Applied Physics
Faculty of Applied Sciences
Department of Radiation, Radionuclides & Reactors
Section Physics of Nuclear Reactors

Calculating the variance of k_{eff} over neutron chains in Monte Carlo criticality calculations

BSc Thesis - T.J.G. van Kruchten - 1324772

Supervisors:
ir. Bart Sjenitzer
dr. ir. Eduard Hoogenboom

-

May 2012 to January 2013

-

Not confidential

Abstract

Cycle-to-cycle correlation is a cause for underestimation of the variance in Monte Carlo criticality calculations. In order to calculate the true variance, this thesis examines a new method to calculate the k_{eff} and its variance with limited changes to the original sampling and routine of the standard calculation. This method consists of computing the k_{eff} and its variance over neutron chains, instead of neutron generations, for a one-dimensional loosely coupled core geometry. The chain-to-chain method, which uses a population control per batch of neutron chains, yields incorrect values of k_{eff} for smaller batch sizes and inconsistent variance values for larger batch sizes. The used population control per batch alters the source distribution significantly and seems to be the cause of the incorrect k_{eff} values.

Contents

1 Introduction

In the field of nuclear reactor physics, the effective multiplication factor k_{eff} of a system plays an important role. This parameter indicates if the chain reaction in a fissile medium is maintained or not. Nuclear engineers use it as an aid in developing new reactors and safety analysis.

One way to calculate this multiplication factor k_{eff}, as opposed to deterministic methods, is by using Monte Carlo methods. Since Monte Carlo methods use random sampling, solutions are accompanied by a variance. Usually when doing a Monte Carlo criticality calculation, it is assumed that the consecutive generations of neutrons are independent. However, this is not the case and therefore the variance of the calculation will be underestimated.

Due to the earlier mentioned importance of the multiplication factor k_{eff}, it is desirable to know the true variance of the calculated multiplication factor k_{eff}. The intention of this thesis is therefore to avoid the dependency of the cycles and to make a more accurate approximation of the true variance. This attempt is made by creating a (simple) Monte Carlo code that calculates the variance between neutron chains instead of calculating the variance between cycles.

1.1 Outline of thesis

In Chapter 2 there will be a compact introduction to the world of nuclear reactor physics as it is essential for writing a Monte Carlo code that resembles the actual physics of neutrons in a reactor.

The basic principles of the Monte Carlo method will be introduced in Chapter 3. Further on in this chapter neutron transport is examined and a approach to simulate this transport with the aid of Monte Carlo is shown. The end of the chapter will contain a numerical example of a standard Monte Carlo criticality calculation.

Then, in Chapter 4 the problem of variance underestimation in Monte Carlo criticality calculations is pointed out and some data is presented to back this up. A new method to calculate the k_{eff} and its variance, the chain-to-chain method, is evaluated for a one-dimensional loosely coupled core geometry. Also, the effect of this chain-to-chain method on the source convergence of some one-dimensional loosely coupled systems is regarded.

Chapter 5 will draw conclusions and make recommendations for potential future research.

2 Reactor physics

2.1 Nuclear fission

Radioactive decay is the process in which a nucleus undergoes a transformation to a different nucleus which in general is accompanied by the emission of other energetic particles. Radioactive decay can happen spontaneously by unstable nuclei, but it can also be induced by letting the nucleus collide with another particle. This is what happens with a fission reaction; a neutron collides with a heavier nucleus that splits in fission fragments, while releasing (possibly!) a number of neutrons and energy in different forms. If a released neutron in its turn induces a fission reaction with another nucleus, a chain reaction is created. The goal of a nuclear reactor is to keep a stable fission chain reaction going as to collect energy. The largest part of the energy (about 80%) comes from the high-speed fission fragments that heat the fissile material which in turn heats a moderator (usually water).

Figure 2.1: Nuclear fission

2.2 Criticality

As mentioned, it is vital for a nuclear reactor to sustain a stable fission chain reaction. A steady-state chain reaction is maintained if, on average, every neutron also induces one new neutron. Then also a constant rate of fission energy is produced. If, on average, one neutron induces *more* than one neutron, the neutron population will grow exponentially. One the other hand, when, on average, one neutron will induce *less* than one neutron, the neutron population will decay exponentially. A ratio, called the multiplication factor k, is defined to quantify this behaviour:

$$k \equiv \frac{\text{Number of neutrons in one generation}}{\text{Number of neutrons in preceding generation}} \tag{2.1}$$

When doing a criticality calculation on a nuclear reactor, the multiplication factor k is being calculated. It is a measure of the state of a nuclear reactor:

$$\begin{aligned} k < 1 &\quad \text{subcritical} \\ k = 1 &\quad \text{critical} \\ k > 1 &\quad \text{supercritical} \end{aligned}$$

Since experiments to determine the multiplication factor k in real life can be very expensive and dangerous, it is a good idea to first run simulations on computers to calculate this value. Exactly this is done in a Monte Carlo criticality calculation.

2.3 Cross sections

In order to describe the transport and different interactions neutrons can have in different material, microscopic and macroscopic cross sections are used;

The probability that a neutron reacts with a nucleus in a thin (in fact so thin that there is only one layer of nuclei) slab of material is called the microscopic cross section:

$$\sigma = \frac{\text{Number of reactions/nucleus/s}}{\text{Number of incident neutrons/cm}^2\text{/s}} = \frac{(R/N_A)}{I} \tag{2.2}$$

Corresponding to which kind of reaction a neutron has with a nucleus, indices are added to the microscopic cross sections as shown in Tab. 1.

Table 1: Different kind of cross sections and their indices

Index	**Meaning**	**Probability of**
σ_t, Σ_t	*total*	neutron-nucleus reaction occurring
σ_s, Σ_s	*scattering*	neutron scattering against a nucleus
σ_a, Σ_a	*absorption*	neutron gets absorbed by nucleus (capture + fission)
σ_c, Σ_c	*capture*	nucleus captures a neutron without release of new neutrons
σ_f, Σ_f	*fission*	nuclear fission occurring

In most cases one does not have such a thin slab of material, but material that is a lot thicker. Here, the nuclei behind the nuclei that encounter the incident neutrons first, are shielded. Since the neutrons will have reactions with earlier seated nuclei, the intensity of the neutron beam will decrease with a position further into the material. The following expression for the intensity of the neutron beam (without derivation) accounts for the shielding from other nuclei in a thicker slab of material:

$$I(x) = I_0 e^{-N\sigma_t x} \tag{2.3}$$

Since $N\sigma_t$ is used frequently in reactor physics to define a material, it has its own symbol called the macroscopic cross section:

$$\Sigma_t \equiv N\sigma_t \tag{2.4}$$

which is interpreted as the probability per unit path length traveled that a neutron will undergo a reaction with the nucleus in the sample [DH76, p. 21]. As with the microscopic cross section, the macroscopic cross section can also carry the different indices as shown in Tab. 1. Macroscopic cross sections (for one energy group) will be extensively used in the Monte Carlo criticality calculations later on when simulating neutron transport. For example: when a neutron-nuclei reaction takes place, the probability that nuclear fission will occur is Σ_f/Σ_t. The average distance a neutron travels before interacting with a nucleus, can also be calculated:

$$\bar{x} \equiv \int_0^\infty xp(x)\mathrm{d}x = \Sigma_t \int_0^\infty xe^{-\Sigma_t x}\mathrm{d}x = \frac{1}{\Sigma_t} \tag{2.5}$$

This is called the neutron mean free path, because it is a measure of the average distance a neutron can travel in the sample before having a neutron-nucleus reaction.

3 Monte Carlo methods for neutron transport

Before using the Monte Carlo method to describe neutron transport, first the general Monte Carlo principles will be explained. After that, the neutron transport equation is prepared for use with the Monte Carlo method. Lastly, all the aspects of a standard Monte Carlo criticality calculation will be addressed and a numerical example is demonstrated.

3.1 Monte Carlo principles

The Monte Carlo method is a stochastic method to solve deterministic problems. There are a lot of different flavors of the Monte Carlo method in use. A short introduction to two general methods will be given.

3.1.1 General Monte Carlo methods

Typically, in a Monte Carlo calculation, one makes use of random numbers to evaluate integrals. There seems to be a general distinction between two types of different approaches. One method (Fig. 3.1a) calculates the averages of randomly selected values of the integrand. For instance, if

$$F = \int_a^b f(x)\mathrm{d}x \tag{3.1}$$

is the integral to be evaluated and x_i is a uniformly distributed random variable between a and b which is generated N times, the integral can be approximated by:

$$F = \frac{b-a}{N}\sum_{i=1}^{N} f(x_i) \tag{3.2}$$

Another approach, is the so-called simulation or rejection method (Fig. 3.1b). Here, the uniformly distributed random numbers are not used as input for the integrand function. Instead, uniformly distributed random numbers (x_i and y_i) are generated. If the position in question lies above the integrand function, it will be *rejected*. If the position in question lies under the integrand function it will be accepted. Then, the ratio of accepted positions against all generated positions is, when multiplied with the area of the predefined square, an approximation of the integral.

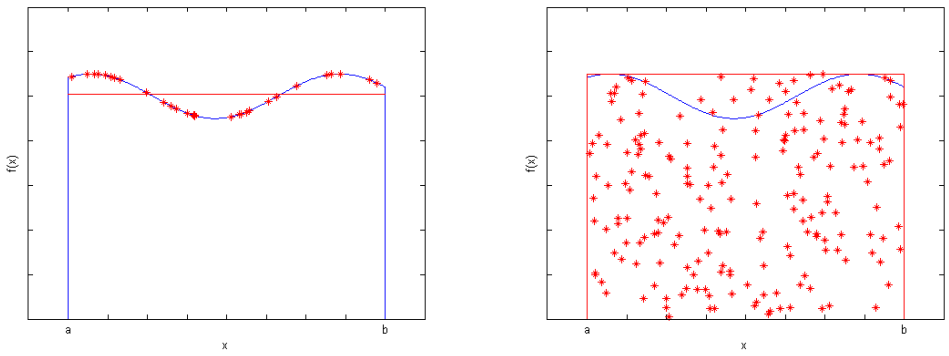
3.1.2 The law of large numbers and the central limit theorem

Since the Monte Carlo method is a stochastic method, its results are subject to statistical uncertainties. This means that the results of these methods have to be accompanied by a confidence interval. The law of large numbers and the central limit theorem provide a foundation for qualitative and quantitative statements about these confidence intervals.

Suppose there are a number of random variables, which all have the same unknown distribution: $X_1, X_2 \ldots X_n$. Looking at X_i as the ith repetition of a measurement. The average of the first n measurements is then:

$$\bar{X} = \frac{X_1 + X_2 + \cdots + X_n}{n} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{3.3}$$

The expectation value of this average is the same as the expectation value of a single random variable: $\mathrm{E}[\bar{X}] = \mu$, but the variance of the average is less than the variance of a single random variable: $\mathrm{Var}(\bar{X}) = \sigma^2/n$. So, using averages is more accurate; the larger the sample n, the smaller the variance. The law of large numbers now states that the sample mean, with a probability that



(a) Averages of random values of integrand

(b) Simulation or rejection technique

Figure 3.1: Graphical interpretation for different Monte Carlo methods

approaches 1 as n increases to infinity, approximates the true mean. This can be proved using Chebyshev's inequality, which roughly states that almost all values of any probability distribution lie within a few standard deviations of the mean. Applying this to the average $\bar{X}$ in mathematical form gives:

$$\mathrm{P}(|\bar{X} - \mu| > \varepsilon) \leq \frac{1}{\varepsilon^2}\mathrm{Var}(\bar{X}) = \frac{\sigma^2}{n\varepsilon^2} \tag{3.4}$$

which claims that the probability that $\bar{X}$ is outside the interval $(\mu - \varepsilon, \mu + \varepsilon)$ is less than or equal to the right-hand of Eq. 3.4. ε can be any positive number. When n goes to infinity, it becomes obvious that the right-hand side becomes zero:

$$\lim_{n\to\infty} \mathrm{P}(|\bar{X} - \mu| > \varepsilon) = 0 \tag{3.5}$$

which is the (weak) law of large numbers.

The central limit theorem is an expansion of the law of large numbers and thrives on the same principle and states that the average $\bar{X}$ (as in Eq. 3.3) of a number of random variables, with the same unknown distribution $X_1, X_2 \ldots X_n$, approaches a normal distribution as n grows large, no matter what the distribution of X_i is:

$$\mathrm{P}\left\{a \leq \frac{\bar{X} - \mu}{\sigma/(n)^{\frac{1}{2}}} \leq b\right\} \to \frac{1}{(2\pi)^{\frac{1}{2}}} \int_a^b e^{-t^2/2}\mathrm{d}t \tag{3.6}$$

3.1.3 Variance and standard deviation

Since the central limit theorem states that the average (Eq. 3.3) approaches the normal distribution for large n, it is now possible to say something sensible about the variance of this average. This *sample variance of the mean* is defined as:

$$s^2_{\bar{X}} = \frac{s^2}{n} \tag{3.7}$$

where s^2 is the sample variance, which is defined as:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \tag{3.8}$$

The factor $1/(n-1)$ is added to ensure that s^2 is an unbiased estimator for the true variance, which can be shown as follows [Dek05]:

$$\begin{aligned} \mathrm{E}[s^2] &= \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 && (3.9)\\ &= \frac{1}{n-1}\sum_{i=1}^{n}\mathrm{Var}(X_i - \bar{X}) && (3.10)\\ &= \frac{1}{n-1}\cdot n \cdot \frac{n-1}{n}\sigma^2 && (3.11)\\ &= \sigma^2 && (3.12) \end{aligned}$$

In a Monte Carlo criticality calculation the average $\bar{X}$ is the final answer of the calculation and $s^2_{\bar{X}}$ gives its variance. Eq. 3.7 can be rearranged in order to reduce the variables that need storing later on when using it in the Monte Carlo code. This leads to the much used expression for the sample variance of the mean:

$$s^2_{\bar{X}} = \frac{1}{n-1}\left\{\frac{1}{n}\sum_{i=1}^{n}X_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n}X_i\right)^2\right\} \tag{3.13}$$

Since the standard deviation is just the square root of the variance, they both are an indication of the accuracy of the calculated value. The standard deviation will be used a lot to give the confidence interval of the Monte Carlo result.

3.2 Neutron transport equation tailored for Monte Carlo

Now that the basic principles of Monte Carlo are explained, it is time to start using it in the field of interest. To arrive at a equation that describes neutron transport and that is also solvable with Monte Carlo, it is possible to integrate the integro-differential form of the neutron transport equation. Another, more intuitive way to achieve this goal is by looking closer at the physics of the actual problem and describing all the relevant actions of a neutron in its lifetime. Before doing so, a mathematical foundation that justifies its use will be presented. The same derivation is used as in Lux and Koblinger [LK91, p. 99]. First, a pair of density functions are introduced:

$$\chi(\boldsymbol{r}, E, \hat{\boldsymbol{\Omega}})\mathrm{d}\boldsymbol{r}\mathrm{d}E\mathrm{d}\hat{\boldsymbol{\Omega}} \tag{3.14a}$$

$$\psi(\boldsymbol{r}, E, \hat{\boldsymbol{\Omega}})\mathrm{d}\boldsymbol{r}\mathrm{d}E\mathrm{d}\hat{\boldsymbol{\Omega}} \tag{3.14b}$$

where Eq. 3.14a, the emission density, represents the expected number of particles *leaving* a collision at position $\boldsymbol{r}$ in a volume element $\mathrm{d}\boldsymbol{r}$ with an energy E in $\mathrm{d}E$ and direction $\hat{\boldsymbol{\Omega}}$ in $\mathrm{d}\hat{\boldsymbol{\Omega}}$. Eq. 3.14b, the collision density, represents the expected number of particles *entering* a collision.

Two kernels are defined in order to describe the two basic processes that are going on in a neutrons lifetime. For a particle that travels through space between collisions, there is the transition kernel;

$$T(\boldsymbol{r}' \to \boldsymbol{r}, E, \hat{\boldsymbol{\Omega}})\mathrm{d}\boldsymbol{r} \tag{3.15}$$

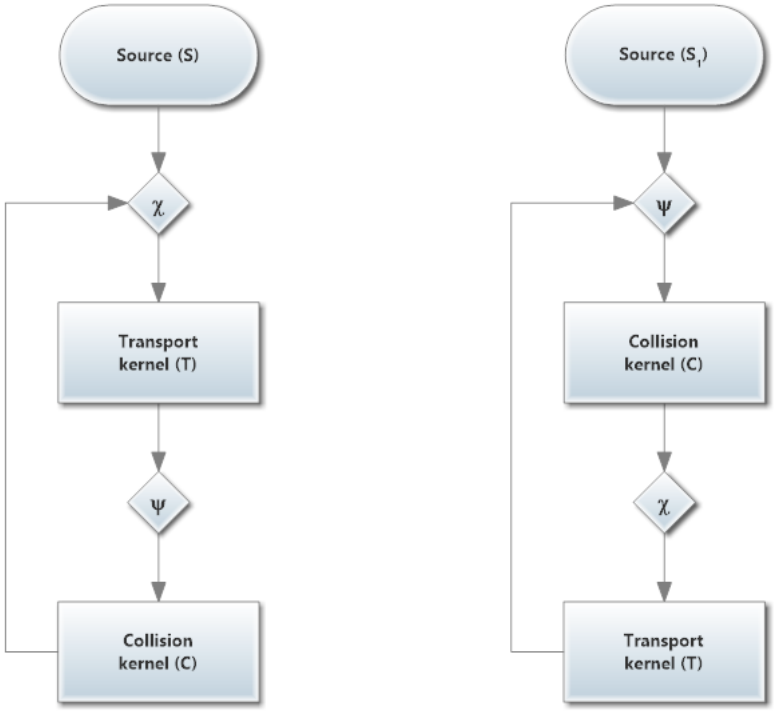
which states the probability that a neutron leaving a collision at $\boldsymbol{r}'$ will enter its next collision at $\boldsymbol{r}$ in $\mathrm{d}\boldsymbol{r}$. Notice that the angle $\hat{\boldsymbol{\Omega}}$ and the energy E of the neutron will remain unchanged. This

is because a neutron has no interaction with other neutrons and between collisions it is assumed that the kinetic energy does not change.

To describe all the possible outcomes at a collision there is the collision kernel:

$$C(\boldsymbol{r}, E' \to E, \hat{\boldsymbol{\Omega}}' \to \hat{\boldsymbol{\Omega}})\mathrm{d}E\mathrm{d}\hat{\boldsymbol{\Omega}} \tag{3.16}$$

which in turn states the probability that a neutron entering a collision at $\boldsymbol{r}$ with energy E' and direction $\hat{\boldsymbol{\Omega}}'$ will leave that collision with energy E in $\mathrm{d}E$ and direction $\hat{\boldsymbol{\Omega}}$ in $\mathrm{d}\hat{\boldsymbol{\Omega}}$. Since the neutrons and nuclei are modeled as point particles, the neutrons enters and exits a collision at the same position. From this approach, which is intuitively represented in Fig. 3.2a the following relations



(a) Combined kernel with transport kernel before collision kernel

(b) Combined kernel with collision kernel before transport kernel

Figure 3.2: Schematic representation of neutron transport

can be obtained:

$$\chi(\boldsymbol{r}, E, \hat{\boldsymbol{\Omega}}) = S(\boldsymbol{r}, E, \hat{\boldsymbol{\Omega}}) + \iint \psi(\boldsymbol{r}, E', \hat{\boldsymbol{\Omega}}')C(\boldsymbol{r}, E' \to E, \hat{\boldsymbol{\Omega}}' \to \hat{\boldsymbol{\Omega}})\mathrm{d}E'\mathrm{d}\hat{\boldsymbol{\Omega}}' \tag{3.17a}$$

$$\psi(\boldsymbol{r}, E, \hat{\boldsymbol{\Omega}}) = \int \chi(\boldsymbol{r}', E, \hat{\boldsymbol{\Omega}})T(\boldsymbol{r}' \to \boldsymbol{r}, E, \hat{\boldsymbol{\Omega}})\mathrm{d}\boldsymbol{r}' \tag{3.17b}$$

where S is an external source. When substituting one equation into the other and vice versa, the particle transport can be described in just one of the densities χ or ψ. Choosing to do this for the

collision density and thus substituting Eq. 3.17a into Eq. 3.17b leads to:

$$\psi(\boldsymbol{r}, E, \hat{\boldsymbol{\Omega}}) = S_1(\boldsymbol{r}, E, \hat{\boldsymbol{\Omega}}) + \iint \psi(\boldsymbol{r}', E', \hat{\boldsymbol{\Omega}}') C(\boldsymbol{r}', E' \to E, \hat{\boldsymbol{\Omega}}' \to \hat{\boldsymbol{\Omega}}) T(\boldsymbol{r}' \to \boldsymbol{r}, E, \hat{\boldsymbol{\Omega}}) \mathrm{d}\boldsymbol{r}' \mathrm{d}E' \quad (3.18)$$

where S_1 is the first-flight collision density due to an external source. This expression could also be derived from Fig. 3.2b [HS11]. If the short-hand notation $P = \boldsymbol{r}, E, \hat{\boldsymbol{\Omega}}$ is used and the combined transport kernel of Fig. 3.2b is defined as $L(P' \to P) = C(\boldsymbol{r}', E' \to E, \hat{\boldsymbol{\Omega}}' \to \hat{\boldsymbol{\Omega}}) T(\boldsymbol{r}' \to \boldsymbol{r}, E, \hat{\boldsymbol{\Omega}})$, the neutron transport equation expressed with the collision density can be simplified to:

$$\psi(P) = S_1(P) + \int \psi(P') L(P' \to P) \mathrm{d}P' \quad (3.19)$$

which is a second-kind Fredholm-type integral. To find a solution with Monte Carlo, the integral is first expanded into a Neumann series; one can express the collision density as a summation, where each term consists of a multidimensional integral that suggests a random walk scheme for its evaluation [CC75, p. 34]. This evaluation is started by substituting $\psi(P')$ in Eq. 3.19 by the following expression:

$$\psi(P') = S_1(P') + \int \psi(P'') L(P'' \to P') \mathrm{d}P'' \quad (3.20)$$

Repeating this procedure for every ψ in all the next integrands and changing the apostrophes to numbers for indexing, the collision density can be expressed as:

$$\psi(P) = \sum_{n=0}^{\infty} \int \cdots \int S_1(P_0) L(P_0 \to P_1) L(P_1 \to P_2) \cdots \cdots L(P_{n-1} \to P) \mathrm{d}P_0 \mathrm{d}P_1 \cdots \mathrm{d}P_{n-1} \quad (3.21)$$

which is a result that is still difficult to solve using Monte Carlo *integration*. Luckily, Monte Carlo *simulation* is the right tool for the job.

3.3 Monte Carlo criticality calculation

To solve the neutron transport equation with a Monte Carlo simulation requires to sample the transport equation. This can be done in a rather intuitive way. By looking at the life of a neutron and the possible interactions it has, particle transport can be simulated without directly using the transport equation. The particle transport itself is a stochastic process, so a direct (analog) simulation should also be stochastic [CC75, p. 41]. This approach actually has the same result as sampling the transport equation.

In the standard Monte Carlo criticality calculation used in this thesis, first a number of predefined source neutrons are uniformly generated in the fissionable material. Each neutron then travels a simulated path length. If this path length reaches outside the system boundaries, this neutron is discarded and the next neutron is sampled. If the path length ends in fissionable material, a collision occurs and the type of interaction as well as the number of induced neutrons is sampled. Next, this routine is repeated for the new induced neutrons creating generations of neutrons that are called cycles. The multiplication factor k_i is calculated for each cycle as shown in Fig. 3.3. The end value for the multiplication factor k_{eff} is the mean value of all these cycle values and the variance of k_{eff} is calculated between these cycles. Some of the important aspects of this calculation are evaluated in more detail in the following sections.

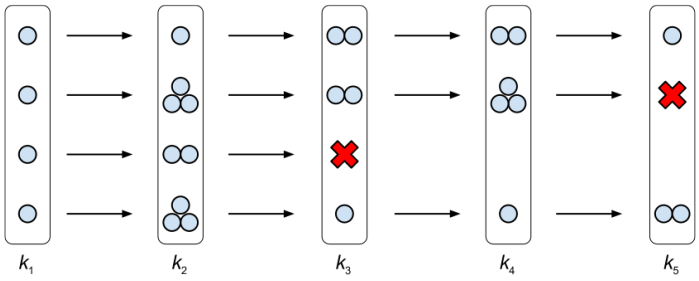


Figure 3.3: Schematic representation of calculating k_{eff} and its variance between cycles

3.3.1 Sampling

In order to describe the way neutrons are generated and the way they interact with the environment, different distribution functions are sampled.

Generating realizations of random variables

The uniformly distributed random number ρ in the domain $[0, 1]$ is widely available for computer codes. Since the sampling of different probability density functions is necessary in a Monte Carlo criticality calculation it is convenient to have the possibility to transform ρ to any distribution that is needed.

To demonstrate the way these realizations are generated, a derivation as in Dekking [Dek05, p. 73] is used. Suppose $F(x)$ is the distribution function of the desirable random number. Assumed is that $F(x)$ is strictly increasing from 0 to 1 on an interval. Then $F(x)$ has an inverse function $F^{\text{inv}}(u)$ and $u \leq F(x)$ is equivalent with $F^{\text{inv}}(u) \leq x$. If the real number u is replaced by the uniformly distributed random variable ρ, the corresponding probabilities are the same:

$$\mathrm{P}(\rho \leq F(x)) = \mathrm{P}(F^{\text{inv}}(\rho) \leq x) \tag{3.22}$$

Since ρ is uniformly distributed, the left-hand side of Eq. 3.22 is equal to $F(x)$, which leads to:

$$F(x) = \mathrm{P}(F^{\text{inv}}(\rho) \leq x) \tag{3.23}$$

which states that the random variable $F^{\text{inv}}(\rho)$ has the distribution function F. To find this random variable $F^{\text{inv}}(\rho)$ one needs to solve $F(x) = \rho$ for x and obtain the wanted random variable with the right distribution.

Sampling the source

The very first neutrons in the source have to be generated. The problem is that in general the true source distribution of the system is unknown. Therefore an initial guess is made. The closer the initial source guess is, the faster the it converges to this proper source distribution. In this thesis the source starts with a flat distribution, which means that the starting positions are uniformly distributed over the one-dimensional system. The first few cycles are not taken into account as to let the starting distribution converge to the true source distribution. Since only these one-dimensional core geometries are used, the sampling of the source is straightforward. To let the

source neutrons start uniformly distributed in the domain $[a, b]$, ρ can be transformed into the right distribution by:

$$f_{\text{source}}(\rho) = a + (b - a)\rho \tag{3.24}$$

If the core geometry consists of two different fissionable regions, as it will later on, first a 50-50% probability can be used to decide in which region to start.

Sampling the track length and new direction

Next, the generated neutron has to travel a certain distance. The probability that a neutron in a material with macroscopic cross section Σ_t has track length s is given by the following probability density function:

$$f(s) = \mathrm{P}(X = s) = \Sigma_t e^{-\Sigma_t s} \tag{3.25}$$

which has the cumulative distribution function:

$$F(s) = \mathrm{P}(X \leq s) = \int_0^s \Sigma_t e^{-\Sigma_t x} \mathrm{d}x = 1 - e^{-\Sigma_t s} \tag{3.26}$$

Since the cumulative distribution function of ρ (as defined before in the domain $[0, 1]$) is just ρ, Eq. 3.26 can be written as:

$$\rho = 1 - e^{-\Sigma_t s} \tag{3.27}$$

Rewriting this last expression yields the track length:

$$s = -\frac{1}{\Sigma_t} \ln(1 - \rho) \tag{3.28}$$

In the event of scattering and fission, the scattered or induced neutron(s) will travel in a new direction. If the particle travels in a plane, the direction can be sampled from $\phi = 2\pi\rho$, which is uniformly distributed on the circumference of a unit circle. The new position of the neutron is then:

$$x_{\text{new}} = x_{\text{old}} + s\cos(\phi) \tag{3.29}$$

$$y_{\text{new}} = y_{\text{old}} + s\sin(\phi) \tag{3.30}$$

Since one dimensional systems are used, only the direction cosine with the x-axis of Eq. 3.29 is of interest. This cosine ranges from -1 to 1 and can also be sampled by $u = 2\rho - 1$. This leads to the (one dimensional) expression for the position:

$$x_{\text{new}} = x_{\text{old}} + us \tag{3.31}$$

Sampling the interactions

When a neutron has traveled a specific track length and has not leaked out of the system, a collision occurs. To determine what happens next at that collision, cross sections are used. Since the scattering, capture and fission cross sections add up to the total cross section $\Sigma_s + \Sigma_c + \Sigma_f \equiv \Sigma_t$, the probabilities that the different actions occur at a collision are:

$$\frac{\Sigma_s}{\Sigma_t}, \frac{\Sigma_c}{\Sigma_t} \text{ and } \frac{\Sigma_f}{\Sigma_t} \tag{3.32}$$

In the event of scattering, a new track length will be sampled. When the neutron is captured, the neutron is no more and the next particle is simulated. With fission occurring, the amount

of induced neutrons is determined and the origin positions are used in the next cycle as starting positions for the induced neutrons. Note that this method is an analog method; it follows the path and life of a neutron as truly as possible. In the next sections some tricks will be added to the analog method, making it a non-analog method. A non-analog method deviates from the physical way a neutron behaves in order to improve the efficiency of the calculation.

3.3.2 Estimators for k_{eff}

Now that the neutron and its interactions are simulated, it is time to extract useful information from the calculation. Since in this thesis only criticality calculations are performed, the quantity of interest is the multiplication factor k_{eff}. From the definition of Eq. 2.1 a way to calculate this quantity can be directly adopted:

$$k_i = \frac{\text{Number of fission neutrons generated during cycle } i}{\text{Number of neutrons started in cycle } i} \tag{3.33}$$

In the event of fission, the number of fission neutrons is calculated with:

$$n_{\text{fis}} = \lfloor \bar{\nu} + \rho \rfloor \tag{3.34}$$

where $\bar{\nu}$ is the number of fission neutrons in the event of fission and which is a material property. ρ is, again, the uniformly distributed random number in the domain $[0, 1]$. The floor brackets are added to ensure a integer value. This way the total number of fission neutrons per cycle can be generated and from those also the multiplication factor k_{eff} per cycle can be calculated.

Another approach to calculating k_{eff},which is statistically favorable, consists of using the expected number of fission neutrons instead of the actual generated number of fission neutrons:

$$\hat{k}_i = \frac{\text{Expected number of fission neutrons generated during cycle } i}{\text{Number of neutrons started in cycle } i} \tag{3.35}$$

There are different ways to estimate the k_i, which correspond to different *estimators*. The path length estimator uses the path length s to estimate the k_{eff}:

$$\hat{k}_i = \frac{1}{N_{i,\text{start}}} \sum_{\substack{\text{paths} \\ \text{in cycle}}} s\bar{\nu}\Sigma_f \tag{3.36}$$

The absorption estimator looks at every absorption in a cycle:

$$\hat{k}_i = \frac{1}{N_{i,\text{start}}} \sum_{\substack{\text{absorptions} \\ \text{in cycle}}} \bar{\nu}\frac{\Sigma_f}{\Sigma_a} \tag{3.37}$$

The collision estimator takes scores at every collision:

$$\hat{k}_i = \frac{1}{N_{i,\text{start}}} \sum_{\substack{\text{collisions} \\ \text{in cycle}}} \bar{\nu}\frac{\Sigma_f}{\Sigma_t} \tag{3.38}$$

where $N_{i,\text{start}}$ is the actual generated neutrons that start at the beginning of the cycle i. These estimators all take scores at different events. This is called tallying. Since the last estimator, the collision estimator, tallies at every collision it is likely to have the most accurate estimate and thus the lowest variance. In every calculation in this thesis, the collision estimator is used to estimate

the k_i. The actual starting number of neutrons is calculated by generating a number of fission neutrons at every collision:

$$n_{\text{fis}} = \left\lfloor \bar{\nu} \frac{\Sigma_f}{\Sigma_t} + \rho \right\rfloor \tag{3.39}$$

The addition of all these contributions in a cycle i gives the number of starting neutrons for the next cycle.

3.3.3 Variance reduction techniques

The goal of a Monte Carlo calculation is to calculate a desired tally with a small enough confidence interval. Therefore several variance reduction techniques that are used in this thesis are introduced.

Population control

If the k_{eff} of the system is larger than 1, the neutron population will grow larger with each cycle. More neutrons need, in general, more computational power. So a growing neutron population is not desirable. A decreasing neutron population will in turn go to zero eventually. This is also undesirable, because then it may not complete enough cycles to have an accurate estimation of the multiplication factor k_{eff}; the variance will be large. In order to control this, the expected number of fission neutrons ν^* is adjusted after each cycle i:

$$\nu^*_{i+1} = \frac{\bar{\nu}}{k_i} \tag{3.40}$$

where $\bar{\nu}$ is the expected number of fission neutrons and k_i the estimate of the multiplication factor for cycle i. This way the number of fission neutrons that are induced is corrected by a factor $1/k_i$ in order to keep the number of neutrons constant. Since the calculation is essentially a stochastic process, the number of generated neutrons can still deviate from the number of expected neutrons. The factor N_{initial}/N_i is therefore added to correct any deviation from the starting number of neutrons:

$$\nu^*_{i+1} = \frac{\bar{\nu}}{k_i} \frac{N_{\text{initial}}}{N_i} \tag{3.41}$$

where N_{initial} is the original starting number of source neutrons and N_i is the number of induced neutrons in every cycle.

Implicit capture and Russian roulette

Instead of letting the probability exist that a neutron is captured by a nucleus, implicit capture is implemented; a source neutron gets a statistical weight of 1 and now at each collision this weight will be adjusted:

$$w_{\text{after}} = w_{\text{before}} \frac{\Sigma_s}{\Sigma_t} \tag{3.42}$$

where Σ_s/Σ_t is the probability that a neutron is *not* absorbed. In this case, neutrons that in an analog way would have died out, can now still generate hits. This way more particles can be directed to positions where normally only a few particles would arrive. This form of modified sampling will reduce the variance.

In order to let neutrons with a small weight and thus small contributions die out, Russian roulette is played with the neutron. When the neutrons weight reaches a minimum value w_{rr}, it has to play Russian roulette. The neutron is terminated with probability:

$$1 - \frac{w}{w_{\text{s}}} \tag{3.43}$$

and survives with probability:

$$\frac{w}{w_{\mathrm{s}}} \tag{3.44}$$

where w_{s} is the survival weight that a neutron gets when it survives. This way neutrons with low contributions will either be terminated or given higher contributions in an unbiased way. For example, if $w = w_{rr}$ and $w_s = 2w_{rr}$, then the neutron has the probability of a half to survive (while obtaining the weight of $2w_{rr}$) and the probability of a half to be terminated. Those two events together have the same expectation value for the weight as the original weight. This way the use of Russian roulette is unbiased. Note that Russian roulette is not a way to reduce the variance. In fact it is the opposite of implicit capture and reduces calculation time. If these low weight particles would not have to play, they would only be stopped if they leaked out of the system. This can take up a lot of costly calculation time on uninteresting particles. In all the Monte Carlo criticality calculations during this thesis $w_{\mathrm{s}} = 2w_{\mathrm{rr}}$ is used with $w_{\mathrm{rr}} = 0.1$.

Final $\hat{k}_i$ and number of fission neutrons

If implicit capture is implemented some adjustments have to be made to the collision estimator:

$$\hat{k}_i = \frac{1}{N_{i,\mathrm{start}}} \sum_{\substack{\text{collisions} \\ \text{in cycle}}} w\bar{\nu}\frac{\Sigma_f}{\Sigma_t} \tag{3.45}$$

Now w represents the weight of the particle in a specific collision. Assumed is that the starting weight of all particles is 1.

The population control is used to adjust the number of actual generated fission neutrons per collision:

$$n_{\mathrm{fis}} = \left\lfloor w\nu^*\frac{\Sigma_f}{\Sigma_t} + \rho \right\rfloor \tag{3.46}$$

where ν^* is the expected number of fission neutrons in the event of fission, that is now adjusted with each cycle.

3.4 Numerical example

To show an example of a standard Monte Carlo criticality calculation, a one-dimensional core geometry has been modeled. This consists of a 30 cm wide homogeneous slab of fissionable material that has been derived from uranium oxide. Its cross sections can be found in Tab. 2. Note that these are not the actual cross sections of uranium oxide, but a mono-energetic approximation.

Table 2: Mono-energetic macroscopic cross sections of uranium oxide and water

Material	Σ_t (cm^{-1})	Σ_s (cm^{-1})	Σ_a (cm^{-1})	Σ_f (cm^{-1})
Uranium oxide	1.60417	1.5783	0.0259	0.0137
Water	2.6296	2.6157	0.0139	-

A Monte Carlo criticality calculation with 600 cycles and 2000 neutrons per cycle has been run. In order to let the source converge, before acquiring any data, 550 active cycles have been used. So the first 50 cycles are not included in the tallying of the collision estimator. An uniform source distribution has been used, as well as population control, implicit capture and Russian roulette as explained in the previous sections. The expected number of fission neutrons generated $\bar{\nu}$ is

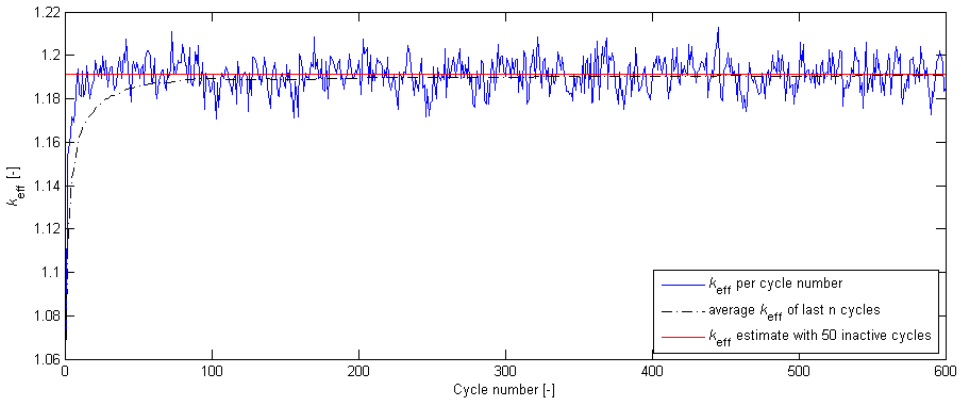


Figure 3.4: Results of a standard Monte Carlo criticality calculation

set to 2.43975. The weight for a neutron entering a game of Russian roulette is set to 0.1 and the survival weight is twice as big. The end result of the calculation gives a k_{eff} of 1.1916 with a standard deviation of $3 \cdot 10^{-4}$. In Fig. 3.4 the values of k_{eff} for each cycle is plotted. It is clear that in the first few cycles the source is not yet converged, therefore the red line represents the final k_{eff}-estimate without taking in account the first 50 cycles.

4 Variance underestimation and chain-to-chain method

4.1 Variance underestimation

4.1.1 Cause of variance underestimation

In order for a sample variance to be an unbiased estimator for the true variance, it is necessary that the random variables, between which the sample variance is calculated, are uncorrelated. However, in a Monte Carlo criticality calculation, the way that in every generation new source neutrons are generated, yields correlated samples; there exists a correlation between the k_{eff}-estimates of the cycles. The variance calculated between these cycle estimates is in turn *under*estimated. The outcome of such a calculation can therefore overly positive accurate results, which is not wanted.

4.1.2 Dominance ratio and core geometry

A Monte Carlo criticality calculation is essentially the solving of an eigenvalue problem by making use of the power iteration method. When doing such a calculation, the k_{eff} is the first, dominant eigenvalue of the system configuration. Since the source distribution of the converged fission source is unknown, an initial guess has to be made. All examples in this thesis use an uniform source distribution. The dominance ratio λ is the ratio of the second eigenvalue to the first:

$$\lambda = \frac{k_2}{k_1} \tag{4.1}$$

If this ratio is close to unity, the convergence to the first eigenstate is slow [Urb95]. Therefore, often the first few cycles of an Monte Carlo criticality calculation are not included for tallying as to make sure that the system has converged. Now it is also generally accepted that systems with dominance ratios close to unity have a strong cycle to cycle correlation and thus show the strongest signs of variance underestimation with the exception of some special cases. Systems that have larger dimensions than the mean free path of neutrons and thus have a weak neutron communication between distant regions of the system exhibit these features. Since these systems are of importance to this thesis the dominance ratio of the simple one-dimensional homogeneous slab reactor of Section 3.4 is calculated with the help of the one-speed diffusion theory model to see if it comes close to unity. The one-speed diffusion theory model learns that the nth eigenvalue of the system is given by:

$$k_n = \frac{\bar{\nu}\Sigma_f}{\Sigma_a + DB_n^2} \tag{4.2}$$

where D is the diffusion coefficient:

$$D = (3\Sigma_t)^{-1} \tag{4.3}$$

and B_n is the nth geometric buckling:

$$B_n = \frac{n\pi}{\tilde{L}} \tag{4.4}$$

with $\tilde{L}$ representing the extrapolated length:

$$\tilde{L} = L + 2 \cdot 0.7104\Sigma_t^{-1} \tag{4.5}$$

The cross sections used are again those of uranium oxide and can be looked up in Tab. 2. Using Eq. 4.2, the first eigenvalue of the system is $k_1 = 1.1916$ (which corresponds with the calculation of Section 3.4) and the second eigenvalue is $k_2 = 0.9688$. So the dominance ratio for this system is 0.8131. Tripling the width of the slab, would yield a dominance ratio of 0.9723 and would

be a better candidate for examining variance underestimation. Still another, more complex, one-dimensional core geometry as depicted in Fig. 4.1 is chosen. These kind of loosely coupled systems are frequently used as benchmark problems because of its slow convergence and high dominance ratio. The dominance ratio for this system is not calculated using the one-speed diffusion theory model because this is not so straightforward as for a slab reactor, but one can find some quantitative results for the growing dominance ratio of a system with a larger slab separation in [Urb95, p. 43]. In this thesis two different systems will be evaluated; the 20-30-20 and the 20-30-18 core geometries. The numbers indicate, respectively, the size in centimeters of the first fissionable slab, the water and the second fissionable slab.

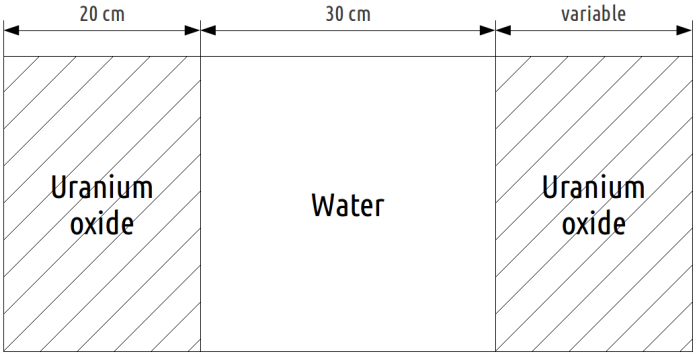


Figure 4.1: One-dimensional loosely coupled core geometry

4.1.3 Variance underestimation in Monte Carlo criticality calculations

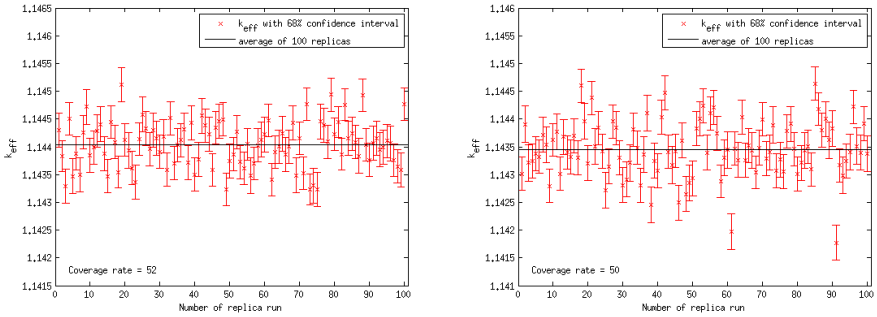
The standard deviation of the mean is the confidence interval of the final value for k_{eff} when the Monte Carlo criticality calculation is run once, since the mean is in this case the k_{eff}-value. If several independent runs, or replicas, would be made, one would expect that the spread, or population standard deviation, of all the obtained k_{eff}-values is the same as the standard deviation of the mean of one single run. This spread or population standard deviation will from now on be referred to as the "true" standard deviation.

In order to point out the variance underestimation, one hundred independent replicas of a Monte Carlo calculation for a 20-30-20 and a 20-30-18 core geometry as seen in Fig. 4.1, will be made. Every replica will run through exactly the same calculation with 2000 starting neutrons and 600 cycles. The replicas will only differ from each other by the use of different random values in the calculation. This will result in one hundred *independent* replicas.

To check for variance underestimation, the mean of the one hundred independent k_{eff}-values is calculated. Next it is verified how often the confidence intervals of the different replicas contain this mean. If there is no underestimation, we would expect that on average 68 of the 100 confidence intervals (for the 1σ-interval) will contain the mean k_{eff}-value of all the replicas. The number of times the confidence interval is contained, is called the coverage rate. A coverage rate below the size of the confidence intervals (68%, 95% and 99%) can be an indication of variance underestimation.

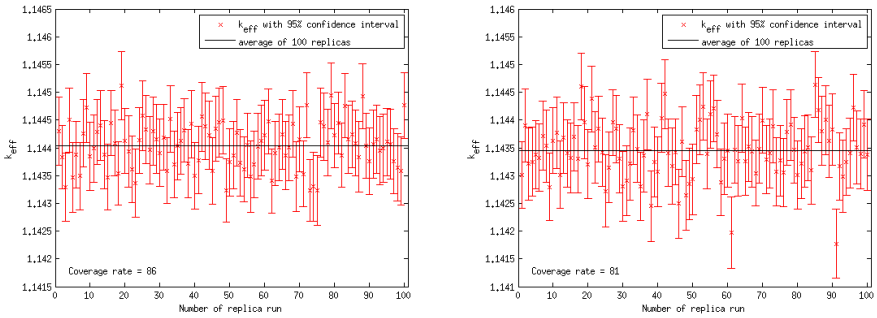
In Fig. 4.2, Fig. 4.3 and Fig. 4.4 one hundred replicas with their confidence intervals (1σ, 2σ and 3σ) and the mean of all the k_{eff}-values are plotted for the two different core geometries. The coverage rates can be found in the lower left corner of the plots and again in Tab. 3. It is clear that

the coverage rates are well below the expected values if there was no cycle-to-cycle correlation. The 20-30-18 core geometry shows the lowest coverage rates of the two systems.



(a) 20-30-20 core geometry (b) 20-30-18 core geometry

Figure 4.2: Actual coverage rates for the 68.3% confidence interval



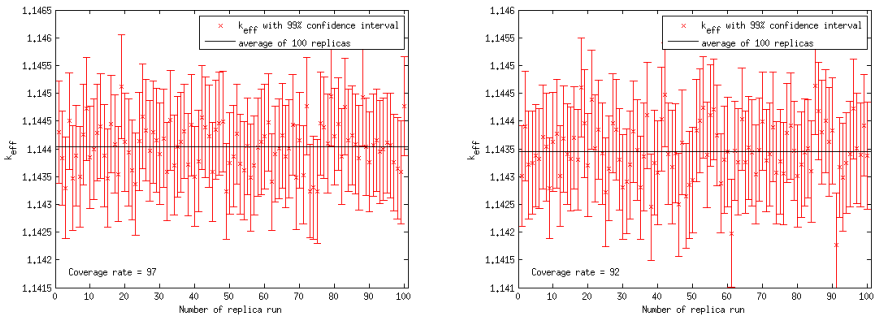
(a) 20-30-20 core geometry (b) 20-30-18 core geometry

Figure 4.3: Actual coverage rates for the 95.4% confidence interval

As to properly conclude that the variance underestimation finds its origin in the correlation between cycles, it is necessary to rule out any statistical possibilities. To do so, one can look at how large the probability is that of a hundred replicas a certain number of times the confidence interval does not contain the mean. This can be done with the help of the binomial distribution:

$$\mathrm{P}(X \leq i) = \sum_{k=1}^{i} \binom{n}{k} p^k (1-p)^{n-k} \tag{4.6}$$

For example, looking at 100 replicas for the 1σ confidence interval, the chance that the coverage rate is only 52 or smaller (as shown in Fig. 4.2a), is 0.0491%. In Tab. 3 the probabilities to have coverage rates as in the numerical examples of Fig. 4.2, 4.3 and 4.4 are displayed. Since all probabilities are well below 1%, the conclusion is drawn that the shown variance underestimation really comes from the intergenerational dependence of the cycles.



(a) 20-30-20 core geometry

(b) 20-30-18 core geometry

Figure 4.4: Actual coverage rates for the 99.7% confidence interval

Table 3: Probabilities of statistical variance underestimation

	20-30-20 geometry		**20-30-18 geometry**	
	Coverage rate	**Probability** (%)	**Coverage rate**	**Probability** (%)
$1\sigma(68.3\%)$	52	0.0491	50	0.0107
$2\sigma(95.4\%)$	86	0.0200	81	$1.4139 \cdot 10^{-5}$
$3\sigma(99.7\%)$	97	0.3515	92	$9.5535 \cdot 10^{-8}$

4.2 Chain-to-chain method

To avoid the cycle-to-cycle correlation in the standard Monte Carlo criticality calculation a new way of calculating the k_{eff} and its variance is introduced. In this new method the cycle-to-cycle correlation is avoided by calculating the k_{eff} and its variance over neutron *chains* instead of calculating it over the *cycles*. A schematic representation of this new method is depicted in Fig. 4.5. Since the number of source neutrons generated in a cycle is correlated with the number of source neutrons in the previous cycle, variance underestimation is present in the standard method. Neutron chains, however, can by no means influence other neutron chains. Therefore, neutron chains will be absolutely uncorrelated and will not give rise to a variance underestimation. To differentiate this attempt at a new method and the standard method, they will be respectively be referred to as the chain-to-chain method and the standard method from now on.

4.2.1 Single neutron chains

A neutron chain consists of all the neutrons that find their origin in the same starting source neutron. So every neutron in a neutron chain has the same original "ancestor". A first attempt is made to calculate a k_{eff} estimate for every neutron chain. So that every single chain will have its own k_{eff} estimate. To incorporate this chain-to-chain method into computer code, every starting neutron is given a number. If a neutron has a collision and will tally a score of $w\bar{\nu}\Sigma_f/\Sigma_t$ with the collision estimator, it will store its origin number with it. Now, the original source neutron of every contribution to the collision estimator is known. The same method applies to the number of actual generated neutrons $n_{\text{fis}} = \lfloor w\nu^*(\Sigma_f/\Sigma_t) + \rho \rfloor$. So, also the origin of every actual generated neutron is known. All the contributions to the collision estimator of one origin neutron are summed up

and divided by all the generated neutrons of the same origin neutron to arrive at the k_{eff} estimate per neutron chain:

$$\hat{k}_{\text{chain}} = \frac{\sum_{\substack{\text{all collisions} \\ \text{in same chain}}} w\bar{\nu}\frac{\Sigma_f}{\Sigma_t}}{\sum_{\substack{\text{all collisions} \\ \text{in same chain}}} \left[w\nu^*\frac{\Sigma_f}{\Sigma_t} + \rho \right]} \tag{4.7}$$

The final value of the k_{eff} is the mean of all chain estimates. The variance of this final value is calculated over the chains with Eq. 3.13. Since the 20-30-18 core geometry showed the largest variance underestimation, this system is used to test the chain-to-chain method on, as the possible gain is the largest.

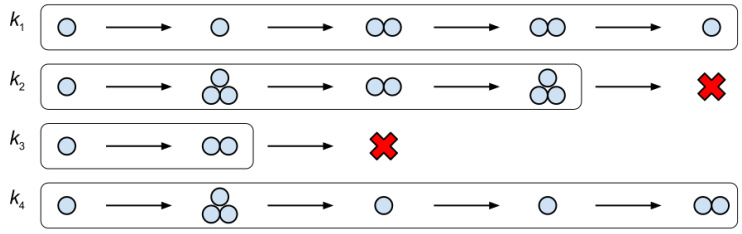


Figure 4.5: Schematic representation of calculating k_{eff} and its variance between neutron chains

Unconverged source

A first attempt of this method tried on the 20-30-18 core geometry with 2000 source particles and 600 cycles. The tallying is started after 50 cycles in order to let the source converge. Note that the starting source neutrons already get their origin number before the first cycle; the system starts with a unconverged source, but the tallying only begins after 50 cycles.

Fig. 4.6 is a plot of the k_{eff} per active neutron chain. The number of neutrons in a specific neutron chain is indicated by the color bar on the right. The final value for the k_{eff} is also plotted by taking the average of all k_{eff} per active neutron chain. This results in a k_{eff} of 1.22 with a standard deviation of $3 \cdot 10^{-2}$.

The standard method is run alongside this chain-to-chain method by also using the standard cycle-to-cycle estimation and not looking at the origin of the estimator contributions and the actual generated fission neutrons. This way the chain-to-chain method can be compared to the standard method. Also one hundred replicas are made in order to calculate the true variance or true standard deviation. This way it is possible to check if the chain-to-chain method, in a single run, approaches the true variance. The standard method yields (on average, because of the replicas) a k_{eff} of 1.1438 with a true standard deviation of $5 \cdot 10^{-4}$. This value is contradictory with the value obtained by the chain-to-chain method and so the chain-to-chain method does not yield the correct value for k_{eff}. The variance is larger, but now it is too large in comparison with the true variance.

Looking at Fig. 4.6 it is clear that there are very few active neutron chains; only 66 of the 2000 starting neutrons has evolved in a neutron chain after 50 cycles. This means that the original source neutron no longer induces any fission neutrons nor does any neutron that finds it origin in that original source neutron; the chain comes to a complete halt. Thus a lot of neutron chains terminate before reaching the 50th cycle where the tallying is started. To look at this behavior in more detail the plot of Fig. 4.7 shows the decrease of active neutron chains per cycle. It is clear that throughout the cycles a rapid decrease of active chains sets in. The chains that do make it

till the 50th cycle do have, in general, relatively long chains and therefore give a quite reasonable estimate. Still, the low number of chains causes a bad overall estimate of the final k_{eff}.

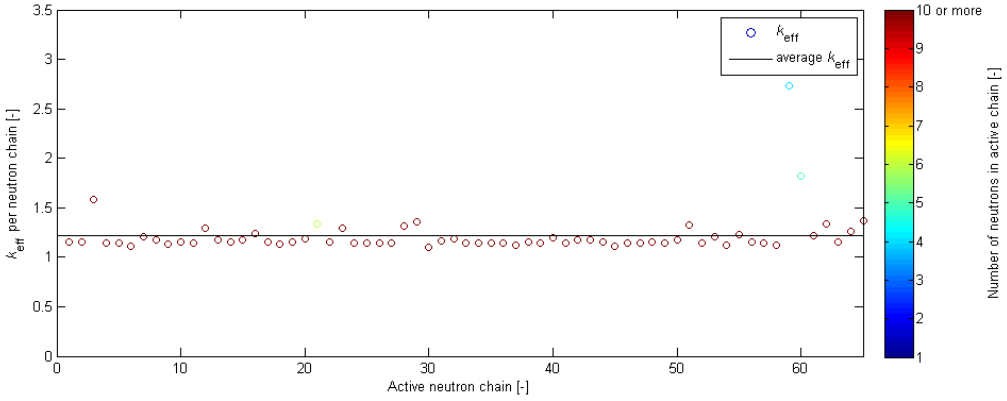


Figure 4.6: k_{eff} of active neutron chains and their neutron numbers starting with unconverged source

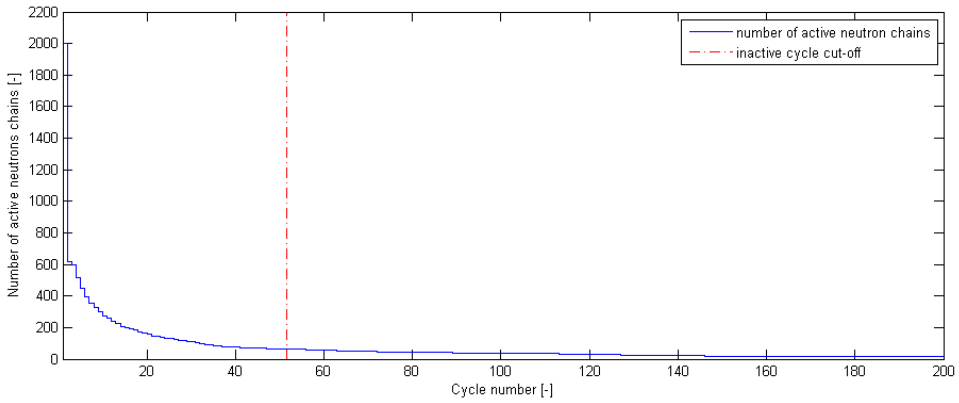


Figure 4.7: Decrease of active neutron chains starting with unconverged source

Converged source

To generate more active neutron chains and to use the particles more efficiently a slightly different approach is tried. Again, the source is let to converge for 50 cycles, but now the induced neutrons of the 50th cycle are taken as origins. This is the same as starting with a converged source and start the tallying immediately. This way less neutron chains are terminated, before the tallying commences.

Fig. 4.8 shows the k_{eff} per active neutron chain for this slightly adjusted approach with a converged source. The color bar on the right is an indication for the number of neutrons in a specific

active neutron chain. The average value of all k_{eff} chain estimates is also plotted, which represents the final k_{eff} value of 1.49 with a standard deviation of $1 \cdot 10^{-2}$.

Again, a standard method calculation is run parallel with the chain-to-chain method in order to compare the two methods. This standard method yields exactly the same result as the standard method in the previous example, because the different way of numbering source neutrons does not effect the standard method. The standard method yields, therefore again, (on average, because of the replicas) a k_{eff} of 1.1438 with a true standard deviation of $5 \cdot 10^{-4}$. The results of the chain-to-chain method are in contradiction with the standard method.

From the decrease of active neutron chains in Fig. 4.9 it is clear that this time more active neutron chains remain (1173). However, a lot of active chains are very short (less than 10 neutrons), as shown by the color coding. These short neutron chains generate bad estimates and explain the incorrect final value for k_{eff}. The large variance can probably be credited to the large variance in the different neutron chain lengths. Different chain lengths will have a larger difference in their k_{eff} estimate.

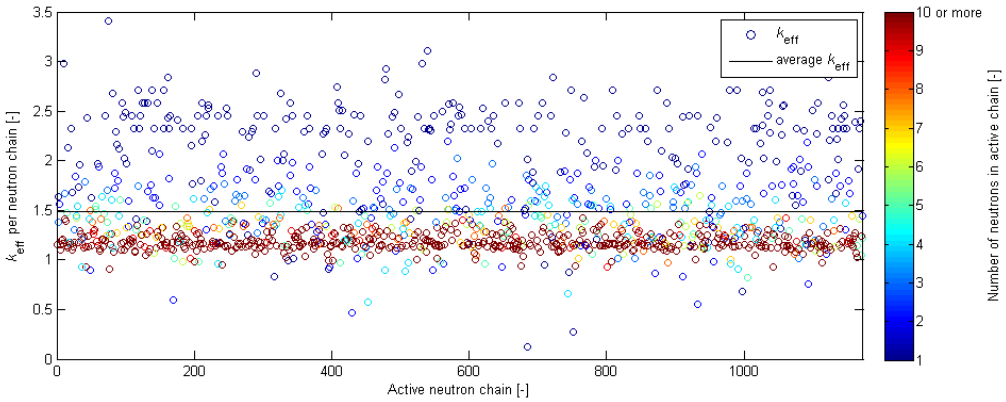


Figure 4.8: k_{eff} of active neutron chains and their neutron numbers starting with converged source

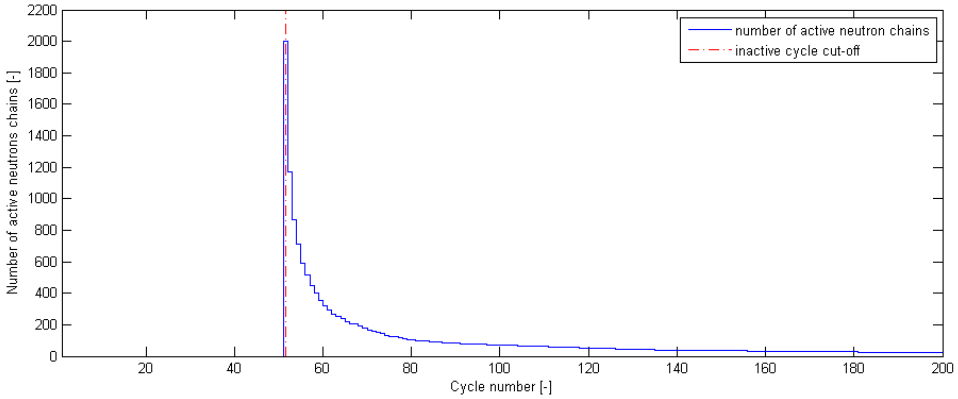


Figure 4.9: Decrease of active neutron chains starting with converged source

Comparison

To compare the decrease of active neutron chains of the two slightly different approaches of the chain-to-chain method, Fig. 4.7 and Fig. 4.9 are plotted together on a log-log scale in Fig. 4.10. The plot of the converged source is translated 50 units to the left in order to align the two plots at the beginning of the active cycles for a better comparison.

In the beginning the unconverged source has a slightly steeper plot, indicating a slightly faster decrease of active neutron histories. However, this small difference is important, because the number of chains that reach the second cycle determines the total number of final active neutron chains. A steep decrease in the first cycle will therefore result in relatively few final active neutron chains; the unconverged source falls off rapidly in the first cycle from 2000 to 617 active chains, whereas the converged source falls off to 1173 chains. This is credited to the converging of the source. Of the 1173 established chains, a lot will terminate in the very first few cycles generating very short chains. This is different for the unconverged source; the short chains, that end somewhere between the first and the fiftieth cycle are not considered. Since the tallying only starts after 50 cycles, it will not include these earlier terminated chains. For the uncoverged source, 6 chains reach the end of all the cycles, as opposed to 8 for the converged source. There seems to be a trade off between few relatively longer chains and more, but also shorter ones. Both chain-to-chain methods, however, do not yield correct results for the multiplication factor k_{eff}.

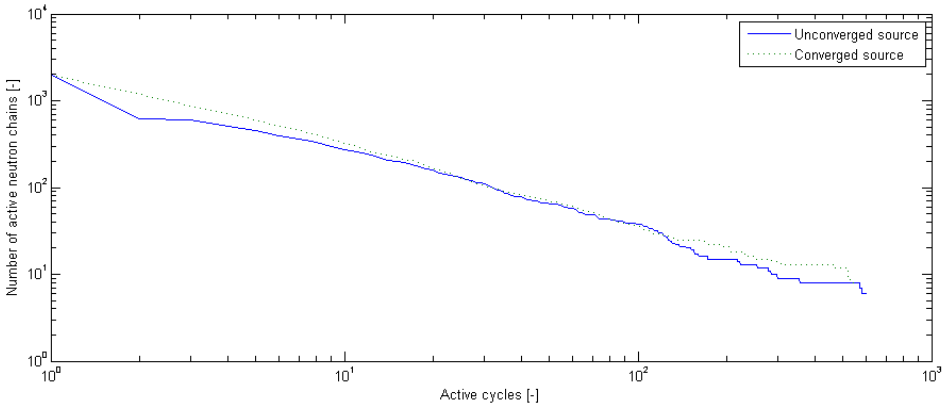


Figure 4.10: Comparison of the unconverged and the converged approach

4.2.2 Batching neutron chains

To even out the bad estimates of the small neutron chains with the much larger ones of the previous converged source example, it is possible to batch different chains together. This is because the average of all the collision contributions and the average of all induced neutrons of a batch will tend to be a better representation than those for just a single chain. This way an estimate for the k_{eff} per batch is made by the following expression:

$$\hat{k}_{\text{batch}} = \frac{\sum_{\substack{\text{all collisions} \\ \text{in same batch}}} w\bar{\nu}\frac{\Sigma_f}{\Sigma_t}}{\sum_{\substack{\text{all collisions} \\ \text{in same batch}}} \left[w\nu^*\frac{\Sigma_f}{\Sigma_t} + \rho \right]} \tag{4.8}$$

The final value for k_{eff} is then acquired by taking the average of all the batch estimates and the variance is calculated over these batches. This is very much the same as the previous chain-to-chain method, but now different neutron chains are grouped together. To illustrate this approach single run results for different batch sizes (200, 100, 40 and 20 neutron chains) are generated and shown in Tab. 4 for the same 20-30-18 core geometry with 2000 initial particles and 600 cycles. The last entry is again the result for the standard method with the true standard deviation acquired by making one hundred replicas.

Now, the results for the two larger batch sizes are in correspondence with the results of the standard method for the k_{eff}. The standard deviation is larger, but it is larger than the true variance calculated with one hundred replicas. The values for the two smaller batch sizes are in contradiction with the results for the standard method. This can be explained by the fact that bad estimates from small chains are no longer evened out; it converges to the single chain approach mentioned earlier.

Table 4: Single run results for batching of neutron chains

Batch size	**Avg k_{eff}**	**Standard deviation**
200	1.142	$1 \cdot 10^{-3}$
100	1.145	$2 \cdot 10^{-3}$
40	1.156	$4 \cdot 10^{-3}$
20	1.171	$6 \cdot 10^{-3}$
-	1.1438	$5 \cdot 10^{-4}$

A closer look at progression of the induced neutrons per batch (for the 10 batches of each 200 starting chains) in Fig. 4.11, reveals that not every batch makes it till the end of the full 600 cycles. For example batches 2,5,7,8,9 and 10 do not contain any neutron chains that survive the 600 cycles. This is not wanted, because longer chains and thus batches that last longer, mean better estimates.

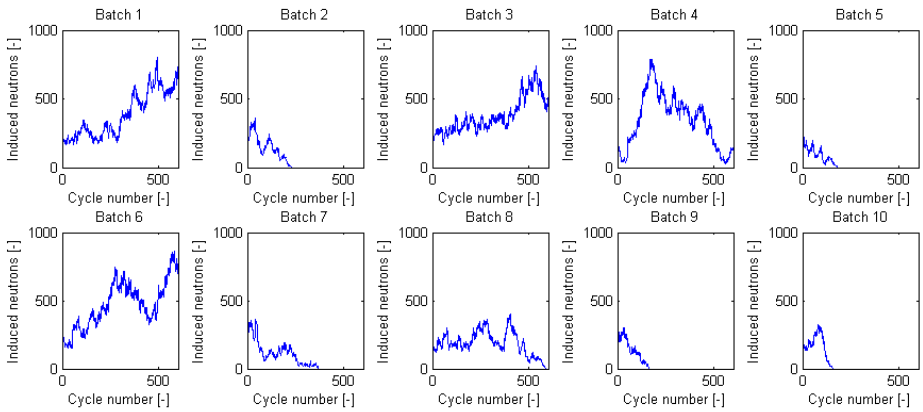


Figure 4.11: Progression of the induced neutrons per cycle for the different batches

Population control per batch

In order to overcome the problem of batches terminating before the end of all the cycles, a population control per batch is introduced; every batch has its own population control. This population control is managed by adjusting the expected number of fission neutrons per cycle *and* per batch:

$$\nu^*_{i+1,\text{batch}} = \frac{\bar{\nu}}{k_{i,\text{batch}}} \frac{N_{\text{batch}}}{N_{i,\text{batch}}} \tag{4.9}$$

where,

$$k_{i,\text{batch}} = \frac{\sum_{j=1}^{i} \left(w\bar{\nu}\frac{\Sigma_f}{\Sigma_t} \right)_{j,\text{batch}}}{\sum_{j=1}^{i} \left\lfloor w\nu^*\frac{\Sigma_f}{\Sigma_t} + \rho \right\rfloor_{j,\text{batch}}} \tag{4.10}$$

This last expression differs slightly with the expression that is used for population control in standard Monte Carlo criticality calculations which uses Eq. 3.45 and Eq. 3.46. Next to the fact that it is now also usable per batch, the summation of all previous collision contributions and previous induced neutrons is used for the current estimate, as opposed to just those of the concerning cycle. This is done because using only the concerning cycle estimates results in heavily varying neutron numbers and in practice leads to the termination of the computer code.

The population control per batch by using Eq. 4.9 and Eq. 4.10 and its consequence on the progression of the induced neutrons per cycle per batch (for the 10 batches of each 200 starting chains) is shown in Fig. 4.12. Now none of the batches has a heavy fluctuating number of induced neutrons per cycle and every batch collectively reaches the end of the cycles. In practice this does not mean that all the chains in a batch reach the end, but mostly that at least one chain in the batch completes all cycles.

It should be noted that by implementation of this new population control the standard method can no longer be run parallel with the chain-to-chain method exactly the same as before. Since now every batch has its own population control, the way neutrons are generated differs from the standard method; it is substantially altered. Still, in order to compare the standard method with the chain-to-chain method, the k_{eff} is evaluated as always in the standard method; by means of Eq. 3.45 and Eq. 3.46.

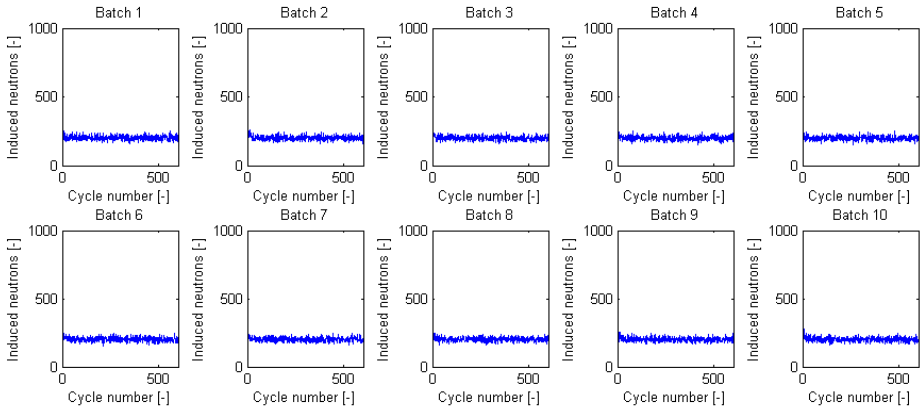


Figure 4.12: Progression of the induced neutrons per cycle for the different batches with population control per batch

This method of batching and population control per batch is tested on the same 20-30-18 core geometry starting with 2000 source neutrons and a converged source after 50 inactive cycles of a total of 600 cycles. Since the way of generating neutrons is no longer exactly the same, also the standard method will yield different results as before. Therefore again one hundred replicas are made with both the chain-to-chain method *with population control per batch* and the standard method. This way the true variance is also established for these methods. This is done for different batch sizes. It should be noted that it becomes more difficult to acquire one hundred replicas for the smaller batch sizes; if the batches get too small, it is probable that, for example, all the 20 consecutive neutron chains in one batch do not make it through the first cycle. If this scenario unfolds, the entire replica run does not yield a result. Therefore, for the smaller batches, more than one hundred runs have been made to arrive at one hundred fruitful replicas.

Tab. 5 and Tab. 6 respectively show the results with the population control per batch for the standard method and the chain-to-chain method. These tables show the average k_{eff} (**Avg** k_{eff}) over all the replicas, the true standard deviation (**True std**), the average standard deviation (**Avg std**), as an indication of the standard deviation per run, and (literally) the sample standard deviation of the one hundred replica standard deviations (indicated with **Std-of-Std**), merely as an indication of how much those single run standard deviations vary relatively. The same results are graphically displayed in Fig. 4.13. The last entry of both tables represent the results for the standard method *without* population control per batch for the purpose of comparison.

Table 5: Replica results for the standard method for different batch sizes with population control per batch

Standard method				
Batch size	**Avg** k_{eff}	**True std**	**Avg std**	**Std-of-Std**
200	1.1430	$10 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
100	1.1413	$14 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
40	1.1371	$17 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
20	1.1348	$13 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$8 \cdot 10^{-6}$
-	1.1438	$5 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$10 \cdot 10^{-6}$

Table 6: Replica results for the chain-to-chain method for different batch sizes with population control per batch

Chain-to-chain method				
Batch size	**Avg** k_{eff}	**True std**	**Avg std**	**Std-of-Std**
200	1.1423	$10 \cdot 10^{-4}$	$8 \cdot 10^{-4}$	$7 \cdot 10^{-4}$
100	1.1406	$14 \cdot 10^{-4}$	$15 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
40	1.1363	$17 \cdot 10^{-4}$	$15 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
20	1.1341	$13 \cdot 10^{-4}$	$12 \cdot 10^{-4}$	$5 \cdot 10^{-5}$
-	1.1438	$5 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$10 \cdot 10^{-6}$

The first thing to notice is the fact that also the standard method is seriously affected by the population control per batch. The k_{eff} gets smaller with smaller batch sizes and the true

standard deviations also vary with different batch sizes. The single run standard deviations are structurally underestimated, which is the original problem of the standard method. According to the true standard deviation of the standard method *without* population control per batch, this underestimation is about $2 \cdot 10^{-4}$. The standard method *with* population control per batch claims that the underestimation is larger, since the true standard deviations are larger by $7 \cdot 10^{-4}$ (smallest underestimation) and $14 \cdot 10^{-4}$ (largest underestimation). The single run standard deviations do not vary a lot, as can be seen from the nearly constant variance of the variance and the nearly straight line in Fig. 4.13. The standard method *with* population control per batch does even yield contradictory results with the standard method *without* population control per batch for the lower three batch sizes using the true standard deviations.

The chain-to-chain method *with* population control per batch does give better results than the previous tried chain-to-chain method approaches; the k_{eff} are closer to the correct value. Still, only the largest batch size yields a k_{eff} that is not contradictory with the standard method *without* population control per batch using the true standard deviations. The single run deviations do however approach the true standard deviations that are calculated with the standard method *with* population control per batch. It should be mentioned that the single run standard deviation varies a lot for the larger batch sizes as can be seen by the relatively large standard deviation of the standard deviations and the heavily fluctuating lines in Fig. 4.13. Consequently, a single run for the larger batches will not yield consistent results. The single run deviations of the smaller batch sizes seem to converge to the true standard deviations calculated with the standard method *with* population control per batch. The single run standard deviations are (on average) larger than the standard deviation for the standard deviation *without* population control per batch.

In short, the chain-to-chain approach yields incorrect values for the k_{eff} for the smaller batches, but has structural larger standard deviations, as intended (although larger than the true standard deviation). The chain-to-chain method for the larger batch size yields a more accurate k_{eff} than the smaller batch sizes, but has (larger) standard deviations that vary too much to use with single run calculations.

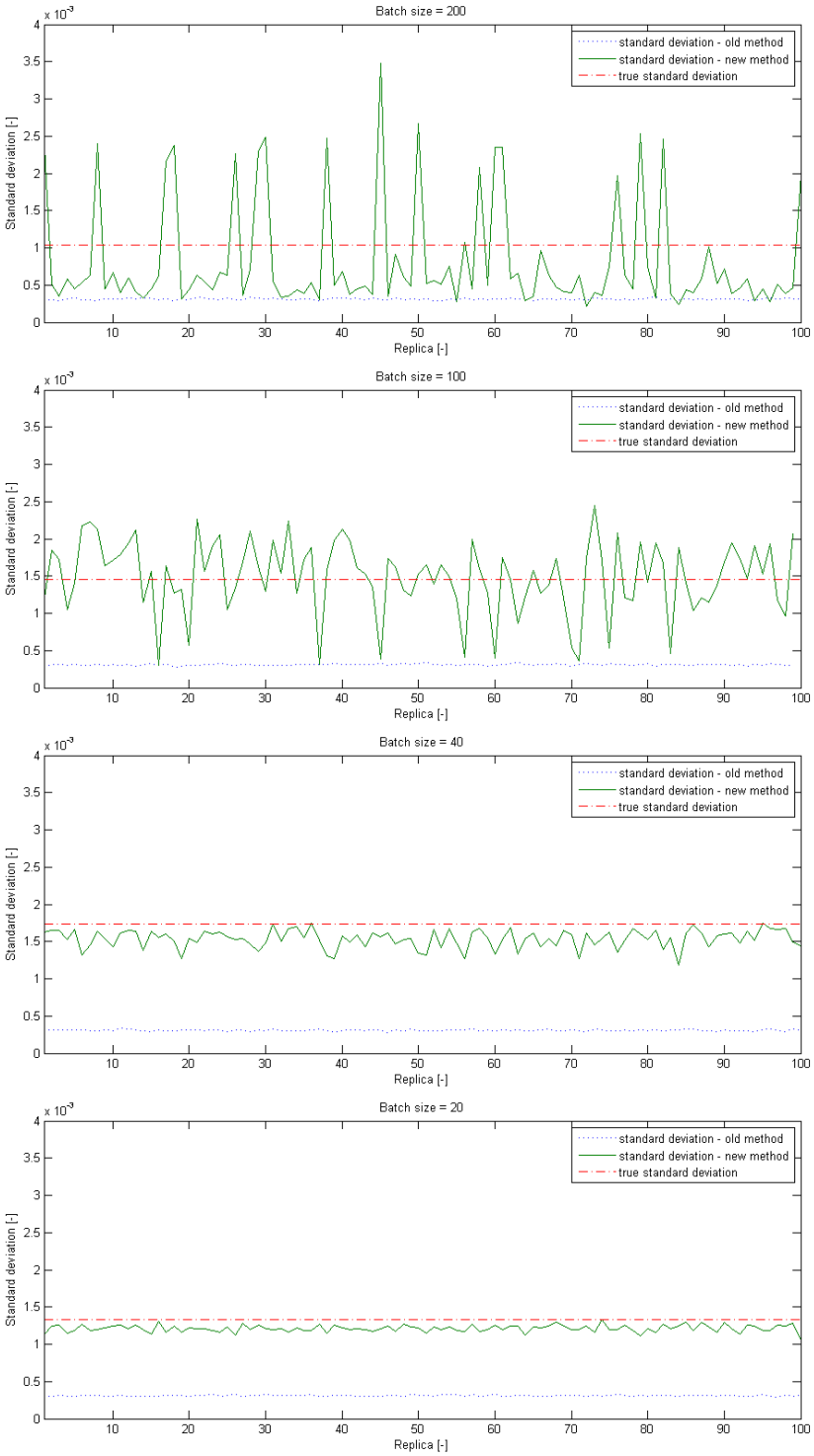


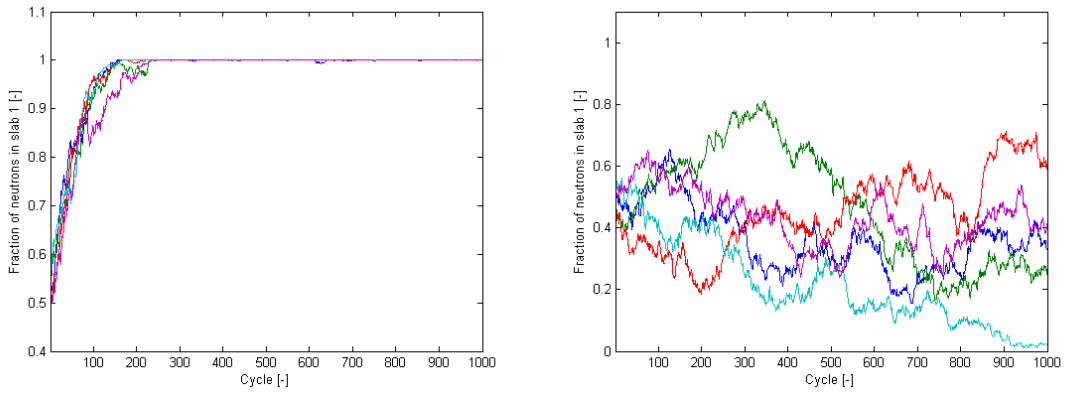
Figure 4.13: Comparison of standard and chain-to-chain method

4.2.3 Fraction of fission neutrons in slabs

Source convergence of loosely coupled systems

Loosely coupled system with a dominance ratio close to unity can give rise to computational problems in a Monte Carlo calculation. When the dominance ratio is close to unity, the decay of the higher-harmonics during the source iteration process is very slow. This can be interpreted as the system having a weak interaction with its own components; the system may not be aware of some of its own components that could have a significant contribution to the k_{eff}. The source distribution may not even reach the true distribution of the system or the tallying is started before the source is converged. This will lead to bad estimates for k_{eff}.

The 20-30-20 and 20-30-18 core geometries are such loosely coupled systems. For these systems the fraction of neutrons in the outer left slab is plotted in Fig. 4.14 as an indication of the source distribution in the system. The standard method *without* population control per batch is used.



(a) 20-30-18 core geometry

(b) 20-30-20 core geometry

Figure 4.14: Five replicas showing the neutron fraction in slab 1 with the standard method

Fig. 4.14a shows the results of 5 independent replica runs for the 20-30-18 core geometry. It is clear that within 200 cycles (nearly) all neutrons are situated in the larger left slab, which is a result of the larger volume of the left slab. This can lead to a bad representation for the k_{eff}, if the right slab is also significant for the total system (and is not identical to the other slab) and too few neutrons reside in that slab.

Fig. 4.14b shows the results of 5 independent replica runs for the 20-30-20 core geometry. Here, the neutron fractions are heavily fluctuating during all 1000 cycles; the source does not seems to converge at all. Since this system has equally large fissile regions, the k_{eff} will not be seriously affected; it does not matter significantly in which slab the neutrons reside.

Effect of chain-to-chain method

To check if the chain-to-chain method *with* population control per batch has any effect on the source convergence, two replicas for each batch size and each system are calculated. The results for the 20-30-20 system are plotted in Fig. 4.15 and the results for the 20-30-18 system is are plotted in Fig. 4.16.

For the 20-30-20 core geometry the chain-to-chain method with population control per batch seems to help the source converge faster than the old method. Note that it does seem to converge

faster, but it is likely that it is converging to an incorrect source distribution. At every batch size the neutron fraction stabilizes eventually. The value to which it seems to converge is not consequent for the different replicas. Again, for this system the fraction of neutrons in each slab will not have a significant impact on the final value of the k_{eff}, but the faster convergence of the source (to the correct source distribution!) will ensure less errors that can be credited to a not yet accurate description of the true source distribution while converging.

Looking at the 20-30-18 system, the chain-to-chain method with population control per batch also seems to lets the source converge. Now, for both replicas the fraction of neutrons in the left slab diminishes as the batch size gets smaller. So, with smaller batch sizes more and more neutrons will be in the smaller slab. This could be an explanation for the descending values of k_{eff} for smaller batch sizes as seen in Tab. 6; apparently too much neutrons reside in the smaller slab as compared to the true source distribution. These neutrons therefore have a too large contribution to the k_{eff}, which is therefore underestimated.

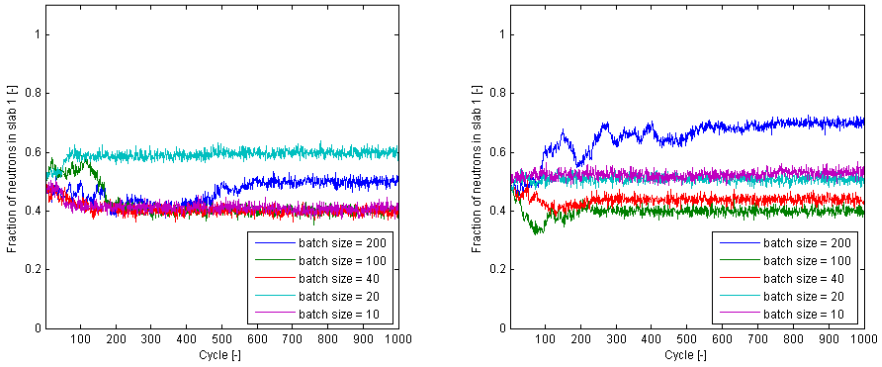


Figure 4.15: Two replicas showing the neutron fraction in slab 1 of a 20-30-20 system for different batch sizes

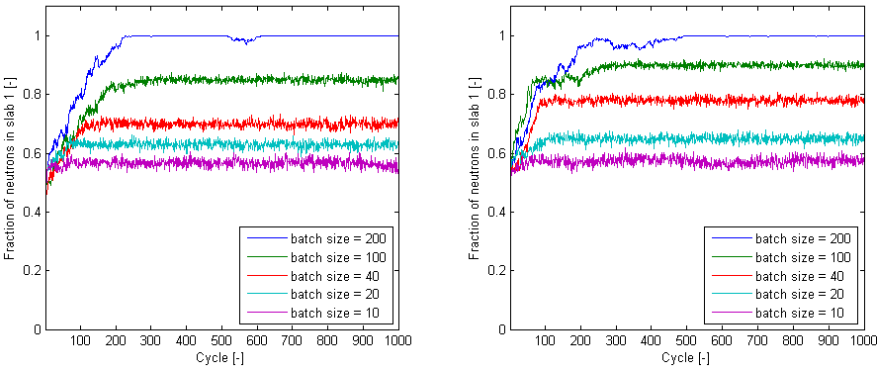


Figure 4.16: Two replicas showing the neutron fraction in slab 1 of a 20-30-18 system for different batch sizes

5 Conclusions & recommendations

5.1 Conclusions

Chain-to-chain method

The chain-to-chain method used per single chain, with the same population control as in the standard method, yields in general incorrect k_{eff} values with a variance that is larger than those of the standard method. The incorrect k_{eff} values are credited to the bad estimates of short neutron chains, which are caused by rapidly decreasing active neutron chains. The large standard deviation is the result of the large variance in the neutron chains itself.

Batching neutron chains together, with the same population control as in the standard method, gives slightly better results for the larger batch sizes, because the batches even out the bad estimates of the shorter chains. The smaller batches suffer more from the bad estimates and have again incorrect k_{eff} values. The standard deviation is for all batch sizes larger than the true standard deviation, again, probably because of the large variance in neutron chain lengths.

Using population control per batch alters the way that the neutrons are generated significantly; the true standard deviations are different for different batch sizes. The k_{eff} values of the chain-to-chain method are not in contradiction with the standard method that also uses the population control per batch. However, the results for the lower batch sizes of the chain-to-chain method are in contradiction with the values of the standard method without population control per batch. The smaller k_{eff} values are explained by the fact that the population control directs too many neutrons to the smaller slab, which have a too large influence on the k_{eff}. All the batch sizes result in standard deviations that are closer to the true standard deviations calculated with the standard method with population control per batch. The standard deviations of the larger batches do itself have a large variance and yield an inconsistent result per single run. The fact that these standard deviations approach the true standard deviations of the standard method with population control per batch is credited to the dismissal of the cycle-to-cycle correlation.

Overall the chain-to-chain method yields incorrect results for the k_{eff} values for the smaller batch sizes and a too inconsistent variance for the larger batch sizes, which makes it an unreliable method so far.

Effect on source convergence

The fraction of neutrons in the larger slab of the 20-30-18 core geometry, decreases with the smaller batch sizes. This means that the population control per batch resulted in a different source distribution; too much particles were directed to the wrong places. This is probably the cause of the smaller values for k_{eff} with smaller batch sizes.

5.2 Recommendations

Since, in general, the chain-to-chain method yields incorrect k_{eff} values, it is sensible to first focus on this shortcoming of the approach. The large variance of the neutron chain lengths is probably the cause of this. The population control per batch is implemented as solution to this problem, but in turn changes the source distribution significantly and yields also incorrect results. For direct further research these two points need attention.

5.2.1 Reducing variance of chain lengths

Larger batch sizes

The batching of neutron chains with use of the population control of the standard method led to better k_{eff} values for the larger batches. Large batches are able to suppress the bad estimates of the short chains. Larger batches may bring better results although probably more particles are needed to generate a sufficient amount of batches. This will, in turn, make the calculation less efficient. It would be interesting to see if the values would eventually correspond with the standard method.

More active chains

A way to more efficiently use the starting particles could be advantageous, since now a lot of the starting source neutrons do not generate an active chain. Maybe the expected number of fission neutrons can be overestimated in the first cycle(s) in order to ensure that every source neutron creates an active neutron chain. Later on in the next cycles, the expected number of fission neutrons can be adjusted as usual. This way every starting source neutron will create an active neutron chain that can be included in the k_{eff} calculation and the calculation will be more efficient.

5.2.2 Source distribution effect of population control

Redirecting particles

The population control per batch is implemented as a solution to the large variance of neutron chain lengths. However, the population control per batch alters the source distribution significantly. Neutrons are redirected to places with low flux and the k_{eff} values are off. In order to prevent this it might be favorable to redirect the neutrons to places where a high contribution to the k_{eff} is made. This could be done by the use of some sort of multiplier that resembles the multiplier that is used in standard calculations. Ideally, the population control per batch is maintained and the particles are also directed to places with a high k_{eff} contribution. If the population control per batch can be perfected, the next step might be to implement a population control per single chain. This way every starting source neutron will result in a large neutron chain that has the possibility to yield better k_{eff} estimates.

Acknowledgements

I am particularly thankful for the help received by Bart Sjenitzer, who is an inexhaustible source of knowledge regarding Monte Carlo calculations and everything Linux.

References

[CC75] L.L. Carter and E.D. Cashwell. *Particle-Transport Simulation with the Monte Carlo Method*. ERDA critical review series. Technical Information Center, Office of Public Affairs, U.S. Energy Research and Development Administration, 1975.

[Dek05] M. Dekking. *A Modern Introduction To Probability And Statistics: Understanding Why and How*. Springer Texts in Statistics. Springer, 2005.

[DH76] J.J. Duderstadt and L.J. Hamilton. *Nuclear Reactor Analysis*. Wiley, 1976.

[HS11] J.E. Hoogenboom and B.L. Sjenitzer. Monte Carlo Lectures. PNR/R^3/TNW/TU Delft, 2011.

[LK91] I. Lux and L. Koblinger. *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*. CRC Press, 1991.

[Urb95] T.J. Urbatsch. *Iterative Acceleration Methods for Monte Carlo and Deterministic Criticality Calculations*. PhD thesis, University of Michigan, 1995.