourg, March 13-16, 2006, to be published (2006).