A zero-variance based scheme for Monte Carlo criticality simulations

A zero-variance based scheme for Monte Carlo criticality simulations

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universtiteit Delft,
op gezag van de Rector Magnificus Prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op maandag 11 oktober 2010 om 10:00 uur
door

Stavros CHRISTOFOROU

Master of Science in Radiation & Environmental Protection, University of Surrey

geboren te Chalkida, Griekenland

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. T.H.J.J van der Hagen

Copromotor:

Dr. ir. J.E. Hoogenboom

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. T.H.J.J van der Hagen,	Technische Universiteit Delft, promotor
Dr. ir. J.E. Hoogenboom,	Technische Universiteit Delft
Prof. dr. ir. C.W.E. van Eijk,	Technische Universiteit Delft
Prof. dr. B.J.M. Heijmen,	Erasmus Universiteit Rotterdam
Prof. dr. ir. G. Jongbloed,	Technische Universiteit Delft
Prof. dr. ir. A.H.M. Verkooijen,	Technische Universiteit Delft
Dr. C. Diop,	Commissariat à l'Énergie Atomique

© 2010, Stavros Christoforou

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without prior permission from the copyright owner.

ISBN 978-960-93-2247-8

Keywords: Monte Carlo, biasing, adjoint, criticality

The research described in this thesis was performed in the section Physics of Nuclear Reactors (PNR), of the department Radiation, Radionuclides & Reactors (R^3), of the Delft University of Technology, Delft, The Netherlands.

PRINTED IN THE NETHERLANDS

Financial support
Part of the work presented in this thesis was financed under the European Commission / EURATOM 6th Framework Programme NURESIM, contract number FI60-CT-2005-516560, effective February 2005 - December 2008.

Contents

CHAPTER 1

INTRODUCTION TO THE MONTE CARLO METHOD

In this thesis we present a new method that we devised in order to tackle variance reduction in Monte Carlo eigenvalue calculations.

The Monte Carlo method is used to evaluate the average or expected behaviour of a system by simulating a large number of events responsible for its behaviour and observing the outcomes. The principle of the Monte Carlo method, statistical simulation, has been in use since the 18$^{\text{th}}$ century. In a well-known experiment(Holgate, 1981), Georges-Louis Leclerc, Compte de Buffon, calculated that if a needle of length L is thrown randomly on a plane with parallel lines of distance D from each other with $D > L$, the probability that it crosses one of the lines is

$$p = \frac{2L}{\pi D}. \tag{1.1}$$

A few decades later, Laplace (1812) suggested that by repeating the process N times and counting the number of crosses, one could calculate π:

$$\frac{N_{\text{crossed}}}{N_{\text{total}}} \approx \frac{2L}{\pi D} \Rightarrow \pi \approx \frac{2L}{D} \frac{N_{\text{crossed}}}{N_{\text{total}}}. \tag{1.2}$$

Although the basis of the method was well-known, the first time it saw practical use was just after World War II, where scientists in the Los Alamos National Lab in

the United States, including John von Neumann, Stan Ulam and Nicholas Metropolis, used it in neutronics in order to improve thermonuclear weapons development (Metropolis and Ulam, 1949). The term *Monte Carlo*, referring to the famous hill in Monaco where the casino is located, was coined by Metropolis, after a relative of Ulam that enjoyed gambling.

With the constant improvement in computing power, Monte Carlo simulation soon became mainstream, and gained popularity in several fields of research. The ability to sample from a large number of possible scenarios and predict the outcome quickly found use in finance, operations research and risk analysis (Fishman, 1996; Rubinstein, 1981). Monte Carlo methods are currently used in a large and diverse number of fields. Statistical physics and molecular modelling make extensive use of Monte Carlo, as do finance and stock market analysis. Risk and reliability assessment are well suited to stochastic approximations, while computer science algorithms, artificial intelligence and game theory inherit its methodology. Finally, particle transport is a natural target field for Monte Carlo, whether that includes ray-tracing for graphics, γ-rays for radiation and biological studies or neutron transport for nuclear and reactor physics, the subject of this thesis.

1.1 Probability distribution functions and sampling

The Monte Carlo method uses random numbers to model problems that can be given a stochastic or probabilistic interpretation (Hammersley and Handscomb, 1964; Manno, 1999). Initially, one has to transform the problem to one that can be solved stochastically and then, using random numbers, simulate a large number of events in order to observe the behaviour of the system. Once we transform our initial problem to a stochastic one, the processes characterizing the problem can be defined by *probability density functions*.

If the probability that a single random variable X inside a domain Z has a value between x and $x+dx$ is

$$p(x < X < x+dx) = f(x)dx\,, \tag{1.3}$$

then $f(x)$ is called the *probability density function*, or pdf in short notation. Since X must have a value in Z, the normalisation condition for $f(x)$ can be easily deduced:

$$\int_Z f(x)dx = 1 \tag{1.4}$$

In order to take samples from a pdf, an easier way is using the *cumulative* distribution function, or CDF:

$$P(x) = \int_{x_{\min}}^{x} f(x')dx' \tag{1.5}$$

Being a cumulative function, $P(x)$ is monotonically increasing in the interval $x_{\min} \leq x \leq x_{\max}$ with boundary values

$$P(x_{\min}) = 0\,, \tag{1.6}$$

$$P(x_{\max}) = 1\,. \tag{1.7}$$

Since $P(x)$ ranges from 0 to 1, it is now simple to sample from it using a uniformly distributed variable ξ on the interval $[0,1]$. There are several methods to sample from $P(x)$: A simple, computationally efficient method, is to generate a tabulated distribution of $P(x)$ values by interpolation and select from them using ξ.

Another method, very popular in the earlier days of Monte Carlo, is the rejection method (von Neumann, 1951). According to the rejection method, samples are taken from an arbitrary distribution $g(x)$, under the only condition that $p(x) < Mg(x)$, where $M > 1$ is an appropriate upper bound on $p(x)/g(x)$. The results are then accepted or rejected depending on whether they fall inside the domain of $p(x)$.

In this work, we are mainly using the inversion method, in which we set the cumulative probability of an event $P(x)$ equal to ξ and solve for $P^{-1}(x)$. By selecting ξ using random numbers, we can then repeatedly sample from $P(x)$ and therefore $p(x)$. An example of the inversion method, and one we shall be using in this thesis, is the selection of a neutron's path length between collisions. The probability that a neutron has an interaction in dx after having travelled a distance x is given by

$$p(x)dx = \Sigma_t e^{-\Sigma_t x} dx\,, \tag{1.8}$$

where Σ_t is the total interaction cross section. The pdf is then

$$p(x) = \Sigma_t e^{-\Sigma_t x}\,, \tag{1.9}$$

and the CDF is

$$P(x) = \int_{0}^{x} \Sigma_t e^{-\Sigma_t x} dx = 1 - e^{-\Sigma_t x}\,. \tag{1.10}$$

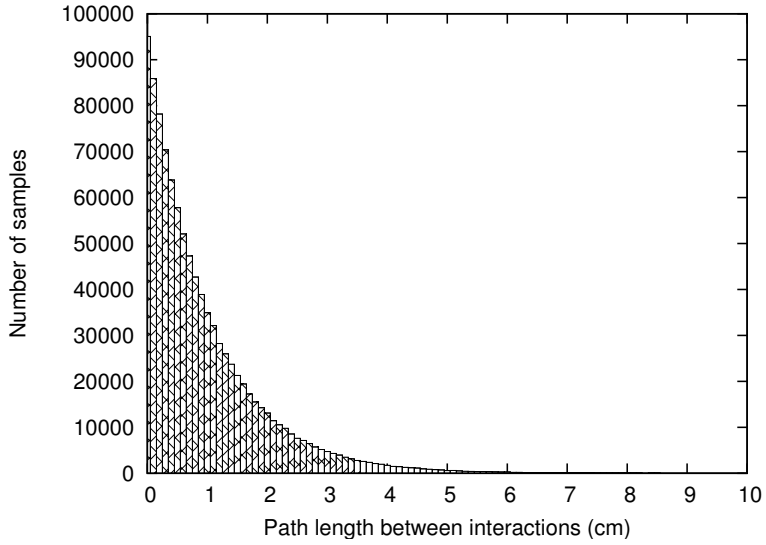

Figure 1.1: Sampling the path length between interactions 10^6 times using random numbers, for a total cross-section of 1cm^{-1}.

We can now sample the path length x by equating the CDF to a random number ξ, uniformly generated between 0 and 1, and solving for x:

$$\xi = 1 - e^{-\Sigma_t x} \Rightarrow x = \frac{\ln(1-\xi)}{\Sigma_t} = \frac{\ln(\xi')}{\Sigma_t}. \tag{1.11}$$

Rather than using $1-\xi$, we can directly use ξ', since ξ and ξ' are selected from a uniform distribution between 0 and 1. In Fig. 1.1, we can see (in histogram form) the result of sampling the path length 10^6 times, using a Σ_t value of 1 cm^{-1} and random numbers uniformly distributed between 0 and 1. As expected, the distribution approaches the analytical solution, which is an exponential.

1.2 Monte Carlo in neutronics

The aim of a Monte Carlo simulation in nuclear reactor physics is to calculate the response of a detector (which can be a physical detector or a virtual detector providing a response) at a point $P = (\mathbf{r}, E, \mathbf{\Omega})$, where $\mathbf{r}$ is the space, E the energy and

$\mathbf{\Omega}$ the direction variable. The response is calculated according to

$$R = \int \eta_\phi(P)\phi(P)dP, \tag{1.12}$$

where $\phi(P) = \phi(\mathbf{r}, E, \mathbf{\Omega})$ is the neutron flux at point P in the phase space, while $\eta_\phi(P)$ is the detector response function (Cashwell and Everett, 1959; Kalos and Whitlock, 1986; Lux and Koblinger, 1991; Spanier and Gelbard, 1969). Since the flux $\phi(P)$ is the solution of the analytical transport equation, we begin by noting the time-independent neutron transport equation, as we will not be dealing with time-dependent problems in this thesis:

$$\begin{aligned}\mathbf{\Omega}\cdot\nabla\phi(\mathbf{r}, E, \mathbf{\Omega}) + \Sigma_t(\mathbf{r}, E)\phi(\mathbf{r}, E, \mathbf{\Omega}) = \\ \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{r}, E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega})\phi(\mathbf{r}, E', \mathbf{\Omega}')dE'd\Omega' + S(\mathbf{r}, E, \mathbf{\Omega}).\end{aligned} \tag{1.13}$$

This is an integro-differential equation, which is not suitable for treatment by Monte Carlo. We need the integral form, which we will now derive by looking at the actual physical processes taking place in neutron transport.

A neutron begins its lifetime at a point P_0 in phase space, selected from the source $S(P_0)$. It then follows a path until its next interaction in phase-space. We calculate its path by using the so called *transition kernel* T, where

$$T(\boldsymbol{r' \to r}, E', \mathbf{\Omega'})dr$$

is the probability that a particle starting a flight path at $\mathbf{r}'$ will have its next collision in dr at $\mathbf{r}$. After the new collision point has been sampled, the collision process must be sampled. This is done by the *collision kernel* C, where

$$C(\boldsymbol{r}, E' \to E, \boldsymbol{\mathbf{\Omega' \to \Omega}})dEd\Omega$$

is the probability that a particle will exit a collision at $\mathbf{r}$ with direction $\mathbf{\Omega}$ in $d\Omega$ and energy E in dE.

In order to make use of those probabilities, we can define a number of event densities, namely the *collision* density $\psi(\mathbf{r}, E, \mathbf{\Omega})$ and *emission* density $\chi(\mathbf{r}, E, \mathbf{\Omega})$, where

$$\psi(P)dV = \psi(\mathbf{r}, E, \mathbf{\Omega})dV \tag{1.14}$$

is the expected number of particles entering a collision at a point $\mathbf{r}$ in dV with energy E and direction $\mathbf{\Omega}$, while

$$\chi(P)dE d\Omega = \chi(\mathbf{r}, E, \mathbf{\Omega})dE d\Omega \tag{1.15}$$

is the expected number of particles starting a flight path at $\mathbf{r}$ with energy E in dE and direction $\mathbf{\Omega}$ in $d\Omega$.

We can use the concept of transition and collision kernel in order to directly define the Boltzmann equation in terms of a Monte Carlo calculation. $\chi(P)$ is defined as

$$\chi(P) = S(P) + \int T(\boldsymbol{r'} \rightarrow \boldsymbol{r}, E', \mathbf{\Omega'})C(\mathbf{r}, E' \rightarrow E, \mathbf{\Omega'} \rightarrow \mathbf{\Omega})\chi(P')dP' \tag{1.16}$$

where $S(P)$ is source of the particles at P. We see that if by starting at a point P', we can sample emission density of the next point P by first sampling the transition kernel $T(\mathbf{r'} \rightarrow \mathbf{r}, E', \mathbf{\Omega'})$ in order to select a new collision point, and after that we sample the collision kernel $C(\mathbf{r}, E' \rightarrow E, \mathbf{\Omega'} \rightarrow \mathbf{\Omega})$ in order to get the next sample of the emission density. If we want to sample the collision density $\psi(P)$, where

$$\psi(P) = S_1(P) + \int C(\mathbf{r'}, E' \rightarrow E, \mathbf{\Omega'} \rightarrow \mathbf{\Omega})T(\mathbf{r'} \rightarrow \mathbf{r}, E, \mathbf{\Omega})\psi(P')dP' \tag{1.17}$$

we can start by sampling the collision kernel (since we are entering a collision, we already know the position of the collision) and then sample the transport kernel for the next collision. Also note that we use the term $S_1(P)$ instead of $S(P)$ since in this case the source density cannot be the initial source density, but the source of first collisions. In order to make the formulation easier, we can combine the transition and collision kernels into the transport kernel K, where

$$K(\mathbf{P'} \rightarrow \mathbf{P}) = T(\mathbf{r'} \rightarrow \mathbf{r}, E', \mathbf{\Omega'})C(\mathbf{r}, E' \rightarrow E, \mathbf{\Omega'} \rightarrow \mathbf{\Omega}) \tag{1.18}$$

or a kernel L, where

$$L(\mathbf{P'} \rightarrow \mathbf{P}) = C(\mathbf{r'}, E' \rightarrow E, \mathbf{\Omega'} \rightarrow \mathbf{\Omega})T(\mathbf{r'} \rightarrow \mathbf{r}, E, \mathbf{\Omega}) \tag{1.19}$$

Now the integral equations for $\chi(P)$ and $\psi(P)$ will be as follows:

$$\chi(P) = S(P) + \int K(P' \rightarrow P)\chi(P')dP' \tag{1.20}$$

$$\psi(P) = S_1(P) + \int L(P' \rightarrow P)\psi(P')dP' \tag{1.21}$$

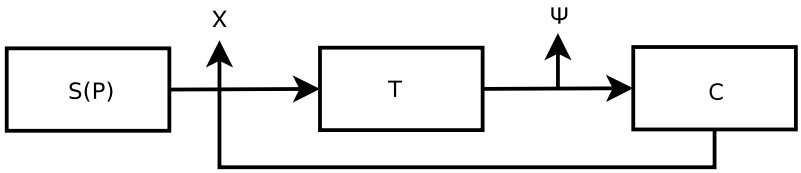

Figure 1.2: Schematic diagram of particle transport, where S is the source and T and C the transition and collision kernels respectively.

Via this scheme it is easy to see when the collision and emission densities can be sampled, as it is shown in Fig. 1.2. By using the relation between flux and collision density,

$$\psi(P) = \Sigma_t(P)\phi(P), \tag{1.22}$$

we can define the detector response in terms of $\psi(P)$:

$$R = \int \eta_\psi(P)\psi(P)dP \tag{1.23}$$

where

$$\eta_\psi(P) = \frac{\eta_\phi(P)}{\Sigma_t} \tag{1.24}$$

is the detector response function with respect to $\psi(P)$. So now, by sampling $\psi(P)$ for a number of histories, we can obtain an estimate of the detector response R.

1.3 Monte Carlo and eigenvalue problems

A large category of problems that can be solved with the Monte Carlo method are the k-eigenvalue or *criticality* problems. In eigenvalue problems, the fixed source component in the neutron transport equation is replaced by the source comprised

of fission neutrons, resulting in the following equation:

$$\begin{aligned}\boldsymbol{\Omega}\cdot\nabla\phi(\mathbf{r},E,\boldsymbol{\Omega})+\Sigma_t(\mathbf{r},E)\phi(\mathbf{r},E,\boldsymbol{\Omega})=&\\ \int_0^\infty\int_{4\pi}\Sigma_s(\mathbf{r},E'\to E,\boldsymbol{\Omega}'\to\boldsymbol{\Omega})\phi(\mathbf{r},E',\boldsymbol{\Omega}')dE'd\Omega'&\\ +\frac{1}{k}\frac{1}{4\pi}\chi_f(E)\int_0^\infty\int_{4\pi}\nu(E')\Sigma_f(\boldsymbol{r},E')\phi(\mathbf{r},E',\boldsymbol{\Omega}')dE'd\Omega'.&\end{aligned} \tag{1.25}$$

Here, $\chi_f(E)$ is the fission spectrum, ν is the average number of neutrons produced per fission, $\Sigma_f(\boldsymbol{r},E)$ is the fission cross-section and k is the *k-eigenvalue*, or *k-effective*.

k-effective does have a physical meaning: it can be seen as the ratio of total production rate of neutrons via fission over total neutron loss rate via leakage and absorption. The value of *k-effective* is what determines the stationarity of a reactor: when $k_{eff}=1$, the system is self-sustaining, or *critical*. $k_{eff}<1$ means a sub-critical system that cannot be sustained without an external source, while $k_{eff}>1$ is a supercritical system that needs to be controlled.

1.4 Comparison between Monte Carlo and deterministic methods

Apart from the Monte Carlo method, another common type of methods can be used in order to solve the transport equation, called *deterministic* methods. Deterministic methods typically require discretising the equations governing neutron transport in all independent variables. The equations are then converted into large algebraic systems and used in order to approximate the solution functions.

In deterministic methods, the energy variable of the transport equation is discretised by limiting the possible energies of neutrons and then dividing them into energy groups. Cross-sections are then calculated per energy group, and the equations are integrated over each group, resulting in the *multigroup* transport equations. The angular variable is usually discretised by one of two methods: in the *discrete-ordinates*, or S_n, method, the Gauss-Legendre quadrature set is used to approximate integrals over angle, while in the *spherical harmonics*, or P_n method,

Item	Deterministic	Monte Carlo
Geometry representation	(Usually) Discrete	Exact
Energy representation	Discrete	Continuous/Exact
Direction representation	Discrete/Series	Continuous
Numerical issues	Convergence	Statistical uncertainty
Amount of information	Large	Limited
CPU memory requirements	Large	Small
CPU time cost	Small	Large
Parallel computing	Complex	Simpler
Scaling to multiple dimensions	Costly	Simple

Table 1.1: A summary of the main differences between Monte Carlo and deterministic methods.

the angular fluxes are expanded in a number of spherical harmonic functions. S_n methods are arguably the closest form to analytically solving the transport equation, and are the most widely-used methods in neutronics. Spatial discretisation is the most important consideration, since the truncation error is the main source of error in a deterministic calculation. There are several methods used, such as the finite differencing, finite element or nodal methods, each with its own strengths and limitations. A detailed analysis of such methods is beyond the scope of this thesis.

We have already discussed the main methodology of the Monte Carlo method, so in order to summarize, we can mention the main differences between the two methods and try to compare the merits of each. A summary of those differences is shown in Table 1.1. The main advantage of the Monte Carlo method is that it is able to model continuous energy, space, and angle in irregular, complicated geometries. While deterministic methods become less exact (because of discretisation error) or less efficient (because of vastly increased CPU and memory usage) with more complex problems, Monte Carlo methods are able to handle them without penalties, other than a reasonable increase in calculation time.

On the other hand, a significant advantage of the deterministic methods is that they are able to provide a large amount of information on the solution functions much faster than Monte Carlo. Since the equations governing transport are solved at every node of the system, one can get information for any point in the domain via the same calculation. Meanwhile, a Monte Carlo simulation is a lot more effective

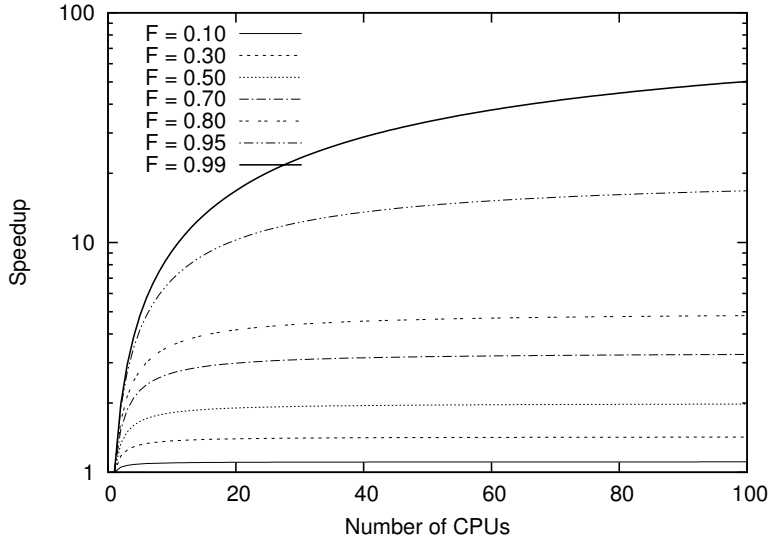

Figure 1.3: Speed-up in parallel computing as defined by Amdahl's law, where F defines the fraction of the total work done in parallel. For small values of F, using more than a dozen CPUs does not improve the performance of the system.

when a limited amount of information, such as a detector response, is sought, and very inefficient when multiple responses from the system are required. In addition, when a large number of particles is required in order to obtain sufficient statistical information in a Monte Carlo simulation, it can be much slower than a comparable deterministic simulation, since the lifetime of all particles must be fully tracked, while the deterministic simulation is only limited by the level of discretisation in all variables.

One advantage of Monte Carlo over deterministic methods that can offset the problem mentioned above is scalability. With the continuous advances in computing, it is now simple to perform large-scale simulations using computing grids. Amdahl's Law (Amdahl, 1967) states that the speed-up in a parallel simulation can be estimated as

$$\text{Speed-up} = \frac{1}{1 - F + \frac{F}{N}}, \tag{1.26}$$

where F is the fraction of work done in parallel and N is the number of CPUs. A

plot of the speed-up versus the number of CPUs can be seen in Fig 1.3. Because of the nature of the Monte Carlo method, multiple independent samples can be calculated at the same time, and therefore F is much higher than deterministic calculations, as it can reach up to 99.5%. In fact, the only limiting factor is the data input/output speed between the systems performing the calculations and the system collecting the results. This allows Monte Carlo simulations to efficiently utilize larger computing clusters and further reduce the gap regarding CPU time cost.

There is also the subject of reproducibility. In deterministic calculations, the name itself suggests that one can reproduce the exact same result by feeding the system with the same initial conditions. In a Monte Carlo calculation, since the outcome of events is governed by random numbers, the results will never be exactly the same, but will form a statistical distribution. In practice, especially when concerning computer code development and debugging, it is advantageous not to generate truly random numbers, but sequences of random numbers that can be reproduced reliably. These numbers are called pseudo-random numbers.

1.5 Aim of current work

The estimation and reduction of statistical error in a Monte Carlo simulation has long been a subject of investigation around the globe. Being the major limitation of such a powerful tool, several methods for the reduction of this error, called *variance reduction* methods, have been proposed. A review of those methods is given in Chapter 2 of this thesis.

The Reactor Physics group of the Delft University of Technology has played an active role in such research. Hoogenboom (1977) investigated the use of adjoint functions in order to reduce the error in Monte Carlo source-detector problems in neutronics. Serov (1996) managed to reduce the error by running forward and adjoint Monte Carlo simulations and correlating the results. Ueki and Hoogenboom (2001) used the correlation between forward and adjoint simulations in order to perform perturbation analysis using Monte Carlo, while Légrády (2005) successfully applied the technique to the field of borehole logging.

However, most of the work involving variance reduction is focused on source-detector problems, and leaves out eigenvalue calculations. In such calculations, additional issues arise when compared to source-detector problems: extra com-

putational time is required to obtain a converged source distribution and a large number of neutron generations must be simulated in order to eliminate the inter-generation bias and obtain acceptable statistics. In addition, since the eigenvalue problem is by default a global problem, Monte Carlo codes struggle to keep up with deterministic ones in terms of calculation time. Finding a solution for such a demanding problem can be very challenging and certainly less trivial than in the case of source-detector simulations, something we are hoping to change with the findings of this thesis.

Our method will attempt to make eigenvalue simulations more efficient not so much by speeding up the calculation, but by reducing the *variance* of the estimate for a given calculation time. We will show that by using the adjoint function as a measure of the importance of a region in the spatial, angular and energy domains, we can direct the simulated particles to those regions that matter for our simulation, therefore reducing our global problem to a more localized one. We will discuss how this is achieved by manipulating the transport kernels and particle weights and how the localization of the contributions increases the information from the system, bypassing the problem of a global simulation.

By applying the scheme to a simple system, it will become clear that the scheme can indeed reduce the variance to zero, but only if specific, non-practical conditions apply. We will then see how the scheme can still significantly decrease the variance in more practical problems by using approximations. Finally, we will also integrate the scheme into a production Monte Carlo code and see that the decrease in variance is still obvious, but at the cost of much higher CPU time cost.

However, before we dwell into the subject of this thesis, we will first introduce in Chapter 2 the concept of variance reduction in a Monte Carlo calculation and discuss the merits and limitations of existing methods.

1.6 Bibliography

G. Amdahl. Validity of the single processor approach to achieving large-scale computing capabilities. In *AFIPS Conference Proceedings*, volume 30, 483–485, 1967.

E. D. Cashwell and C. J. Everett. *A Practical Manual on the Monte Carlo Method for Random Walk Problems*. Pergamon Press, 1959.

G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, 1996.

J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Methuen & Co., 1964.

P. Holgate. Studies in the history of probability and statistics XXXIX: Buffon's cycloid. *Biometrika*, **68**, 712, 1981.

J. E. Hoogenboom. *Adjoint Monte Carlo Methods in Neutron Transport Calculations*. Ph.D. thesis, Delft University of Technology, 1977.

M. H. Kalos and P. A. Whitlock. *Monte Carlo Methods*. John Wiley & Sons, 1986.

P. S. Laplace. *Théorie analytique des probabilités*. V. Courcier, Paris, 1812.

D. Légrády. *Improved Monte Carlo Methods With Application to Borehole Logging Simulations*. Ph.D. thesis, Delft University of Technology, 2005.

I. Lux and L. Koblinger. *Monte Carlo particle transport methods: neutron and photon calculations*. CRC Press, Boca Raton, 1991.

I. Manno. *Introduction to the Monte Carlo method*. Akadémiai Kiadó, Budapest, 1999.

N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, **44**, 335, 1949.

J. von Neumann. Various techniques used in connection with random digits. In *Monte Carlo Methods, Applied Mathematics Series*, volume 12, 36. National Bureau of Standards, 1951.

R. Y. Rubinstein. *Simulation and the Monte Carlo method*. John Wiley & Sons, 1981.

I. Serov. *Estimation of Detector Responses by Midway Forward and Adjoint Monte Carlo Coupling in Nuclear Systems*. Ph.D. thesis, Delft University of Technology, 1996.

J. Spanier and E. M. Gelbard. *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley Publishing Co., 1969.

T. Ueki and J. E. Hoogenboom. Exact Monte Carlo perturbation analysis by forward-adjoint coupling in radiation transport calculations. *Journal of Computational Physics*, **171**, 509, 2001.

CHAPTER 2

VARIANCE REDUCTION IN MONTE CARLO

Monte Carlo simulation is a stochastic process, and as such there is always a statistical error associated with the result. The determination and reduction of this error is a major field of current research and the main subject of this thesis.

In this chapter we will deal with the basis of statistical analysis in Monte Carlo. We will mention the main quantities required and give a definition of variance reduction. We will introduce the concept of importance and relate it to variance reduction. We will then show the significance of the adjoint function as importance function and will mention a number of commonly used variance reduction techniques, some utilizing the adjoint function. Finally, most of the current variance reduction methods in Monte Carlo only apply to shielding problems and very few are used when dealing with multiplying systems. We will discuss the shortcomings of the current methods regarding criticality problems and propose some improvements.

2.1 Estimated accuracy of the Monte Carlo results

As discussed in Chapter 1, the Monte Carlo method solves integral equations by replacing them by sums of a large number of samples and calculating the mean values of the samples. We now need to know what the mean of the samples represents, and how can we define the statistical accuracy of the approximation. We can use two important theorems in statistics in order to answer these questions, the *law of large numbers* (Bernoulli, 1713) and the *central limit theorem* (Laplace, 1812).

If we define the *sample mean* of a series of samples $(m_1, m_2, \dots m_N)$ as

$$\overline{m} = \frac{1}{N}\sum_{i=1}^{N} m_i \,, \tag{2.1}$$

according to the law of large numbers, the *expected value* $E(m)$ equals

$$E(m) = \lim_{x\to\infty} \frac{1}{N}\sum_{i=1}^{N} m_i \,. \tag{2.2}$$

Therefore, as the number of samples approaches infinity, the mean value converges to the expected value we want to calculate. The variance is estimated by

$$\sigma_m^2 = \frac{1}{N-1}\sum_{i=1}^{N}(m_i - \overline{m})^2 \,, \tag{2.3}$$

where the term $\frac{1}{N-1}$ ensures that σ_m^2 is an unbiased estimate of the variance. The variance of the mean $\sigma^2 \equiv \sigma_{\overline{m}}^2$ can be calculated from

$$\sigma^2 = \frac{\sigma_m^2}{N} = \frac{1}{N(N-1)}\sum_{i=1}^{N}(m_i - \overline{m})^2 \,. \tag{2.4}$$

In practice, a more convenient form of equation (2.4) is used, which does not require storage of all the m_i until the final estimate:

$$\sigma^2 = \frac{1}{N-1}\left[\frac{1}{N}\sum_{i=1}^{N} m_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} m_i\right)^2\right] . \tag{2.5}$$

Using this form, only the square of the sum and the sum of the squares of the samples are required. The *standard deviation* σ of the mean is now defined as the

square root of the variance. The error in a Monte Carlo simulation is typically given as the *relative* standard deviation $\sigma_{\overline{m}}/\overline{m}$.

In addition, the central limit theorem states that the sample mean of a large number of identically distributed independent random variables x_i with finite expectation values μ and variance σ^2, is itself a random variable that approximately follows a *Gaussian* (also called *normal*) distribution, irrespective of the shape of the distribution of x_i. The pdf of x has the following form:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.6}$$

The fact that, for sufficiently large N, $\overline{m}$ follows a normal distribution allows the use of *confidence intervals*, which give a more practical description of the statistical accuracy of the simulation.

$$P(\mu - a\sigma < x < \mu + a\sigma) = \int_{\mu-a\sigma}^{\mu+a\sigma} p(x)dx \tag{2.7}$$

In a normal distribution around 68% of the values fall between 1σ of the mean, while around 95% will be no further than 2σ from the mean.

2.2 Aim of variance reduction methods

As the name suggests, the target of any variance reduction method is to reduce the variance of the estimate. As we can see from equation (2.4),

$$\sigma \sim \frac{1}{\sqrt{N}}. \tag{2.8}$$

Therefore, one can argue that simply increasing the number of histories will provide the reduction in variance needed.

On the other hand, in a Monte Carlo run, the computational time is directly proportional to the number of histories N, for sufficiently large N and assuming fully independent histories. Therefore, using the naive approach of increasing N in order to get the reduction expected might not be practical or even possible, due to time constraints.

A better definition of the target in variance reduction could then be to reduce the variance of a system for a given simulation time, or speed up the calculation for a target σ.

However, some schemes directly reduce the variance but increase the total time of the simulation, while others reduce the simulation time, thus allowing more particle histories to be simulated (therefore reducing the total variance). Therefore, it makes sense to compare the performance of a Monte Carlo scheme not by comparing σ^2 or T, but by using a combination of the two. We can define the *figure of merit* (or *efficiency*) of a simulation(Glynn and Whitt, 1992) as

$$FOM \equiv \frac{1}{\sigma_{\text{rel}}^2 T}, \tag{2.9}$$

where σ_{rel} is the relative standard deviation and T the total time of the simulation. In a normal, analogue Monte Carlo simulation, the figure of merit should remain approximately constant, regardless of the number of histories used. Therefore, we can now better define the target of any variance reduction scheme as to increase the figure of merit of a simulation. Throughout this text, and in general, it is the efficiency increase of a calculation that we will be referring to as variance reduction.

2.3 Basic variance reduction methods

From equation (2.9) we can deduct that there are two main paths towards variance reduction, either reducing the simulation time or reducing the variance, while the other one increases at a lesser rate. As there is a large number of techniques that can be used for either path, a full review of the variance reduction techniques available is beyond the scope of this thesis. We will now give a short overview of the most important methods used and direct the reader to more in-depth reviews (Haghighat and Wagner, 2003; Lux and Koblinger, 1991; Spanier and Gelbard, 1969).

2.3.1 Importance sampling

Intuitively, one can think of increasing the FOM of a Monte Carlo simulation by encouraging particle histories which are more likely to contribute to the required response, while the particle histories less likely to contribute are somehow discarded, or given less consideration. This way, at least one of the aforementioned

conditions (reduction of variance or simulation time) can be satisfied. This can be done by altering the pdf that describe the various processes sampled.

We start by recalling the target of a Monte Carlo simulation, to calculate the expected value of a function $f(x)$:

$$E_p(f) = \int_a^b f(x)p(x)dx\,, \tag{2.10}$$

with $p(x)$ the pdf of $f(x)$. Now, if we sample a modified pdf $p^*(x)$, we are no longer calculating $E_p(f)$ but rather we are calculating the expected value of the modified function $E_{p^*}(f^*)$. In order for the result to be unbiased, i. e. for the estimator's expected value and the true value of $E_p(f)$ to be the same, we need to compensate for the change by assigning a weight $w(x) = p(x)/p^*(x)$. Now we are estimating $f^*(x) = f(x)w(x) = f(x)p(x)/p^*(x)$ and the expected value of f^* is the same as the original function:

$$E_{p^*}(f^*) = \int_a^b f^*(x)p^*(x)dx = \int_a^b f(x)w(x)p^*(x)dx = \int_a^b f(x)p(x)dx = E_p(f) \tag{2.11}$$

However, the variance is not the same, since:

$$\begin{aligned} V(f^*) &= E_{p^*}(f^{*2}) - E_{p^*}(f^*)^2 = \int_a^b f^{*2}(x)p^*(x)dx - E_p(f)^2 \\ &= \int_a^b f^2(x)\frac{p^2(x)}{p^{*2}(x)}p^*(x)dx - E_p(f)^2 = \int_a^b \left[\frac{p(x)}{p^*(x)}\right] f^2(x)p(x)dx - E_p(f)^2\,. \end{aligned} \tag{2.12}$$

If we choose $p^*(x)$ appropriately so that $\frac{p(x)}{p^*(x)} < 1$ in regions of interest, it is likely from Eq. (2.12) that the variance is reduced, that is,

$$V(f^*) < V(f). \tag{2.13}$$

Since $p(x)$ and $p^*(x)$ are pdfs, they are normalised to integral 1, which means that there should also be regions where $\frac{p(x)}{p^*(x)} > 1$, or where the contribution of the region to the variance is increased. Hence, we have managed to reduce the variance in our region of interest by increasing it in different regions of the system.

If we take $p^*(x) = f(x)p(x)/E_p(f)$, with $p^*(x)$ appropriately normalised and $E_p(f) \neq 0$, we can obtain a result with zero variance:

$$V(f^*) = \int_a^b \left[f(x)\frac{p(x)}{p^*(x)} - E_p(f) \right]^2 p^*(x)dx = 0 \tag{2.14}$$

Of course, this choice of $p^*(x)$ is not very practical, since it requires knowledge of $E_p(f)$, which is the target of our simulation.

The question now becomes how can we know which $p^*(x)$ gives the highest FOM for our system and which particle histories are more likely to contribute. The answer to this question can only be given after we give particles a measure of that "significance", and only in terms of variance (since simulation time depends on many more parameters). This measure is called *importance* of a particle.

The importance $I(P)$ of a particle is defined as the expected contribution to the detector response of a particle starting a flight path at point $P = P(\mathbf{r}, E, \mathbf{\Omega})$ (Lewins, 1965).

If we modify the pdf describing the interactions of a particle throughout its lifetime by the appropriately normalised importance of the particle, we obtain a biased pdf:

$$p^*(x) = p(x)I(x) \tag{2.15}$$

In order to keep our final estimate unbiased, we then apply a weight of

$$w(x) = \frac{p(x)}{p^*(x)} = \frac{1}{I(x)} \tag{2.16}$$

to the particle when we are sampling $p^*(x)$. We will see in the following sections how to obtain $I(P)$ and how it will affect the sampling process and the response estimate we get from the detector.

2.3.2 Implicit capture

In highly-absorbing systems, such as thermal reactors, the particles can get absorbed after only a few interactions, therefore never reaching the scoring region. In such cases, an easy way to reduce the variance of the simulation is to extend the lifetime of particles by removing absorption and replacing it by a change in

the particle weight that accounts for the absorption probability. This technique is called *implicit capture*.

If the absorption probability is Σ_a/Σ_t, the new weight of a particle after a scattering event is

$$w' = w\,(1 - \frac{\Sigma_a}{\Sigma_t}). \tag{2.17}$$

Since the particle cannot be absorbed, the particle history will continue indefinitely, unless it can escape from the system, or we apply a lower weight boundary at which particles are killed. Implicit capture always reduces the variance, but the total figure-of-merit may not improve, as the simulation time is increased because of the longer particle histories. It is, however, widely used because of its simplicity and ease of implementation.

2.3.3 Russian roulette

Russian roulette prevents particles with low expected contributions to the detector response from being simulated for too long, thus decreasing the total simulation time without an appreciable change in variance. When a particle has its statistical weight w drop below a predefined limit w_{RR}, the particle undergoes Russian roulette with probability of survival

$$p_{\text{surv}} = \frac{w}{w_s}. \tag{2.18}$$

Should the particle survive, its statistical weight is increased to w_s. Using this weight correction, the average weight of particles after Russian roulette is

$$\overline{w} = \frac{w}{w_s} w_s + \left(1 - \frac{w}{w_s}\right) 0 = w \tag{2.19}$$

which ensures a fair game. When the survival weight w_s is set to twice the threshold weight w_{RR}, a common selection, a particle at the weight threshold has exactly 0.5 chance of survival.

Since Russian roulette manipulates weights, it is of no use in an analogue Monte Carlo calculation, where particle weights do not exist, but must be used together with implicit capture or another method that changes the particle weight.

Russian roulette can be applied at any time during the life of a particle, usually after an interaction has taken place. The weight w_s may also depend on the position, energy and direction of a particle. Russian roulette always increases variance

since it cuts off histories that could still contribute to the detector, but it also always reduces the simulation time when compared to an implicit capture scheme without weight thresholds.

2.3.4 Splitting

Splitting is a simple but effective method to decrease the variance of a simulation by increasing the number of possible scores at some stage during the simulation. When the simulation weight w of a particle exceeds a limit defined by the user, the particle is split into a number of particles N, each with weight w/N. The particles are then followed during their lifetime and the scores obtained added to that of the original particle. Again, as with Russian roulette, splitting only works with methods where the weight of the particles changes.

Due to the need for following the extra particles generated during the process, splitting always increases the simulation time, however, the variance is decreased because of the extra scores accumulated.

Splitting and roulette are very common techniques in Monte Carlo simulation and not only because of their simplicity. Since they only deal with variance reduction via population control and do not modify pdfs, they can be used in addition to most other techniques for an added effect.

2.3.5 Weight windows

One of the most used and effective variance reduction methods is the *weight-window* method (Booth, 1982; Booth and Hendricks, 1984). Space and energy dependent windows are generated, and the particles that are outside the window when they cross a point in phase-space undergo splitting or Russian roulette, depending on the ratio of the particle weight to the weight boundaries of the window.

The weight-window method simultaneously deals with both the direct decrease of variance via a large number of samples (through splitting) and the decrease of simulation time via Russian roulette, and is therefore a very effective variance reduction technique.

The weight windows can be set intuitively or via the use of importances. The

latter can be a non-trivial and time consuming task, but yields excellent results if used correctly.

2.3.6 Exponential transform

The exponential transform method (Clark, 1966), also called *path length stretching*, alters the path of a particle towards preferred directions through the use of a modified cross-section Σ^*, where

$$\Sigma^*(\mathbf{r}, E) = \Sigma_t(\mathbf{r}, E)[1 - k\mathbf{\Omega}_{\mathbf{det}} \cdot \mathbf{\Omega}], \tag{2.20}$$

where $\Sigma^*(\mathbf{r}, E)$ is the transformed total cross-section, $0 < k < 1$ is the biasing strength parameter, $\mathbf{\Omega}$ is the current direction of a particle and $\mathbf{\Omega}_{\mathbf{det}}$ the direction of the region where the particle score is calculated, i.e. the preferred direction. The distance s to the next collision is now sampled from

$$p^*(s) = \Sigma^* e^{-\Sigma^* s}. \tag{2.21}$$

Using this formulation, we can see that when $\mathbf{\Omega}_{\mathbf{det}} \cdot \mathbf{\Omega}$ approaches unity, i.e. when the particle is moving towards the (virtual) detector, the interaction cross-section becomes smaller, so that the particle travels further towards the detector before a collision. Accordingly, when $\mathbf{\Omega}_{\mathbf{det}} \cdot \mathbf{\Omega}$ approaches -1, the cross-section becomes large, so that the particle travels a small distance before a possible scattering event sends it towards the detector. The weight of the particle must then be modified, in order to compensate for the cross-section change. The new weight is

$$w = w' \frac{p(s)}{p^*(s)} = w' \frac{\Sigma_t e^{-\Sigma_t s}}{\Sigma^* e^{-\Sigma^* s}} = w' \frac{e^{-\Sigma_t k \mathbf{\Omega}' \cdot \mathbf{\Omega} s}}{1 - k\mathbf{\Omega}_{\mathbf{det}} \cdot \mathbf{\Omega}} \tag{2.22}$$

From Eq. (2.22), we can see that correlated sampling can lead to high particle weights if $k\mathbf{\Omega}_{\mathbf{det}} \cdot \mathbf{\Omega} \to 1$, so it should be used with care.

2.3.7 Stratified sampling

The *stratified sampling* method (Neyman, 1934) attempts to reduce the variance by dividing the full integration space into subspaces, performing a Monte Carlo integration in each subspace, and adding up the results in the end. This allows a larger number of samples to be taken in places where under-sampling could occur,

which would normally result in underestimated variance. However, the technique is quite sensitive to the choice of subspaces, since an incorrect choice can actually increase the variance by oversampling regions of lower importance.

2.3.8 Forced collision

Forced collision (Cashwell and Everett, 1959) is a method used to sample collisions in media where the mean free path is large enough that too few collisions are sampled for satisfactory statistics, since particles escape very quickly from the system. This is done by altering the statistical weight of a particle according to the collision probability within a region and then sample a collision inside that region with probability 1. The usual way to achieve this is by splitting the original particle in two, with weights relative to the collision probability. In this way, the "original" particle can continue its trajectory with its weight reduced by the non-collision probability

$$P_{nc} = e^{-\Sigma_t d}\,, \tag{2.23}$$

where d is the distance to the boundary of the medium. Meanwhile, the other particle is forced to collide by selecting its path length s using a biased pdf

$$p^*(s) = \Sigma_t \frac{e^{-\Sigma_t s}}{1 - e^{-\Sigma_t d}}\,, \qquad 0 \le s \le d \tag{2.24}$$

The particle's weight is then reduced appropriately and Russian roulette is used to ensure that the calculation time is not increased significantly by the technique.

2.3.9 Other methods

A different approach to variance reduction has been shown by Becker et al. (2007). In this approach, the Monte Carlo simulation is performed as a correction to an initial deterministic estimate of the flux. Therefore the method does not solve the true Boltzmann equation by Monte Carlo, but rather uses virtual particles called "correctons", which are nevertheless related to real neutrons by a simple mathematical expression. The correcton flux has a much lower spatial variation than the physical neutron flux, which can significantly reduce the variance in deep penetration problems, where the flux between the source and detector decreases by several orders of magnitude.

Another fundamental technique is *correlated sampling* (Spanier and Gelbard, 1969). This technique is used in order to avoid small changes in the expected response (usually the result of a perturbation in the system) being masked by the larger statistical errors because of a limited number of samples. The technique uses the same random number sequences in order to estimate integrals via the Monte Carlo method. This way, the difference in the response is most likely not due to the statistical error but due to the difference between the integrals.

Correlated sampling is not a variance reduction technique in the same sense as the others here, but it can nevertheless significantly assist in determining the statistical accuracy of Monte Carlo simulations of perturbed systems.

2.4 Adjoint functions and importance

As we have seen in Chapter 1, the detector response in a forward Monte Carlo simulation can be calculated from the collision density ψ by

$$R = \iiint \eta_\psi(\mathbf{r}, E, \mathbf{\Omega})\psi(\mathbf{r}, E, \mathbf{\Omega}) d\mathbf{r}\, dE\, d\mathbf{\Omega} = \int \eta_\psi(P)\psi(P) dP, \tag{2.25}$$

where η_ψ is the detector response function with respect to ψ and P is a point in phase-space. However, we can also calculate the detector response using the adjoint form of Eq. (1.21):

$$\psi^*(P) = \eta_\psi(P) + \int L(P \to P')\psi^*(P') dP'. \tag{2.26}$$

Using this formulation, R is given by

$$R = \int S_1(P)\psi^*(P) dP. \tag{2.27}$$

If we compare Eqs. (2.25) and (2.27), we see that $\psi^*(P)$ is the contribution of a particle entering a collision at P to the detector response R. Therefore, we define $\psi^*(P)$ as the importance of a neutron entering a collision at P.

The detector response can be similarly defined using the adjoint emission density $\chi^*(P)$ as

$$R = \int S(P)\chi^*(P) dP, \tag{2.28}$$

where

$$\chi^*(P) = \eta_\chi(P) + \int K(P \to P')\chi^*(P')dP' \tag{2.29}$$

is the importance of a neutron leaving a collision/source at P and starting a new flight path. Here, η_χ is the detector response function with respect to χ. From equations (2.26) and (2.29), through transformation of the kernels, we arrive to the following relation between ψ^* and χ^*:

$$\psi^*(P) = \eta_\psi(P) + \iint C(\mathbf{r}, E \to E', \mathbf{\Omega} \to \mathbf{\Omega}')\chi^*(\mathbf{r}, E', \mathbf{\Omega}')dE'd\Omega', \tag{2.30}$$

The fact that the adjoint form of the collision and emission densities can be explained as the importance of particles entering or exiting a collision (or the source), respectively, means that we can make use of them as the importance function $I(P)$ in order to bias the pdfs and reduce the variance of our estimate. The significance of the adjoint function as importance function was initially shown by Coveyou et al. (1967), with further proof given by Hoogenboom (1979) and Noack (1979). We will discuss the adjoint functions in more detail in Chapter 3, when we develop our variance reduction scheme.

2.5 Variance reduction using adjoint functions

Since the basic variance reduction methods described in Sec. 2.3 are well established and widely used, the use of adjoint functions is now the main focus of research in Monte Carlo variance reduction. One of the most important aspects of any scheme is the ease of use, since manual generation and optimization of the importance functions can be a daunting task for a Monte Carlo code developer, let alone a user of the code. Therefore, in most cases, the adjoint solution from a deterministic calculation is used as an estimate of the importance. We will now shortly mention some characteristic methods currently in use and direct the reader to a more detailed review by Haghighat and Wagner (2003).

2.5.1 Tang's method

One of the first attempts to use deterministically-derived adjoint functions was made by Tang (1976). He used importance functions obtained from a 2D discrete

ordinates adjoint calculation as biasing functions for the source location, transition and collision kernel. Because of the direct biasing, the technique was limited to multigroup configurations.

2.5.2 Automated importance generation

Since the introduction of the weight-window technique, many efforts have been made to optimize the setting of the weight thresholds. Booth and Hendricks (1984) initially suggested the use of forward Monte Carlo, but since the adjoint function was established as an appropriate importance function, research shifted towards the use of adjoint functions for determining the weight thresholds for weight windows application. In addition, since a deterministic code can output space-, energy- and angle-dependent adjoint fluxes (and therefore importances), a large number of techniques have been developed that try to couple an adjoint deterministic calculation with a forward Monte Carlo calculation for variance reduction purposes.

Some of the initial attempts were performed by using adjoint diffusion calculations to generate space- and energy-dependent weight windows (Mickael, 1995; Miller et al., 1990). Although diffusion calculations are very efficient, they are not well suited for some of the problems Monte Carlo deals with (such as duct streaming problems) because of their inability to deal with highly absorbing systems or systems where vacuum regions exist.

The *AVATAR* methodology (Van Riper, 1997) uses the adjoint solution from a 3-D deterministic discrete-ordinates code, in order to generate space-, energy- and angle-dependent weight windows.

The CADIS (*Consistent Adjoint Driven Importance Sampling*) methodology, developed by Wagner (1997), focuses on the automated generation of weight windows for a forward Monte Carlo simulation via a completely automated system coupling Monte Carlo and deterministic runs. The system initially generates geometrical meshes and multigroup cross-sections for use by the deterministic code. The adjoint deterministic calculation is then ran and the results are used as inputs for the weight-window generator. Finally, the Monte Carlo simulation is performed, making use of the automatically generated weight windows. The CADIS methodology is now successfully used in the SCALE/MAVRIC code system (Oak Ridge National Laboratory, 2009).

A different approach towards an automated weight-window generator has been

demonstrated by Shahdatullah et al. (2006). The authors use a weight-window generator based on the finite elements method in order to process the output of an adjoint deterministic calculation.

2.5.3 Local Importance Function Transform

Turner and Larsen (1997a) have developed a method utilizing the local importance function transform. The method is based on the exponential transform technique (see section 2.3.6). The exact adjoint solution is approximated by a piecewise-continuous function containing parameters that are obtained from a deterministic adjoint calculation, and the transport and collision processes of the transformed Monte Carlo problem bias the source distribution, distance to collision, and selection of post-collision energy groups and directions. The method claims to be especially effective in shielding systems, surpassing the efficiency of the implicit capture scheme by a factor of 10^6 (Turner and Larsen, 1997b). However it is limited to multigroup energy configurations.

2.5.4 Variational Variance Reduction

Barrett and Larsen (2001) have introduced a variance reduction method which is based on the variational method (Becker, 1964). The *Variational Variance Reduction* (VVR) method employs a variationally-motivated functional that processes global, "low-quality" forward and adjoint flux estimates into "high-quality" estimates, such as the transmitted current through a shield or an eigenvalue. The VVR functionals are more costly to evaluate than standard functionals used to obtain Monte Carlo estimates, but they are more accurate. VVR can be used in combination with other methods, such as the AVATAR method discussed earlier.

Another important feature of the VVR method is that every Monte Carlo particle influences the estimate, even ones that do not reach the detector. This is because every Monte Carlo particle contributes to the "global" variationally-derived terms of the functional. The global character of the method allows it to improve the variance in cases where other schemes fail, such as shielding systems with large optical thickness or eigenvalue calculations.

2.5.5 Other methods

There are several other methods worth mentioning when discussing variance reduction with adjoint functions.

One such method is the adjoint Monte Carlo method (Kalos, 1968), in which the so-called adjoint particles start at the detector region and gain energy as they travel towards the source where they are detected. In this respect, we can visualise those particles as neutrons travelling backwards in phase-space and the real source and real detector as the adjoint detector and adjoint source, respectively. At the adjoint detector, an adjoint function is obtained with which numerically the same result can be derived as with forward Monte Carlo.

The advantage of this technique is that in many cases the source of the system is very large compared to the detector. Consequently, in the adjoint calculation, many more particles will arrive at the adjoint detector (i.e. the forward source) than in the forward calculation, hence decreasing the variance without significant additional cost in CPU time.

Another method called the *Midway method*, proposed by Serov et al. (1999), combines forward and adjoint Monte Carlo simulations. The responses in both simulations are calculated on a surface enclosing either the source or the detector (but not both) and the results are coupled in order to obtain the final estimate. The advantage of the method is that the volume encompassed by the Midway surface can be significantly larger than both the source and detector volumes, therefore improving on both the forward and adjoint methods individually. However, matching the responses is a non-trivial process.

2.6 Shortcomings regarding criticality

All of the variance reduction methods discussed before have been developed with shielding calculations in mind. The reason, as we discussed in Chapter 1, is that the Monte Carlo technique works better when a limited amount of information is sought. Variance reduction techniques further enforce this notion by biasing particles towards limited regions of higher importance, such as detectors. However, criticality problems are by nature problems where global information is required, since the fission source of a new particle generation needs to be constructed throughout the system, rather than detected at a remote point. In contrast, since deterministic

techniques solve the transport equation everywhere in the system, they do not differentiate between criticality and source-detector systems in that respect.

In addition, using an adjoint-based variance reduction method for criticality problems brings some conceptual problems. As we discussed in Sec. 2.4, the adjoint function can be interpreted as the expected contribution of a particle to the final estimate. However, in a criticality calculation, we are estimating ratios of successive generations. Therefore, it is much harder for us to determine the importance of a particle, since it affects future generations, rather than only contributing to the estimate of the current one.

Still, there are some ways towards solving the issue. As an example, techniques which use global information in order to estimate responses, such as the correcton method and techniques based on the variational method, can be used in an eigenvalue simulation in the same manner as in a source-detector simulation, and can therefore also be effective in eigenvalue problems. Also, techniques such as implicit capture, which allow longer total particle trajectories, can still be useful, since their effect in the simulation is global, rather than localized to a scoring region. However, particle weights do not change as much in criticality calculations as in shielding ones, therefore such techniques do not provide significant reduction in variance.

In this thesis, we present a radical approach to variance reduction in eigenvalue simulations. As we will see in Chapter 3, we can formulate an eigenvalue problem as a virtual source-detector problem, which allows the use of adjoint-based variance reduction. The advantage of this technique is that it can be combined with most, if not all, of the above techniques in order to gain an appreciable reduction in the variance.

2.7 Bibliography

C. L. Barrett and E. W. Larsen. Variationally-based variance reduction method for Monte Carlo neutron transport calculations. *Annals of Nuclear Energy*, **28**, 457, 2001.

M. Becker. *The Principles and Applications of Variational Methods*. M.I.T. Press, Cambridge, MA, 1964.

T. L. Becker, A. B. Wollaber and E. W. Larsen. A hybrid monte carlo-deterministic

method for global particle transport calculations. *Nuclear Science and Engineering*, **155**, 155, 2007.

J. Bernoulli. *Ars conjectandi, opus posthumum: acc. Tractatus de seriebus infinitis,et epistola Gallice scripta de ludo pilae reticularis*. Impensis Thurnisiorum, Basileae, 1713.

T. E. Booth. Automatic importance estimation in Monte Carlo calculations. In *Transactions of the American Nuclear Society*, volume 41, 308–309, 1982.

T. E. Booth and J. S. Hendricks. Importance estimation in Monte Carlo calculations. *Nuclear technology/fusion*, **5**, 90, 1984.

E. D. Cashwell and C. J. Everett. *A Practical Manual on the Monte Carlo Method for Random Walk Problems*. Pergamon Press, 1959.

F. H. Clark. The exponential transform as an importance sampling device - a review. Technical Report ORNL-RSIC-14, Oak Ridge National Laboratory, 1966.

R. R. Coveyou, V. R. Cain and K. J. Yost. Adjoint and importance in Monte Carlo application. *Nuclear Science and Engineering*, **27**, 219, 1967.

P. W. Glynn and W. Whitt. The asymptotic efficiency of simulation estimators. *Operations Research*, **40**, 505, 1992.

A. Haghighat and J. C. Wagner. Monte Carlo variance reduction with deterministic importance functions. *Progress in Nuclear Energy*, **42**, 25, 2003.

J. E. Hoogenboom. Optimum biasing of integral equations in Monte Carlo calculations. *Nuclear Science and Engineering*, **70**, 210, 1979.

M. H. Kalos. Monte Carlo integration of the adjoint gamma-ray transport equation. *Nuclear Science and Engineering*, **33**, 284, 1968.

P. S. Laplace. *Théorie analytique des probabilités*. V. Courcier, Paris, 1812.

J. Lewins. *Importance, The Adjoint Function: The Physical Basis of Variational and Perturbation Theory in Transport and Diffusion Problems*. Pergamon Press, 1965.

I. Lux and L. Koblinger. *Monte Carlo particle transport methods: neutron and photon calculations*. CRC Press, Boca Raton, 1991.

M. W. Mickael. Fast, automated, semideterministic weight windows generator for MCNP. *Nuclear Science and Engineering*, **119**, 34, 1995.

P. C. Miller et al. The use of an inbuilt importance generator for acceleration of the Monte Carlo code MCBEND. In *International Conference on the Physics of Reactors*, 124–132, 1990.

J. Neyman. On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, **97**, 558, 1934.

K. Noack. On the relation between adjoint and importance in Monte Carlo solutions of linear particle transport problems, part I. *Kernernergie*, **22**, 346, 1979.

Oak Ridge National Laboratory. SCALE, a modular code system for performing standardized computer analyses for licensing evaluations. Technical Report ORNL-TM-2005/39, Version 6, Vols. I-III, Oak Ridge National Laboratory, 2009.

I. V. Serov, T. M. John and J. E. Hoogenboom. Midway forward-adjoint coupling method for neutron and photon Monte Carlo transport. *Nuclear Science and Engineering*, **133**, 55, 1999.

M. S. Shahdatullah et al. Monte Carlo variance reduction using finite element adjoint weight windows. In *PHYSOR-2006 - American Nuclear Society's Topical Meeting on Reactor Physics*, 2006.

J. Spanier and E. M. Gelbard. *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley Publishing Co., 1969.

J. S. Tang. *Methods of Monte Carlo biasing using two-dimensional discrete ordinates adjoint flux*. Ph.D. thesis, University of Tennessee, 1976.

S. A. Turner and E. W. Larsen. Automatic variance reduction for three-dimensional Monte Carlo simulations by the local importance function transform - I: Analysis. *Nuclear Science and Engineering*, **127**, 22, 1997a.

S. A. Turner and E. W. Larsen. Automatic variance reduction for three-dimensional Monte Carlo simulations by the local importance function transform - II: Numerical results. *Nuclear Science and Engineering*, **127**, 36, 1997b.

K. A. Van Riper. AVATAR - Automatic variance reduction in Monte Carlo calculations. In *Proc. Joint Int. Conf. Mathematical Methods and Supercomputing for Nuclear Applications*, volume 1, 661–670, 1997.

J. C. Wagner. *Acceleration of Monte Carlo Shielding Calculations with an Automated Variance Reduction Technique and Parallel Processing.* Ph.D. thesis, The Pennsylvania State University, 1997.

CHAPTER 3

ZERO VARIANCE AND CRITICALITY

A natural target in the quest for variance reduction is a scheme that estimates a response with zero variance, i.e. the true answer to our problem. Although a Monte Carlo scheme with zero-variance sounds like an oxymoron, such schemes can indeed be devised, but they are not very practical for day-to-day simulations. In this chapter, we will devise a zero-variance scheme for criticality calculations and discuss how it can form the basis for a variance reduction technique.

3.1 The zero-variance scheme

The idea of a zero-variance scheme is not new in Monte Carlo. Already back in 1949, both Goertzel (1949) and Kahn (1949) were discussing a scheme, in which appropriate biasing of the transition and collision kernels with an appropriate importance function could lead to a result with zero variance. Almost 20 years later, Coveyou et al. (1967) showed the significance of the adjoint function as importance function. Zero-variance schemes for various estimators using the adjoint function, starting with the last-event estimator, were later introduced by Hoogenboom (1979) and Noack (1979).

Dwivedi (1982) and Gupta (1983) tried to generate a universal zero-variance scheme that applies to more estimators. However, Hoogenboom (2008b) has re-

cently shown that only one specific zero-variance scheme exists for a given estimator.

A different approach towards a solution with zero variance was shown by Booth (1989). Instead of biasing the transport kernels, he selected random numbers from a biased phase space. In addition, he explored the zero-variance solution for multiple correlated tallies (Booth, 1998).

Lux and Koblinger (1991) briefly touched upon the subject of a zero-variance scheme, via the use of the *moments equations*, similar to the earlier authors. A more readable approach, as well as a full literature review on the subject is presented by Hoogenboom (2008b), where the reader is directed to for more background information.

3.1.1 Theory

As we discussed in Chapter 2, an appropriate biasing function is the adjoint function $\phi^*(P)$. However, since we are using the emission and collision densities, it is the adjoint form of those that we should be using for biasing the transport of particles.

Following the lifetime of a particle, we start with its generation at the source. Following the zero-variance scheme, particles should be selected by a distribution biased by the adjoint emission density $\chi^*(P)$, rather than the true source $S(P)$. Therefore, the biased source function is

$$\bar{S}(\boldsymbol{r}, E, \boldsymbol{\Omega}) = S(\boldsymbol{r}, E, \boldsymbol{\Omega}) \frac{\chi^*(\boldsymbol{r}, E, \boldsymbol{\Omega})}{R}, \tag{3.1}$$

where the expected detector response R, given by Eq. (2.28), is introduced for normalization. The simulation weight of the particles should be appropriately changed by

$$W_{\bar{S}} = \frac{S(P)}{\bar{S}(P)} = \frac{R}{\chi^*(\boldsymbol{r}, E, \boldsymbol{\Omega})}, \tag{3.2}$$

in order to ensure an unbiased estimate.

In order to bias the transport kernel $K(P' \to P)$, it is simpler and more natural to split the process in two. Initially, we bias the transition kernel $T(\boldsymbol{r'} \to \boldsymbol{r}, E', \boldsymbol{\Omega'})$ by the adjoint collision density $\psi^*(\boldsymbol{r}, E', \boldsymbol{\Omega'})$, since it represents the importance of particles entering a collision at $(\boldsymbol{r}, E', \boldsymbol{\Omega'})$. Following, we bias the collision kernel

$C(\boldsymbol{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega})$ by the adjoint emission density $\chi^*(\boldsymbol{r}, E, \boldsymbol{\Omega})$, which can be derived using Eqs (2.26) and (2.30):

$$\chi^*(\boldsymbol{r}, E, \boldsymbol{\Omega}) = \int T(\boldsymbol{r} \to \boldsymbol{r'}, E, \boldsymbol{\Omega}) \psi^*(\boldsymbol{r'}, E, \boldsymbol{\Omega})\, dV'. \tag{3.3}$$

The biased transition kernel is then formulated as follows:

$$\overline{T}(\boldsymbol{r'} \to \boldsymbol{r}, E', \boldsymbol{\Omega}') = T(\boldsymbol{r'} \to \boldsymbol{r}, E', \boldsymbol{\Omega}') \frac{\psi^*(\boldsymbol{r}, E', \boldsymbol{\Omega}')}{\chi^*(\boldsymbol{r'}, E', \boldsymbol{\Omega}')}. \tag{3.4}$$

Again, a weight correction of

$$W_{\overline{T}} = \frac{T(\boldsymbol{r'} \to \boldsymbol{r}, E', \boldsymbol{\Omega}')}{\overline{T}(\boldsymbol{r'} \to \boldsymbol{r}, E', \boldsymbol{\Omega}')} = \frac{\chi^*(\boldsymbol{r'}, E', \boldsymbol{\Omega}')}{\psi^*(\boldsymbol{r}, E', \boldsymbol{\Omega}')} \tag{3.5}$$

is required. Similarly, for the biased collision kernel:

$$\overline{C}(\boldsymbol{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}) = C(\boldsymbol{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}) \frac{\chi^*(\boldsymbol{r}, E, \boldsymbol{\Omega})}{\psi^*(\boldsymbol{r}, E', \boldsymbol{\Omega}') - \eta_\psi(\boldsymbol{r}, E')}. \tag{3.6}$$

The normalization is obtained from the relation between χ^* and ψ^*:

$$\psi^*(\boldsymbol{r}, E, \boldsymbol{\Omega}) = \eta_\psi(\boldsymbol{r}, E) + \int C(\boldsymbol{r}, E \to E', \boldsymbol{\Omega} \to \boldsymbol{\Omega}') \chi^*(\boldsymbol{r}, E', \boldsymbol{\Omega}')\, dE'\, d\Omega' \tag{3.7}$$

Therefore, the weight correction that needs to be applied for an unbiased estimate of R is:

$$W_{\overline{C}} = \frac{C(\boldsymbol{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega})}{\overline{C}(\boldsymbol{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega})} = \frac{\psi^*(\boldsymbol{r}, E', \boldsymbol{\Omega}') - \eta_\psi(\boldsymbol{r}, E')}{\chi^*(\boldsymbol{r}, E, \boldsymbol{\Omega})}. \tag{3.8}$$

As has been proven for source-detector studiesHoogenboom (2008b) and as we will show here for criticality studies, if we bias the kernels by the adjoint functions as above, we reach a result with zero variance. However, it is obvious that the zero-variance scheme is mostly of theoretical value. Indeed, in practice, if one wishes to obtain the adjoint functions χ^* and ψ^* in order to bias the forward solution, one must solve the adjoint problem, thus making the forward calculation redundant. However, the scheme can still lead to a decrease in variance (never to zero, of course) when approximate adjoint functions are used. That way, obtaining a computationally cheap estimate of the adjoint function could help reduce the variance of the expensive forward Monte Carlo simulation.

We will now show that the scheme leads to zero variance also in a criticality case, first theoretically, then in practice, using a simplified transport model in a homogeneous slab reactor.

3.2 Proof of the zero-variance scheme for criticality

In a multiplying system, we can interpret the physical meaning of the eigenvalue k_{eff} as the ratio of source neutrons of two successive neutron generations:

$$k_{eff} = \frac{\int S_{n+1}(P)\,dP}{\int S_n(P)\,dP}. \tag{3.9}$$

The new fission source $S_{n+1}(P)$ is calculated from the current generation n by

$$S_{n+1}(P) = \frac{1}{4\pi}\chi_f(E)\int \frac{\nu\Sigma_f(\boldsymbol{r},E')}{\Sigma_t(\boldsymbol{r},E')}\psi_n(\boldsymbol{r},E',\boldsymbol{\Omega}')\,dE'\,d\boldsymbol{\Omega}', \tag{3.10}$$

where $\chi_f(E)$ is the fission energy spectrum, ν is the expected number of neutrons produced by one fission and Σ_f is the fission cross-section. Using this formulation, for a certain generation we in fact treat a criticality simulation as a source-detector problem, the detector response function of which we will see below.

For generation n we solve the equation

$$\chi_n(P) = S_n(P) + \int K(P'\to P)\chi_n(P')\,dP', \tag{3.11}$$

with fission treated as absorption when sampling the collision kernel C. In order to sample the kernel K, the transition kernel is sampled first, followed by the collision kernel. Starting with a normalised source $S_n(P)$, we can regard k_{eff} as the detector response

$$R = k_{eff} = \int S_{n+1}(P)\,dP = \int \frac{\nu\Sigma_f(P)}{\Sigma_t(P)}\psi_n(P)\,dP \tag{3.12}$$

if the source distribution is converged, so for sufficiently large n. The equation for the collision density of the n-th generation is

$$\psi_n(P) = S_{1,n}(P) + \int L(P'\to P)\psi_n(P')\,dP' \tag{3.13}$$

with $S_{1,n}$ the source of first collisions for the n-th generation.

Now, if we follow the standard method of deriving the adjoint equation, the source term in the equation adjoint to Eqs. (3.12) and (3.13) for ψ^* becomes

$$\eta_\psi(P) = \frac{\nu\Sigma_f(P)}{\Sigma_t(P)} \tag{3.14}$$

and the adjoint function we are looking for is given by

$$\psi^*(P) = \frac{\nu\Sigma_f(P)}{\Sigma_t(P)} + \int L(P \to P')\psi^*(P')\,dP', \tag{3.15}$$

which is an adjoint source-detector problem. Note that the adjoint function from Eq. (3.15) is different from the adjoint eigenfunction of the criticality problem and that Eq. (3.15) is independent of n.

Since we are concerned with a k_{eff} problem, the contribution of the particle to the score at each fission event is the particle's fission weight at that time, namely

$$w_f(P) = w\ \eta_\psi(P) \tag{3.16}$$

In our case, since we use adjoint functions in order to bias the source and transport kernels, we apply weight factors to the particle after each interaction in order to keep the result unbiased. With P_k the coordinates of a particle entering the k-th collision the contribution to the estimator from a particle's history will now be

$$\widehat{R}_{col} = \sum_{k=1}^{\infty} w(P_0, P_1, P_2, \ldots, P_k)\eta_\psi(P_k). \tag{3.17}$$

We can try to construct the scoring sequence of a particle, keeping in mind that the particle will score every time after a transition and before scattering takes place. The neutron weight at the source event is given by W_S according to Eq. (3.2). Between two successive events for starting a flight path at (r_{k-1}, E_k, Ω_k) and at $(r_k, E_{k+1}, \Omega_{k+1})$, the neutron weight is multiplied by $W_{\overline{T}}W_{\overline{C}}$ according to Eqs. (3.5) and (3.8).

$$W_{\overline{L}}(r_{k-1}, P_k, E_{k+1}, \Omega_{k+1}) = W_{\overline{T}}W_{\overline{C}} = \frac{\chi^*(r_{k-1}, E_k, \Omega_k)}{\chi^*(r_k, E_{k+1}, \Omega_{k+1})}\frac{\psi^*(P_k) - \eta_\psi(P_k)}{\psi^*(P_k)} \tag{3.18}$$

Before scoring at the next collision site another weight factor $W_{\overline{T}}$ is applied for selecting the path length. To simplify the equation we introduce the quantity

$$\overline{\kappa}(P) = \frac{\psi^*(P) - \eta_\psi(P)}{\psi^*(P)}. \tag{3.19}$$

$\overline{\kappa}$ can be considered as the equivalent of the non-absorption probability for biased neutron transport with the transport kernel $\overline{L}$

$$\overline{L}(P' \to P) = L(P' \to P)\frac{\psi^*(P)}{\psi^*(P')} \tag{3.20}$$

as

$$\overline{\kappa}(P) = \int \overline{L}(P \to P')\, dP' = \frac{1}{\psi^*(P)} \int L(P \to P')\psi^*(P')\, dP' = 1 - \frac{\eta_\psi(P)}{\psi^*(P)}. \tag{3.21}$$

Now the neutron weight when entering the k-th collision becomes

$$\begin{aligned} w(P_1, P_2, \ldots, P_k) &= \frac{R}{\chi^*(r_0, E_1, \Omega_1)} W_{\overline{L}}(r_0, P_1, E_2, \Omega_2) \times \ldots \\ &\quad \ldots \times W_{\overline{L}}(r_{k-2}, P_{k-1}, E_k, \Omega_k) W_{\overline{T}}(r_{k-1}, P_k) \\ &= R \prod_{i=1}^{k-1} \kappa(P_i) \frac{1}{\psi^*(P_k)} \end{aligned} \tag{3.22}$$

and the score over a neutron history becomes

$$\widehat{R}_{col} = \sum_{k=1}^{\infty} R \prod_{i=1}^{k-1} \kappa(P_i) \frac{\eta_\psi(P_k)}{\psi^*(P_k)} = R \left\{ \frac{\eta_\psi(P_1)}{\psi^*(P_1)} + \kappa(P_1) \frac{\eta_\psi(P_2)}{\psi^*(P_2)} + \ldots \right\}. \tag{3.23}$$

Using Eq. (3.21) we can calculate $1 - \widehat{R}_{col}/R$, to get:

$$\begin{aligned} 1 - \frac{\widehat{R}_{col}}{R} &= \overline{\kappa}(P_1) - \overline{\kappa}(P_1)\left[1 - \overline{\kappa}(P_2)\right] - \overline{\kappa}(P_1)\overline{\kappa}(P_2)\left[1 - \overline{\kappa}(P_3)\right] - \ldots \\ &= \overline{\kappa}(P_1) - \overline{\kappa}(P_1) + \overline{\kappa}(P_1)\overline{\kappa}(P_2) - \overline{\kappa}(P_1)\overline{\kappa}(P_2) + \overline{\kappa}(P_1)\overline{\kappa}(P_2)\overline{\kappa}(P_3) - \ldots \\ &= \overline{\kappa}(P_1)\overline{\kappa}(P_2)\overline{\kappa}(P_3)\ldots \\ &= 0, \end{aligned} \tag{3.24}$$

since $\overline{\kappa}(P_i) < 1$ for an infinite number of collision points i. Therefore, at all histories, $\widehat{R}_{col} = R$, and the variance will be zero, despite each history being different from the others. The scheme requires the use of a collision estimator for scoring, which means that it would not be valid in the special case of a point detector.

3.3 Demonstration of the scheme

In order to demonstrate the scheme, there are some simplifications we need to make. Since we need to use the converged and normalised source distribution, as well as the exact adjoint functions, we need to be able to calculate the analytical solutions for the above parameters, in order to reach zero variance. To this end, we have used the two-direction model, thoroughly discussed by Hoogenboom (2008a). Since the two-direction model uses several simplifications, it is therefore necessary to initially show the use of adjoint functions for biasing in a multigroup, discrete direction scheme, rather than the general continuous energy and direction Monte Carlo scheme.

3.3.1 Biasing with the adjoint functions

In our scheme, when using a multigroup treatment, the energy of neutrons generated by fission at a point $P(\mathbf{r}, g, \mathbf{\Omega})$ is selected from the fission spectrum χ_f biased by the direction averaged adjoint function $\chi^*(\mathbf{r}, g)$. In general terms:

$$\overline{S}(P) = S(P)\frac{\chi^*(P)}{\int S(P)\chi^*(P)dP}. \tag{3.25}$$

Since we want to have an unbiased estimate for k_{eff}, we need to weigh the particles appropriately. As we have seen earlier, the correction weight is simply the ratio of the initial and modified pdfs, and now becomes

$$W_{\overline{S}} = \frac{S(P)}{\overline{S}(P)} = \frac{\int S(P)\chi^*(P)dP}{\chi^*(P)}. \tag{3.26}$$

The particle direction is then selected from the isotropic distribution biased by the directional adjoint function, in a way similar to collision biasing, shown later. As stated earlier, after biasing we need to alter the statistical weight of the particle. In this case, it needs to be set inversely proportional to the adjoint function for the selected energy group and direction. This requires appropriate normalisation of the weights of all particles in a new batch. In practice, when using a multigroup

treatment, the initial weight is set equal to

$$w_{\text{init}} = \frac{\sum_{g''} \chi_f(g'')\chi^*(\mathbf{r}, g'')}{\chi^*(\mathbf{r}, g, \mathbf{\Omega})} \tag{3.27}$$

For biasing the transition kernel to select a new collision site, the scheme requires the adjoint function ψ^* for particles entering a collision at P as the biasing function. The new collision site is therefore selected from the biased transition kernel

$$\overline{T}(\mathbf{r}' \to \mathbf{r}, g, \mathbf{\Omega}) = \frac{T(\mathbf{r}' \to \mathbf{r}, g, \mathbf{\Omega})\psi^*(\mathbf{r}, g, \mathbf{\Omega})}{\int T(\mathbf{r}' \to \mathbf{r}'', g, \mathbf{\Omega})\psi^*(\mathbf{r}'', g, \mathbf{\Omega}) dV''} \tag{3.28}$$

In practice, the new path length needs to be selected from the normalised probability

$$\xi = \frac{\int_0^s T(\mathbf{r} \to \mathbf{r} + s'\mathbf{\Omega}, g, \mathbf{\Omega})\psi^*(\mathbf{r} + s'\mathbf{\Omega}, g, \mathbf{\Omega}) ds'}{\int_0^{s_{max}} T(\mathbf{r} \to \mathbf{r} + s'\mathbf{\Omega}, g, \mathbf{\Omega})\psi^*(\mathbf{r} + s'\mathbf{\Omega}, g, \mathbf{\Omega}) ds'}, \tag{3.29}$$

where ξ is a random number, uniformly distributed between 0 and 1, $\mathbf{\Omega}$ is the direction and g is the current energy group of the particle. After selecting ξ, we can now calculate the value of $\mathbf{r}$ using a root-finding iterative method until $\mathbf{r}$ converges. A weight factor is then applied, in order to keep the final result unbiased:

$$W_{\overline{T}} = \frac{\int_0^{s_{max}} T(\mathbf{r} \to \mathbf{r} + s'\mathbf{\Omega}, g, \mathbf{\Omega})\psi^*(\mathbf{r} + s'\mathbf{\Omega}, g, \mathbf{\Omega}) ds'}{\psi^*(\mathbf{r}, g, \mathbf{\Omega})} \tag{3.30}$$

In order to apply the biasing scheme to the collision kernel, we have to bias it by $\chi^*(P)$. The biasing is done in two steps - initially, we select the new energy group g after the collision with probability

$$\overline{p}(g|\mathbf{r}, g') = \frac{\Sigma_s(\mathbf{r}, g' \to g)\chi^*(\mathbf{r}, g)}{\sum_{g''} \Sigma_s(\mathbf{r}, g' \to g'')\chi^*(\mathbf{r}, g'')}. \tag{3.31}$$

After that, we need to select an outgoing direction, with probability

$$\overline{p}(\mathbf{\Omega}|\mathbf{r},g',g)=\frac{\Sigma_s(\mathbf{r},\mathbf{\Omega}'\rightarrow\mathbf{\Omega}|g',g)\chi^*(\mathbf{r},g,\mathbf{\Omega})}{\int\Sigma_s(\mathbf{r},\mathbf{\Omega}'\rightarrow\mathbf{\Omega}''|g',g)\chi^*(\mathbf{r},g,\mathbf{\Omega}'')d\mathbf{\Omega}''}\,. \tag{3.32}$$

The (combined) weight factor to be applied for the scattering biasing will then be

$$W_{\overline{C}}=\frac{\sum\limits_{g''}\Sigma_s(\mathbf{r},g'\rightarrow g'',\mathbf{\Omega}'\rightarrow\mathbf{\Omega})\chi^*(\mathbf{r},g'',\mathbf{\Omega})}{\chi^*(\mathbf{r},g,\mathbf{\Omega})}\,. \tag{3.33}$$

3.3.2 The two-direction model

The two-direction model is a model that limits the transport of particles to the $\pm x$ direction. That way, although it remains a true transport model, the equations describing the particle transport become differential, diffusion type equations, but still suitable for Monte Carlo simulation. This means that, for simple geometries, both forward and adjoint solutions can be obtained analytically, which makes the model an excellent, although strongly simplified one for testing theories for variance reduction.

By limiting our transport to the $\pm x$ directions, the transition and collision kernels become extremely simple, especially if we assume a mono-energetic case. In that case, the distance between two interaction points x' and x becomes $s=|x-x'|$, while for the collision kernel all we need to select is a direction cosine $\mu=\pm1$.

Let us first derive our detector response $R=k_{eff}$ analytically, using the two-direction model. From Eq. (3.12), we have

$$R=\int\nu\Sigma_f(x)\phi(x)dx\,. \tag{3.34}$$

We can calculate $\phi(x)$ by solving the eigenvalue differential equation for the two-direction modelHoogenboom (2008a):

$$\frac{1}{\Sigma_{tr}}\frac{d^2\phi(x)}{dx^2}-\Sigma_a\phi(x)+\frac{1}{k_{eff}}\nu\Sigma_f(x)\phi(x)=0\,, \tag{3.35}$$

where the transport cross-section Σ_{tr} is defined as

$$\Sigma_{tr}\equiv\Sigma_t-\overline{\mu}_0\Sigma_s\,. \tag{3.36}$$

and $\overline{\mu}_0$ the averaged cosine of the scattering angle. The solution of the eigenvalue equation in a homogeneous slab system of half-width a is proportional to $\cos(Bx)$ with the square root of the geometrical buckling B determined by the boundary conditionHoogenboom (2008a)

$$\left|\frac{d\phi(x)}{dx}\right|_{\pm a} = \Sigma_{tr}\phi(\pm a) \tag{3.37}$$

leading to the condition for B

$$B\tan(Ba) = \Sigma_{tr} \tag{3.38}$$

The normalised fission source distribution is

$$S(x) = \frac{B\cos(Bx)}{2\sin(Ba)}, \qquad |x| \le a\,. \tag{3.39}$$

and, using Eq. (3.35) and (3.39), the corresponding neutron flux $\phi(x)$ is equal to:

$$\phi(x) = \frac{B\Sigma_{tr}\cos(Bx)}{2\sin(Ba)(B^2+\Sigma_a\Sigma_{tr})}\,, \tag{3.40}$$

which, using Eq. (3.34), gives us the final theoretical response:

$$k_{th} = R = \frac{\nu\Sigma_f\Sigma_{tr}}{B^2+\Sigma_a\Sigma_{tr}}\,. \tag{3.41}$$

In practice, for one cycle, k_{eff} is calculated as follows:

$$k_{cycle} = \sum_{n=1}^{N} R_n\,, \tag{3.42}$$

where N is the number of particles followed in the cycle, and R_n is the contribution of a particle to the detector response:

$$R_n = \sum_{\text{collisions}} w\frac{\nu\Sigma_f}{\Sigma_t}\,, \tag{3.43}$$

where w is the statistical weight of the particle at the collision.

In order to simulate this process by Monte Carlo, we need to sample the kernels. We can write the two kernels (for a homogeneous medium) as follows:
The transition kernel will be

$$T_{\pm}(x' \to x) = \Sigma_t e^{-\Sigma_t|x-x'|}\,, \tag{3.44}$$

while the collision kernel becomes equal to

$$C(x,\mu' \to \mu) = \begin{cases} \frac{\Sigma_{\rightarrow}}{\Sigma_t}, & \mu'\mu = +1 \\ \frac{\Sigma_{\leftarrow}}{\Sigma_t}, & \mu'\mu = -1 \end{cases} \tag{3.45}$$

where $\Sigma_{\rightarrow}$ and $\Sigma_{\leftarrow}$ are the forward and backward scattering cross section, respectively, with

$$\Sigma_{\rightarrow} + \Sigma_{\leftarrow} = \Sigma_s \tag{3.46}$$

$$\Sigma_{\rightarrow} - \Sigma_{\leftarrow} = \overline{\mu}_0 \Sigma_s \,, \tag{3.47}$$

Hence, $\overline{\mu}_0$ follows from

$$\overline{\mu}_0 = \frac{\Sigma_{\rightarrow} - \Sigma_{\leftarrow}}{\Sigma_{\rightarrow} + \Sigma_{\leftarrow}} \,. \tag{3.48}$$

In the case of isotropic scattering, $\overline{\mu}_0 = 0$ and therefore

$$\Sigma_{\rightarrow} = \Sigma_{\leftarrow} = \frac{1}{2}\Sigma_s \,. \tag{3.49}$$

From the equations above, it is clear that the model simplifies sampling of the transition and collision kernels significantly. In addition, the scheme can still be applied to heterogeneous systems, as well as with multigroup energy treatment, although the analytical solutions become more involved. However, if we want to implement the zero-variance scheme in the model, the most important issue is obtaining the adjoint functions.

3.3.3 Adjoint equations in the two-direction model

For a homogeneous system, the differential equation for $\chi^*(x) = \frac{1}{2}\left[\chi^*_+(x) + \chi^*_-(x)\right]$ has the same form as the one for the adjoint function $\phi^*(x)$Hoogenboom (2008a):

$$-\frac{1}{\Sigma_{tr}} \frac{d^2\chi^*(x)}{dx^2} + \Sigma_a \chi^*(x) = \eta_\phi \,. \tag{3.50}$$

Solving the equation for a 1-D slab reactor with half-width a in the two-direction model, using the boundary conditionHoogenboom (2008a)

$$\frac{1}{\Sigma_{tr}} \left| \frac{d\chi^*(x)}{dx} \right|_a = \chi^*(a), \tag{3.51}$$

gives the analytical solution for $\chi^*(x)$:

$$\chi^*(x) = \frac{\nu\Sigma_f}{\Sigma_a}\left[1 - \frac{\Sigma_{tr}\cosh\sqrt{\Sigma_a\Sigma_{tr}}x}{\Sigma_{tr}\cosh\sqrt{\Sigma_a\Sigma_{tr}}a + \sqrt{\Sigma_a\Sigma_{tr}}\sinh\sqrt{\Sigma_a\Sigma_{tr}}a}\right]. \tag{3.52}$$

The directional adjoint $\chi^*_\pm(x)$ is then equal to Hoogenboom (2008a)

$$\chi^*_\pm(x) = \chi^*(x) \pm \frac{1}{\Sigma_{tr}}\frac{d\chi^*(x)}{dx}. \tag{3.53}$$

$\psi^*(x)$ can be obtained by $\chi^*(x)$:

$$\psi^*(x) = \eta_\psi + \frac{\Sigma_s}{\Sigma_t}\chi^*(x), \tag{3.54}$$

while $\psi^*_\pm(x)$ can be obtained from $\chi^*_\pm(x)$:

$$\psi^*_\pm(x) = \eta_\psi + \frac{\Sigma_\rightarrow}{\Sigma_t}\chi^*_\pm(x) + \frac{\Sigma_\leftarrow}{\Sigma_t}\chi^*_\mp(x). \tag{3.55}$$

Using the above equations, $\chi^*(x)$ was calculated for a homogeneous 1-D slab reactor of half-width $a = 10$ cm. The following set of one-group cross-sections was used: $\Sigma_t = 1.1$ cm^{-1}, $\Sigma_s = 0.6$ cm^{-1} and $\nu\Sigma_f = 0.625$ cm^{-1}, while isotropic scattering ($\overline{\mu}_0 = 0$) was assumed. The resulting $\chi^*(x)$ can be seen in Fig. 3.1.

As $\chi^*_\pm(x)$, the adjoint emission density for the $\pm x$ direction, becomes zero at the right and left boundary, respectively, the probability of particles to escape the system when using the biased transition kernel $\overline{T}$ according to Eq. (3.4) goes to zero. In the centre of the slab, the importance for both directions is exactly the same (since our system is symmetric around $x = 0$), and therefore the adjoint emission density is flat. Setting $\frac{d\chi^*(0)}{dx} = 0$ in Eq. (3.53) results in $\chi^*_-(0) = \chi^*_+(0) = \chi^*(0)$. Note that if our system was infinite, $\chi^*_\pm(0)$ would be equal to $k_\infty = \frac{\nu\Sigma_f}{\Sigma_t}$, since it is the expected contribution of a particle starting a flight path at $x = 0$. In our finite slab, the value of $\chi^*_\pm(0)$ is slightly lower, since there is still a small but finite leakage probability at $x = 0$. When the particle approaches the boundary, the leakage probability increases and therefore $\chi^*_+(x)$ or $\chi^*_-(x)$ decreases accordingly.

3.3.4 Sampling the biased kernels

In order to obtain a k_{eff} response using the zero-variance scheme, we need to properly bias the source, transition and collision kernels and then sample from

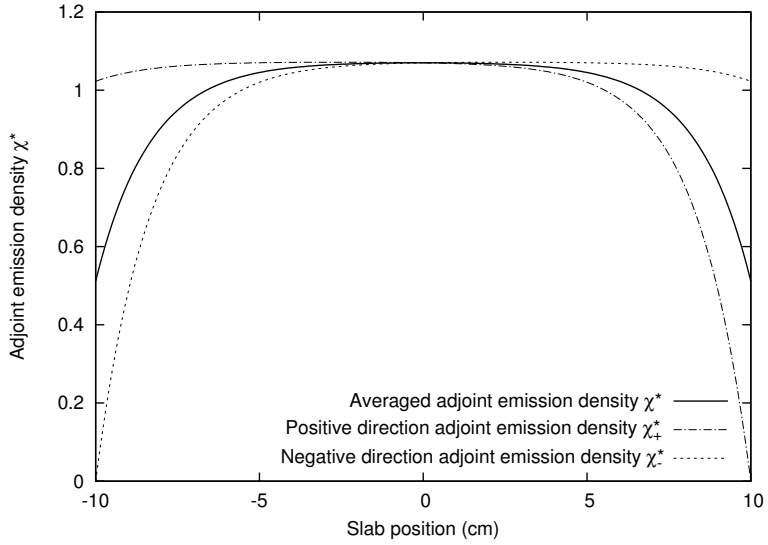

Figure 3.1: Adjoint emission density $\chi^*(x)$ for a slab reactor.

the biased kernels. By applying the fact that the optimum weighting factor is the adjoint emission density, we have already seen in Eq. (3.1) that the optimum source biasing is equal to

$$\overline{S}_\mu(x) = S_\mu(x) \times \frac{\chi^*_\mu(x)}{R} = \frac{1}{2} S(x) \frac{\chi^*_\mu(x)}{R}, \quad \mu = \pm 1 \tag{3.56}$$

with the factor R included to have a normalised biased source.

Using Eq. (3.39), (3.41), (3.52) and (3.53), we have an analytical form for $\overline{S}_\mu(x)$. The starting position of the neutron is selected from $\overline{S}(x)$:

$$\overline{S}(x) = \sum_\mu \overline{S}_\mu(x) = S(x) \frac{\chi^*(x)}{R} \tag{3.57}$$

which can be sampled using a root-finding method. Once the position is selected, the direction is sampled with probability

$$\overline{p}_\mu = \frac{\chi^*_\mu(x)}{\chi^*(x)} \tag{3.58}$$

The weight of the particle then needs to be altered in order to account for the biasing. The new weight is equal to

$$w_{\overline{S}} = \frac{S_\mu(x)}{\overline{S}_\mu(x)} = \frac{R}{\chi^*_\mu(x)} \tag{3.59}$$

in accordance with Eq. (3.2). The biased transition kernel from Eq. (3.4) is equal to

$$\begin{aligned} \overline{T}_\mu(x' \to x) &= T_\mu(x' \to x) \frac{\psi^*_\mu(x)}{\chi^*_\mu(x')} \\ &= e^{-\Sigma_t |x-x'|} \frac{\Sigma_t \eta_\psi + \Sigma_{\to} \chi^*_\mu(x) + \Sigma_{\leftarrow} \chi^*_{-\mu}(x)}{\chi^*_\mu(x')}, \qquad \mu(x'-x) > 0 \end{aligned} \tag{3.60}$$

where $\chi^*_\mu(x')$ is a normalisation factor for the kernel $\overline{T}$. In order to select from the biased kernel, we have:

$$\rho = \overline{P}(x) = \int_{x'}^{x} \overline{T}_\mu(x' \to x'') dx'' \tag{3.61}$$

where ρ is a random number, uniformly distributed between 0 and 1. Since selection from Eq. (3.61) is not possible via direct inversion, we calculate the value of x using a root-finding iterative method, such as the regula falsi method, until x converges. The neutron weight must be multiplied by a weight factor

$$W_{\overline{T}} = \frac{\chi^*_\mu(x')}{\psi^*_\mu(x)} \tag{3.62}$$

By simplifying the biased collision kernel from Eq. (3.6) to fit the two-direction model with appropriate normalization, we get:

$$\overline{C}(x, \mu' \to \mu) = C(x, \mu' \to \mu) \frac{\chi^*_\mu(x)}{\psi^*_{\mu'}(x) - \eta_\psi}. \tag{3.63}$$

By use of Eq. (3.45) and (3.55) we arrive at the final form of the biased collision kernel $\overline{C}$:

$$\overline{C}(x, \mu' \to \mu) = \begin{cases} \frac{\Sigma_{\to} \chi^*_\mu(x)}{\Sigma_{\to} \chi^*_{\mu'}(x) + \Sigma_{\leftarrow} \chi^*_{-\mu'}(x)}, & \mu'\mu = +1 \\ \frac{\Sigma_{\leftarrow} \chi^*_\mu(x)}{\Sigma_{\to} \chi^*_{\mu'}(x) + \Sigma_{\leftarrow} \chi^*_{-\mu'}(x)}, & \mu'\mu = -1 \end{cases} \tag{3.64}$$

We can select from the collision kernel by simply calculating the probability of a particle to change direction or not after a collision, since we are only interested in the $+x$ and $-x$ directions. The neutron weight must be multiplied by a weight factor

$$W_{\frac{C}{C}} = \frac{C}{\overline{C}} = \frac{\psi^*_{\mu'}(x) - \eta_\psi}{\chi^*_\mu(x)} \tag{3.65}$$

The zero-variance scheme, using the biasing functions above, should give a result with zero variance for each neutron history. We will now see if this is obtained in practice.

3.3.5 Numerical Demonstration

For demonstration of the principle of the zero-variance scheme, we have written a simple code that simulates a Monte Carlo criticality calculation. The simulation scheme for one cycle (or batch) of particles is shown in Fig. 3.2. The equations used are exactly the same ones as for a source-detector problem, but now using the detector response function $\eta_\psi = \frac{\nu\Sigma_f}{\Sigma_t}$.

In the scheme, particles were selected from the analytically calculated, biased converged neutron source. The selection of the interaction location was done via the biased transition kernel $\overline{T}$, while the biased collision kernel $\overline{C}$ determined the direction after scattering. The exact analytical forms of the adjoint equations were used for the biasing of the kernels.

Because of the implicit capture used and the biasing of both the transition and collision kernels, there are no leakage or absorption events in the system (leakage is included in Fig. 3.2 for generality), so the only way to terminate a particle's history was by Russian roulette below a pre-set weight limit. Because of all this, the contribution of each particle should be exactly the same and equal to the expected detector response if the Russian roulette threshold goes to zero. We used a batch of 2000 particles, while the geometry and cross-section values are the same as stated earlier in this text.

The results, shown in Fig. 3.3, show the average standard deviation for a cycle using the converged source distribution, for different Russian roulette threshold weights. When the threshold is relatively high, as seen in the right part of Fig. 3.3, particles have had a relatively low number of interactions before being killed, which

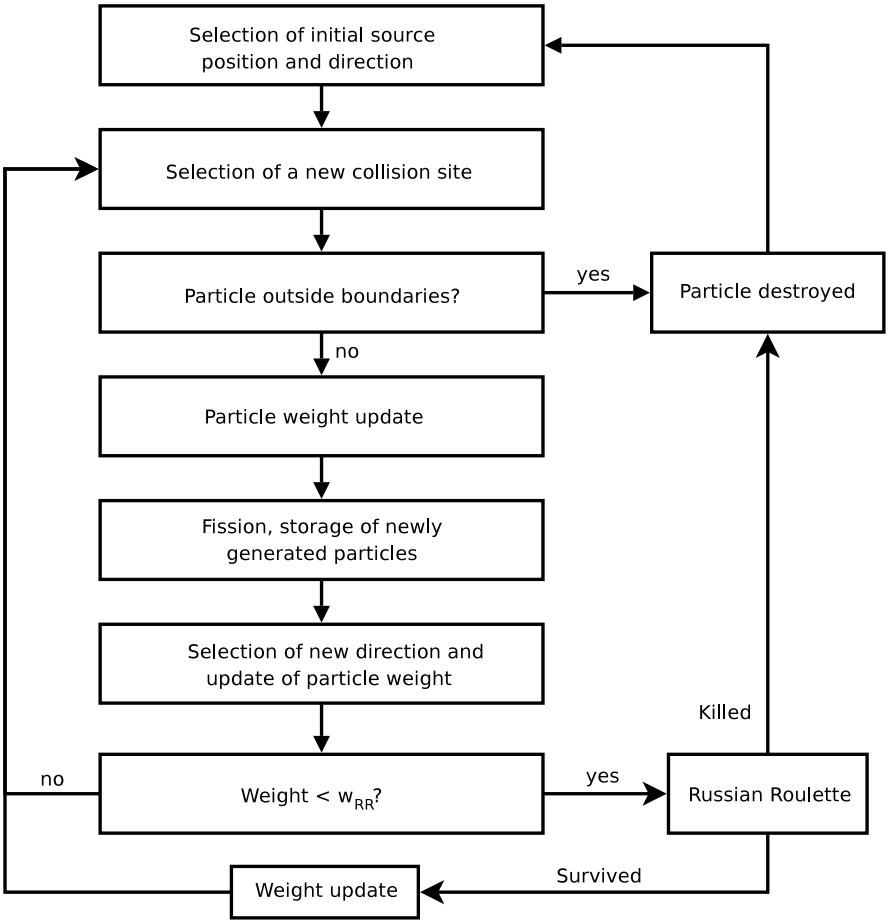

Figure 3.2: The Monte Carlo simulation scheme for one batch of particles.

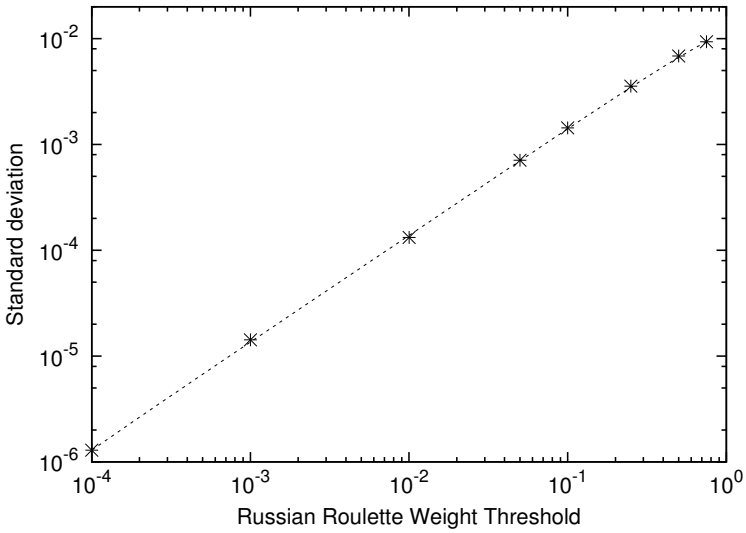

Figure 3.3: k_{eff} standard deviation for varying Russian roulette weight thresholds.

limits the amount of information they provide, and therefore increases the variance. When we move towards lower RR limits, i.e. to the left, particles provide more information (whereas information translates to scores), thus reducing the total variance of the simulation. It is obvious that, by using an appropriately low weight threshold, we can approximate a zero variance at least to the limit of computing accuracy.

However, *approximate* implies that, in fact, a true zero variance scheme cannot exist. We will now see which theoretical and practical issues might arise that introduce a finite variance to our scheme, and in the next chapter we will discuss in detail the effect of each in the performance of the scheme.

3.4 The need for approximations

The first objection to a true zero variance scheme is already seen at the proof of the scheme itself, in Eq. (3.24): An infinite number of collision points is required in order to get zero variance. This is of course not possible to use in any practical form, so Russian roulette is used in order to terminate the histories at low weights.

The use of Russian roulette (which cuts off histories that could still contribute to the detector) therefore brings us the first source of variance, which is dependent on the Russian roulette threshold weight, as can clearly be seen in Fig. (3.3).

In addition, we need to bias the source $S(P)$ by $\chi^*(P)$ in order to reach zero variance, as seen in Eq. (3.1). Since we want to have an unbiased estimate for k_{eff}, we need to weigh the particles appropriately, as per Eq. (3.2). According to the zero-variance scheme, it is in fact this quantity ($\int S(P)\chi^*(P)dP$) that we estimate with zero variance. However, it is only when the true analytical source distribution is used that we can equate this integral to R, as per Eq. (2.28). If not, what we end up with after each cycle is a detector estimate with zero variance, but that detector estimate does not represent k_{eff}. So although the zero-variance proof given in Sect. 3.2 still holds, if we start with an approximation, we end up with an approximation. This problem will be discussed further at the next chapter of this thesis.

In addition to the theoretical problems discussed above, in practical simulations there are more issues that need to be considered before determining an optimum strategy for our variance reduction scheme.

Using extremely low Russian roulette thresholds might reduce variance, but it can disproportionally increase the total CPU time of the simulation. It is therefore necessary to decide whether absolute minimum variance is preferred compared to a possibly higher figure of merit and modify the input weight thresholds accordingly.

An additional issue arises from the complexity of the simulation setup itself. In a practical Monte Carlo simulation, a far more realistic model is used, rather than the simplified 2-direction model we used for our demonstration. This model requires special treatment in position (3-D transport), angle (anisotropy in scattering) and energy (continuous energy configurations, resonance treatment). In such configurations, several functions we used, such as the the converged source distribution, the detector response and the adjoint functions used for biasing, are very hard (or impossible) to calculate analytically.

Also, the adjoint functions used to bias the simulation are in general not exact. This means that, even if all the previous requirements for the zero-variance scheme were satisfied, the biasing with the non-exact adjoint function still leads to a non-zero variance. Note that using the non-exact adjoint function does not lead to a biased k_{eff} estimate, if the weight of the particles is still appropriately changed to compensate for the biasing. The better we approximate the true analytical form of

the adjoint functions, the smaller the variance for one cycle is. Since the adjoint functions are in general output by a deterministic code, it is important to know how the accuracy of the adjoint functions used for biasing affects the performance of the scheme, in order to obtain the maximum FOM for our purpose.

Finally, although the adjoint functions can be approximated, in order for the scheme to give zero variance, the converged source distribution $S(P)$ should be exact, and since in a practical Monte Carlo calculation we start with an approximated source, additional problems regarding variance arise.

In the next chapter, we will see how these approximations affect the nature of the scheme, their effect on variance reduction and whether they still allow the scheme to be of practical use in more realistic configurations.

3.5 Bibliography

T. E. Booth. Zero-variance solutions for linear Monte Carlo. *Nuclear Science and Engineering*, **102**, 332, 1989.

T. E. Booth. Simultaneous Monte Carlo zero-variance estimates of several correlated means. *Nuclear Science and Engineering*, **129**, 199, 1998.

R. R. Coveyou, V. R. Cain and K. J. Yost. Adjoint and importance in Monte Carlo application. *Nuclear Science and Engineering*, **27**, 219, 1967.

S. R. Dwivedi. Zero variance biasing schemes for Monte Carlo calculation of neutron and radiation transport problems. *Nuclear Science and Engineering*, **80**, 172, 1982.

G. Goertzel. Quota sampling and importance functions in stochastic solution of particle problems. Technical Report 434, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1949.

M. C. Gupta. A class of zero-variance biasing schemes for Monte Carlo reaction rate estimators. *Nuclear Science and Engineering*, **83**, 187, 1983.

J. E. Hoogenboom. Optimum biasing of integral equations in Monte Carlo calculations. *Nuclear Science and Engineering*, **70**, 210, 1979.

J. E. Hoogenboom. The two-direction neutral-particle transport model: a useful tool for research and education. *Transport Theory and Statistical Physics*, **37**, 65 , 2008a.

J. E. Hoogenboom. Zero-Variance Monte Carlo schemes revisited. *Nuclear Science and Engineering*, **160**, 1, 2008b.

H. Kahn. Modifications of the Monte Carlo method. Technical Report P-132, Rand Corporation, 1949.

I. Lux and L. Koblinger. *Monte Carlo particle transport methods: neutron and photon calculations*. CRC Press, Boca Raton, 1991.

K. Noack. On the relation between adjoint and importance in Monte Carlo solutions of linear particle transport problems, part I. *Kernernergie*, **22**, 346, 1979.

CHAPTER 4

REDUCING THE VARIANCE

As we discussed in Chapter 3, despite the theoretical feasibility of a zero-variance scheme for criticality, we were only able to apply it to a single-cycle estimate. There are several theoretical and practical reasons why such a scheme cannot be used to its full extent in a practical criticality simulation, in which the fission source is constructed by simulating successive generations of particles, until a converged distribution is achieved. A brief mention of those factors was given at the end of Chapter 3. We will start our investigation with the assumption that the full zero-variance scheme can be applied in practice, analyse the various factors and see the problems that they could pose to the application of the zero-variance scheme in a multiple-cycle, real criticality simulation.

4.1 Russian roulette threshold

This factor has already been discussed in Sect. 3.4: The zero-variance scheme we have devised requires an implicit capture scheme, and in order to achieve true zero variance, infinite particle histories. Since this is not practical, Russian roulette can be used in order to terminate the histories at low weights. The effect of Russian roulette is that information that could still contribute to the detector response is possible discarded, therefore introducing variance into the system. Still, by

lowering the threshold setting for application of Russian roulette, the variance it introduces can be lowered to values limited only by the limits of CPU arithmetic precision, as can be seen in Fig. 3.3.

4.2 Fission source iterations

In the previous chapter, the analytical expressions for the fission source distribution and the adjoint functions were available, from which we could sample the starting positions of neutrons for a single, converged cycle and perform the exact biasing of the kernels. However, this will never be the case in practice and we will now discuss the consequences if these functions are not available in an exact analytical form. If we have available the adjoint function in approximate form, for instance as solution of an approximate deterministic calculation, we can still use these functions for biasing and obtain an unbiased estimate of k_{eff} by applying appropriate weight factors. However, we cannot sample an approximate source distribution as we cannot correct it by a weight factor to get an unbiased estimate.

If the converged source distribution is not known, the usual procedure is to apply iterations using successive generations or cycles according to Eqs. (3.11) and generate the source for the next cycle from Eq. (3.10), starting from an arbitrary initial source distribution. This can also be applied in the case of zero-variance biasing, but it requires a few modifications.

According to Eq. (3.42), our k_{eff} estimate is the average of the individual contributions of the particles to the estimate for a cycle. As we have seen in Sect. 3.2, when the exact biasing is used, the contribution of each particle should be exactly the same and equal to the expected detector response, leading to a zero-variance result, as we have already proven.

The distribution of particle contributions, when using the two-direction model and for the system described earlier can be seen in Fig. 4.1. The majority of particles, as expected, has a contribution equal to the theoretical k_{eff}. The particles that deviate have a higher contribution because of repeated survival of the Russian roulette. In fact, if we use a much lower Russian roulette threshold, down to the limit of computing accuracy, and depending on the source sampled, the distribution can approach a δ-function at $x = k_{theory}$, thus reaching a result with zero variance.

Let us now assume that the same scheme can be directly applied to successive

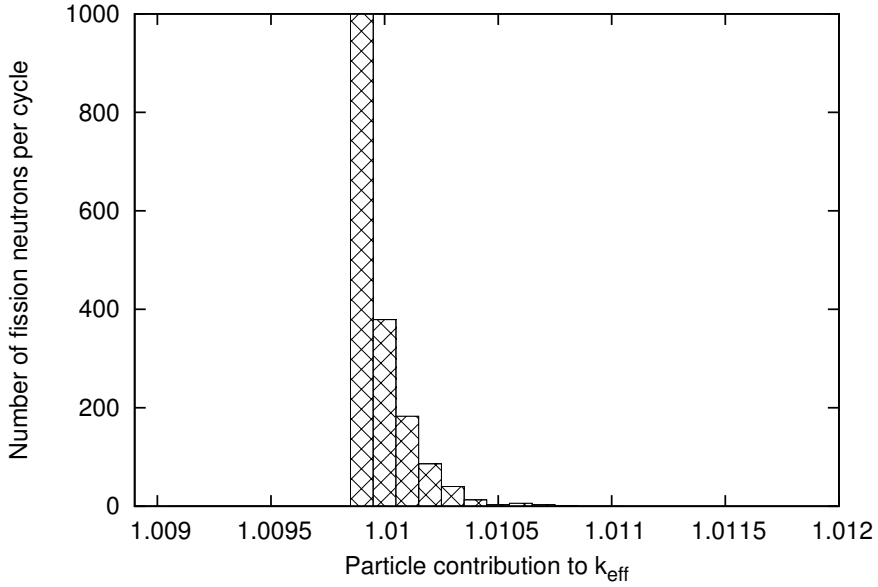

Figure 4.1: Distribution of contributions to k_{eff} for the first cycle, using a Russian roulette threshold of 10^{-3}. The analytically calculated value for k_{eff} is 1.00991.

cycles. The fission source for each cycle is now generated from the neutrons simulated in the previous cycle. In our initial zero-variance scheme, let us call it single-cycle scheme, the source $S(P)$ is calculated analytically, which allows for the calculation of the integral in Eq.(3.2), and therefore R. This, as we have seen in Chapter 3, means that we actually start from R, in order to estimate R with zero variance! Using the proof of the scheme in Sect. 3.2, we can see that what we end up with after each cycle is a detector estimate with zero variance, but that detector estimate does not represent k_{eff}, unless the analytical source distribution is used.

Now, in a criticality simulation, the source of a generation is constructed from the fission events of the preceding generation, until the system converges to the true (in a statistical sense) source distribution. This means that the numerator in Eq. (3.2) is not equal to the detector response R, if not yet converged. In order to be able to still apply a zero-variance scheme, we normalise the integral to the sum of starting particle weights, which is then included in the estimation of k_{eff}. This is in

contrast to sampling the biased analytical source distribution, where according to Eq. (3.2), the expected initial weight of each particle is 1. From now on, this new scheme we have devised will be referred to as the multiple-cycle scheme, and is the scheme we will be using in practice for the rest of this thesis.

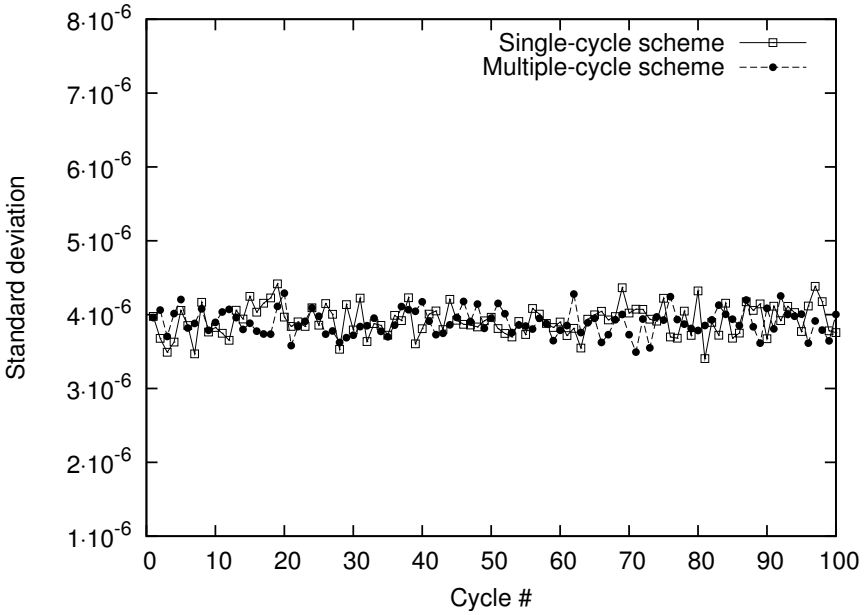

Figure 4.2: Standard deviation per cycle in the single- and multiple-cycle schemes, using a Russian roulette threshold of 10^{-4}. The standard deviation per cycle is on average the same and only limited by the Russian roulette threshold.

Since in the multiple-cycle scheme we are still biasing the transition and collision kernel by the analytically calculated adjoint function, the scheme should give a zero-variance result *within a cycle*. Indeed, in Fig. 4.2, we can see a comparison of the standard deviation per cycle between the single- and multiple-cycle schemes, after convergence, for a Russian roulette threshold of 10^{-4}. Both schemes perform similarly, with the limiting factor being the Russian roulette threshold. The important difference between the schemes is that in the multiple-cycle scheme, the result is no longer the exact solution being replicated at each cycle. Rather, the initial estimate converges slowly to the final solution, with a number of cycles required

in order to converge the source distribution and eliminate the bias in k_{eff}. This result can be seen in Fig. 4.3. Meanwhile, in our single-cycle scheme, it makes no

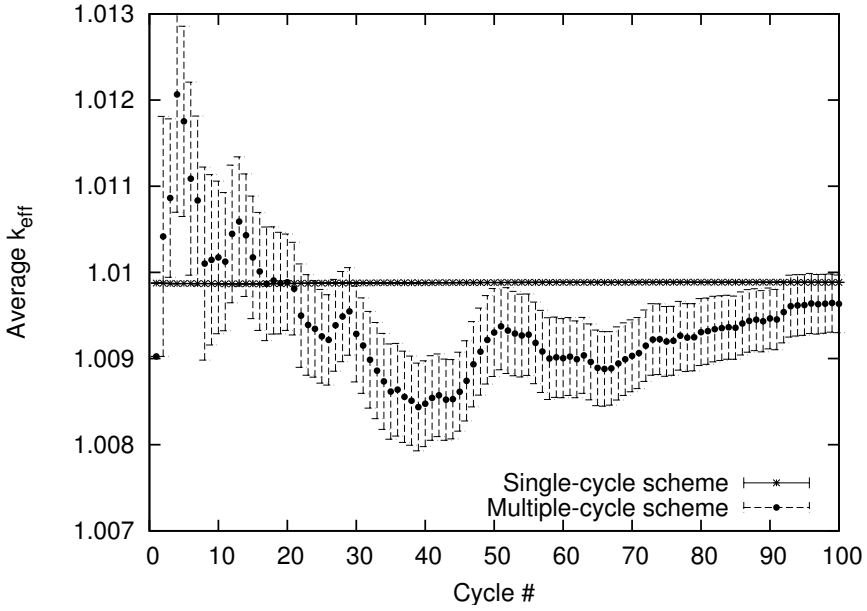

Figure 4.3: Average k_{eff} per cycle for the two calculation schemes. In the single-cycle scheme, the analytical result is effectively replicated at each cycle, while in the multiple-cycle scheme, the estimate converges after a number of cycles to the statistically expected one. Note that the error bars in the single-cycle scheme are too small to discern in the graph.

difference whether we use one or more cycles. We are starting from an analytically calculated R and, through biasing with the exact adjoint function, we are able to get to the same R via Monte Carlo transport. Therefore, the fact that the distribution of the particles is a statistical distribution plays no role.

Looking at the above results, one might pose the following question, regarding the multiple-cycle scheme: Since we are starting the calculation with the exact source (with an arbitrary normalization), using exact biasing, why do we get fluctuating results which are statistically inferior to the 'true' zero-variance scheme?

A first attempt towards an answer can be made looking at the generation of new

particles at fission events. The number of new particles per neutron history must be an integer, since we can only simulate an integer number of particles, and that integer is generated via the use of a random number. The result is that the number of new particles per fission follows a broad discrete distribution, as seen in Fig. 4.4, which introduces a finite variance to the calculation. This can be compared to the distribution of particle contributions in the true zero-variance scheme, as seen in Fig. 4.1.

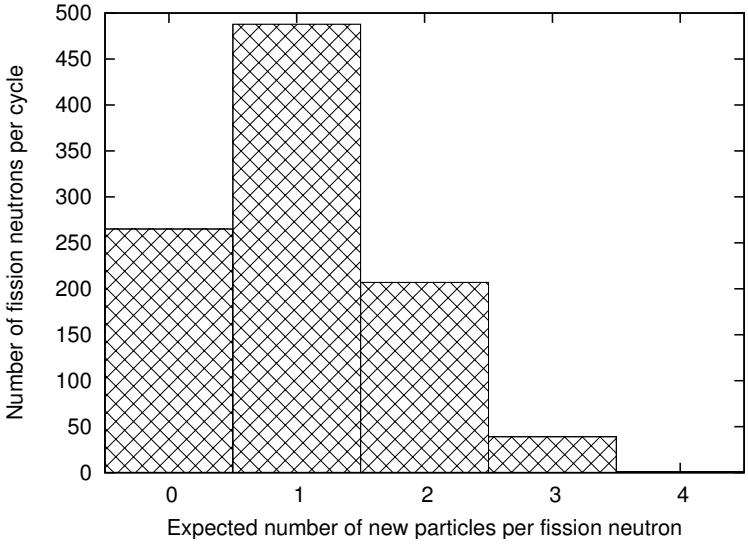

Figure 4.4: Distribution of expected number of new particles generated per neutron history for a batch of 1000 particles.

However, this problem could be theoretically solved by sampling a large number of new fission neutrons, either by sampling more histories per cycle or by artificially increasing the number of new neutrons per fission. Yet, the variance still does not drop to zero. We therefore need to look at what happens at the beginning of every cycle.

4.3 Weight normalisation

For a neutron entering a collision with weight w, the weight of fission neutrons for the next cycle is given by

$$w_f = w \frac{\nu\Sigma_f(\boldsymbol{r}, E)}{\Sigma_t(\boldsymbol{r}, E)} \tag{4.1}$$

Playing a Russian roulette game will result in 0, 1 or possibly more fission neutrons of a preselected weight. The zero-variance scheme for a single cycle requires that the weight of a source particle equals $R_{\text{est}}/\chi^*(\boldsymbol{r}, E, \boldsymbol{\Omega})$ as is the case in Eq. (3.2). As the theoretical value of R is not known exactly, we have to use an estimate of the effective multiplication factor R_{est}, which can be obtained from a previous cycle calculation. At this stage the energy E and direction $\boldsymbol{\Omega}$ of the fission neutron are not yet known. Therefore, we use the adjoint function weighed by the fission spectrum $\chi_f(E)$ and (in the case of fission) isotropic direction probability

$$Q^*(\boldsymbol{r}) = \iint \frac{1}{4\pi}\chi_f(E)\chi^*(\boldsymbol{r}, E, \boldsymbol{\Omega})\, dE\, d\Omega = \int \chi_f(E)\chi^*(\boldsymbol{r}, E)\, dE \tag{4.2}$$

or in multigroup form

$$Q^*(\boldsymbol{r}) = \sum_g \int \frac{1}{4\pi}\chi_f(g)\chi^*(\boldsymbol{r}, g, \boldsymbol{\Omega})\, d\Omega = \sum_g \chi_f(g)\chi^*(\boldsymbol{r}, g). \tag{4.3}$$

To generate fission neutrons with a weight $R_{\text{est}}/Q^*(\boldsymbol{r})$ the number of fission neutrons is chosen equal to

$$n_f = \left[w_f \frac{Q^*(\boldsymbol{r})}{R_{\text{est}}} + \rho \right] \tag{4.4}$$

with $[x]$ denoting the integer part of x and ρ a uniformly distributed random number between 0 and 1, which effectively performs the Russian roulette to get an integer number of fission neutrons. The position and weight of the fission neutrons are stored in a bank and this data is retrieved when starting the next cycle. When starting a neutron history in the next cycle the neutron energy and direction have to be sampled, biased by the adjoint function. For the energy selection, the biased probability density function is

$$\overline{p}(E|\boldsymbol{r}) = \frac{\chi_f(E)\chi^*(\boldsymbol{r}, E)}{Q^*(\boldsymbol{r})}. \tag{4.5}$$

In multigroup form:

$$\overline{p}(g|\boldsymbol{r}) = \frac{\chi_f(g)\chi^*(\boldsymbol{r},g)}{Q^*(\boldsymbol{r})}. \tag{4.6}$$

This requires the weight to be multiplied by a factor

$$W_g = \frac{p(g)}{\overline{p}(g)} = \frac{\sum_{g'} \chi_f(g')\chi^*(\boldsymbol{r},g')}{\chi^*(\boldsymbol{r},g)} = \frac{Q^*(\boldsymbol{r})}{\chi^*(\boldsymbol{r},g)}. \tag{4.7}$$

The biased pdf for the direction is

$$\overline{p}(\boldsymbol{\Omega}|\boldsymbol{r},g) = \frac{1}{4\pi}\frac{\chi^*(\boldsymbol{r},g,\boldsymbol{\Omega})}{\chi^*(\boldsymbol{r},g)}, \tag{4.8}$$

which requires a weight factor

$$W_{\boldsymbol{\Omega}} = \frac{p(\boldsymbol{\Omega})}{\overline{p}(\boldsymbol{\Omega})} = \frac{Q^*(\boldsymbol{r},g)}{\chi^*(\boldsymbol{r},g,\boldsymbol{\Omega})}. \tag{4.9}$$

The initial weight factor then becomes

$$w_{\text{init}} = \frac{R_{\text{est}}}{Q^*(\boldsymbol{r})} \times \frac{Q^*(\boldsymbol{r})}{\chi^*(\boldsymbol{r},g)} \times \frac{\chi^*(\boldsymbol{r},g)}{\chi^*(\boldsymbol{r},g,\boldsymbol{\Omega})} = \frac{R_{\text{est}}}{\chi^*(\boldsymbol{r},g,\boldsymbol{\Omega})}. \tag{4.10}$$

In a two-direction model, these equations simplify to the ones mentioned in the previous sections. A starting weight of each neutron in any cycle inversely proportional to the (exact) adjoint function guarantees a zero-variance estimate (if the transition and collision kernels are biased by the exact adjoint functions). However, the result of the estimate itself will be R_{est} instead of the true value R.

If the theoretical value of R was known, this value could be used instead of R_{est} and an unbiased zero-variance estimate would have been obtained over the next cycle. If only an estimate of R is known, we have to normalise the scores in the next cycle to the averaged initial weight of the neutrons $\overline{w}_{\text{init}}$:

$$\overline{w}_{\text{init}} = \frac{1}{N_f}\sum_i \frac{R_{\text{est}}}{Q^*(\boldsymbol{r}_i)}, \tag{4.11}$$

with $\boldsymbol{r}_i$ the positions where a fission neutron is selected for the next cycle and N_f the total number of fission neutrons for the next cycle. The averaged initial weight of a neutron in a next cycle is a random quantity as it depends on the selection

of the collision sites in the current cycle and the outcome of the Russian roulette to accept or discard the fission neutron. Note that the average is taken over the fission neutron weight before selecting its energy and direction. It would have been normal to average the initial weight after the biased selection of energy and direction, which gives the actual starting weight of a neutron in a next generation, but this would have add only more variance to the averaged initial weight due to the selection of energy and direction and the corresponding weight factors.

According to the zero-variance proof given in Sect. 3.2, the estimate of the detector response in the next section, i.e. k_{eff} will have zero variance if proper biasing of the transition and collision kernels is done and no histories are cut off due to Russian roulette. The estimate itself is, however, not the true k_{eff} value, but R_{est} as used in the initial weight of a fission neutron instead of the true value k_{eff}. During the next cycle a new estimate of k_{eff} is obtained by summing up the contributions of all particles throughout the cycle and then normalised by the averaged initial weight $\overline{w}_{\text{init}}$.

This causes the estimate of k_{eff} to vary from cycle to cycle, which means it does not have zero variance when considered over more cycles. It will only become a zero-variance process if the averaged initial weight $\overline{w}_{\text{init}}$ can be estimated with zero variance.

A first step towards reducing the variance in $\overline{w}_{\text{init}}$ has been introduced already by averaging the initial neutron weight before selecting energy and angle. We can go further by using an expected value estimator instead of the average of actual initial neutron weights from Eq. (4.11). For a neutron with weight w starting a flight path at P the expected number of fission neutrons generated in a volume element dV' at $\boldsymbol{r'}$ is equal to $w\,T(\boldsymbol{r}\rightarrow\boldsymbol{r'},E,\boldsymbol{\Omega})\eta_{\psi}(\boldsymbol{r'},E)\,dV'$.

Although the biased kernel $\overline{T}$ is actually sampled in the Monte Carlo simulation, the neutron weight is corrected afterwards, so effectively the physical transition kernel T is sampled. Hence, the expected number of fission neutrons over all possible path lengths starting at P is

$$\overline{n}_f(P) = w \int\limits_{\text{reactor}} T(\boldsymbol{r}\rightarrow\boldsymbol{r'},E,\boldsymbol{\Omega})\eta_{\psi}(\boldsymbol{r'},E)\,dV' \tag{4.12}$$

Note that integration is up to the outer boundary of the reactor because the biased transition kernel $\overline{T}$ is zero outside the reactor. As we want fission neutrons for the next cycle to have an initial weight equal to $R_{\text{est}}/\chi^*(\boldsymbol{r},E,\boldsymbol{\Omega})$ the probability for

fission is multiplied by the inverse. Therefore, the expected total weight of fission neutrons over all possible path lengths is

$$\overline{w}_f(P) = w \int_{\text{reactor}} T(\boldsymbol{r} \to \boldsymbol{r'}, E, \boldsymbol{\Omega}) \eta_\psi(\boldsymbol{r'}, E) \frac{\chi^*(\boldsymbol{r'}, E, \boldsymbol{\Omega})}{R_{\text{est}}} \, dV' \tag{4.13}$$

All fission neutrons generated for the next cycle are considered as independent. To estimate the average initial weight of neutrons of the next cycle we add up all contributions $n_f(P_i)$ and $w_f(P_i)$, respectively, from all neutrons starting a flight path during their history and from all neutron histories. Then the expected value estimate of the initial weight becomes

$$\overline{w}_{\text{init}} = \frac{\sum_i^{N_{\text{fs}}} \overline{w}_f(P_i)}{\sum_i^{N_{\text{fs}}} \overline{n}_f(P_i)}, \tag{4.14}$$

with N_{fs} the total number of positions from which a neutron starts a flight path during the current cycle. It is not simple to determine the variance in $\overline{w}_{\text{init}}$ as the numerator and denominator are statistically dependent. However, as their correlation is very high (from an actual Monte Carlo simulation it turned out to be > 0.99) we can expect that the variance is much lower than that for the averages of n_f and w_f itself.

4.4 Adjoint functions

The adjoint functions used to bias the simulation are generally not exact. This means that, even if all the previous requirements for the zero-variance scheme were satisfied, the biasing with the non-exact adjoint function still leads to a non-zero variance. In addition, the weight correction for the particles after biasing the kernels can produce abnormal weights, especially in phase-space regions where the adjoint function varies rapidly. This can influence the contribution of particles to the detector response and therefore the final variance of the response, so special care should be taken when setting Russian roulette or splitting limits, to avoid the loss of particle information. We will be discussing the effect of the approximate adjoint biasing later in this, and in subsequent chapters.

4.5 Consequences of using approximations

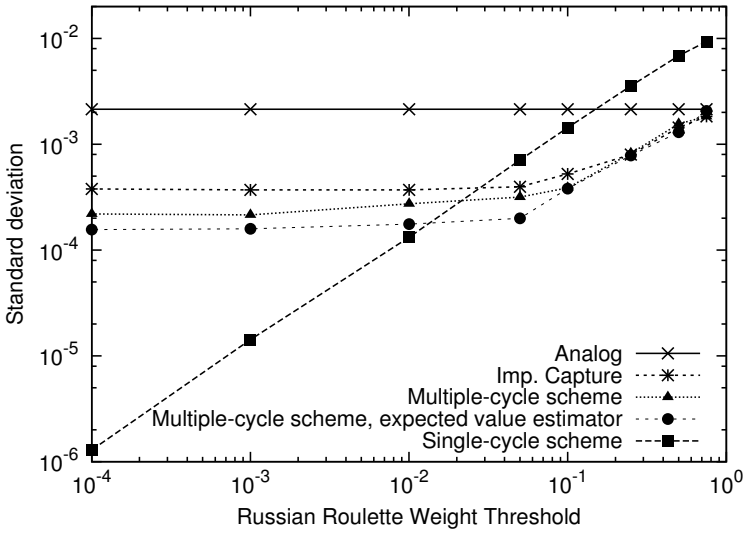

Figure 4.5: Dependence of the standard deviation on the Russian Roulette threshold for the various cases investigated. The single-cycle scheme illustrates the true zero-variance scheme, while the practically applicable multiple-cycle scheme shows significant improvement compared to the implicit capture scheme.

In order to see what the effect on the variance of each of the factors discussed above is, we ran a series of calculations with varying Russian roulette thresholds. The two-direction model was used, while for the calculation 100 active cycles of 2000 particles were simulated, with 10 initial cycles discarded to allow for the fission source to converge, while the statistical weight of a particle after survival was set to twice the threshold weight. The standard deviation was then calculated, with the results being shown in Fig. 4.5.

The case shown in Fig. 4.5 as the single-cycle scheme, corresponds to Fig. 3.3, which is a run with the maximum amount of information available (i.e. the true zero-variance scheme): Particles are selected from the true source distribution, while the theoretical response of the system and the exact adjoint function are used for biasing. The only parameter varied is the Russian roulette threshold. We can

see that the decrease in variance is almost linearly (on a log-log scale) related to the decrease in the weight threshold used, as we have seen earlier. As the source particles are selected from the true distribution and are not the fission particles from the previous cycle, this line represents the absolute minimum standard deviation we can get, limited by numerical precision of the computer used, and does not decrease with more cycles used, unlike the following cases. It should therefore not be directly compared with the following cases, since they represent the standard deviation based on a sample of cycle contributions, and not particle contributions.

In the case indicated as the multiple-cycle scheme, particles are selected from a flat initial distribution, while the theoretical detector response is set to an initial guessed value and updated throughout simulation of successive cycles. The exact adjoint function is still used for biasing. We can see that there is a minimum plateau, generated by the initial weight distribution and averaging, below which reducing the Russian roulette weight threshold offers no further advantage. Note that the scheme produces higher variance for high Russian roulette threshold values. This is because the initial weight change, that compensates for the source biasing, could cause the weight of a particle to immediately drop below the Russian roulette limit. This has a chance to kill the particle before it has had any contribution, leading to a higher variance.

The case indicated as the expected value estimator one is the same as the above, but using the expected value for the average initial neutron weight, as shown in Eq. (4.14). As expected, there is an improvement over the original multiple-cycle scheme, around a factor of 1.5, while the same plateau behaviour is observed.

In the unbiased case only implicit capture is used for the calculation. We can see that implicit capture offers no improvement in the variance below a Russian roulette threshold of 0.1. The reason is that although implicit capture offers more possibilities of scoring through longer particle trajectories, it does not bias the particle transport towards higher importance regions in phase space. From this line we can also justify the conservative, rather high values used by commercial Monte Carlo codes in setting Russian roulette thresholds.

Finally, the case indicated as analogue shows the results for a true analogue Monte Carlo simulation, with the exception of the use of a mean value $\overline{\nu}$ for the number of particles generated through fission. Since no variance reduction methods are used, the variance is constant for a set number of particles N, and represents the upper variance limit. In practice, it should only be used if no other methods are available, since even a simple variance reduction method, such as the use of

implicit capture, can significantly improve the figure of merit, even taking into account the corresponding time penalty.

4.5.1 Biasing individual kernels separately

Initially, we only considered the comparison between a fully biased calculation and a fully unbiased one. However, in practice, collision biasing is not as taxing on CPU time as transition biasing, therefore we also investigated the cases where no biasing is applied to the transition kernel, but adjoint biasing is applied to the collision kernel, and vice versa.

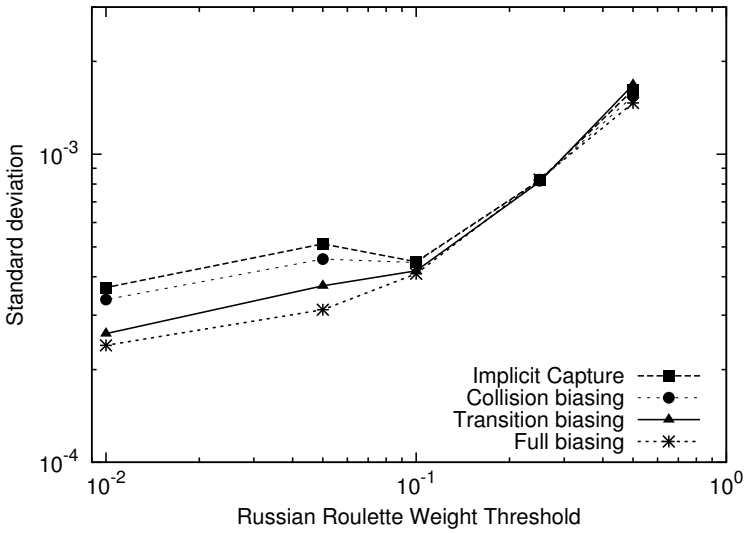

Figure 4.6: Dependence of the variance on the Russian Roulette threshold for the different biasing forms. Transition biasing is more effective since it has the added effect of prohibiting leakage, which increases the information available.

The results are shown in Fig. 4.6. We can see that transition biasing is more effective than collision biasing, which can be explained by the fact that it prohibits leakage of neutrons, unlike collision biasing. However, one thing not shown here is the figure of merit, which is much higher for the collision biasing scheme, due to

the much simpler calculations involved. We will be dealing with the efficiency of the schemes in the next chapter of this thesis.

4.5.2 Effect of the adjoint function's discretisation

Of practical importance is the discussion of the adjoint function used for biasing. In all but the simplest cases, the analytical forms of the adjoint equations are very hard to solve. Therefore, another method must be used in order to acquire the adjoint functions. In practice, this is done using the output of a deterministic calculation. However, the fact that approximate adjoint functions are used means that the scheme has to be altered, since the kernels must be normalised and the weight corrections changed in order to account for the normalisation.

Suppose we derived an approximate adjoint function $\tilde{\chi}^*(\boldsymbol{r}, g, \boldsymbol{\Omega}_m)$ from a deterministic multigroup discrete-ordinates calculation with g indicating the energy group and $\boldsymbol{\Omega}_m$ the discrete direction. Note that the directional adjoint function as normally obtained from a discrete-ordinates calculation must be multiplied by the weight w_m of the quadrature set corresponding to direction $\boldsymbol{\Omega}_m$ as used in the discrete-ordinates quadrature set.

Theoretically, it is possible to use adjoint functions with discrete directions in order to bias a continuous direction Monte Carlo calculation. However, as the practical implementation, described later in this thesis, is not as simple, we are going to use discrete directions also during Monte Carlo transport. The only difference is that rather than selecting a direction from a continuous distribution at the particle source and after each scattering event, a discrete direction m is now selected from the possible scattering directions with probability w_m.

In the zero-variance based scheme, when using a multigroup treatment in the Monte Carlo simulation, the energy group of a neutron generated by fission at a point $\boldsymbol{r}$ is selected according to

$$\overline{p}(g|\boldsymbol{r}) = \frac{\chi_f(g)\tilde{\chi}^*(\boldsymbol{r}, g)}{\sum_{g'} \chi_f(g')\tilde{\chi}^*(\boldsymbol{r}, g')}. \tag{4.15}$$

and the discrete direction from

$$\overline{p}(\boldsymbol{\Omega}_m|\boldsymbol{r}, g) = \frac{\tilde{\chi}^*(\boldsymbol{r}, g, \boldsymbol{\Omega}_m)}{\tilde{\chi}^*(\boldsymbol{r}, g)}, \tag{4.16}$$

The correction weight now becomes

$$W_{\overline{S}} = \frac{\sum_{g''} \chi_f(g'')\tilde{\chi}^*(\boldsymbol{r}, g'')}{\tilde{\chi}^*(\boldsymbol{r}, g, \boldsymbol{\Omega}_m)} \tag{4.17}$$

When biasing the transition kernel, we now need to take into account the fact that it is no longer normalised by χ^*. The adjoint function $\tilde{\psi}^*$ is obtained from $\tilde{\chi}^*$ according to Eq. (3.7):

$$\tilde{\psi}^*(\boldsymbol{r}, g, \boldsymbol{\Omega}_m) = \eta_\psi(\boldsymbol{r}, g) + \sum_{g'} \sum_{m'} \frac{\Sigma_s(\boldsymbol{r}, g \to g', \boldsymbol{\Omega}_m \to \boldsymbol{\Omega}_{m'})}{\Sigma_t(\boldsymbol{r}, g)} \tilde{\chi}^*(\boldsymbol{r}, g', \boldsymbol{\Omega}_{m'}) \tag{4.18}$$

Now the biased transition kernel becomes

$$\overline{T}(\boldsymbol{r'} \to \boldsymbol{r}, g, \boldsymbol{\Omega}_m) = \frac{T(\boldsymbol{r'} \to \boldsymbol{r}, g, \boldsymbol{\Omega}_m)\tilde{\psi}^*(\boldsymbol{r}, g, \boldsymbol{\Omega}_m)}{\int T(\boldsymbol{r'} \to \boldsymbol{r''}, g, \boldsymbol{\Omega}_m)\tilde{\psi}^*(\boldsymbol{r''}, g\boldsymbol{\Omega}_m)\, dV''} \tag{4.19}$$

In practice, the new path length needs to be selected from the normalised probability as follows:

$$\rho = \frac{\int_0^s T(\boldsymbol{r'} \to \boldsymbol{r'} + s'\boldsymbol{\Omega}_m, g, \boldsymbol{\Omega}_m)\tilde{\psi}^*(\boldsymbol{r'} + s'\boldsymbol{\Omega}_m, g, \boldsymbol{\Omega}_m) ds'}{\int_0^{s_{\max}} T(\boldsymbol{r'} \to \boldsymbol{r'} + s'\boldsymbol{\Omega}_m, g, \boldsymbol{\Omega}_m)\tilde{\psi}^*(\boldsymbol{r'} + s'\boldsymbol{\Omega}_m, g, \boldsymbol{\Omega}_m) ds'}, \tag{4.20}$$

where ρ is a random number, uniformly distributed between 0 and 1 and $s_{\max}$ is the maximum distance from $\boldsymbol{r'}$ to the outer boundary in the direction $\boldsymbol{\Omega_m}$. If $s_{\max}$ is large the integration may be truncated in practice to a sufficient number of mean free paths.

After selecting ρ, we can now calculate the value of s and hence $\boldsymbol{r} = \boldsymbol{r'} + s\boldsymbol{\Omega}_m$ using a root-finding iterative method until $\boldsymbol{r}$ converges. The weight factor will now be

$$W_{\overline{T}} = \frac{\int_0^{s_{\max}} T(\boldsymbol{r'} \to \boldsymbol{r'} + s'\boldsymbol{\Omega}_m, g, \boldsymbol{\Omega}_m)\tilde{\psi}^*(\boldsymbol{r'} + s'\boldsymbol{\Omega}_m, g, \boldsymbol{\Omega}_m) ds'}{\tilde{\psi}^*(\boldsymbol{r}, g, \boldsymbol{\Omega}_m)}. \tag{4.21}$$

For the collision kernel, we perform the biasing in two steps. Initially, we select the new energy group g after the collision with probability

$$\overline{p}(g|\boldsymbol{r},g') = \frac{\Sigma_s(\boldsymbol{r},g' \to g)\tilde{\chi}^*(\boldsymbol{r},g)}{\sum_{g''}\Sigma_s(\boldsymbol{r},g' \to g'')\tilde{\chi}^*(\boldsymbol{r},g'')}. \tag{4.22}$$

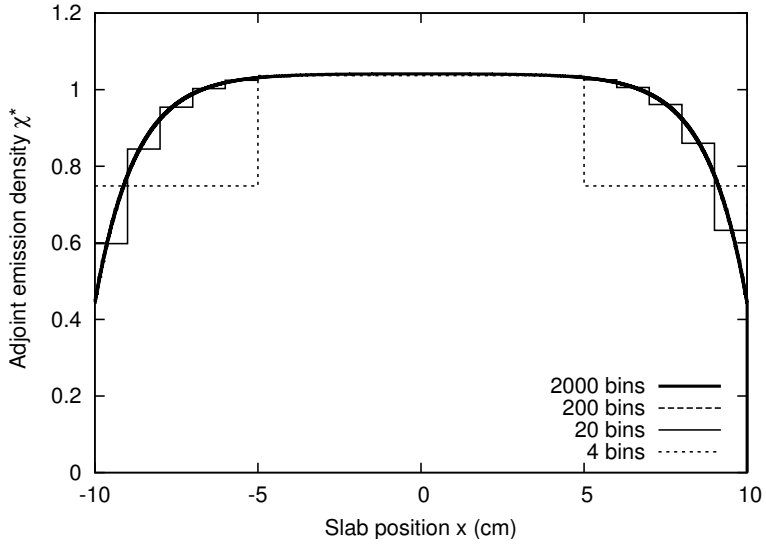

Figure 4.7: Effect of discretisation on the form of the adjoint function $\chi^*(x)$. A fine discretisation approaches the analytical form very well, while a coarse one introduces significant error towards the boundaries of the system, where $\chi^*(x)$ changes rapidly.

After that, we need to select an outgoing direction, with probability

$$\overline{p}(\boldsymbol{\Omega}_m|\boldsymbol{r},g',g) = \frac{\Sigma_s(\boldsymbol{r},\boldsymbol{\Omega}_{m'} \to \boldsymbol{\Omega}_m|g',g)\tilde{\chi}^*(\boldsymbol{r},g,\boldsymbol{\Omega}_m)}{\sum_{m''}\Sigma_s(\boldsymbol{r},\boldsymbol{\Omega}_{m'} \to \boldsymbol{\Omega}_{m''}|g',g)\tilde{\chi}^*(\boldsymbol{r},g,\boldsymbol{\Omega}_{m''})}. \tag{4.23}$$

The (combined) weight factor to be applied for the collision biasing will then be

$$W_{\overline{C}} = \frac{\sum_{g''}\sum_{m''}\Sigma_s(\boldsymbol{r}, g' \to g'', \boldsymbol{\Omega}_{m'} \to \boldsymbol{\Omega}_{m''})\tilde{\chi}^*(\boldsymbol{r}, g'', \boldsymbol{\Omega}_{m''})}{\tilde{\chi}^*(\boldsymbol{r}, g, \boldsymbol{\Omega}_m)}. \tag{4.24}$$

We can now use approximated adjoint functions, in order to see whether the approximation affects the variance of our calculations. As a demonstration, we used bin-averaged adjoint function, generated by binning the exact values calculated using Eq. (3.52). The result of the averaging can be seen in Fig. 4.7.

Despite the fact that the piecewise constant adjoint using a large number of bins (up to 2000) approximates the true adjoint extremely well, the fact that an approximation is used means that the scheme has to be altered slightly, since the kernel sampling is now done via a discrete, rather than a continuous distribution function. The exact workings of the scheme, and application to a more complex system, will be the subject of the next chapter of this thesis.

Once again, the same cross-section data and system parameters were used for the calculation. The results can be seen in Table 4.1. It is interesting that in the only case

Number of bins	σ_{cycle}
0 (Imp. Capture)	3.71×10^{-3}
4	5.52×10^{-3}
20	1.24×10^{-3}
200	3.02×10^{-4}
2000	2.73×10^{-4}

Table 4.1: Average σ per cycle for different levels of adjoint function discretisation, for a Russian roulette threshold of 10^{-2}.

that does not accurately represent the shape of the adjoint function, the standard deviation per cycle is in fact higher than the case where only implicit capture was used. This is a direct result of the non-optimum weighting of the particles, which then causes a very broad distribution in contributions to the final score, and therefore a higher variance. This effect is especially pronounced towards the boundaries of our system, where the difference between the analytically calculated and the approximated adjoint function can lead to large variations in the weight corrections for particles that traverse the region, resulting in an increase, rather than decrease of the variance.

If a finer discretisation is used, so that the digression from the analytical form near the boundaries of the system is smaller, the results show significant improvement. Taking into account that we are discussing criticality problems here, which are by default global problems, approximating the shape, rather than the exact value of the adjoint function, could be sufficient. Indeed, in the last two cases (200 and 2000 bins), the reduction in variance is not worth the extra computational time spent in order to create the finer grid and calculate the integrals in Eq. (4.20).

4.6 Concluding remarks

We have seen that zero variance cannot be obtained in practice, due to the need for renormalisation and averaging of particle weights at the beginning of each successive generation. However, a reduction in variance does indeed occur even when using approximate adjoint functions that have been obtained computationally.

The analysis of the various sources of variance in a Monte Carlo simulation also produces some interesting results regarding the points one should focus on when trying to improve the variance in a calculation.

Analogue schemes should in general be avoided, since a simple variance reduction method, such as implicit capture, produces better results with minimal implementation effort and CPU time overhead. When using an implicit capture scheme, lowering the Russian roulette threshold below $0.1w_0$, where w_0 is the starting simulation weight of particles, results to no noticeable improvement in variance, while the simulation slows down from the additional paths simulated. Therefore, when no other methods are used along with implicit capture, a conservative (on the high side) approach in the threshold is the best, regarding the efficiency of the simulation.

Depending on the quality of the adjoint function, biasing should improve on the implicit capture case. However, one important thing to note is that the nature of the biasing scheme using the approximate adjoint function makes it more taxing on CPU. This means that the efficiency of the biasing is heavily dependent on the computational scheme used, as well as the method used to approximate the adjoint functions (piecewise constant, linear or spline interpolation, etc.). During our numerical tests, we managed to raise the figure of merit by a whole order of magnitude, by simply optimizing the root-finding algorithm for calculating the adjoint at a point x. Another option considered was the biasing of the transport

kernel based on the slope of the adjoint function. That way, biasing would be applied only where needed (near interfaces, voids, etc.), while in large homogenized regions it would be omitted for gains in CPU time.

In the next chapter we will deal with the implementation of biasing using approximated adjoint functions obtained from a deterministic, the efficiency of such an implementation, and how to improve it.

CHAPTER 5

A PRACTICAL APPLICATION

As we have seen in Chapter 4 of this thesis, using the two-direction model, we could analytically obtain all the information required for a zero-variance scheme. We also saw how, in a criticality calculation, a number of factors affect the final variance and lead to a non-zero variance for the final k_{eff} estimate. In this chapter, we will start building up a more complex model by removing some of the simplifications applied earlier and see how the scheme performs when used for a practical Monte Carlo application.

5.1 Obtaining the adjoint functions

Since we are no longer considering the two-direction model, analytically obtaining the adjoint functions is extremely hard, if not impossible. Therefore, we are going to make use of a deterministic code, which can output discrete values of the adjoint function. We will then have to modify the scheme to take into account the discretisation and see its effect on our calculation.

To that end, the XSDRN code (Greene and Craven Jr, 1969) was initially selected as the deterministic code to use. XSDRN is a 1-D discrete-ordinates transport code which can, upon option, solve the adjoint forms of the 1-D transport equation. resulting in the angular and total $\chi^*(x)$. We can then directly use the output of

XSDRN in order to bias the source and transport kernel. We initially tested XSDRN against the analytical results of the two-direction model, using the configuration shown in Chapter 4. XSDRN could easily accommodate the two-direction model, since we only needed to set the direction cosines to ± 1, each being selected with probability 1/2. In our simple geometry, XSDRN gave excellent results for the values of the adjoint function, that compared well with the analytical solutions, both in isotropic and anisotropic scattering calculations.

5.2 Implementation for more realistic problems

In order to apply a more realistic model to our simulations, several limitations of the two-direction model were removed and the test code created for demonstration of the zero-variance scheme was extended, in order to accommodate those changes, starting with the input and use of discrete adjoint function values.

A multigroup model was used in energy for the Monte Carlo calculations, in order to better observe the effect of the scattering kernel biasing. In addition, transport calculations could now be done in any direction, rather than the two directions used earlier. However, we chose to use discrete rather than continuous directions for transport in the Monte Carlo code. By using the same set of discrete directions as the ones defined by the deterministic code, we could directly compare the results for debugging purposes and significantly simplify input/output operations between the two codes. Preliminary tests showed differences of less than 1 % between using for instance 16 or more directions and a continuous direction model for our simple geometries, so the results shown here also hold for a continuous direction model.

As initially we did not have a deterministic code available that could generate the angular dependent adjoint function in 3-D, we limited our geometrical capabilities to 1-D geometries.

5.3 Source and kernel sampling in the biased scheme

As we have previously discussed in Sect. 5.1, the adjoint functions are input as discrete values at predetermined points in a grid. However, biasing the kernels requires that the adjoint function is available at any point x (in a 1-D geometry). In order to achieve that, we have to interpolate between the grid points where the

adjoint functions are available. In this thesis we will make use of two methods of interpolation: constant value throughout a bin, where a bin is taken as the space between grid points, and linear interpolation of values between two successive grid points.

When using the bin-wise constant adjoint function, the value is taken as the average value of the adjacent grid points. For $x_0 < x < x_1$,

$$\chi^*(x) = \frac{\chi^*(x_0) + \chi^*(x_1)}{2}. \tag{5.1}$$

As we have seen in the previous chapter, if a coarse adjoint grid is used, the shape of the adjoint function is not approximated very well. To prevent that, the adjoint function at any point x can also be approximated by linearly interpolating the values at the grid points adjacent to x as follows:

$$\chi^*(x) = \chi^*(x_0) + (x - x_0)\frac{\chi^*(x_1) - \chi^*(x_0)}{x_1 - x_0}. \tag{5.2}$$

5.3.1 Sampling the biased source

In order to select from the biased source distribution in the first cycle of the calculation, we can isolate the source components in space, energy and direction as follows:

$$S(x, g, \mu) = S(x)\chi_f(g)w_m, \tag{5.3}$$

where x is the position of the new particle and $\chi_f(g)$ is the fission energy spectrum (in group-wise form). The term w_m represents the weighting factor of the m direction in the deterministic calculation and, in Monte Carlo terms, the probability that the direction m with cosine μ_m with the x-axis is selected as the direction at the source or after a scattering event. This means that

$$\sum_m w_m = 1. \tag{5.4}$$

For the first cycle only, since for later ones the initial location of a particle is already set, we select x from the biased fission source distribution:

$$\overline{S}(x) = \frac{S(x)\sum_g \chi_f(g)\chi^*(x, g)}{\int \sum_g S(x')\chi_f(g)\chi^*(x', g)dx'}. \tag{5.5}$$

The correction weight will then be:

$$W_x = \frac{\sum_g \chi_f(g) \int S(x')\chi^*(x',g)dx'}{\sum_g \chi_f(g)\chi^*(x,g)}. \tag{5.6}$$

Afterwards, we need to bias the initial energy group selection. This is done via biasing the fission spectrum:

$$\overline{S}(g|x) = \frac{\overline{S}(x,g)}{\overline{S}(x)} = \frac{\chi_f(g)\chi^*(x,g)}{\sum_{g'} \chi_f(g')\chi^*(x,g')}. \tag{5.7}$$

Again, we need to weigh the particles with weight factor

$$W_g = \frac{\sum_{g'} \chi_f(g')\chi^*(x,g')}{\chi^*(x,g)}. \tag{5.8}$$

Finally, we need to bias the selection of initial direction of the particle:

$$\overline{S}(\mu|g,x) = \frac{\overline{S}(x,g,\mu)}{\overline{S}(g,x)} = \frac{w_m\chi^*(x,g,\mu_m)}{\sum_m w_m\chi^*(x,g,\mu_m)}. \tag{5.9}$$

The weighed sum of the direction-dependent adjoint functions $\chi^*(x,g,\mu_m)$ is simply the scalar $\chi^*(x,g)$:

$$\sum_m w_m\chi^*(x,g,\mu_m) = \chi^*(x,g). \tag{5.10}$$

Therefore, the new correction weight is

$$W_\mu = \frac{\chi^*(x,g)}{\chi^*(x,g,\mu_m)}. \tag{5.11}$$

The total weight correction now is

$$W = W_x W_g W_\mu = \frac{\sum_g \chi_f(g) \int S(x')\chi^*(x',g)dx'}{\chi^*(x,g,\mu_m)} \tag{5.12}$$

Since we alter the starting weight of the particles, we need to renormalise the source at each cycle, which is done during fission, according to Eq. (4.4).

5.3.2 Sampling the biased collision kernel

Sampling the collision kernel is done in the same way as with the analytically-obtained adjoint functions, and therefore according to Eq. (3.31) and (3.32). The weight correction is shown in Eq. (3.33).

5.3.3 Sampling the biased transition kernel

In order to sample the biased transition kernel, the adjoint collision density $\psi^*(x)$ must be used. We can convert $\chi^*(x)$ from our deterministic output to the required $\psi^*(x)$, assuming isotropic scattering, using the discretised form of Eq. (3.7):

$$\begin{aligned}\psi^*(x,g,\mu_m) &= \frac{\nu\Sigma_f(x,g)}{\Sigma_t(x,g)} + \frac{1}{\Sigma_t(x,g)}\sum_{g'}\Sigma_{g\to g'}\sum_m w_{m'}\chi^*(x,g',\mu_{m'}) \\ &= \frac{\nu\Sigma_f(x,g)}{\Sigma_t(x,g)} + \frac{1}{\Sigma_t(x,g)}\sum_{g'}\Sigma_{g\to g'}\chi^*(x,g').\end{aligned} \tag{5.13}$$

This means that the collision density $\psi^*(x,g,\mu_m)$ is the same for all values of μ_m, for given (x,g). We will now show how to sample from the biased transition kernel, using the two methods of interpolation discussed earlier. Since the normalization integrals for the transition kernel biasing depend on the interpolation used, different methodologies have to be used for each. Note that, since we are only concerned with neutrons, the energy of a particle does not change in between collisions, therefore for simplicity we will denote $\psi^*(g,x)$ as $\psi^*(x)$.

Biasing using bin-wise constant adjoint function

In the case of bin-wise constant adjoint function, assuming that bin i ranges between x_{i-1} and x_i, $\psi^*(x)$ is given as a set of values $\psi^*_i(i=1,N)$, where N is the number of bins. For a particle starting its flight path at x' and having its next collision at x, the transition kernel is given by

$$T(x'\to x) = \mathrm{e}^{-z_{i-1}}\Sigma_t(i)\mathrm{e}^{-\Sigma_t(i)\frac{(x-x_{i-1})}{\mu}}, \qquad x_{i-1}<x<x_i \tag{5.14}$$

with

$$z_i = \sum_{j=1}^{i}\Sigma_t(j)\frac{(x_j-x_{j-1})}{\mu}, \tag{5.15}$$

which is the number of mean free paths up to $x = x_i$; e^{-z_i} is the probability for a neutron to reach x_i without any collision, while $\Sigma_t(j)$ is the total cross-section inside bin j. Now the biased transition kernel becomes

$$\overline{T}(x' \to x) = \frac{\psi^*(x) T(x' \to x)}{A_{\text{norm}}} = \frac{\psi_i^* e^{-z_{i-1}} \Sigma_t(i) e^{-\Sigma_t(i)\frac{(x-x_{i-1})}{\mu}}}{A_{\text{norm}}}, \qquad x_{i-1} < x < x_i \tag{5.16}$$

Although the transition kernel is not limited to x_N, we take $\psi^*(x) = 0$ for $x > x_N$, since the value of the adjoint function is 0 outside the system. Now, the normalization constant becomes

$$A_{\text{norm}} = \int_0^{x_N} \overline{T}(x)\mathrm{d}x = \sum_{i=1}^{N} \psi_i^* \int_{x_{i-1}}^{x_i} \overline{T}(x)\mathrm{d}x = \sum_{i=1}^{N} \psi_i^* e^{-z_{i-1}} (1 - e^{-\Sigma_t(i)\frac{\Delta x_i}{\mu}}). \tag{5.17}$$

To sample $\overline{T}$ we have to solve the equation

$$\begin{aligned} \rho &= \int_0^x \overline{T}(x')\mathrm{d}x' \\ &= \frac{1}{A_{\text{norm}}} \left\{ \sum_{k=1}^{i-1} \psi_k^* e^{-z_{k-1}} \left(1 - e^{-\Sigma_{t_k} \frac{\Delta x_k}{\mu}}\right) + \psi_i^* e^{-z_{i-1}} \left(1 - e^{-\Sigma_t(i)\frac{(x-x_i)}{\mu}}\right) \right\}, \end{aligned} \tag{5.18}$$

where the first term inside the summation is the probability to reach point x_{i-1} and the second term is the probability, starting from x_{i-1}, to reach x. To determine the mesh interval i we define the probability for selecting interval k by

$$P_k^* = \frac{1}{A_{\text{norm}}} \int_{x_{k-1}}^{x_k} \overline{T}(x')\mathrm{d}x' = \frac{1}{A_{\text{norm}}} \psi_k^* e^{-z_{k-1}} (1 - e^{-\Sigma_t(\frac{\Delta x_k}{\mu})}). \tag{5.19}$$

We then select i from the cumulative probabilities $P^*_{\text{cum},i-1} < \rho < P^*_{\text{cum},i}$ with

$$P^*_{\text{cum},i} = \sum_{k=1}^{i-1} P_k^* \tag{5.20}$$

Then, using the same value of ρ, we can solve Eq. (5.18). This results in

$$x = x_{i-1} - \frac{\mu}{\Sigma_t(i)} \ln \left[1 - (\rho - P^*_{\text{cum},i-1}) \frac{A_{\text{norm}}}{\psi_i^*} e^{z_{i-1}} \right]. \tag{5.21}$$

Using Eq. (5.14), we correct the weight of the particles by the following factor:

$$W = \frac{T(x' \to x)}{\overline{T}(x' \to x)} = \frac{A_{\text{norm}}}{\psi_i^*} \tag{5.22}$$

Biasing using linearly interpolated adjoint function

In the case of linearly interpolated adjoint function, we use the same methodology as with the bin-wise constant approximation, with one important difference: There is now an extra term, due to the linear dependence on x, that makes the functions slightly more complicated. The biased kernel now is:

$$\overline{T}(x' \to x) = \frac{\mathrm{e}^{-z_{i-1}} \psi^*(x) \Sigma_t(i) \mathrm{e}^{-\Sigma_t(i)\frac{(x-x_{i-1})}{\mu}}}{A_{\text{norm}}}, \qquad x_{i-1} < x < x_i \tag{5.23}$$

For ease of use we can transform the equations to equations of the form $y = ax + b$ as follows:

$$\psi^*(x) = c_{i-1} + a_i x, \qquad x_{i-1} \leq x < x_i \tag{5.24}$$

where

$$a_i = \frac{\psi^*(x_i) - \psi^*(x_{i-1})}{x_i - x_{i-1}} \tag{5.25}$$

and

$$c_{i-1} = \psi^*(x_{i-1}) - a_i x_{i-1}. \tag{5.26}$$

Sampling $\overline{T}(x' \to x)$ requires selecting from the following expression:

$$\begin{aligned} \rho &= \int_0^x \overline{T}(x')\mathrm{d}x' \\ &= \frac{1}{A_{\text{norm}}} \left\{ \sum_{k=1}^{i-1} \int_{x_{k-1}}^{x_k} \psi^*(x'')T(x' \to x'')dx'' + \int_{x_{i-1}}^{x} \psi^*(x'')T(x' \to x'')dx'' \right\}. \end{aligned} \tag{5.27}$$

Solving in a similar way as was used in the previous section, we get:

$$\begin{aligned} P_k^* = \frac{1}{A_{\text{norm}}} \int_{x_{k-1}}^{x_k} \overline{T}(x')\mathrm{d}x' = \frac{1}{A_{\text{norm}}} \Bigg\{ &(c_{k-1} + \frac{a_k \mu}{\Sigma_t(k)} + a_k x_{k-1}) \mathrm{e}^{-\Sigma_t(k)\frac{(x_{k-1}-x')}{\mu}} \\ &- (c_{k-1} + \frac{a_k \mu}{\Sigma_t(k)} + a_k x_k) \mathrm{e}^{-\Sigma_t(k)\frac{(x_k-x')}{\mu}} \Bigg\}. \end{aligned} \tag{5.28}$$

We then find the bin of the final location of the particle, selecting from the cumulative probabilities as per Eq. (5.20). In order to sample the path of the particle inside the final bin k, because of the complexity of Eq. (5.27), we have to use a root-finding method. The final correction weight is

$$W = \frac{T(x' \to x)}{\overline{T}(x' \to x)} = \frac{A_{\text{norm}}}{\psi^*(x)} \tag{5.29}$$

5.4 Application of the scheme

5.4.1 A Loosely Coupled System

In order to demonstrate the zero-variance scheme in a more realistic test configuration, we examined the case of a loosely-coupled system. The system examined is described in the OECD/NEA benchmarks on source convergence (Blomquist, 2003), and more specifically case 4 in benchmark 3. It was chosen as, in addition to the variance reduction, it allows us to investigate the effect of biasing on source convergence. It is composed of a one-dimensional slab (infinite in the y- and z-directions), as shown in Fig. 5.1.

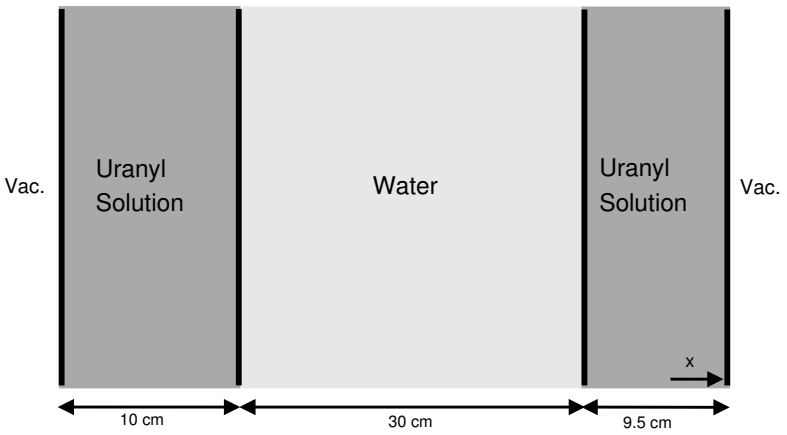

Figure 5.1: Geometry of the loosely coupled system. The system is infinite in the y and z dimensions

A vacuum boundary condition is imposed on both sides of the slab. The slab is divided into 3 sections: two slabs of variable width containing a Uranyl solution, separated by a water slab, the width of which was set to 30 cm. In this setup, k_{eff} converges faster than the fission source distribution, leading to incorrect results when an insufficient number of initial cycles has been discarded. In addition, when the system is symmetric, or nearly symmetric, there is a periodic shift in the distribution of the source between the two Uranyl slabs, making convergence even harder to predict.

Material	Group	$\Sigma_t(\text{cm}^{-1})$	$\Sigma_a(\text{cm}^{-1})$	$\Sigma_f(\text{cm}^{-1})$	ν	χ_f
Uranyl	1	0.26715	0.00132	0.00036	2.77609	0.76153
	2	1.10219	0.00592	0.00104	2.43630	0.23847
	3	2.88911	0.06055	0.03434	2.43783	0.00000
Water	1	0.26297	0.00064			
	2	1.22258	0.00056			
	3	3.27209	0.01891			

Table 5.1: 3-group parameters of the test problem.

Uranyl				Water			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.16595	0.09987	0.00000	1	0.15973	0.10260	0.00000
2	0.00000	1.03720	0.05907	2	0.00000	1.13860	0.08342
3	0.00000	0.00026	2.82830	3	0.00000	0.00008	3.25310

Table 5.2: $\Sigma_{g' \to g}$ in cm^{-1} for the materials used in the test problem, where g′ is the incoming and g the outgoing energy group.

Using the compositions given in the benchmark specification, a reduced set of zone-mixed cross-sections for the 2 materials was prepared through the use of the SCALE code system (2006). For simplicity, only the P0 scattering matrix was used, which means that all scattering was isotropic. XSDRN was then used in order to generate the adjoint functions, which were passed to the Monte Carlo code for use in biasing. The cross-section data used for our problem can be seen in Tables 5.1 and 5.2.

5.4.2 Results

For our test calculation, we opted for a system where the 3 slabs had widths of 10, 30 and 9.5 cm. By making the water slab much larger, we should be able to see how biasing affects the transport of particles (and therefore information) between the two fissile slabs. In addition, making the system nearly, but not exactly symmetric means that the source should reach convergence, albeit slowly. The adjoint functions were initially obtained by running an adjoint S_N calculation using a bin width of 0.5 cm, resulting in 100 grid points that were used in order for the interpolation of the adjoint function. The CPU time cost of the deterministic calculation has been added to that of the biased Monte Carlo cases, although it was negligible compared to the total cost of the calculation.

A series of calculations was then run. Each consisted of 550 successive cycles, with 1000 initial particles. 50 cycles were discarded as an initial estimate at the start of each calculation in order to allow for source distribution to converge. Therefore, 500 active cycles were used in the final results. In order to be able to get consistent results for the FOM, two runs with the same data and random number sequence were done per calculation. The CPU time was then averaged.

Case	k_{eff}	σ	Relative FOM
Implicit Capture	0.7702	7.9369×10^{-4}	1.00
Bin-wise constant χ^*	0.7718	3.7268×10^{-4}	1.15
Linearly interpolated χ^*	0.7709	2.7971×10^{-4}	1.73

Table 5.3: Comparison of standard deviation σ and figure of merit in biased and unbiased simulations. The deterministically calculated value of k_{eff} is 0.7712, while the figure of merit is shown relative to that of the implicit capture case.

The results can be seen in Table 5.3. Using implicit capture, we were able to obtain a result that matched very well the deterministic one, within statistics. Biasing with the bin-wise constant adjoint function resulted in a standard deviation 2 times lower, or a quadruple decrease in variance, but with an increased CPU that was also almost 4 times the implicit capture one, therefore increasing the figure of merit by around 15 %. Linearly interpolating the values gave an even larger reduction in variance with a slightly increased CPU time cost, compared to the bin-wise constant approximation.

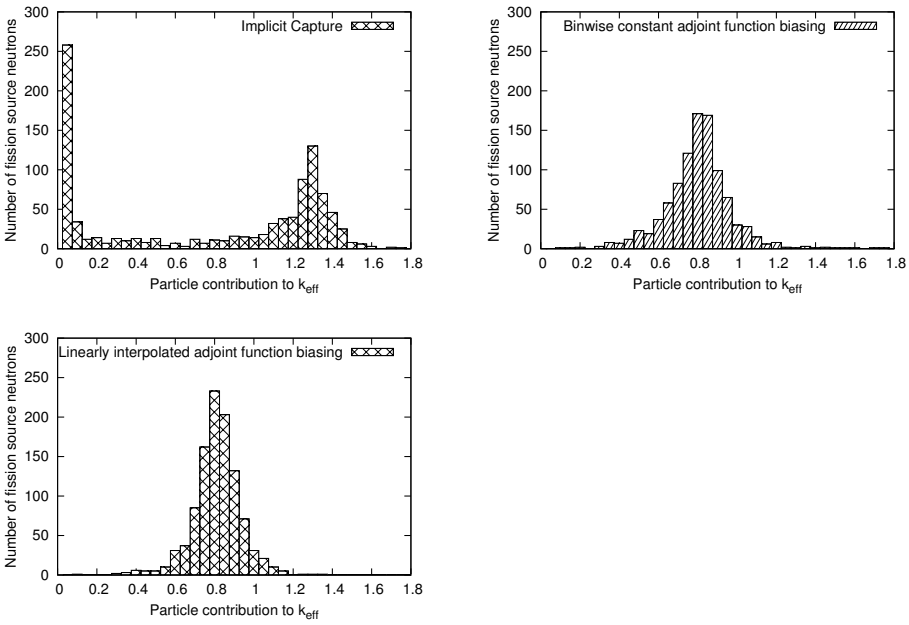

Figure 5.2: Distribution of particle contributions to k_{eff} for cycle number 550 of each calculation, divided by the initial weight.

5.4.3 Weight analysis

The spatial distribution of the particles is important, since it affects the convergence of the fission source and the eigenvalue k_{eff}. In addition, the variance of the eigenvalue in a cycle is strongly dependent on the spread of the individual particle contributions. It is therefore of great importance in the scheme to ensure that the distribution of the contributions is such that the variance in the result remains low. In the proof of the zero-variance scheme, we have already shown that, in ideal conditions, the contribution of each particle is exactly equal to the expected eigenvalue, within computational and Russian roulette threshold constrains.

We can see the distribution of the weighted contribution of the particles for the last cycle of each calculation in Fig. 5.2. Indeed, in the implicit capture case particle contributions are well spread out, with several contributions being very low or zero due to particles leaking from the system. On the contrary, the two biased cases show very similar behaviour, with the contributions concentrated closer to

the final result, leading to a smaller standard deviation in a cycle. However, one would expect that since the contributions per particle are very similar, the bin-wise constant approximation should have similar variance to the linear interpolation one. In addition, being faster, the figure of merit should be even higher in the former case. But as we have seen in Table 5.3, this is not the case. What could then be the culprit?

In order to find the answer, we should pay attention to the standard deviation per cycle, shown in Fig. 5.3. We can see that, although both biasing methods have lower

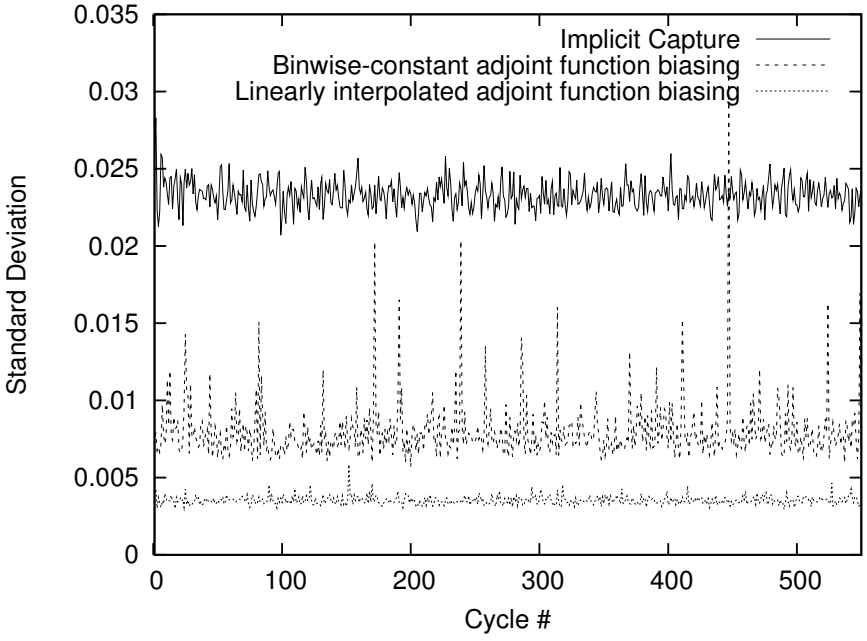

Figure 5.3: Standard deviation per cycle for the 3 cases investigated. The spikes that appear in the biased cases are caused by poor approximation of the adjoint functions leading to abnormal particle weights.

standard deviation per cycle than the implicit capture case, several "spikes" appear in the bin-wise constant approximation. These spikes are the result of differences between the exact $\psi^*(x)$ and the constant one that we have calculated per bin. If the difference between the two is large, as is the case at the edges of the system,

the weighting factors change significantly, allowing particles to gain abnormally high weight, and therefore give higher than the expected contributions to the k_{eff}. This leads to significant differences in k_{eff} between cycles, and therefore a higher variance. In the linear interpolation approximation, the weight of the particles is much better controlled, leading to an almost complete disappearance of spikes and a smoother, almost constant standard deviation per cycle.

5.4.4 Source convergence and entropy

In a Monte Carlo calculation, the fission source distribution must be converged before one can reliably begin statistical sampling for k_{eff}. Usually, k_{eff} convergence is taken as an estimate of the source convergence in order to determine the number of inactive cycles in a simulation. However, k_{eff} being an integral quantity, it can converge faster than the source. This is especially true in systems where the ratio of the first two eigenvalues, also called the dominance ratio, is close to 1. In those systems, the proximity of the two eigenvalues creates a strong correlation between cycles, making source convergence difficult to achieve(Brown, 2009).

In order to reliably diagnose whether source convergence has been achieved in a simulation, Ueki (2005) suggests the use of the the Shannon entropy of the source distribution. The Shannon entropy H is defined as

$$H_S \equiv -\sum_{i=1}^{N} p_i \times \log_2(p_i), \tag{5.30}$$

where N is the number of spatial bins we divide our geometry into and

$$p_i = \frac{\text{number of source particles in bin i}}{\text{total number of source particles}}. \tag{5.31}$$

Convergence of the Shannon entropy of the source distribution is a much more reliable criterion for fission source convergence compared to k_{eff} and an easy one to test for.

In Fig. 5.4, we can see the entropy of the source for the 3 schemes investigated. We can see that the source does not appear to be converged until after 100 cycles, although k_{eff} has converged earlier. However, more important is the behaviour of the biased calculations. In the bin-wise constant approximation, the evolution of the source is exactly the same as in the unbiased case. However, in the linear

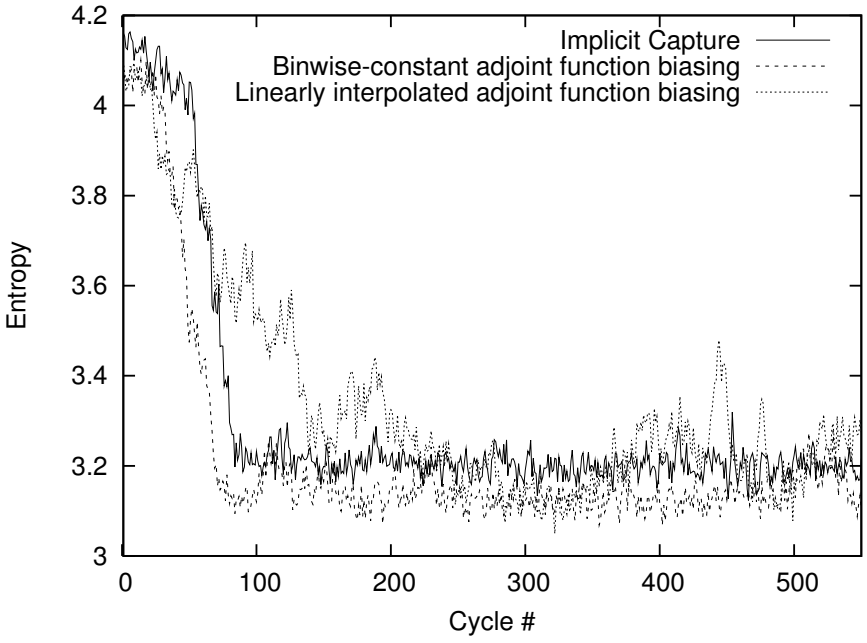

Figure 5.4: Evolution of the source entropy for the loosely coupled system with and without biasing.

interpolation approximation, we can see that there are spikes that indicate that the source might not have converged yet. In order to understand this phenomenon, we need to look at the source of the fluctuations in entropy.

In a nearly symmetric system, such as the one we are using, the value of k_{eff} is dictated by the larger fissile volume. In fact, the number of fissions taking place in the right slab is only 2 % of the total one, leading to the fission source being generated almost exclusively in the left slab. If we plot the fission fraction in the right slab throughout the calculation, as seen in Fig. 5.5, we see that in fact only in the linear approximation do we have a non-zero fraction almost throughout the calculation. It seems that the biasing scheme with the linearly interpolated adjoint function allows for particles to reach the right hand slab more often and therefore fission there. This does not seem to be the case with the other two methods, although the bin-wise constant approximation allows for a non-zero fraction in

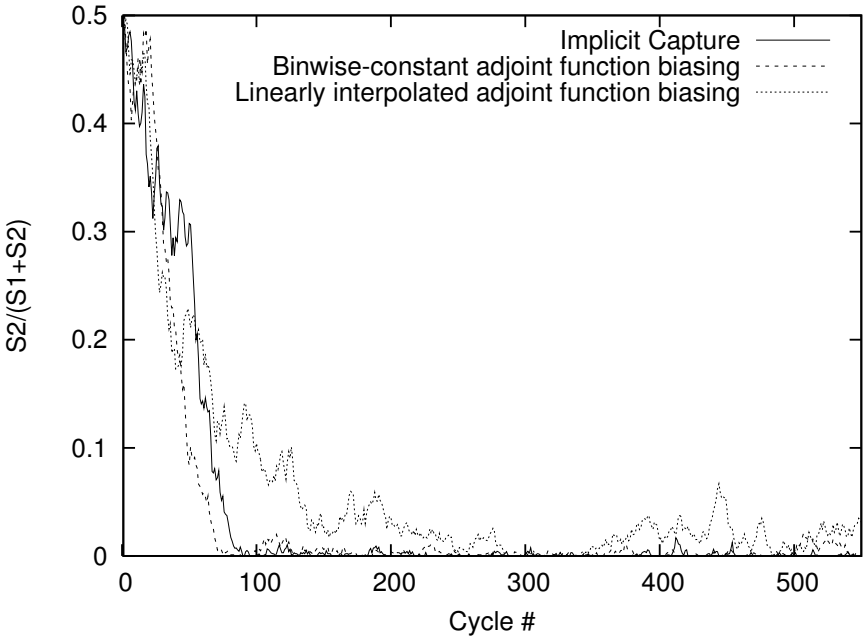

Figure 5.5: Fission fraction of the right slab of the loosely coupled system.

some stages of the calculation. However, the unpredictability of such a system does not allow for final conclusions, since only a change in random numbers is enough to trigger a different behaviour in the transport of particles from one slab to another, distorting our interpretation of the results.

5.4.5 Efficiency of the scheme

In Table 5.3, we included the relative figure of merit for the different calculations. Although in some cases (such as safety calculations) the absolute variance is of more importance, it is always desired to obtain the result in a reasonable amount of time. However, when calculating the speed of a calculation, several factors must be taken into account, some of which have no relation to the physical problems or the schemes used to simulate those problems, but are directly related to the computational implementation of the schemes.

As an example, since the code created to demonstrate the zero-variance scheme was not intended for any production use, our focus was on ease of implementation and debugging capabilities, rather than efficiency. In that aspect, a direct comparison of the efficiency between the biased and unbiased schemes is rather subjective, as a different implementation of the scheme can provide different efficiency results because of the change in CPU time. However, we can reach some general conclusions by comparing the different biasing schemes, and observing the parameters we have discussed in the previous section of this chapter.

In terms of speeding up the calculation, regardless of the implementation used, several parameters can be adjusted. Already in Fig. 4.5, we saw that the minimum plateau appears at a Russian roulette threshold of around 5×10^{-2}. It is therefore advised not to use lower thresholds, since there is minimal, if any, gain for the extra time spent.

In addition, the level and methodology of discretisation of the adjoint function play an important role in the final efficiency. Biasing with bin-wise constant adjoint function can be very efficient in terms of time, since the values are stored and retrieved from computer memory, rather than being recalculated at every step. In fact, in systems where few spatial, energy or direction variables are used, one can store the biased cumulative distributions for the sampling of kernels in advance, leading to a practically zero CPU time overhead, compared to an unbiased calculation. In a calculation where the reduction in standard deviation is an order of magnitude, this means a gain of two orders of magnitude in efficiency, compared to the unbiased case. However, one must try to correctly define the resolution of the grid: a coarser grid decreases CPU time but, as we have discussed before, could increase the variance in areas where the adjoint value differs significantly between grid points.

Linear approximation can be used with much coarser grids than the bin-wise constant approximation, since it provides more accurate values for the adjoint function, even near the grid boundaries. This can offset the CPU time disadvantage of extra calculations required, since during transition biasing, the most time-consuming part of the simulation, particles have to traverse fewer regions where a recalculation or retrieval of information is required. In addition, the algorithm used for calculation of the final location can yield a change in efficiency. As an example, using the Newton-Raphson method rather than the regula falsi method for finding the exact interaction point inside a bin yields a speed-up of around 10 %, since the number of iterations required is much smaller, even taking into account the

calculation of the function's derivative, necessary for the former method.

5.5 Concluding remarks

In this chapter we have used our zero-variance based scheme in practice, although with a simplified configuration. We have seen that the order of variance reduction in a more realistic configuration is similar to that in the two-direction model, which is a positive result for the validity of the model. In addition, the different approximations used in biasing were shown to have an effect in almost all aspects of a simulation - time, variance and convergence of the source distribution.

An important aspect to note is that one should be careful when setting the Russian roulette threshold in a biased calculation. As we have stated in the previous section, setting the threshold below 10^{-2} affects efficiency, since CPU time increases without an appropriate decrease in variance. However, we should note that the initial weight of the particles depends on the adjoint functions. Therefore, one should also be careful not to set the threshold to values higher than 10^{-1}, in which case particles may undergo Russian roulette at birth or soon afterwards, significantly affecting the amount of information obtained, and therefore the final variance.

In the next chapter we will see how we can integrate the scheme in a full-scale production code and use it in a criticality calculation.

5.6 Bibliography

R. Blomquist. OECD/NEA source convergence benchmark program: Overview and summary of results. In *ICNC2003 proceedings*, 278, 2003.

F. B. Brown. A review of Monte Carlo criticality calculations - convergence, bias, statistics. In *International Conference on Mathematics, Computational Methods & Reactor Physics, M & C 2009*, 2009.

N. M. Greene and C. W. Craven Jr. XSDRN, a discrete ordinates spectral averaging code. Technical Report ORNL-TM-2500, Oak Ridge National Laboratory, 1969.

Oak Ridge National Laboratory. SCALE, a modular code system for performing standardized computer analyses for licensing evaluation. Technical Report ORNL-TM-2005/39, Version 5.1, Vols. I-III, Oak Ridge National Laboratory, 2006.

T. Ueki. Information theory and undersampling diagnostics for Monte Carlo simulation of nuclear criticality. *Nuclear Science and Engineering*, **151**, 283, 2005.

CHAPTER 6

IMPLEMENTATION OF THE SCHEME IN A PRODUCTION MONTE CARLO CODE

The variance reduction scheme has been shown to work as expected in our own code, however, it is important to be able to use it in an everyday Monte Carlo calculation, rather than just as a theoretical possibility. In this chapter, we will discuss the implementation of the scheme in two widely-used Monte Carlo codes: TRIPOLI-4 (Both et al., 2003) and MCNP5 (X-5 Monte Carlo Team, 2003).

Generation of adjoint functions in 3-D geometry

Since the implementation of the scheme was now intended for a more realistic calculation, a deterministic code that could output adjoint functions for a 3-D geometry was required. Our code of choice was PARTISN4 (Alcouffe et al., 2005).

The PARTISN4 code solves the multigroup form of the linear Boltzmann transport equation using the S_N method. It works in 3-D (x-y-z or r-z-θ) geometries, while vacuum, reflective, periodic, white, or inhomogeneous boundary conditions can be solved. General anisotropic scattering and inhomogeneous sources are permitted. PARTISN4 was mainly selected because of its ease of use, and its proven capabilities in producing adjoint functions from source-detector calculations.

During testing of PARTISN4, we saw that the code could also successfully replicate the results from XSDRN, and since the cross-section format was simpler in PARTISN4, as cross-sections could be entered directly into the input file in ASCII format, we decided to use it exclusively for 1, 2 and 3-D geometries.

6.1 The TRIPOLI-4 code

6.1.1 Description of TRIPOLI-4

TRIPOLI-4 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 l'Energie Atomique (CEA), Saclay, France, with a recent version being publicly available from the OECD Nuclear Energy Agency (NEA) Data Bank.

TRIPOLI-4 can be run on many different computer platforms, has parallel capabilities and can handle both continuous-energy cross sections, as well as homogenized and self-shielded multigroup cross-sections from the APOLLO-2 lattice code. A graphical user interface is also available for generation and verification of input files.

The description of sources is very flexible, as various combinations of space, energy, direction and time dependent sources can be described. There is also the possibility to use analytical functions to represent the source dependence in any variable. Various standard estimators are available to calculate particle fluxes in specified volumes, 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-4 has a criticality mode for calculating the effective multiplication factor k_{eff}, as well as a perturbation option.

Finally, a for us important feature of TRIPOLI-4 is its ability to apply particle weighting schemes to reduce the variance of estimators in a shielding calculation. An importance function can be defined for the whole space, energy, direction and time domain, and is then used to control the weight of the particle by splitting or Russian roulette. An importance function can also be used to bias the transport kernel to direct particles towards a predefined position, thus providing the maximum detector response for the calculation.

6.1.2 Implementation of the scheme in TRIPOLI-4

TRIPOLI-4 is written in the C++ computer language. Our implementation of the zero-variance scheme was therefore based on a new C++ class that dealt with biasing kernels by the adjoint functions. Methods were written for the code to be able to retrieve the adjoint functions from either the exact, analytical distribution (for zero-variance testing purposes) or from discrete, tabulated data. An input method was written that read into the code adjoint data in ASCII format, matching the print output format of PARTISN4.

The source, transition and collision routines of the code were modified in order to accommodate the biasing, which could be enabled upon an option in the TRIPOLI-4 input file. As the zero-variance scheme works for a collision estimator, the code was modified in order to avoid trying to combine the output of different estimators for the final k_{eff} estimate.

In order to have consistency in the results, a set of multigroup cross-section data to match PARTISN4 data was generated and kindly provided by CEA in the GENDF format, which TRIPOLI-4 can read directly. The cross-section data for the deterministic codes was once again generated using the SCALE code system.

Initially, we implemented and tested the exact zero-variance scheme, using analytically calculated adjoint functions for biasing. The results successfully matched those of our own code. We also had success in applying collision biasing to the loosely coupled core calculation (see Sect. 5.4.1), using adjoint functions generated by PARTISN4. This was not a problem in the 1-D geometry, since we could simply set the surfaces to match grid meshes, so the code could still apply its own routines.

However, despite the assistance received from CEA, which we are grateful for, we were not able to successfully implement transition biasing with discrete adjoint functions, not even in 1-D geometry. There were numerous problems with the co-existence of an adjoint biasing grid with the geometrical structures of the code. More specifically, we had problems with tracking, both during particle simulation and adjoint function generation phases of the calculation. There were several problems in updating the cross-section data after each virtual surface crossing, as well as calculating distances to virtual surfaces, when the adjoint function mesh was involved.

Since we were not able to solve these issues in a reasonable time frame, we decided to abandon the implementation in TRIPOLI-4 and focus on the implemen-

tation in MCNP5, described in the following section.

6.2 The MCNP5 code

6.2.1 Description of MCNP5

MCNP5 is a general-purpose Monte Carlo N-Particle code, developed at the Los Alamos National Laboratory (LANL), that can be used for neutron, photon, electron, or coupled neutron/photon/electron transport. The latest version of MCNP5 can be obtained directly from the Radiation Safety Information Computational Center (RSICC). MCNP5 is arguably the most often used Monte Carlo code when it comes to particle transport. Its applications include, but are not limited to, radiation protection and dosimetry, radiation shielding, radiography, medical physics, nuclear criticality safety, detector design and analysis, nuclear oil well logging, accelerator target design, fission and fusion reactor design, decontamination and decommissioning.

The code can handle arbitrary three-dimensional configuration of materials in geometric cells, while cross-section data can be used in point-wise (continuous) or group-wise formats. It has a powerful system to describe sources, whether point, surface, volume or criticality, and tallies, whether current, flux, charge, energy or energy deposition. Geometry and tally plotting is available, while the code boasts a large array of variance reduction techniques, including an extensive system for weight window generation and application.

MCNP5 has already been used as the basis for hybrid deterministic-Monte Carlo schemes. In fact, most of the schemes discussed in Sect. 2.5.2 are applied to MCNP. We therefore considered it the prime target for application of our scheme.

6.2.2 Implementation of the scheme in MCNP5

The structure of the scheme implementation in MCNP5 was similar to that of TRIPOLI-4, with some differences due to the fact that MCNP5 is written in FORTRAN.

Since PARTISN4 was used for the generation of the adjoint functions, the binary output file containing the adjoint functions had to be processed and converted via

an in-house code to an ASCII file, to be read by MCNP5 via a special option in the input file. Since the output from PARTISN4 is that of an adjoint calculation, the directions had to be reversed in order to convert them to the equivalent MCNP5 ones, while the groups were printed out in a reverse order, which had to be taken into account. Later, an additional option was added to MCNP5, in order to be able to input the adjoint functions directly from the PARTISN4 binary format. This was necessary as using linear interpolation to calculate the adjoint functions for biasing requires the adjoint functions at several points in the grid, rather than the piecewise-constant ones, making the amount of data for a 3-D calculation significantly larger.

A number of FORTRAN subroutines were written that dealt with biasing the source and collision and transition kernels. The particle tracking routine required some changes, since we wanted the code to allow particles to be tracked all the way to the boundaries of the system, so that the biased probability table is generated. Because of that, special attention was given to treatment of boundary conditions in the code. Also, as the code allows for repeated geometrical structures, we had to take them into account since particle transport is done in the local geometrical structures, while retrieval and use of the adjoint functions is done in the global geometry. In addition, the code could easily accommodate discrete directions, using the PARTISN4 output file as an input for the number and data for those directions.

Regarding creation and use of the adjoint function mesh, the current framework in the code that deals with weight window meshes was used in the code. Compared to TRIPOLI-4, this had the advantage that the code already had routines available to retrieve or recalculate any data necessary upon entering a virtual mesh, a geometrical surface or a combination of the two.

After implementing our changes to the code, we were able to successfully run cases using the two-direction model, as well as cases using multiple directions in 1-D (loosely coupled core) and in 2-D (a 5×5 MOX cell cluster). We therefore decided to move to a more realistic calculation of a cluster of fuel assemblies, what is sometimes called a *mini-core* calculation with MCNP5.

6.3 A mini-core calculation with MCNP5

6.3.1 Description of the problem

In our mini-core calculation, a 3×3 cluster of 9 PWR fuel assemblies is used, with boundary conditions set to vacuum. Each assembly consists of a 17×17 array of heterogeneous fuel pins, control rod guide tubes with the rods fully extracted and a central instrumentation tube. For simplicity the central instrumentation tube, which normally contains measuring devices, is assumed to have the same geometry and composition as the control rod guide tube. The dimensions for both the fuel and the water hole cells are shown in Fig. 6.1.

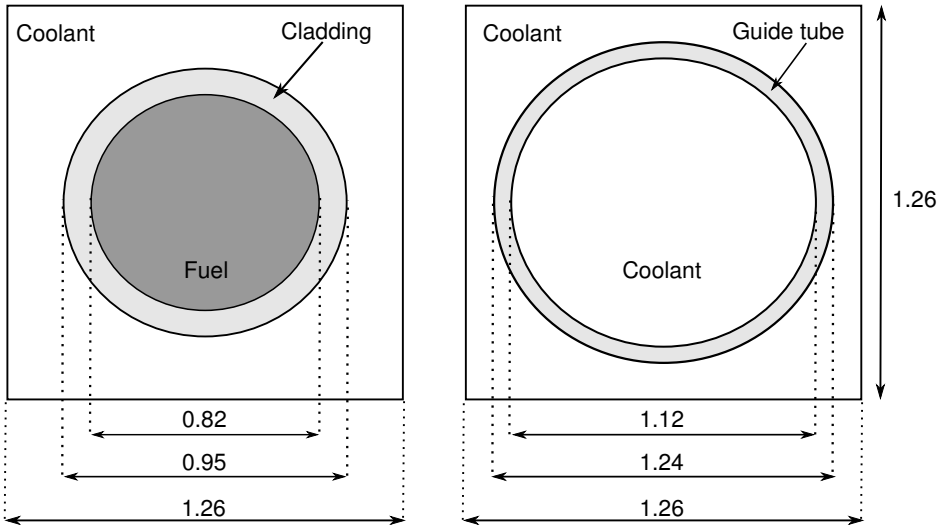

Figure 6.1: Geometry of the fuel cell (left) and the water hole (right). All values are given in units of cm.

Each MOX assembly contains fuel pins with three different plutonium enrichments, 3.2%, 5.2% and 7.80%, defined as MOX-1, MOX-2 and MOX-3, respectively, with a density of 10.25 g/cm^3. In the lattice, the 8 outer assemblies contain a mixture of MOX-1, MOX-2 and MOX-3 fuel as shown in Fig. 6.2, while the central assembly replaces all fuel types with MOX-1 type fuel.

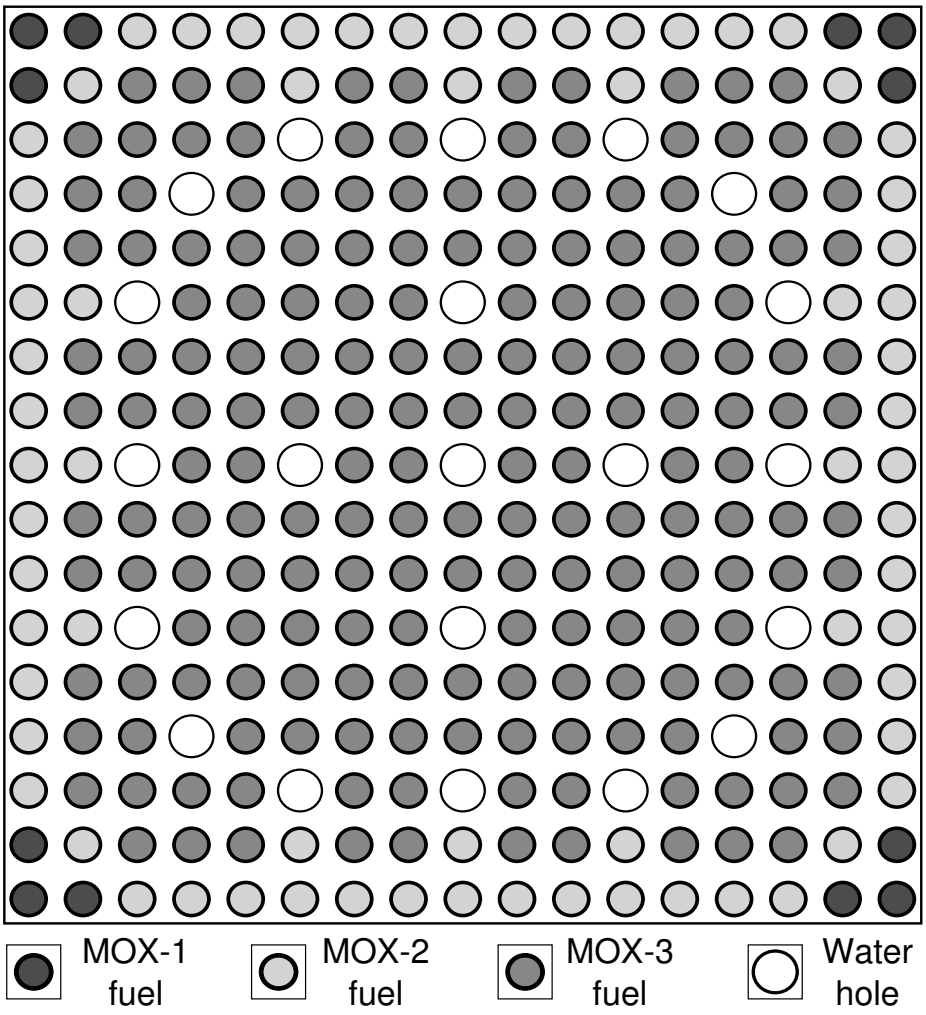

Figure 6.2: A MOX type 17×17 assembly. The light-coloured cells on the outside of the assembly contain MOX-2 type fuel, the darker-coloured ones in the centre contain MOX-3 type fuel, while the cells at the corners contain MOX-1 type fuel. Coolant between the fuel cells and inside the guide tubes is represented by white colour. In the central assembly, all three types of fuel have been replaced with MOX-1 fuel, while the location of the water holes remains the same.

The composition of Uranium and Plutonium in the MOX fuel is given in Table 6.1. For simplicity, the isotopic composition of the cladding is assumed to be natural Zr, with a density of 5.77 g/cm^3. This is also the case with the guide tubes, but with a density of 6.55 g/cm^3. The coolant in both types of cells is water with a density of 0.7164 g/cm^3, containing 500 ppm of boron with an assumed boron isotopic composition of 18 wt % ^{10}B and 82 wt % ^{11}B.

Uranium		Plutonium			
U-234	0.01%	Pu-238	2.10%	Pu-241	9.30%
U-235	0.25%	Pu-239	54.50%	Pu-242	6.40%
U-238	99.74%	Pu-240	25.00%	Am-241	2.70%

Table 6.1: Composition of MOX fuel for the mini-core problem.

Using the data given here, the geometry and composition inputs were manually generated for both MCNP5 and PARTISN4.

6.3.2 Cross-section preparation

In order to provide better comparison to PARTISN4, the cross-sections were homogenised for each different material. This was done via multiple continuous-energy MCNP5 calculations and a special program called MgXsect (Hoogenboom et al., 2007), using specific material zones for cross-section tallying, for 3 energy groups. The temperature of the fuel was 813 K, while the cladding, guide tubes and coolant were at 578 K. The upper energy boundaries of the groups were at 20 MeV, 100 keV and 0.625 eV for groups 1, 2 and 3, respectively.

Since PARTISN4 would only be used for the generation of adjoint functions, we chose to use simplified, homogenised cells for the deterministic calculation. Six different types of cells were therefore created: 3 types for the MOX fuel in the outer assemblies, a water hole / guide tube cell, plus the water hole and MOX-1 fuel cell in the central assembly. The cross-section data was again generated using the same MCNP5/MgXsect calculation, then converted and used as direct ASCII input in PARTISN4, since it provides the option.

Because of the way MCNP5 uses the cross-sections, when generating them there was an issue with $(n,2n)$ reactions, because of which the sum of all interaction cross-sections was higher than Σ_t. In order to solve that problem and because the

probability for $(n,2n)$ was minute, we took it into account implicitly, by setting that cross-section to zero and then increasing the weight of a particle at each collision by the $(n,2n)$ probability.

The tables containing the cross-section data for the problem can be found in Appendix A.

6.3.3 Generation of adjoint functions

In order to generate adjoint functions using PARTISN4, we had to run the code in adjoint source-detector mode, using a special option in the code in order to remove fission (and therefore get the correct importances per source particle). The adjoint source was then normalised to $\nu\Sigma_f$ of the forward k_{eff} calculation:

$$S^*_{\text{tot}} = \sum_g \int \nu\Sigma_f(g) dV \tag{6.1}$$

Note that the source term in the adjoint integro-differential equation solved by PARTISN4 is not η_ψ but η_ϕ, with the relation between the two given in Eq. (1.24).

A coarse mesh of one bin per cell was generated, since we wanted to test the performance of the scheme when rather imprecise adjoint functions are used for biasing. Running PARTISN4 generated directional and scalar adjoint functions for all 3 groups. We can see the output for the scalar, group dependent χ^* in Fig. 6.3. The CPU time cost for this calculation was extremely low, as it was completed in less than 2 seconds, on a modern dual-core Intel CPU.

Despite the fact that $\nu\Sigma_f$ is higher for the MOX-3 type fuel, we can see that the value of the adjoint function is higher in the central assembly and at the assembly interfaces, where only MOX-1 and MOX-2 are present. While the higher value in the central assembly is somewhat expected, the one at the interfaces seems counter-intuitive. A possible explanation is that at the assembly interfaces, the neighbouring assembly has an importance as high as the current assembly, therefore the adjoint function values corresponding to the directions away from the assembly are also high. This compares well with the value of the scalar adjoint function at the outer boundaries, where it is no longer higher than in the inner parts of the assembly. In fact, because of the zero values of the adjoint function in the directions leading outside of the system, the value of the scalar adjoint function drops significantly.

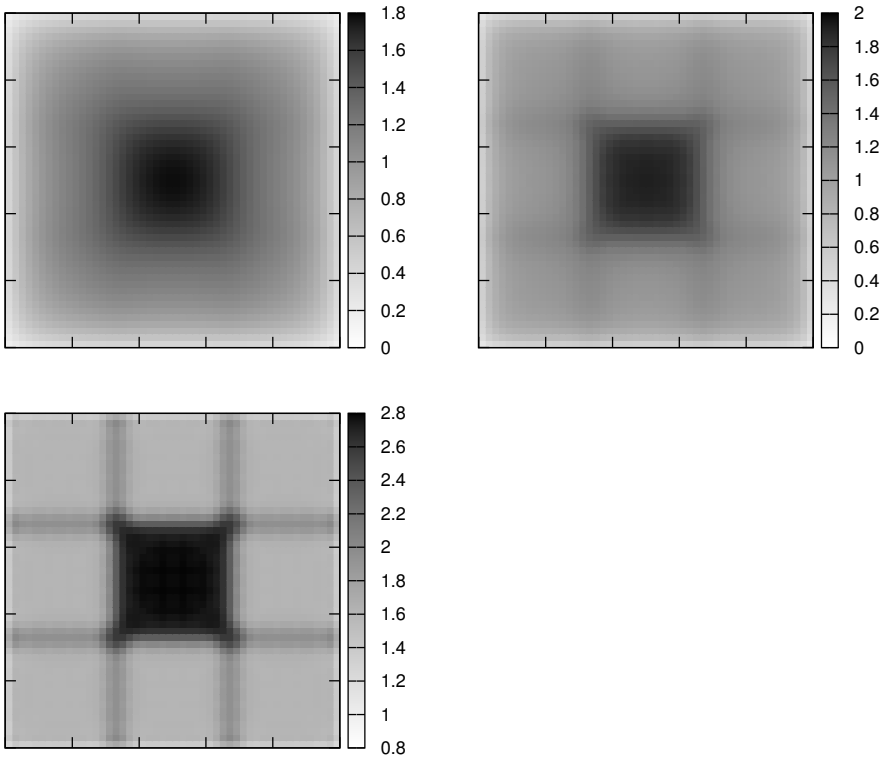

Figure 6.3: Scalar $\chi^*(g)$ for energy group g=1 (top left), 2 (top right), and 3 (bottom left). The higher values in group 3 reflect the importance of fissions taking place in that energy group.

6.4 Results

6.4.1 k - eigenvalue

The calculation was run for a total of 1000 active cycles, with 50 initial (inactive) cycles to allow for the fission source to converge. 2000 particles per cycle were used, while the Russian roulette survival weight was set to 0.02, resulting in a threshold weight of 0.01. The results can be seen in Table 6.2, while in Fig. 6.4, we can see the evolution of k_{eff} for the 1000 active cycles of the calculation.

Type of calculation	k_{eff}	st. dev.
Implicit capture (no biasing)	0.98978	0.00035
Biasing with piecewise-constant adjoint function	0.98914	0.00023
Biasing with linearly-interpolated adjoint function	0.98908	0.00022

Table 6.2: Calculation results for the mini-core system for 1000 active cycles of 2000 particles with a Russian roulette threshold weight of 0.01.

Both unbiased and biased calculations converge rather quickly, even for the variation shown in the first hundred active cycles, if we take into account the scale of the plot. Also, the results are all within 3σ of each other, which indicates that there were no errors (which would be apparent in the particle contributions) in the implementation of either biasing method.

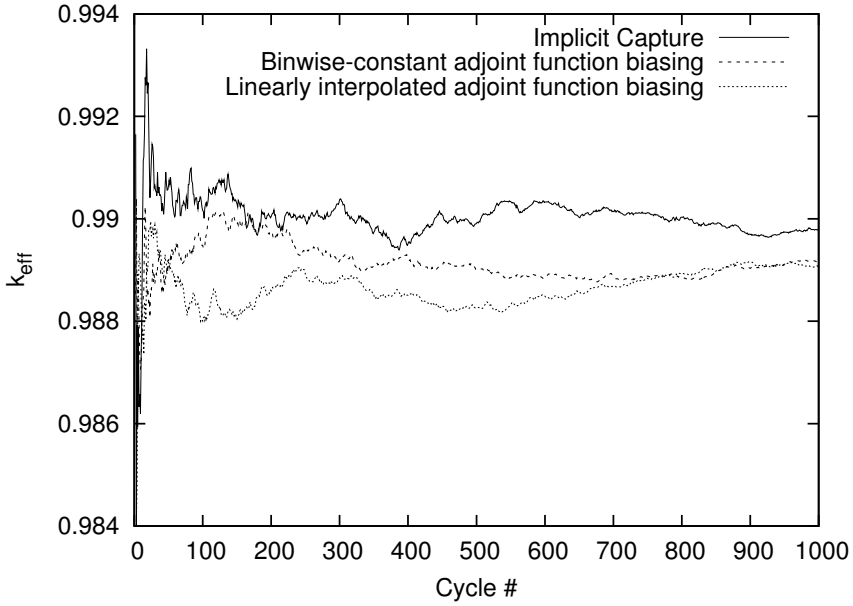

Figure 6.4: k_{eff} during the active cycles for the different methods used in the mini-core calculation. According to the MCNP code output, convergence has already been achieved at the first active cycle.

6.4.2 Variance

It is more useful to look at the evolution of the standard deviation of the calculation during the active cycles, shown in Fig. 6.5, where we see that the biasing scheme performs consistently better than the implicit capture scheme. If a 10^{-3} relative

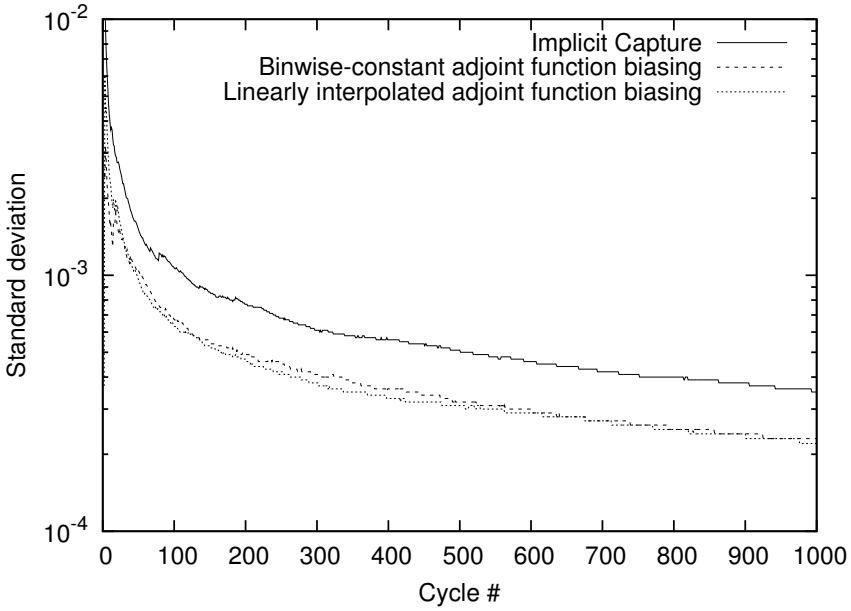

Figure 6.5: Standard deviation during the active cycles for the different methods used in the mini-core calculation. Note the logarithmic scale used in the representation of standard deviation.

standard deviation was required, the biased calculation would achieve it in exactly 50 cycles, while the implicit capture one requires almost 3 times as many.

It is interesting to see that in this case, the two different types of interpolation produce almost identical results, unlike the results shown in Fig. 5.3 for the loosely coupled core system. This can be explained by the fact that, during the calculation of the adjoint functions, the cells are homogenised. That way, the adjoint function is not significantly different at the boundaries of each cell, which means that linear interpolation is not much more accurate, compared to the piecewise-constant one.

However, since the extra CPU time necessary for calculation of the linearly interpolated adjoint functions was not significant, our suggestion would be to always use this type of implementation. One should only fall back to the piecewise-constant adjoint functions when the number of mesh cells is so large that the recalculation of the adjoint functions becomes a problem.

6.4.3 Efficiency

We earlier discussed the reduction in variance the scheme resulted in, but not of any increase in the efficiency of the calculation. This is because we found that our implementation made the code run extremely slow when running biased calculations. In fact, testing the time of the MCNP5 calculation to that of our own code, using the loosely coupled core as a testing system, we saw that MCNP5 was over an order of magnitude slower.

Of course, this was not performed as a direct comparison of the code systems, since our code was built just for showcasing the biasing scheme, while MCNP5 has a significantly broader range of applications. However, it was able to show us that our implementation in MCNP5 was certainly lacking optimisations that could bring the performance of the biasing scheme on par with the current capability of the code.

By profiling the code, it is quite obvious that most of the time is spent during the tracking of the particle throughout the geometry. This is even more pronounced in the biased calculations, where the particle must be tracked all the way to the boundaries of the system (or, theoretically, to infinity if the system uses reflective boundaries). In our scheme, we chose to stop particle tracking after 10 mean free paths, since the probability of the particle to reach any further without interactions approaches zero.

In addition, we noticed that whenever a particle would cross a virtual surface, the code immediately had to search for both the cell and adjoint function mesh the particle was now in, in order to be able to obtain the material data and the adjoint functions, both necessary for the biasing process.

As our main interest lied in showcasing the reduction in variance obtained by the scheme, we did not try to perform further optimisations towards the efficiency of the coding. Some pointers and suggestions for that are given in the next chapter.

6.4.4 Source convergence

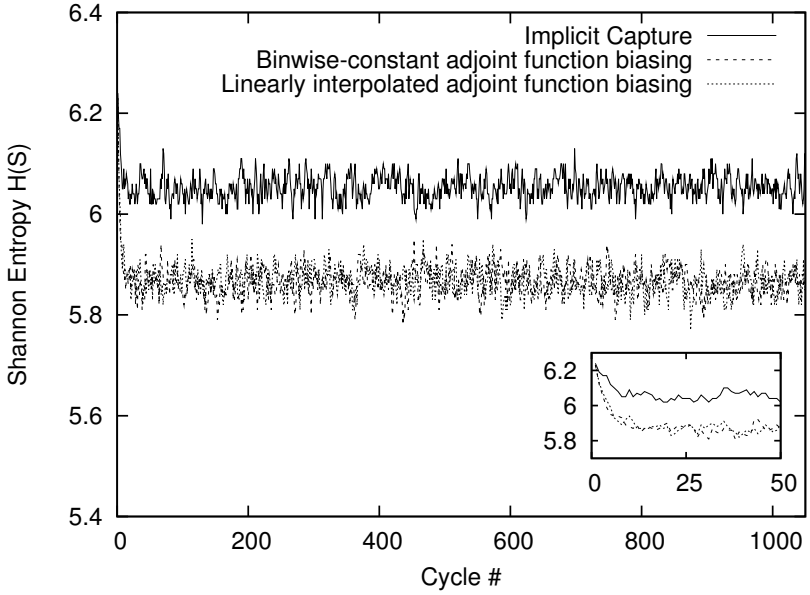

Figure 6.6: Shannon entropy of the fission source. The smaller inset figure shows the initial 50 (inactive) cycles of the calculation magnified. Note that the two cases where biasing is used strongly overlap.

MCNP5 automatically calculated the Shannon entropy of the source, with the results for all 1050 cycles shown in Fig. 6.6.

It is clearly seen that the evolution of the source distribution does not differ between the biased and unbiased calculations. This indicates on one hand that the scheme does not offer a speed-up in the source convergence for this system, but on the other hand biasing with the adjoint functions does not adversely affect the fundamental behaviour of the calculation.

This is in contrast to the loosely coupled system, where the behaviour of the calculation did not follow the same patterns when biasing was used, but as explained earlier, that system is a special case, while in general the biasing results reflect the ones we see in this mini-core system.

6.5 Conclusion

In this chapter we showed how to implement the scheme in a production code, in order to be able to run practical nuclear reactor criticality problems. By implementing the biasing scheme in the MCNP5 Monte Carlo code and running a mini-core calculation with it, we demonstrated that the reduction in variance still exists for practical nuclear reactor calculations. In addition, the source distribution convergence is not affected by the scheme. However, since the code lacked optimisations for speed, we were not able to demonstrate an appropriate increase in the efficiency of the calculation, since despite the halving in variance, there was significantly higher CPU time cost.

6.6 Bibliography

R. E. Alcouffe et al. PARTISN: A time-dependent, parallel neutral particle transport code system. Technical Report LA-UR-05-3925, Los Alamos National Laboratory, 2005.

J. P. Both et al. User manual for version 4.3 of the TRIPOLI-4 Monte Carlo method particle transport computer code. Technical Report CEA-R-6044, CEA Saclay, 2003.

J. E. Hoogenboom, V. A. Khotylev and J. M. Tholammakkil. Generation of multigroup cross sections and scattering matrices with the Monte Carlo code MCNP5. In *Joint International Topical Meeting on Mathematics and Computations and Supercomputing in Nuclear Applications, M and C + SNA 2007*, 2007.

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, 2003.

CHAPTER 7

CONCLUSIONS & RECOMMENDATIONS

7.1 Final conclusions and remarks

Zero variance theoretically achievable for criticality

In this thesis we devised a zero-variance scheme for a criticality calculation and proved its existence. Using a two-direction model and biasing using analytically calculated source and adjoint functions, the variance of the calculation would approach zero, the only limit being the Russian roulette threshold, dictated by computational limits. It was shown that zero variance cannot be obtained in practice, due to the need for renormalisation and averaging of particle weights at the beginning of each successive generation of neutrons. However, a significant reduction in variance would occur even using approximate adjoint functions that have been obtained computationally.

Performance of the scheme in a simple system

The scheme was tested in an own code using a loosely coupled system configuration with discrete directions and a group-wise configuration in energy. The result showed that the order of variance reduction in a more realistic configuration is

similar to that in the two-direction model, a positive result for the validity of the model.

Different approximations for the adjoint function were shown, with the adjoint function generated by linear interpolation of discrete values produced by a deterministic calculation resulting in almost double the figure of merit, compared to an implicit capture calculation. In addition, the different approximations used in biasing were shown not to have negative effects in any aspects of a calculation - time, variance and convergence of the source distribution.

Scalability to a larger test problem

Finally, we tested the scheme in a practical Monte Carlo calculation, that of a mini-core configuration. In order to emulate its use in a production environment, an attempt to implement the scheme in a production Monte Carlo code was made. We were only able to succeed for limited, 1-D geometry in the TRIPOLI-4 code, but we succeeded in using the scheme for a calculation of a mini-core using the MCNP5 code, with the adjoint functions generated using the deterministic PARTISN4 code. The results confirmed that the scheme leads to a halving in variance even for a more complex system, but at the cost of significantly higher CPU usage. This leads to the scheme not providing the expected increase in figure of merit, something that can be corrected by optimizing the implementation in the code.

7.2 Recommendations for further work

Improving the variance of the average initial weight

As we have seen in chapter 3 of this thesis, the main reason why the zero-variance scheme will not work to its full extent in a multiple cycle simulation is because of the residual variance in the initial weight of the fission source. We have already made several attempts to decrease that variance, such as averaging the initial neutron weight before selecting energy and angle, and then using an expected value estimator for the average initial weight of the source particles. However, there should be more methods or tricks that will allow us to further reduce the variance, while still keeping the final estimate unbiased.

In fact, during testing of the scheme in the MCNP5 code, a rather strange effect was observed: artificially raising the values of the adjoint function by around 20% would bring improvements in the variance of k_{eff}, despite actually increasing the variance per cycle (as it should, since the scheme is not working optimally). This was despite verifying that the adjoint function was normalised properly, and thus that all input data was correct. We were not able to explain this phenomenon, but it is a clear indication that we have not reached the floor of variance in a calculation, but there are still gains to be explored.

Proving the scheme for different estimators

In this thesis, we have proven that there is a zero-variance scheme in a criticality calculation, for a collision estimator. However, as Hoogenboom (2008) has proven for source-detector problems, there is a unique zero-variance scheme for each possible estimator in a Monte Carlo simulation. Therefore, it will be of interest to prove that a zero-variance scheme using a different estimator, such as a track-length estimator, exists also for for criticality. The methodology should in theory remain the same: starting with an unbiased estimate of the score throughout the length of its track, it should be possible to bias the sampling in a way that the initial estimate with zero variance is obtained.

Using continuous distributions for energy and direction selection in a biased calculation

Although we derived the scheme in a most general way, the implementation was done using discrete directions and energy groups. When it comes to everyday calculations, especially safety ones, it is important to be able to use the most accurate data at one's disposal. The next step should therefore be implementing the scheme using continuous distributions in energy and direction during selection from source or scattering events.

Energy selection should be rather straightforward: the initial energy of a particle or its energy after collision is sampled using the same methods as in the unbiased case, but all probabilities are biased by the appropriate adjoint for the group that the energy corresponds to.

Direction selection is more involved, since both the cosine of scattering angle

μ and azimuthal angle ϕ have to be selected after each event. Initially, the cosine of scattering angle μ is calculated in the centre of mass frame of reference, but biased by an adjoint with direction cosines in the laboratory frame of reference, hence the correct transformations should take place. Even then, it is important to note that since the scattering angle will never coincide with a discrete direction from the deterministic set, an algorithm for finding the closest discrete direction to the continuous one and applying biasing based on that should be created. Finally, after μ has been selected, the new ϕ must be selected. This means that the sets of discrete μ, ϕ must be put in a way that the whole 4π sphere is covered, something far from trivial.

Anisotropy in scattering, something that has been left out of our implementation for simplicity reasons, should also be taken care of. Although it was not included in the examples for practical reasons, mainly for ease in cross-section generation, the theory accommodates it without issues.

Improving performance via better implementation in code

As we discussed earlier in this thesis, although the biasing scheme results in a lower variance for a calculation, the computational time spent on the biasing routines can render the scheme useless, if the total figure of merit of the calculation is lowered. It is therefore of high importance to ensure that the implementation in the Monte Carlo code is strongly optimized.

As we have already discussed, tracking particles through the geometry is especially time-consuming. An adaptive particle tracking scheme, where the biased transition kernel and the random number are taken into account before determining the cut-off point for the generation of the biased probability table, might be a better option in order to ensure that all particles are tracked properly and without unnecessary CPU cost.

Another optimisation would be to arrange the code in a way that during the initialisation phase, the geometry and the adjoint mesh are combined, so that each cell is now a geometrical union of the Monte Carlo geometrical cell and the adjoint mesh. That way, the code does not have to recalculate the values of the adjoint function after all mesh or geometrical surface crossings, but they can now be attached to the virtual cell and recalled from memory when needed. The number of crossings might be larger, but since no recalculations are needed, there should be a

considerable boost in speed, at the expense of a modest (and certainly worthwhile) increase in computer memory usage.

As a general suggestion regarding performance, an integrated solution (where the same code system is used for generating the adjoint function and using them in a Monte Carlo calculation) is a much better alternative to the implementation in a current code. Most current Monte Carlo codes would require significant changes in order to accommodate the adjoint biasing grid, while significant work has to go into preparation of separate inputs for the Monte Carlo and deterministic codes, generating the adjoint functions and then converting them to the proper format for the Monte Carlo code. Such a solution is currently used in the MAVRIC code, part of the SCALE6 code system, but only for shielding calculations. It would therefore be of interest to extend such a code system to criticality calculations using the scheme we have discussed in this thesis.

7.3 Epilogue

Monte Carlo schemes have come a long way since their first practical implementation, over 60 years ago. The combined efforts of scientists from all over the world in theoretical schemes have given Monte Carlo codes a tremendous boost in applicability, backed by a large increase in raw CPU power, which the method inherently depends on. The current trend towards multiple-CPU, parallel environments, is especially suited to the Monte Carlo method and further extends the improvement.

Reading the above, one could think that the simplest thing to do would be to sit back and wait for the inevitable progress in computing power, which will allow for faster and more accurate Monte Carlo calculations, in shorter time than ever before. However, as we have also seen in this thesis, there is still room for improvement in the theoretical front, which is applied to, and can take advantage of, any improvements in computational power in the future. Improved as it might be, an analogue Monte Carlo calculation will always depend on the number of samples in order to yield results, while as we have seen in this thesis, biased schemes could achieve the same results with much fewer, theoretically even one, samples. Being able to get results with near-zero variance from a calculation as exact (in terms of physics) as a Monte Carlo one would give a large boost in research on safety design of new types of nuclear reactors, and not only in a scientific sense. Being able to show to the public that we can predict the behaviour of a system as feared as a

nuclear reactor with almost zero error margins would further renew the interest that mainly economical reasons have sparked the last few years.

That is not to say, of course, that such an advancement in Monte Carlo would make deterministic methods obsolete. Monte Carlo codes are still in an early state when concerning calculations that deterministic codes have been able to perform for quite some time, such as time-dependent calculations or calculations with external, usually thermohydraulic feedback. In addition, the advantage of deterministic calculations in computational speed is still very large, and their ability to provide reliable results in systems where large amount of information is required at many places, like in full core calculations where pin-by-pin data may be required. This is where hybrid methods come into play: by being able to use data from fast but sometimes simplistic deterministic calculations in order to speed up the Monte Carlo calculation, they can improve it to the point where Monte Carlo is already considered an indispensable tool for research, and not only a benchmarking tool against the latest trend in deterministic methods.

The future for Monte Carlo simulations in nuclear physics looks bright, but it should not be left standing on the shoulders of the computational advancements of our era. Continuing work in the theoretical field, being able to edge ever closer to an estimate with zero variance, will immensely help the development of Monte Carlo tools for simulations and will hopefully further contribute to the nuclear renaissance of our times.

7.4 Bibliography

J. E. Hoogenboom. Zero-Variance Monte Carlo schemes revisited. *Nuclear Science and Engineering*, **160**, 1, 2008.

APPENDIX A

CROSS-SECTION DATA FOR THE MINI-CORE CALCULATION

The cross-section data used for the MCNP5 calculation of the mini-core can be found in the following tables. All cross-sections are given in units of cm^{-1}, while ν and χ_f are unitless.

Tables A.1 and A.2 contain the data used in the PARTISN4 calculation for generation of the adjoint functions. Note that, as described in the thesis, fission is not allowed in the PARTISN4 calculation, so $\nu\Sigma_f$ is not actually used, but only mentioned here for the sake of completeness.

Tables A.3 and A.4 contain the MCNP5 cross-section data for the central fuel assembly of the mini-core geometry, while tables A.5 and A.6 contain the data for the outer fuel assemblies of the geometry.

MOX-1 fuel cell				MOX-2 fuel cell			
Group	Σ_t	Σ_a	$\nu\Sigma_f$	Group	Σ_t	Σ_a	$\nu\Sigma_f$
1	0.31159	0.00285	0.00570	1	0.31050	0.00308	0.00642
2	0.78109	0.02334	0.00872	2	0.78510	0.02771	0.01358
3	1.11250	0.18332	0.28129	3	1.19230	0.24748	0.39320

MOX-3 fuel cell				Water hole & Guide tube			
1	0.31070	0.00338	0.00736	1	0.30948	0.00022	0.00000
2	0.79115	0.03242	0.01880	2	0.92060	0.00102	0.00000
3	1.27470	0.31907	0.51714	3	1.19630	0.01743	0.00000

MOX-1 fuel cell (central FA)				Water hole & Guide tube (central FA)			
1	0.31287	0.00286	0.00572	1	0.31144	0.00022	0.00000
2	0.78229	0.02479	0.00907	2	0.92281	0.00105	0.00000
3	1.12090	0.18248	0.27997	3	1.22300	0.01876	0.00000

Table A.1: Cross-section data in cm^{-1} for the PARTISN4 calculation of the homogenised cells, used for generation of the adjoint functions. Unless otherwise noted, the data is given for the outer fuel assemblies of the geometry.

MOX-1 fuel cell				MOX-2 fuel cell			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.26722	0.04152	0.00000	1	0.26650	0.04092	0.00000
2	0.00000	0.72994	0.02781	2	0.00000	0.73021	0.02718
3	0.00000	0.00081	0.92841	3	0.00000	0.00092	0.94390

MOX-3 fuel cell				Water hole & Guide tube			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.26616	0.04117	0.00000	1	0.24601	0.06325	0.00000
2	0.00000	0.73236	0.02637	2	0.00000	0.87631	0.04328
3	0.00000	0.00100	0.95462	3	0.00000	0.00071	1.17810

MOX-1 fuel cell (central FA)				Water hole & Guide tube (central FA)			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.26782	0.04219	0.00000	1	0.24697	0.06425	0.00000
2	0.00000	0.72793	0.02957	2	0.00000	0.87389	0.04788
3	0.00000	0.00064	0.93778	3	0.00000	0.00055	1.20360

Table A.2: $\Sigma_{g' \to g}$ in cm^{-1} for the PARTISN4 calculation of the homogenised cells, used for generation of the adjoint functions. Here, g' is the incoming and g the outgoing energy group. Unless otherwise noted, the data is given for the outer fuel assemblies of the geometry.

Material	Group	Σ_t	Σ_c	Σ_f	ν	χ_f
MOX-1 fuel	1	0.32741	0.00207	0.00582	2.85363	2.85734
	2	0.55037	0.06420	0.00982	2.84609	0.01265
	3	0.98879	0.25276	0.34451	2.85734	0.00000
MOX-1 cladding	1	0.24849	0.00037			
	2	0.30163	0.00241			
	3	0.24825	0.00454			
MOX-1 coolant	1	0.31585	0.00019			
	2	1.01460	0.00074			
	3	1.35082	0.02010			
Guide tube	1	0.28279	0.00046			
	2	0.34233	0.00281			
	3	0.28214	0.00539			
Coolant inside guide tube	1	0.31638	0.00019			
	2	1.01618	0.00076			
	3	1.36517	0.02069			
Coolant outside guide tube	1	0.31667	0.00019			
	2	1.01647	0.00077			
	3	1.36804	0.02081			

Table A.3: MCNP5 cross-section data of the materials in the central fuel assembly. Cross-sections are given in units of cm^{-1}.

MOX-1 fuel				MOX-1 cladding			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.31531	0.00421	0.00000	1	0.24689	0.00123	0.00000
2	0.00000	0.47521	0.00114	2	0.00000	0.29889	0.00034
3	0.00000	0.00077	0.39076	3	0.00000	0.00053	0.24317

MOX-1 coolant				Guide tube			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.24128	0.07438	0.00000	1	0.28088	0.00148	0.00000
2	0.00000	0.96173	0.05213	2	0.00000	0.33912	0.00040
3	0.00000	0.00061	1.33010	3	0.00000	0.00053	0.27622

Coolant inside guide tube				Coolant outside guide tube			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.24129	0.07490	0.00000	1	0.24152	0.07496	0.00000
2	0.00000	0.96052	0.05490	2	0.00000	0.96039	0.05531
3	0.00000	0.00056	1.34391	3	0.00000	0.00054	1.34669

Table A.4: $\Sigma_{g' \to g}$ in cm^{-1} of the central fuel assembly materials used in the MCNP5 calculation, where g' is the incoming and g the outgoing energy group.

Material	Group	Σ_t	Σ_c	Σ_f	ν	χ_f
MOX-1 fuel	1	0.32732	0.00207	0.00582	2.85533	0.98735
	2	0.54733	0.06051	0.00950	2.84630	0.01265
	3	0.98835	0.25419	0.34292	2.85760	0.00000
MOX-1 cladding	1	0.24809	0.00037			
	2	0.30389	0.00245			
	3	0.24798	0.00433			
MOX-1 coolant	1	0.31438	0.00019			
	2	1.01227	0.00069			
	3	1.33020	0.01909			
MOX-2 fuel	1	0.32678	0.00206	0.00645	2.88513	0.98735
	2	0.56136	0.06830	0.01428	2.86203	0.01265
	3	1.28921	0.37475	0.52439	2.86672	0.00000
MOX-2 cladding	1	0.24771	0.00037			
	2	0.30375	0.00241			
	3	0.24785	0.00424			
MOX-2 coolant	1	0.31391	0.00019			
	2	1.01183	0.00068			
	3	1.32279	0.01874			
MOX-3 fuel	1	0.32652	0.00206	0.00724	2.91757	0.98634
	2	0.57868	0.07754	0.02021	2.87176	0.01266
	3	1.66398	0.52606	0.74928	2.87185	0.00000
MOX-3 cladding	1	0.24736	0.00037			
	2	0.30368	0.00241			
	3	0.24779	0.00418			
MOX-3 coolant	1	0.31380	0.00019			
	2	1.01191	0.00068			
	3	1.31885	0.01855			
Guide tube	1	0.28165	0.00042			
	2	0.34346	0.00278			
	3	0.28166	0.00500			
Coolant inside guide tube	1	0.31449	0.00019			
	2	1.01329	0.00070			
	3	1.33351	0.01917			
Coolant outside guide tube	1	0.31497	0.00019			
	2	1.01385	0.00071			
	3	1.33930	0.01943			

Table A.5: MCNP5 cross-section data of the materials in the outer fuel assemblies. Cross-sections are given in units of cm^{-1}.

MOX-1 fuel				MOX-1 cladding			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.31514	0.00428	0.00000	1	0.24652	0.00120	0.00000
2	0.00000	0.47626	0.00105	2	0.00000	0.30113	0.00031
3	0.00000	0.00102	0.39022	3	0.00000	0.00070	0.24295

MOX-1 coolant				MOX-2 fuel			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.24080	0.07339	0.00000	1	0.31410	0.00417	0.00000
2	0.00000	0.96377	0.04781	2	0.00000	0.47785	0.00094
3	0.00000	0.00074	1.31036	3	0.00000	0.00109	0.38898

MOX-2 cladding				MOX-2 coolant			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.24613	0.00122	0.00000	1	0.24037	0.07335	0.00000
2	0.00000	0.30105	0.00029	2	0.00000	0.96448	0.04667
3	0.00000	0.00072	0.24290	3	0.00000	0.00081	1.30323

MOX-3 fuel				MOX-3 cladding			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.31308	0.00414	0.00000	1	0.24578	0.00121	0.00000
2	0.00000	0.48004	0.00089	2	0.00000	0.30099	0.00028
3	0.00000	0.00124	0.38740	3	0.00000	0.00077	0.24284

MOX-3 coolant				Guide tube			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.24043	0.07318	0.00000	1	0.27982	0.00142	0.00000
2	0.00000	0.96496	0.04627	2	0.00000	0.34033	0.00034
3	0.00000	0.00085	1.29946	3	0.00000	0.00069	0.27596

Coolant inside guide tube				Coolant outside guide tube			
$g' \setminus g$	1	2	3	$g' \setminus g$	1	2	3
1	0.24066	0.07364	0.00000	1	0.24070	0.07407	0.00000
2	0.00000	0.96366	0.04892	2	0.00000	0.96351	0.04963
3	0.00000	0.00075	1.31359	3	0.00000	0.00071	1.31916

Table A.6: $\Sigma_{g' \to g}$ in cm^{-1} of the outer fuel assembly materials used in the MCNP5 calculation, where g' is the incoming and g the outgoing energy group.

Summary

The ability of the Monte Carlo method to solve particle transport problems by simulating the particle behaviour makes it a very useful technique in nuclear reactor physics. However, the statistical nature of Monte Carlo implies that there will always be a variance associated with the estimate obtained. One of the main targets of research in Monte Carlo is to decrease this variance as much as possible, without heavily taxing the computational time of the calculation. Towards this target of variance reduction, most Monte Carlo simulations today include some form of biasing.

Since the adjoint function can be used as a measure of the importance of a region in the spatial, angular and energy domains of the calculation, it is frequently used as a biasing function for the calculation, in order to direct the simulated particles towards the most important regions of the simulation and reduce the variance of the estimate in those regions. In fact, it has been proven that using the appropriate biasing, one can get the variance of the Monte Carlo estimate down to zero, at least in theory. However, it is not straightforward how to reduce the variance in a criticality calculation, where the reactor serves both as the source and "detector" of particles simulated, therefore having a global problem that cannot easily be localised.

In this thesis we devised a zero-variance scheme for a criticality calculation. Treating the criticality problem as a source-detector one, we showed that, when using a collision estimator and biasing the transport kernels by the adjoint function, with an appropriate manipulation of particle weights, it is possible to reach an estimate with zero variance. A simplified two-direction model was used for demonstration, proving that when biasing using analytically calculated source and adjoint functions, the only limit to the reduction of variance of the estimate obtained from a single neutron generation is the Russian roulette threshold. This threshold can in theory get down to zero, therefore reducing the variance to zero, but in practice it is dictated by computational limits.

However, zero variance cannot be obtained in a criticality calculation in practice, due to the need for renormalisation and averaging of particle weights at the beginning of each successive neutron generation. Still, a halving in variance was observed using the two-direction model, even when using approximate adjoint functions that have been obtained computationally.

The scheme was also tested in an own code using a 1-D loosely coupled system configuration with discrete directions and a group-wise configuration in energy. The results showed that the order of variance reduction in a more realistic configuration is similar to that in the two-direction model. Different approximations for the adjoint function were shown, with the adjoint function generated by linear interpolation of discrete values produced by a deterministic calculation resulting in almost double the figure of merit, compared to an implicit capture calculation.

In order to emulate its use in a production environment, an attempt to implement the scheme in a production Monte Carlo code was made. We were only able to succeed for limited, 1-D geometry in the TRIPOLI-4 code, which we attempted first, as the structure of the code did not allow us to implement the full scheme. Therefore, we attempted to implement it in the MCNP5 code. We were successful, and using the adjoint function values generated by the deterministic PARTISN4 code, we tested the scheme in a practical Monte Carlo calculation for a 17x17 fuel cell "mini-core" configuration.

The variance in the estimate was half that of an implicit capture case, indicating that the scheme can be useful for practical nuclear reactor calculations. In addition, the convergence of the source distribution was not adversely affected by the biasing. However, since the code lacked optimisations for speed, we were not able to demonstrate an appropriate increase in the efficiency of the calculation, since there was significantly higher CPU time cost. This leads to the scheme not providing the expected increase in figure of merit, something that can be corrected by optimizing the implementation in the code.

Samenvatting

De Monte-Carlomethode is een erg nuttige techniek in de kernreactorfysica, omdat deeltjestransportproblemen hiermee kunnen worden opgelost door middel van het simuleren van het gedrag van de deeltjes. Echter de statistische aard van Monte Carlo zorgt er voor dat een schatter ook altijd een variantie heeft. Eén van de hoofddoelen van Monte-Carlo-onderzoek is het reduceren van deze variantie, zonder dat dit te veel ten koste gaat van de rekentijd. De meeste Monte-Carlosimulaties gebruiken tegenwoordig een vorm van *biasing* om dit doel, de variantie reductie, te bereiken.

Aangezien de geadjungeerde functie gebruikt kan worden als maat voor de importantie van een gebied in het spatiële, richtings en energetische domein in de berekening, wordt zij vaak gebruikt als *bias*functie voor de berekening, zodat de gesimuleerde deeltjes naar de belangrijkste gebieden van de simulatie worden gestuurd en de variantie wordt verminderd in deze gebieden. Sterker nog, het is bewezen dat, wanneer de juiste *bias* wordt gebruikt, de variantie van een berekening naar nul kan worden teruggebracht, tenminste in theorie. Het is echter niet vanzelfsprekend hoe de variantie in een criticaliteitsberekening, waar de reactor zowel bron als "detector" van de gesimuleerde deeltjes is, zou moeten worden verminderd, omdat dit een globaal probleem is, dat moeilijk te lokaliseren is.

In dit proefschrift hebben we een nul-variantieschema ontwikkeld voor een criticaliteitsberekening. Door het criticaliteitsprobleem te behandelen als een bron-detector probleem, hebben we laten zien dat het mogelijk is om een schatter te krijgen met een variantie nul, als er een botsingsschatter wordt gebruikt, als de transport kernen zijn *gebiast* door de geadjungeerde functie en met de juiste manipulatie van de gewichten van de deeltjes. Een gesimplificeerd twee-richtingenmodel is gebruikt als demonstratie, waarmee is bewezen dat de enige beperking voor het reduceren van de variantie de drempel van de Russische roulette is, als er voor de *bias* de analytisch berekende bron en geadjungeerde functies gebruikt worden. Deze drempel kan in theorie to nul worden teruggebracht, zodat de variantie ook naar nul gaat, maar in de praktijk wordt deze grens bepaald door de grenzen van de computerberekening.

In de praktijk echter kan de nul-variantie niet worden gehaald in een criticaliteitsberekening, omdat er gerenormaliseerd moet worden en het gewicht van alle deeltjes aan het begin van alle opeenvolgende cycli moet worden gemiddeld. Toch is er een halvering van de variantie waargenomen in het twee-richtingenmodel, zelfs als

er een met de computer benaderde geadjungeerde functie is gebruikt.

Dit systeem is ook getest in een eigen code, met een 1-D zwak gekoppeld systeem met daarin discrete richtingen en een configuratie met energiegroepen. De resultaten lieten zien dat de orde van grootte van de variantievermindering in een realistischer configuratie vergelijkbaar is met die in het twee-richtingenmodel. Verschillende benaderingen voor de geadjungeerde functie worden getoond, waarbij de geadjungeerde functie een *figure of merit* geeft die bijna twee maal zo hoog is dan wanneer er alleen *implicit capture* wordt gebruikt. De geadjungeerde functie is gegenereerd door middel van lineaire interpolatie van discrete waarden die zijn geproduceerd door een deterministische berekening.

Om het gebruik van deze methode in een productieomgeving te emuleren, is een poging ondernomen om de methode in een productiecode te implementeren. We zijn er alleen in geslaagd om een beperkte 1-D geometrie in de TRIPOLI-4 code te implementeren, hetgeen we eerst geprobeert hebben omdat de structuur van de code het niet toestond om het volledige systeem te gebruiken. Daarom hebben we geprobeerd om het systeem in de MCNP5 code te implementeren. Dit was succesvol en we hebben, gebruikmakend van de geadjungeerde functiewaardes, die door de deterministische code PARTISN4 zijn gegenereerd, het systeem getest in een praktijk Monte-Carlo berekening voor een 17x17 brandstof cel “mini-kern” configuratie.

De variantie in de schatter was de helft van de variantie in het geval van *implicit capture*, wat er op wijst dat dit systeem bruikbaar kan zijn voor praktische nucleaire reactorberekeningen. Daar komt bij dat de convergentie van de bronverdeling niet nadelig werd beïnvloed door de *biasing*. Echter, omdat de code niet was geoptimaliseerd voor snelheid, zijn we niet in staat geweest om een bijpassende verbetering van de efficiëntie van de berekening te demonstreren, omdat de rekentijd significant hoger was. Dit leidt ertoe dat het systeem niet de verwachte verhoging van het *figure of merit* bracht, iets dat kan worden opgelost door de implementatie verder te optimaliseren.

Acknowledgements

First of all, I would like to thank Tim van der Hagen and Eduard Hoogenboom for giving me the opportunity to work in Delft and pursue this research. Especially Eduard, without whose help this thesis would certainly not have been completed, since as my daily supervisor he was always available and willing to help with any problem that came up during all these years, research or otherwise related.

Many thanks to the PNR group in the reactor institute in Delft, who created a very nice environment for me to work during my 4+ years there. All the PhDs, visitors and MSc students at the office, with whom we shared knowledge, laughs and a beer or two: Wilfred, Károly, Luca, Stuart, Ming, Johan, Jitka, Amer, and many more, thank you all. Bart Sjenitzer, who had to listen to my complaints and comments while completing his BSc and MSc theses under my supervision, and for actually completing his theses under my supervision. Also the staff members, Jan Leen, Danny, Martin, Piet, Dick, Camiel and August for always being there when I needed them. Brian, Nele & Jerome for our fruitful discussions but also our always entertaining gatherings, outings or trips, when work got the better of us all! And of course, special thanks goes to Ine Olsthoorn, for sorting out all problems that needed sorting out during my PhD thesis, however big or small.

The TRIPOLI-4 group in CEA, Saclay, for their help with coding whenever it was needed and for their hospitality during the meetings and working periods there.

The group from the Dutch course in Delft, the first foreigners I met who I shared my problems and hopes with: Seda, Callista, Agnieszka, Alyona, Anand, thank you all. The Latin group of friends were my home away from home, and for that they have my eternal friendship and gratitude: Carlos, Christian & Flavia, Gianluca & Diana, Daniel & Ana, Andrea & Alessandra, Hiro & Cristina, thank you all so much for your friendship and your hospitality, I hope we will continue to share ever happier moments together.

Of course, none of all these would have happened without my family in Greece, who supported me in every way possible and helped me get through 10 years abroad, and many more at home. And on the occasion, I would like to welcome its latest member to the family, my brother's first child! A special thanks goes to my friends back in Greece who didn't forget who I was whenever I got back there, especially to George Souroupidis for his enthusiastic support.

Last but certainly not least, I would like to thank Vicky for putting up with me

throughout the last 10 years of my life. It is not only this PhD, but many many more things that I would never have done, if I did not have her at my side. For this, and simply for being at my side all those years, thank you.

List of Publications

S. Christoforou and J. E. Hoogenboom. A zero-variance-based scheme for variance reduction in Monte Carlo criticality. *Proceedings of PHYSOR 2006, ANS Topical Meeting on Reactor Physics*, Vancouver, Canada, September 10-14, 2006.

S. Christoforou et. al.. Implementation of an approximate zero-variance scheme in the TRIPOLI Monte Carlo code. *Proceedings of PHYSOR 2006, ANS Topical Meeting on Reactor Physics*, Vancouver, Canada, September 10-14, 2006.

S. Christoforou and J. E. Hoogenboom. Transition and collision biasing in a variance reduction scheme for criticality calculations. *Proceedings of Joint International Topical Meeting on Mathematics and Computations and Supercomputing in Nuclear Applications (M & C + SNA 2007)*, Monterey, USA, April 15-19, 2007.

S. Christoforou and J. E. Hoogenboom. Investigating the minimum achievable variance in a Monte Carlo criticality calculation. *Proceedings of International Conference on the Physics of Reactors, "Nuclear Power: A Sustainable Resource"*, Interlaken, Switzerland, September 14-19, 2008.

S. Christoforou and J. E. Hoogenboom. Using a zero-variance scheme to accelerate the fission source convergence in a Monte Carlo calculation. *Proceedings of International Conference on Mathematics, Computational Methods & Reactor Physics (M & C 2009)*, Saratoga Springs, New York, USA, May 3-7, 2009.

S. Christoforou and J. E. Hoogenboom. A Zero-Variance Based Scheme for Monte Carlo Criticality Calculations. Accepted in *Nuclear Science and Engineering*.

Curriculum Vitae

Stavros Christoforou was born on September 30, 1981 in Amarousio, Greece.

He grew up in Chalkida, Greece, where he received his secondary school diploma in 1999 from Lykeio Kanithou. In 2000, he moved to the United Kingdom to begin his university education in the University of Surrey, where he obtained his degree in Physics with Nuclear Astrophysics in 2003.

After being awarded the Society for Radiological Protection scholarship in 2003, he enrolled for the Master of Science degree in Radiation & Environmental Protection in the same university, which he received with distinction in 2004.

In January 2005 he started his PhD research in the Physics of Nuclear Reactors group in TU Delft under the supervision of dr. ir. J.E. Hoogenboom. The results of this research are described in this thesis. During his time there, he was the recipient of the 2nd best student paper award in the PHYSOR 2008 conference in Interlaken, Switzerland.