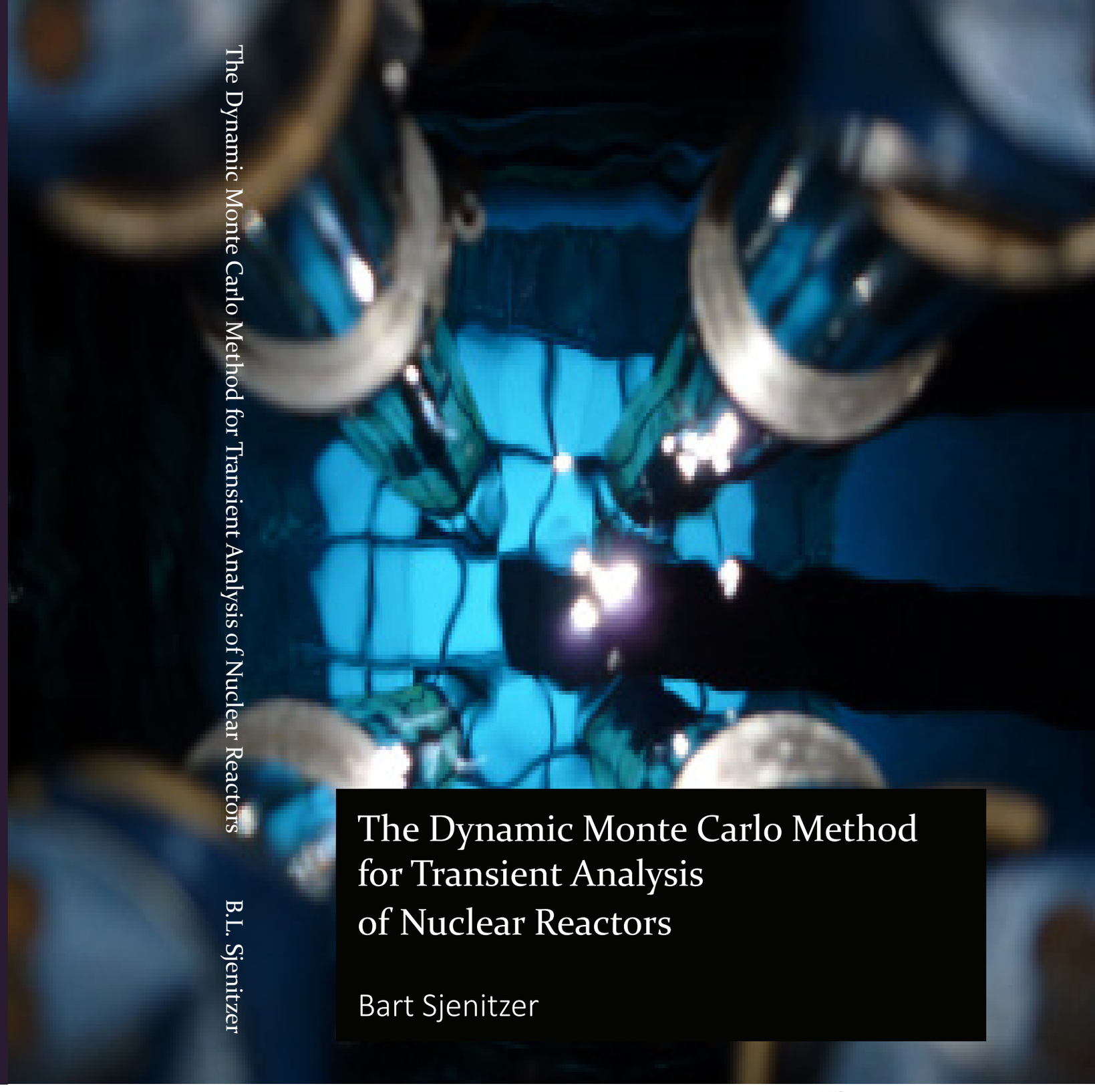


The Dynamic Monte Carlo Method for Transient Analysis of Nuclear Reactors

B.L. Sjenitzer

The Dynamic Monte Carlo Method for Transient Analysis of Nuclear Reactors

Bart Sjenitzer

TU Delft Delft University of Technology

The Dynamic Monte Carlo Method for Transient Analysis of Nuclear Reactors

The Dynamic Monte Carlo Method for Transient Analysis of Nuclear Reactors

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 vrijdag 05 juli 2013 om 10:00 uur
door

Bart Laurens SJENITZER

Natuurkundig ingenieur

geboren te Amsterdam

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. F.J. Beekman,	Technische Universiteit Delft
Prof. dr. A.J. Koning,	Uppsala University
Prof. F.H.J. Redig,	Technische Universiteit Delft
Dr. F. Malvagi,	Commissariat à l'Énergie Atomique
Prof. dr. ir. C.W.E. van Eijk,	Technische Universiteit Delft

ISBN 978-90-8891-657-1

Keywords: Nuclear, Monte Carlo, Safety, Transient, Dynamic, Coupled, Neutron, Transport

The research described in this thesis was performed in the section Nuclear Energy and Radiation Applications (NERA), of the department of Radiation Science and Technology (RST), of the Delft University of Technology, Delft, the Netherlands.

Printed by: Proefschriftmaken.nl || Uitgeverij BOXPress
PRINTED IN THE NETHERLANDS

voor Anne

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

Contents

Part I

Introduction

1

INTRODUCTION

Picture: G. Schoonewille

For nuclear safety it is of the highest importance to be able to predict the exact power production in a nuclear reactor during a transient. Such a transient might occur during regular operation, but also in an accident scenario. It is of vital importance to calculate the temperature during such a transient accurately, in order to predict fuel damage and other consequences. When performing this kind of dynamic analyses there are multiple problems to be solved, involving both thermal-hydraulics and neutronics which can be tightly coupled.

The main part of this thesis describes the development of a novel method, utilising Monte Carlo simulation to calculate the neutronics during a transient, which is more accurate than the present state-of-the-art methods. This novel Monte Carlo method is then coupled to a thermal-hydraulics solver and a demonstration is given of a coupled transient in a mini core.

NURISP project

This work is part of the European FP7 project NURISP, which develops the nuclear reactor simulation platform NURESIM (Chauliac et al., 2011) and a large part belongs to task T1.1.4 of the NURISP project: *Development of the basic Monte Carlo techniques for long-time kinetic and dynamic calculations*. This task is part of subproject 1 (SP1), Core Physics, and in this subproject it is part of work package 1 (WP1.1), Advanced Monte Carlo Core Physics development.

1.1 State-of-the-art transient-analysis methods

Up to now, it was only possible to calculate the neutronic part of a transient using deterministic or hybrid methods. Currently many state-of-the-art methods are based on the improved quasi-static method, but this method is now well over 50 years old (Henry, 1958) and this method relies on the factorisation of the neutron flux in a shape and an amplitude function. The shape function can be solved by performing a steady-state calculation using, for example, discrete ordinates (C. Bentley, R. DeMeglio, M. Dunn, *et al.*, 1997; Goluoglu and Dodds, 2001) or Monte Carlo (Bentley et al., 1997; Shayesteh and Shahriari, 2009; Yun et al., 2008). Factorisation is not always a good approximation, especially not with strong transients, when the positive or negative reactivity is larger than a dollar and when the flux profile changes.

Another issue with these methods is the need of kinetic parameters to solve the time-dependent part. These parameters are time dependent and there are many different definitions and options to calculate them (Carta et al., 2011; Meulekamp and Van Der Marck, 2006; Verboomen et al., 2006), with each definition yielding a slightly different result. Furthermore, although many optimizations have been proposed (Dulla et al.; Meulekamp et al., 2005; Picca et al., 2011), these methods still rely on the factorization of the flux.

Also direct methods are used for transient analysis, but they depend on simplified transport models, e.g. diffusion theory (Boer et al., 2010) and S_N methods (Alcouffe and Baker, 2005). These methods all discretise phase space: splitting the problem not only geometrically, but also angularly and energetically. The geometry is split into cells, the direction in discrete angles and the neutron energy in groups. All these parts of the phase space are connected and the result of one part of phase space is the boundary condition of the next part and for the outer boundaries a boundary condition is needed. The solution is being iterated until it has converged. The main difference between the methods is in the way the transport from one cell to another is modelled. Here the key features of a few methods are described and more extensive descriptions can be found in Bell and Glasstone (1979) and Lewis and Miller. (1984).

With diffusion theory, the neutrons are assumed to diffuse through the system according to Fick's law. Neutron diffusion ignores the possible angular dependency of the neutron flux and when a large density gradient is present it is not accurate. For more details about diffusion theory, see Duderstadt and Hamilton (1976).

More advanced methods model the neutron transport itself. These methods need to discretise the angular dependence of the neutron flux. The neutrons will be transported along discrete angles, defined by spherical harmonics (P_n-theory) or discrete ordinates (S_n-theory). Also the method of characteristics, which solves the integral neutron transport equation, uses discrete angles. These methods describe the anisotropic behaviour of neutron transport better than diffusion theory, but it can prove difficult to use enough angles to describe the neutron transport. If too few angles are used, ray effects can occur (Cho and Chang, 2009).

However, the main limitation of these direct deterministic methods is the required discretisation. The neutron flux exists in a 7-dimensional phase space (x,y,z,ω_1,ω_2,E,t) and when each of these dimensions is discretised in 100 intervals, the problem already consists of $(10^2)^7 = 10^{14}$ unknowns. This yields an astronomically large system of equations, several orders of magnitude beyond the capacity of any existing computer to solve (Larsen, 2011). Therefore a stochastic method might be an interesting option.

1.2 Stochastic methods for reactor analysis

A stochastic method does not try to find a deterministic solution of a problem, but tries to find a stochastic process which has the same mean result as the original problem. In physics this results often in simulating the actual physics, although non-analog techniques can be applied to increase the efficiency of the simulation.

This method yields, just like the deterministic ones, a numerical approximation of the solution. However, with this method the accuracy can be exactly determined and increased if necessary; the error is only statistical in nature. Therefore Monte Carlo is used regularly

in reactor physics to find steady-state solutions for a nuclear reactor and to perform source-detector calculations.

Time-dependent Monte Carlo is applied in other fields, such as implicit Monte Carlo for non-linear radiation transport (Fleck Jr. and Cummings Jr., 1971; Martin and Brown, 2001) or neutrino transport (Abdikamalov et al., 2012). Some investigations on time-dependent neutron transport were performed by Kaplan (1958). His paper proposes some methods which can be used for transient analysis, such as population control at the time boundaries, moving weight windows and collision biasing. However, it was not possible at that time to perform an analysis on the kinetic or dynamic behaviour of a nuclear reactor.

The major drawback of the Monte Carlo method is its computational cost. However, the required computer time is being reduced rapidly via two ongoing developments. The first development is the rapid increase in computer power, which is described by Moore's law, which predicts a doubling of computing power every 18 months. Although this law is not exact, in some periods the development was slower (Schaller, 1997), it does illustrate the exponential increase in computing power. Recently, the processors have not increased in speed, but the number of processors has been increasing instead. These developments do not reduce the computational cost, but do reduce the cost of the computation.

Monte Carlo can also take full advantage of the extra processors, since the method is made of independent processes, from which an average is calculated and these processes can all be simulated on separate processors. Although the scaling to many processors is not as straight forward as it may seem (Hoogenboom, 2012), Monte Carlo codes can be adapted to run in a massively parallel fashion (Romano and Forget, 2013).

With more computing power, the results of the Monte Carlo method automatically improve. The simplest way of reducing the variance is simply to calculate more samples. Identifying this automated improvement, Kord Smith calculated in 2003 that with the techniques of that time, it would be possible to perform a routine Monte Carlo calculation of an entire core, with 1 % accuracy in 2030 and he challenged the Monte Carlo community to demonstrate that this could be achieved much sooner (Smith, 2003).

In fact, the method had already been much improved over the years, with many researchers who are working on variance reduction techniques, i.e. techniques to reduce the variance without increasing the computational effort. Many of these variance reducing methods are hybrid methods, using a deterministic calculation to improve the Monte Carlo method. One of the more effective improvements is the FW-CADIS method by Wagner et al. (2007), which calculates deterministically the best weight windows for a complex system. Also a zero-variance scheme (Christoforou and Hoogenboom, 2011; Christoforou, 2010) has been investigated, which also uses a deterministic method to bias the Monte Carlo simulation in such a way that, if the deterministic solution is perfect, there will be no variance in the final result of the Monte Carlo calculation. A more exotic method is the correcton method, which actually redefines the problem and uses Monte Carlo techniques to calculate the multiplicative error of a deterministic solution (Becker et al., 2007; Huisman,

2012; Hennink, 2012).

To monitor the advancement in the speed of Monte Carlo calculations a benchmark has been set up by Hoogenboom and Martin (2009) and although the Kord Smith challenge has not been met, the Monte Carlo community is getting close (Kelly et al., 2012).

With this increased computer power and advanced Monte Carlo methods, new possibilities become available. Therefore it is the right moment to investigate another drawback of the Monte Carlo method, which is its limitation to the two simulation modes: source-detector and criticality. Although this limitation is not mentioned in literature, this is a large hiatus, which will be addressed in this thesis.

1.3 Aim and structure of this thesis

As explained, the current methods for transient analysis, deterministic and hybrid, although in general fast, still require numerous approximations, e.g. homogenisation, diffusion theory, few-group cross sections and factorisation. With such approximations it is difficult to predict the error in the final result and their validity for each specific case must be determined separately, which is especially difficult for new and unique reactor designs. For transient analysis, which is crucial for safety calculations, it is desirable to have an higher-fidelity method. Therefore, the Monte Carlo method is an interesting option to perform reliable and accurate transient analysis.

In this thesis a new and unique dynamic Monte Carlo method is developed, which extends the possibilities of the Monte Carlo method and which enables the possibility to analyse the transient behaviour of a nuclear reactor, without any approximations to the geometry or any discretisation. This method can perform transient analysis on the milliseconds to minutes scale and can handle changing system properties, feedback effects, delayed neutrons and prompt neutron fluctuations.

Monte Carlo calculations are, however, computationally expensive. Therefore the aim of this thesis is not to develop a tool for the design of reactors, but a tool to perform validation calculations on reactors, which is much cheaper than performing experiments.

The first part of the development of this novel method is the development of new Monte Carlo simulation techniques. In Chap. 2 an innovative way of sampling precursors and delayed neutrons is developed. This enables the modelling of long-lived particles such as delayed-neutron precursors together with short-lived particles such as prompt neutrons. Next, in Chap. 3 a new variance reduction technique is introduced to reduce the variance caused by the fluctuations in prompt neutron chain lengths.

Then, in the second part of this thesis, the development of the dynamic method is addressed. In this part the focus is more on the implementational details of the dynamic Monte Carlo method. The simulation scheme has to be adjusted, the initial conditions must

be calculated and the tallying is adjusted. First, a purpose-built kinetic Monte Carlo code is developed and tested in a simple system in Chap. 4 and then the method is extended to be generally applicable in Chap. 5. Next, the kinetic model is coupled to thermal-hydraulic feedback, making the method truly dynamic in Chap. 6. Finally, Chap. 7 gives conclusions and an outlook to possible further developments of and improvements in the method.

1.4 Bibliography

E. Abdikamalov et al. A new Monte Carlo method for time-dependent neutrino radiation transport. *The Astrophysical Journal*, **755**, 111, 2012.

R. E. Alcouffe and R. S. Baker. *A time-dependent, parallel neutral particle transport code system.* LANL, LA-UR-05-3925 edition, 2005.

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**, 2007.

G. Bell and S. Glasstone. *Nuclear reactor theory.* Van Nostrand Reinhold Company, 1979.

C. Bentley et al. Development of a hybrid stochastic/deterministic method for transient, three-dimensional neutron transport. *Transactions ANS*, **76**, 221, 1997.

B. Boer et al. Validation of the DALTON-THERMIX code system with transient analyses of the HTR-10 and application to the PBMR. *Nuclear Technology*, **170**, 306, 2010.

C. Bentley, R. DeMeglio, M. Dunn, *et al.* Development of a time-dependent three-dimensional deterministic neutron transport method. *Transactions ANS*, **76**, 219, 1997.

M. Carta et al. Calculation of the effective delayed neutron fraction by deterministic and Monte Carlo methods. *Science and Technology of Nuclear Installations*, **2011**, 2011.

C. Chauliac et al. NURESIM - A European simulation platform for nuclear reactor safety: Multi-scale and multi-physics calculations, sensitivity and uncertainty analysis. *Nuclear Engineering and Design*, **241**, 3416, 2011.

N. Cho and J. Chang. Some outstanding problems in neutron transport computation. *Nuclear Engineering and Technology*, **41**, 2009.

S. Christoforou. *A Zero-variance Based Scheme for Monte Carlo Criticality Simulations.* Ph.D. thesis, Delft University of Technology, 2010.

S. Christoforou and J. E. Hoogenboom. A zero-variance-based scheme for Monte Carlo criticality calculations. *Nuclear Science and Engineering*, **167**, 91, 2011.

J. J. Duderstadt and L. J. Hamilton. *Nuclear Reactor Analysis.* Wiley, New York, 1976.

S. Dulla, E. H. Mund and P. Ravetto. The quasi-static method revisited. *Progress in Nuclear Energy*, **50**, 908, ????

J. Fleck Jr. and J. Cummings Jr. An implicit Monte Carlo scheme for calculating time and frequency dependent nonlinear radiation transport. *Journal of Computational Physics*, **8**, 313, 1971.

S. Goluoglu and H. L. Dodds. A time-dependent, three-dimensional neutron transport methodology. *Nuclear Science and Engineering*, **139**, 248, 2001.

A. Hennink. *Correcton method for neutron transport calculations with real time flux estimation*. Bachelor thesis, Delft University of Technology, 2012. PNR-131-2012-004.

A. F. Henry. The application of reactor kinetics to the analysis of experiments. *Nuclear Science and Engineering*, **3**, 52, 1958.

J. Hoogenboom. Is Monte Carlo embarrassingly parallel? In *International Conference on the Physics of Reactors 2012, PHYSOR 2012: Advances in Reactor Physics*, volume 2, 936–951, 2012.

J. E. Hoogenboom and W. R. Martin. A proposal for a benchmark to monitor the performance of detailed Monte Carlo calculation of power densities in a full size reactor core. In *Proceedings of American Nuclear Society - International Conference on Mathematics, Computational Methods and Reactor Physics 2009, M and C 2009*, 2009.

M. Huisman. *Variance reduction in a three-dimensional space using the correcton method*. Bachelor thesis, Delft University of Technology, 2012. PNR-131-2011-001.

E. L. Kaplan. *Monte Carlo Methods for equilibrium solutions in neutron multiplication*. UCRL-5275-T. Lawrence Livermore National Laboratory, 1958.

D. J. Kelly, T. M. Sutton and S. C. Wilson. MC21 analysis of the Nuclear Energy Agency Monte Carlo performance benchmark problem. In *Proceedings of PHYSOR-2012 - American Nuclear Society Topical Meeting on Reactor Physics*. Knoxville, 2012.

E. W. Larsen. Hybrid Monte Carlo-deterministic methods for neutral particle transport problems. Frédéric Joliot/Otto Hahn Summer School, 2011.

E. Lewis and W. Miller. *Computational Methods of Neutron Transport*. John Wiley & Sons, Inc, 1984.

W. Martin and F. Brown. Comparison of Monte Carlo methods for nonlinear radiation transport. In *Proceedings of ANS Topical Meeting, International Topical Meeting on Mathematics and Computation*, 2001.

R. K. Meulekamp, J. C. Kuijper and M. Schikorr. Point Genetics: A New Concept to Assess Neutron Kinetics. *Nuclear Science and Engineering*, **149**, 237, 2005.

R. K. Meulekamp and S. C. Van Der Marck. Calculating the effective delayed neutron fraction with Monte Carlo. *Nuclear Science and Engineering*, **152**, 142, 2006.

P. Picca, R. Furfaro and B. D. Ganapol. A hybrid transport point-kinetic method for simulating source transients in subcritical systems. *Annals of Nuclear Energy*, **38**, 2680, 2011.

P. K. Romano and B. Forget. The OpenMC Monte Carlo particle transport code. *Annals of Nuclear Energy*, **51**, 274, 2013.

R. Schaller. Moore's law: past, present and future. *Spectrum, IEEE*, **34**, 52 , 1997.

M. Shayesteh and M. Shahriari. Calculation of time-dependent neutronic parameters using Monte Carlo method. *Annals Nuclear Energy*, **36**, 901, 2009.

K. Smith. Reactor core methods. In *Invited lecture at the M&C 2003 International Conference*, 2003.

B. Verboomen, W. Haeck and P. Baeten. Monte Carlo calculation of the effective neutron generation time. *Annals of Nuclear Energy*, **33**, 911, 2006.

J. C. Wagner, E. D. Blakeman and D. E. Peplow. Forward-weighted CADIS method for global variance reduction. In *Transactions of the American Nuclear Society*, volume 97, 630–633, 2007.

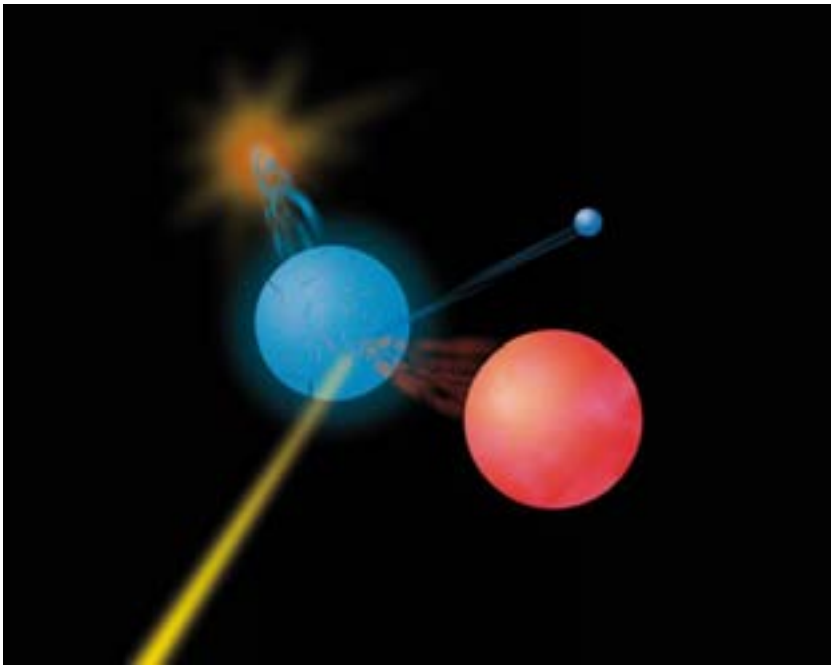
S. Yun, J. W. Kim and N. Z. Cho. Monte Carlo space-time reactor kinetics method and its verification with time-dependent SN method. In *Proceedings of PHYSOR-2008 - American Nuclear Society Topical Meeting on Reactor Physics*. Interlaken, 2008.

Part II

New Monte Carlo Techniques

2

Simulation of Precursors



Picture: Boxpress

Precursors play a crucial role in nuclear reactor physics. They are the reason why reactors can be controlled by mechanical means. In this chapter first the physics of precursors will be discussed in section 2.1. Then the sampling techniques used to simulate precursors in a Monte Carlo calculation are developed in section 2.2 and in section 2.3 variance reduction techniques are created, for controlling the precursor population. Finally the sampling of a source distribution is addressed in section 2.4 and there is a conclusion on the precursor simulation in section 2.5.

2.1 Physics of nuclear fission

2.1.1 Neutron production

When simulating the kinetic or dynamic behaviour of nuclear reactors delayed neutrons and therefore precursors play an important role. In static calculations all fission neutrons are considered to be released without any delay (prompt). A prompt neutron is released within a time scale of approximately 10^{-14} s (Keepin, 1965), this is a shorter time span than the accuracy of a simulation and therefore the release can be considered instant. In reality not all neutrons are prompt, some of the neutrons are released after a considerable delay.

The total number of neutrons emitted at a fission event can vary from zero up to even more than six. The average number of neutrons is denoted by $\overline{\nu}$. The number of neutrons released can be described as a Gaussian distribution around this number $\overline{\nu}$ with a standard deviation of 1.08 (Terrell, 1957). The value of the standard deviation is independent of $\overline{\nu}$.

Prompt neutrons

A prompt neutron is emitted by an exited fission product (Bohr and Wheeler, 1939). The energy level of the exited fission product is much higher than the binding energy of a neutron. One of the options is then to emit a neutron as depicted in Fig. 2.1. Here E^* is the energy of the exited state of the fission product, which is much larger than the binding energy of a neutron E_{Bn}. Now a gamma or a neutron can be emitted, both reducing the energy of the nucleus. When a neutron is emitted, the nucleus changes and has a new the binding energy E_{Bn}. In the figure A represents the number of nucleons and Z is the atomic number of the isotope. This process can happen multiple times until the energy of the fission product is below the binding energy of a neutron. The average number of prompt neutrons produced is denoted by $\overline{\nu_p}$.

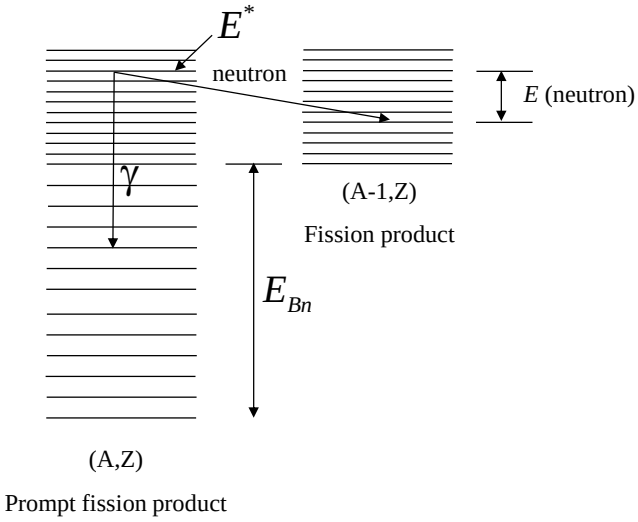


Figure 2.1: A prompt neutron can be produced when the energy of a fission product is higher than the binding energy.

Delayed Neutrons

Delayed neutrons are also emitted by the fission products of a fission event. In this case however, the fission product decays via beta emission. This process is depicted in Fig. 2.2. After this beta decay, again the energy of the resulting nucleus can be higher than the binding energy of a neutron, directly resulting in the release of a neutron. The beta decay is not instantaneous, but happens after some time, delaying the release of the neutron. Therefore this neutron is called a delayed neutron.

The fission product which will emit the neutron is called a delayed neutron precursor, or in short a precursor. Not all nuclei that have a beta-decay will emit a delayed neutron, because it is also possible to emit a gamma instead of a neutron. Also if the beta particle has too much energy, the energy of the resulting nucleus can be too low to emit a neutron. By definition, only the fission products that eventually will decay and produce a delayed neutron are called precursors. The average number of delayed neutrons released per fission is called $\overline{\nu_d}$.

The so-called β-fraction is defined as the delayed fraction of total number fission neutrons released at a fission event:

$$\beta \equiv \frac{\overline{\nu_d}}{\overline{\nu}} \tag{2.1}$$

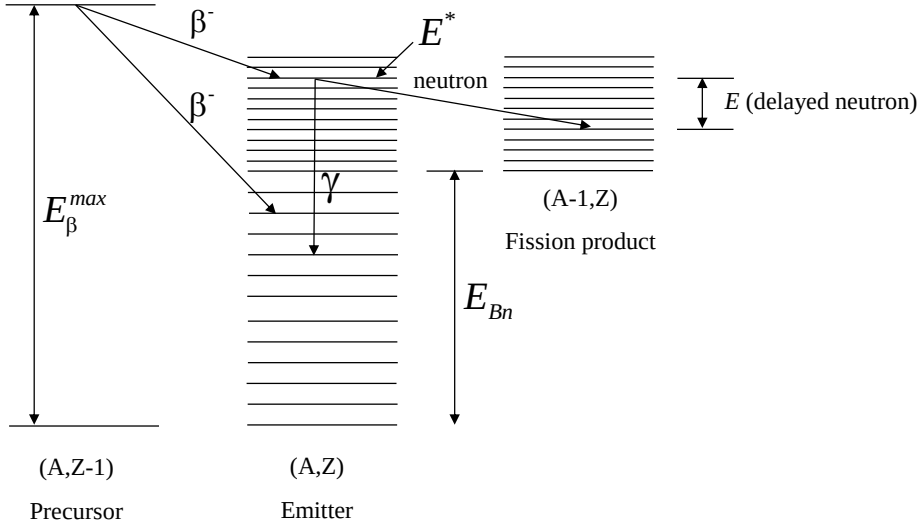


Figure 2.2: A delayed neutron can be produced after the beta-decay of an unstable isotope.

2.1.2 Precursor decay

The decay of a precursor is a stochastic process, which can be described by an exponential distribution:

$$p(t) = \lambda e^{-\lambda t} \tag{2.2}$$

where $p(t)$ is the probability the precursor has its beta-decay at time t and λ is the decay constant of the precursor. The average lifetime of a precursor is:

$$\overline{l} = \int_0^\infty t\, p(t) \mathrm{d}t = \frac{1}{\lambda} \tag{2.3}$$

There are many different precursors, a few hundred have been identified so far (Pfeiffer et al., 2002), and each of them has its own decay time. A few well known examples are ^{87}Br, ^{88}Br, ^{137}I. To make this more practical for reactor applications, precursors are combined into six or eight precursor families. These families all have their own characteristic, average precursor lifetime. The precursor families are considered to have a pure exponential decay probability with decay constant λ_i.

The creation of precursors happens in a fraction of all the fission reactions, so the net production of precursors becomes, as explained in Duderstadt and Hamilton (1976):

$$\frac{\partial C_i(\mathbf{r},t)}{\partial t} = \beta_i \overline{\nu} \Sigma_f \phi(\mathbf{r},t) - \lambda_i C_i(\mathbf{r},t) \tag{2.4}$$

Table 2.1: Delayed fraction and decay constants of ^{235}U when the incident neutron has an energy of 1eV

Family i	λ (s^{-1})	β_i/β
1	0.0127	0.0340
2	0.0283	0.1501
3	0.0425	0.0992
4	0.1330	0.2001
5	0.2925	0.3122
6	0.6665	0.0932
7	1.6348	0.0872
8	3.5546	0.0240

Table 2.2: Delayed fraction and decay constants of ^{239}Pu when the incident neutron has an energy of 1eV

Family i	λ (s^{-1})	β_i/β
1	0.0127	0.0288
2	0.0283	0.2250
3	0.0425	0.0951
4	0.1330	0.1490
5	0.2925	0.3510
6	0.6665	0.0370
7	1.6348	0.0974
8	3.5546	0.0168

Here β_i is the delayed fraction, considering only family i. C_i is the precursor concentration of the ith family. All β_i together form the total delayed fraction, $\sum \beta_i = \beta$, and all precursor concentrations together give the total precursor concentration, $\sum C_i = C$. The first term on the right hand side describes the precursor production rate density and the second term the decay rate density.

2.1.3 Precursor yield at fission

The yields of precursors and precursor families are material constants. There are huge differences between the main fissionable isotopes (Wimett et al., 1957). For example ^{239}Pu yields on average fewer delayed neutrons than ^{235}U, making a system with plutonium more difficult to control. Examples of typical yields for some common isotopes are given in Tables 2.1 and 2.2. The data for these tables and the following plots is taken from the JEFF3.1.1 data library (A. Santamarina, D. Bernard, P. Blaise, *et al.* , 2009).

In reality there is also a small material dependence for the average decay constant of the precursor families, but this difference is relatively small, often smaller than the uncertainty in the data. Therefore it is common practice to define the decay constant per family independent of energy and material. This can also be seen in the tables.

The yield of prompt neutrons at a fission is not only material dependent, it is also energy dependent. A typical dependence can be seen in Fig. 2.3. The neutron yield is fairly constant for a large range of incident neutron energies, but for high energies, the prompt yield starts to increase linearly, whereas the delayed yield starts to decrease.

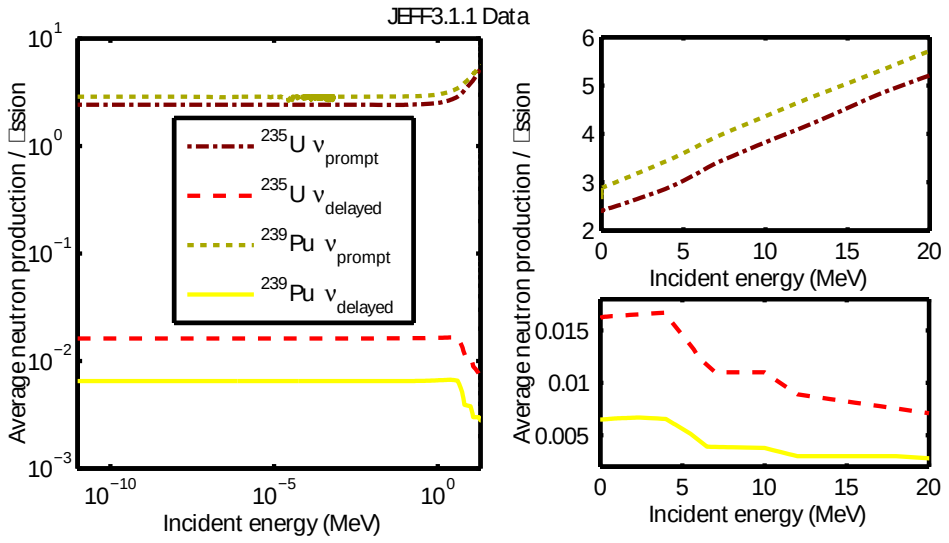


Figure 2.3: The prompt and delayed neutron production at a fission event for ^{235}U and ^{239}Pu. The nuclear data is from the JEFF3.1.1 library (A. Santamarina, D. Bernard, P. Blaise, *et al.* , 2009).

2.1.4 The start of a prompt fission chain

When a system is at steady state, the number of precursors compared to the number of neutrons can be approximated. Let's assume a system with one precursor family with decay constant $\lambda = 0.0785\ \mathrm{s}^{-1}$, $\beta = 0.00685$ and a mean generation time of $\Lambda = 7 \times 10^{-5}$ s. The mean generation time is the average time it takes a neutron to produce one new neutron. At steady state the production rate of precursors is equal to its decay rate:

$$\lambda C = \frac{\beta}{\Lambda} n \tag{2.5}$$

yielding a precursor concentration of

$$C = \frac{\beta}{\Lambda\lambda} n \tag{2.6}$$

This shows that for every neutron present there will be approximately 10^3 precursors.

In a time interval from t to $t+\Delta t$ both prompt neutrons and delayed neutrons will initiate chains of successive fission events. The delayed neutrons are produced by precursor decay during the time interval and they can induce fission, physically initiating a prompt fission

chain. Prompt neutrons do not physically initiate a prompt fission chain, but they do initiate the chain in this specific interval; they are the neutrons crossing the time boundary at t. The fraction of prompt neutron chains initiated by a prompt neutron in a time interval is given by:

$$\frac{n_0}{n_0 + d_0} \tag{2.7}$$

Here n_0 is the number of prompt neutrons at the start of the interval and d_0 is the number of delayed neutrons that are created during the time interval. The number of delayed neutrons can be calculated as:

$$d_0 = C_0 \int_t^{t+\Delta t} \lambda e^{-\lambda(t'-t)} \mathrm{d}t' \tag{2.8}$$

This turns into

$$d_0 = \frac{\beta}{\Lambda\lambda} n_0 (1 - e^{-\lambda \Delta t}) \tag{2.9}$$

Substituting this equation into Eq. (2.7) leads to

$$\frac{n_0}{n_0 + d_0} = \frac{1}{1 + \frac{\beta}{\Lambda\lambda}(1 - e^{-\lambda\Delta t})} \tag{2.10}$$

When using the same system properties as above and a time interval of 100 ms, approximately 10 % of the neutron chains is initiated by the prompt neutrons of the previous interval and 90 % is started by delayed neutrons, which demonstrates the importance of the delayed neutrons.

2.2 Sampling precursors

The fact that precursors have on average a much longer lifetime than neutrons, makes it possible to control a reactor with mechanical control rods. A neutron has a lifetime of approximately 10^{-4} s for a light water reactor and 10^{-7} s for a fast reactor, but the lifetime of a precursor can go up to 10^2 s. When making a crude approximation it can be demonstrated that the small fraction of delayed neutrons makes a large difference in the dynamic behaviour of a nuclear reactor.

The effective average lifetime, ℓ_{eff}, of a neutron can be calculated combining the average lifetime of prompt and delayed neutrons:

$$\ell_{eff} = (1 - \beta)\ell + \left(\sum_i \beta_i \frac{1}{\lambda_i} + \ell \right) \tag{2.11}$$

Here ℓ is the average lifetime of a prompt neutron. This yields an effective average lifetime of approximately 0.1 s, which is much more controllable with a mechanical moving device than in the case of 10^{-4} s.

For Monte Carlo simulation however, this difference in time scales poses a big challenge. Tallies are scored only when there are neutrons moving around and not when waiting for a precursor to decay. The time, during which neutrons are moving around is given by the lifetime of a prompt fission chain.

Lifetime of a prompt fission chain

The flight path of a neutron will end, when the neutron induces fission, is captured, or leaks out of the system. In a critical system most of the neutrons will end by inducing fission, creating new fission neutrons. When assuming only one prompt neutron can be created per fission, the probability to create such a new neutron is given by:

$$P_p = k_{eff}(1-\beta) \tag{2.12}$$

Here k_{eff} is the effective multiplication factor. The newly formed neutron also has a probability of P_p to create a new prompt neutron. The probability to create a chain of length n is given by:

$$P(n) = (1-P_p)\sum_{i=1}^{n} P_p^{(i-1)} \tag{2.13}$$

The average chain length is now given by:

$$\overline{n} = \sum_{n=1}^{\infty} nP(n) = (1-P_p)\sum_{n=1}^{\infty} nP_p^{(n-1)} = \frac{1}{1-P_p} \tag{2.14}$$

When assuming β to be 0.00685 and the system to be critical, the average chain length becomes close to 150 neutrons. The average lifetime of a neutron in a light water reactor is approximately 10^{-4} s, making the average lifetime of a prompt fission chain in the order of 10^{-2} s.

The lifetime of a precursor can range roughly from 10^{-2} s to 10^{2} s. This demonstrates that for each fission chain there is also a period, which may be orders of magnitude larger, without neutrons. If the reactor is at critical state, each prompt fission chain will on average produce one precursor, which will start a new prompt neutron chain after decay. During this time there are no neutrons of this chain, and there will be no power produced by this chain, see Fig. 2.4.

In a real power reactor this effect is averaged out by the large number simultaneous fission chains present in the reactor. If the total power of a reactor is 1 GW and the average energy released per fission is 3.2×10^{-11} J(200 MeV) the total number of fissions per second is approximately 3×10^{16}. Sadly, a Monte Carlo simulation does not have the luxury of sampling that many particles. Therefore the precursors will have to be simulated in a clever way.

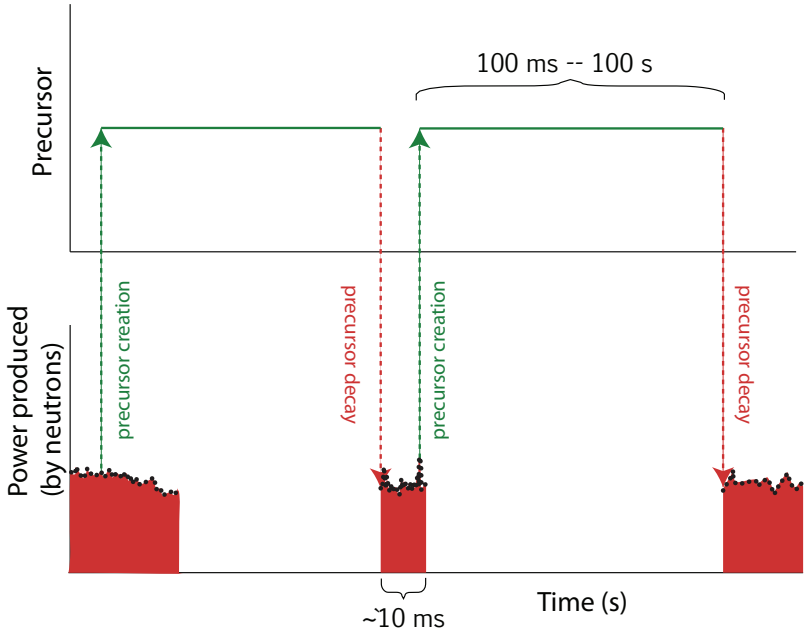


Figure 2.4: A schematic representation of a neutron chain with precursors. The difference in lifetime between precursors and prompt fission chains creates variance in the power production.

2.2.1 Combined precursor

The first step to simulate precursors more effectively than analog, is to combine all families into one Monte Carlo particle: the combined precursor. The probability of a precursor decaying in $\mathrm{d}t$ around time t is defined by

$$P_i(t)\mathrm{d}t = \lambda_i e^{-\lambda_i(t-t_0)}\mathrm{d}t \tag{2.15}$$

Here t_0 is the time when the precursor was created.

Now combine all precursor families into one with a decay probability of

$$P_{\text{combined}}(t) = \sum_i f d_i \lambda_i e^{-\lambda_i(t-t_0)} \tag{2.16}$$

Here fd_i is the fraction of delayed neutrons of the i-th family. This is defined by

$$fd_i = \frac{\beta_i}{\beta}, \tag{2.17}$$

In special cases this must be defined differently (see section 2.4).

Using the combined precursor, the variance caused by the different decay times of the different precursor families is addressed. The combined precursor incorporates the decay probability from all precursor families and is still exact. A precursor no longer has pure exponential decay when combining the different decay probabilities and therefore the age of the combined precursor becomes a factor.

2.2.2 Weight of neutrons from forced decay

To solve the problem of long periods without precursor decay and hence no prompt neutrons, the time is divided into intervals and the tallies of the Monte Carlo calculation are also scored in these intervals. Each precursor is forced to produce a weighted delayed neutron in every time interval, see Fig 2.5. This is described by Legrady and Hoogenboom (2008). The probability that the precursor has a forced decay at time t inside a time interval between t_1 and $t_1+\Delta t$ is chosen to be uniformly distributed:

$$\overline{p}(t) = \frac{1}{\Delta t} \tag{2.18}$$

To keep the game unbiased, the statistical weight of the resulting delayed neutron must be adapted so the probability of the forced decay times the weight equals the probability of a natural decay at that time. With the statistical weight of the precursor, w_C, considered, the weight of the delayed neutron becomes:

$$w_d(t) = w_C \frac{p(t)}{\overline{p}(t)} = w_C \Delta t \sum_i fd_i \lambda_i e^{-\lambda_i (t-t_0)} \qquad \text{with } t_1 < t < t_1+\Delta t \tag{2.19}$$

Forced decay starts a prompt fission chain in every time interval to ensure the presence of fission chains in all intervals.

2.2.3 Energy of delayed neutrons

When sampling a energy for the resulting delayed neutron, it should be taken into account that the different precursor families have a slightly different energy spectrum (Ott and Neuhold, 1985). Since these energy spectra are given per family it is best to select a precursor family from which the delayed neutron is originating and select an energy from the

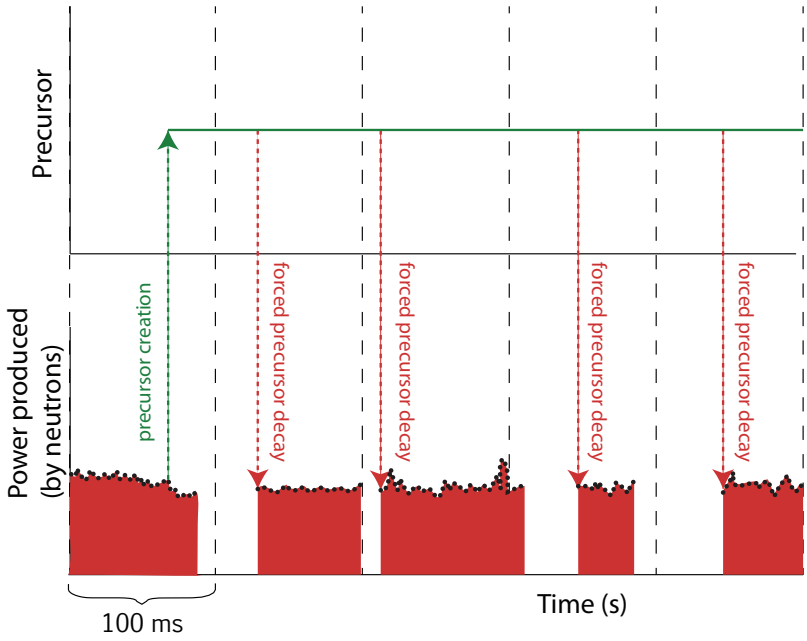


Figure 2.5: A schematic representation of a precursor with forced decay. Now there are scores in all time intervals.

spectrum, applicable to that family. The distribution of probabilities between the families changes over time and is given by:

$$P_i(t) = \frac{fd_i \lambda_i e^{-\lambda_i (t-t_0)}}{\sum_j fd_j \lambda_j e^{-\lambda_j (t-t_0)}} \tag{2.20}$$

The probabilities at the time of decay must be used to select the correct family for the energy spectrum.

2.2.4 Demonstration of improved precursor sampling

To demonstrate the effectiveness of the newly developed sampling methods, the following simple example is illustrative. In this example there is a precursor concentration at $t = 0$ and the delayed neutron source is sampled in time intervals of 100 ms. The distribution of the precursor families is β_i/β and the decay constants per precursor family are taken

Table 2.3: The precursors are divided into six families. Here the fractions and decay constants per precursor family i are given. Also the total delayed fraction and inversely weighted average decay constant are shown.

Family	λ (s^{-1})	β
1	0.0127	0.000260
2	0.0317	0.001459
3	0.1156	0.001288
4	0.3110	0.002788
5	1.4000	0.000877
6	3.8700	0.000178
av/tot	0.0784	0.006850

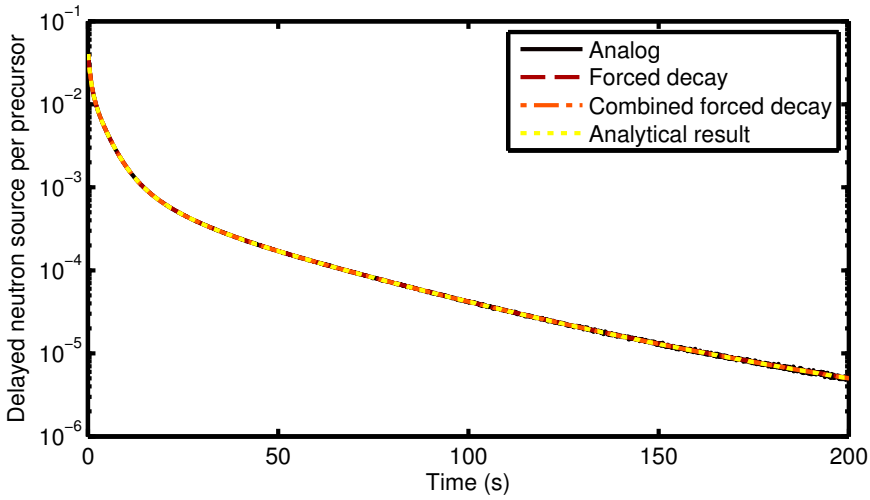


Figure 2.6: The average delayed neutron source in a time interval of 100 ms per source precursor is calculated using four different methods. All methods yield an unbiased result.

from Dam et al. (2005) and are given in Table 2.3. In this example three cases are tested: the analog case, forced decay and forced decay with a combined precursor.

In the analog case, first one of the precursor families is selected and then a decay time is selected. For the forced decay case, also a precursor family is selected, but then the precursor is forced to produce a delayed neutron in each time interval. In the last case the precursor represents all precursor families and this combined precursor is forced to produce a delayed neutron in each time interval.

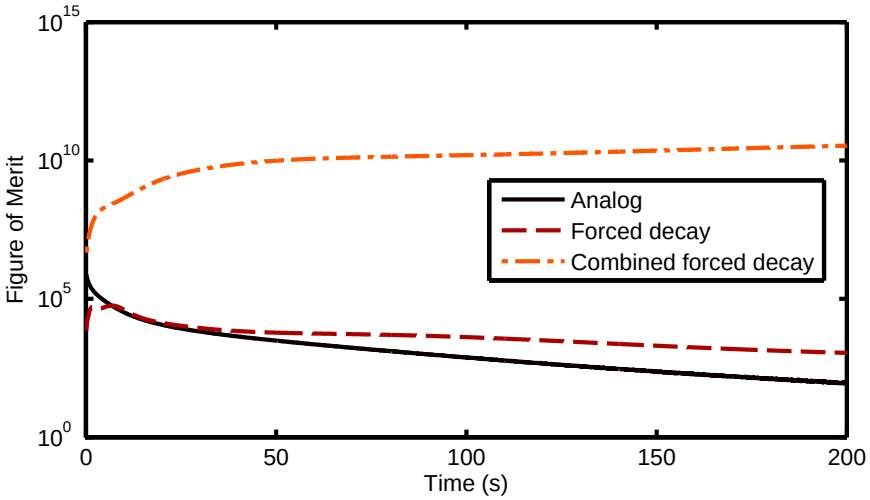


Figure 2.7: The figure of merit for all three methods of sampling precursor decay.

In Fig. 2.6 the delayed source per precursor is plotted with a time interval of 100 ms. All lines in this figure are on top of each other, demonstrating that all methods yield an unbiased result. It can already be seen that the analog case has more variance towards the larger time scales than the cases with forced decay, when inspecting the plot more closely. The black line starts to show more noise, which can be observed as a wider black line at the end of the plot. However, this is still within the statistical error of the analog case.

Fig. 2.7 shows the Figure of Merit (FoM) for the different methods. The FoM is calculated using the relative uncertainty of a time interval (σ_r) and the calculation time of the total simulation (T).

$$\text{FoM} = \frac{1}{\sigma_r^2 T} \tag{2.21}$$

It shows that forced decay is more effective than the analog case, but it becomes much more effective when combining all precursor families.

In this example only the delayed source is calculated, but in a real simulation, the calculation time of the precursor decay most likely does not contribute significantly to the total calculation time. Therefore also the relative uncertainty per delayed neutron is plotted in Fig. 2.8. The calculation time is proportional to the total number of delayed neutrons produced, since these start the prompt fission chains. The method with a combined precursor and forced decay has a relative uncertainty which is also several orders smaller than for the other methods.

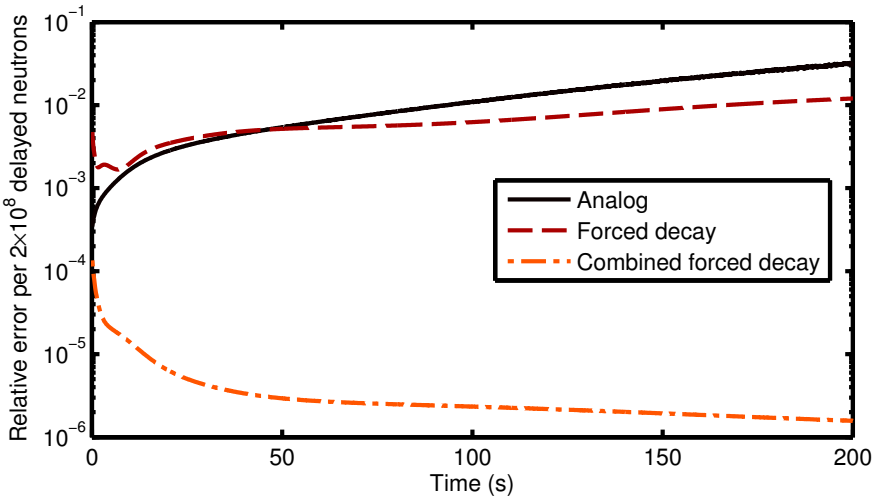


Figure 2.8: The relative uncertainty for all three methods of sampling precursor decay.

2.3 Precursor population control

With the technique described in the previous section, all combined precursors are stored to have a decay in all time intervals, with a decreasing delayed neutron weight, if the time intervals are equal in size. At the same time, new combined precursors are created by the fission reactions in the simulation. This is a situation with only production of combined precursors and no termination. Even in a sub-critical system the number of combined precursors would continue to increase. This calls for population control.

2.3.1 Precursor weights

Population control for precursors is somewhat different from population control for neutrons. There are three different statistical weights that can be distinguished for precursors: the main statistical weight of the precursor, the expected weight of the delayed neutron and the timed statistical weight.

The first weight is the weight which can be altered when applying population control. The other two weights are derived weights, which can be used to determine the importance of the precursor. They represent physical quantities, the expected delayed neutron source strength and the precursor concentration, respectively.

Main precursor weight

The first quantity is the main statistical precursor weight, denoted by w_C. This is the base weight from which the weight of the delayed neutron is calculated. This weight represents the number of physical precursors this Monte Carlo precursor represented at time of creation. This is also the weight which can be adjusted by variance reduction techniques. The main weight will not change over time, it is only changed by variance reduction techniques. The other weights are derived from this weight.

Expected delayed neutron weight

The precursor has interaction with the system via its delayed neutrons, which are generated by forced decay. It is logical to use the weight of these neutrons for population control. However, this weight is dependent on the exact decay time, which is sampled uniformly over the time interval.

To remove the weight fluctuation caused by the sampling of the exact decay time, the expected weight of a delayed neutron is defined:

$$w_{d,av} = \frac{1}{\Delta t} \int_{t_1}^{t_1+\Delta t} w_d(t) \mathrm{d}t \tag{2.22}$$

Here t_1 is the start of the next time interval. Using (2.19), $w_{d,av}$ becomes:

$$w_{d,av} = w_C \sum_i f d_i (e^{-\lambda_i (t_1 - t_0)} - e^{-\lambda_i (t_1 + \Delta t - t_0)}) \tag{2.23}$$

The evolution of the expected delayed neutron weight is shown in Fig. 2.9 by the dotted line. This shows the decrease of the statistical importance of a precursor over time.

Timed weight of the precursor

The third statistical weight is the timed weight of the precursor. This is the weight the precursor represents at time t. As time progresses the probability of a real precursor decaying increases and therefore the precursor particle represents less physical precursors. The timed weight is determined by the probability it did not have its decay before time t:

$$w_{timed} = w_C \left(1 - \int_{t_0}^{t} \sum_i f d_i \lambda_i e^{-\lambda_i (t - t_0)} \mathrm{d}t \right) = w_C \sum_i f d_i e^{-\lambda_i (t - t_0)} \tag{2.24}$$

The evolution of the timed precursor weight is represented by the dashed line in Fig. 2.9.

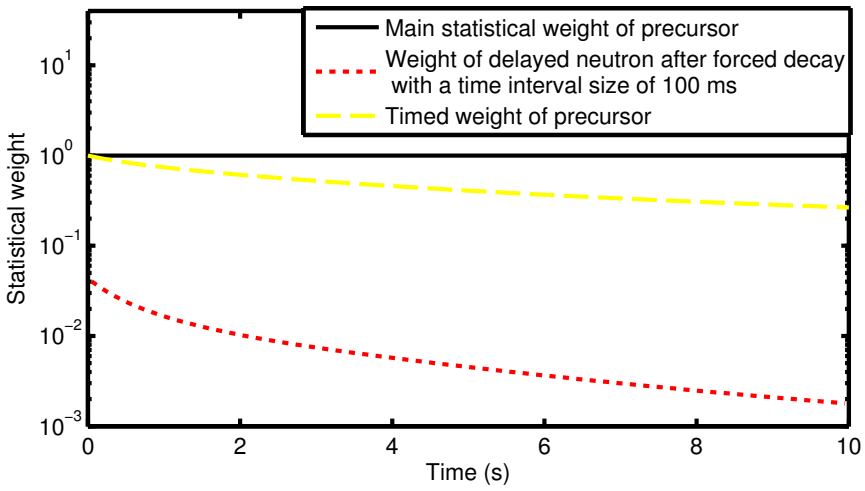


Figure 2.9: The different statistical weights of a precursor when using forced decay. The precursor is created at $t = 0$ and the time interval is 0.1 s.

2.3.2 Weight windows for precursors

A common method for variance reduction is the use of weight windows. These keep the weights of the different particles in the same range, to ensure that all particles have a similar contribution. The precursor weight which is monitored is either the timed weight or the expected delayed neutron weight.

Russian roulette

If the expected delayed neutron weight is below the Russian roulette threshold for neutrons, w_l, and the survival weight for neutrons would be w_{surv}, then Russian roulette can be played on the precursor. The surviving probability and survival weight can be calculated using:

$$w_{C,\text{new}} = w_C \frac{w_{\text{surv}}}{w_{d,\text{av}}} \qquad \rho < \frac{w_{d,\text{av}}}{w_{\text{surv}}} \tag{2.25}$$

$$w_{C,\text{new}} = 0 \qquad \rho > \frac{w_{d,\text{av}}}{w_{\text{surv}}} \tag{2.26}$$

with ρ a random number with a uniform distribution between 0 and 1.

Splitting

Splitting can be handled in a similar way, although it will be less common due to the decreasing importance of the precursors. In this case the precursor will be split when the expected delayed neutron weight is above the splitting threshold. The number of precursors in which it will be split is given by:

$$n_{\text{split}} = \left[\frac{w_{d,\text{av}}}{w_{\text{split}}} \right] \tag{2.27}$$

with the square brackets denoting the integer part of the fraction. The new precursor weight then becomes:

$$w_{C,\text{new}} = \frac{w_C}{n_{\text{split}}} \tag{2.28}$$

2.3.3 Combing technique

An other technique for population control, which is less well known, is the combing technique (Booth, 1996). The applicability of this method for population control in Dynamic Monte Carlo has been investigated by Kuilman (2012).

The idea of the combing method is to adjust the number of particles, while retaining the total particle weight. This is achieved by making a list of all particle weights and then selecting M particles which are equally spaced in weight. This is depicted in Fig. 2.10.

The length of the comb is defined as the total weight of all K particles, W:

$$W = \sum_{k=1}^{K} w_k \tag{2.29}$$

Now the average weight of a particle will be the length divided by the new total number of particles M

$$w_{av} = \frac{W}{M} \tag{2.30}$$

This is the distance between the teeth of the comb.

Each location of a tooth in the comb selects a particle, which will be stored and given weight w_{av}. If two teeth select the same particle, it is stored twice. In Fig. 2.10 this is the case for particles 3 and 5. If a particle is not selected, it is discarded, in this case particle 4.

The location of the first tooth is determined by a random number. The locations of the teeth now become:

$$t_m = \rho\, w_{av} + (m-1) w_{av} \qquad \text{with } m = 1, \ldots, M \tag{2.31}$$

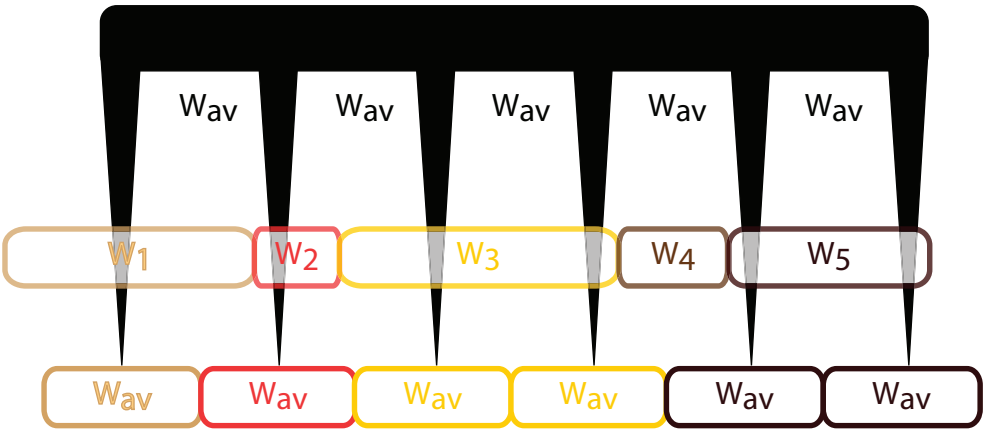


Figure 2.10: Combing the particles. All new particles have weight w_{av}. In this case one copy of particles 1 and 2 are stored, two copies of particles 3 and 5 are stored and particle 4 is terminated. The total weight is conserved.

with ρ a random number with a uniform distribution between 0 and 1.

The location a tooth hits is therefore uniformly distributed over the entire weight domain. Now the probability a tooth hits a weight interval of size w_k is given by:

$$P_{hit,k} = \frac{w_k}{w_{av}} \tag{2.32}$$

If this probability becomes larger than one, more than one tooth will hit the particle and the particle will be sampled multiple times.

The expected weight of a particle is conserved, using Eq. (2.32):

$$E[w_{k,after}] = P_{hit,k} w_{av} = w_k \tag{2.33}$$

Precursor combing

Combing precursors is more complicated, since there is a difference in the weight which must be altered and the weight which is monitored. To accomplish this, a variation to the importance-weighted comb of Booth (1996) is made.

The weight which must be conserved is the timed precursor weight, since this is the weight which represents the physical quantity of the total number of precursors present at time t. With T_k defined as the timed weight operator for particle k:

$$T_k = \sum_i f d_{i,k} e^{-\lambda_{i,k}(t-t_{0,k})} \tag{2.34}$$

and using Eq. (2.24) the timed weight of combined precursor k becomes:

$$w_{timed,k} = T_k w_{C,k} \tag{2.35}$$

and the total timed weight is

$$W_{timed} = \sum_{k=1}^{K} T_k w_{C,k} \tag{2.36}$$

The total timed weight is equal to the total precursor concentration $C(t)$. In this case the tooth size is

$$w_{timed,av} = \frac{W_{timed}}{M} \tag{2.37}$$

and the probability to select a specific precursor is given by

$$P_{hit,timed,k} = \frac{w_{timed,k}}{w_{timed,av}} \tag{2.38}$$

The weight of the selected precursor becomes:

$$w_{C,k,after} = \frac{w_{timed,av}}{T_k} \tag{2.39}$$

When combining the last two equations it can be shown that the expected main precursor weight is conserved and therefore also the expected delayed neutron weight:

$$E[w_{C,k,after}] = P_{hit,timed,k} \frac{w_{av}}{T_k} = w_{C,k} \tag{2.40}$$

In a similar fashion it can be shown that it is also possible to comb over the expected delayed neutron weight, while conserving the timed precursor weight. To do this only the factor T_k must be changed accordingly. Also importance, based on user knowledge or adjoint calculations can be taken into account in this factor.

2.4 Precursor source distribution

Now that we can simulate precursors, it is time to create a starting distribution for the precursors. It is possible to let the user input a precursor distribution, but this can be cumbersome and the exact distribution might be unknown.

In general a transient will start from a steady state. This might be a critical system or a sub-critical system with an external source. The first task is to determine this steady state distribution.

2.4.1 Geometrical distribution

To determine the initial distribution of neutrons and precursors, first a criticality calculation is done until the fission source is converged. For a critical system this is the proportional to the steady state neutron distribution. From this fission source distribution the precursor and prompt neutron distribution can be obtained.

Prompt neutron fraction

When the system is at steady state ($k_{eff} = 1$), the precursors density can be calculated using Eq. (2.4) in stationary form, $\frac{\partial C_i}{\partial t} = 0$:

$$C_{i0}(\boldsymbol{r}) = \frac{\beta_i}{\lambda_i} \nu \Sigma_f \phi(\boldsymbol{r}) \tag{2.41}$$

For all precursor families together this becomes:

$$C_0(\boldsymbol{r}) = \sum_i \frac{\beta_i}{\lambda_i} \nu \Sigma_f \phi(\boldsymbol{r}) \tag{2.42}$$

with an inversely weighted λ

$$\lambda^b \equiv \frac{\beta}{\sum \frac{\beta_i}{\lambda_i}} \tag{2.43}$$

this becomes

$$C_0(\boldsymbol{r}) = \frac{\beta}{\lambda^b} \nu \Sigma_f \phi(\boldsymbol{r}) \tag{2.44}$$

The fraction of prompt neutrons is defined as:

$$\frac{n_0(\boldsymbol{r})}{n_0(\boldsymbol{r}) + C_0(\boldsymbol{r})} = \frac{\frac{1}{\nu}\phi(\boldsymbol{r})}{\frac{1}{\nu}\phi(\boldsymbol{r}) + \frac{\beta}{\lambda^b}\nu\Sigma_f\phi(\boldsymbol{r})} = \frac{1}{1 + \frac{\beta}{\lambda^b}\nu\nu\Sigma_f} \tag{2.45}$$

Note, this fraction is between prompt neutrons and precursors, whereas Eq. (2.10) gives the fraction between prompt and delayed neutrons. Also this fraction does not use the generation time Λ which is difficult to determine exactly (Verboomen et al., 2006). Instead it only uses well-defined quantities, like the neutron speed and the total fission cross section.

The values used in this work for a mono-energetic case are taken from Appendix A.1. The neutron speed is chosen to be 10 times lower than thermal speed, because this will give the artificial system a generation time of $\Lambda = 72.7$ μs, which is more realistic for a thermal nuclear reactor. With these numbers the fraction of prompt neutrons in steady state becomes 0.08 %.

When we use weighted neutrons and precursors, the number of weighted precursors is calculated from the flux of weighted neutrons, $\phi_w(\boldsymbol{r})$, times the average weight of the neutrons:

$$\overline{w_{timed}} C_0(\boldsymbol{r}) = \frac{\beta}{\lambda^b} \nu\Sigma_f \phi_w(\boldsymbol{r}) \overline{w_n} \tag{2.46}$$

Now use this in (2.45) to obtain:

$$\frac{\frac{1}{v}\phi_w(\boldsymbol{r})}{\frac{1}{v}\phi_w(\boldsymbol{r}) + \frac{\beta}{\lambda^b}\nu\Sigma_f\phi_w(\boldsymbol{r})\overline{w_n}/\overline{w_{timed}}} = \frac{1}{1+\frac{\beta}{\lambda^b} v\nu\Sigma_f \overline{w_n}/\overline{w_{timed}}} \tag{2.47}$$

2.4.2 Family distribution

With exponential decay a precursor has no age: there is always the same probability that a precursor has its decay in the next time interval, no matter what time it has lived before. However, with combined families a precursor does not have a pure exponential decay probability any more, as stated in section 2.2.1.

In this case the combined precursor represents all precursor families. Each precursor family is represented by a fraction of the combined precursor:

$$\text{fraction of family } i = \frac{C_i}{C} \tag{2.48}$$

The precursors from different families decay with different decay constants. Therefore the fractions per family that one combined precursor represents evolve. The precursor concentration which is represented by the combined precursor is the original precursor concentration minus the decay:

$$C_i(t) = \frac{\beta_i}{\beta}\left(1 - \int_{t_0}^{t} \lambda_i e^{-\lambda_i(t-t_0)} \mathrm{d}t\right) \tag{2.49}$$

$$C(t) = \sum_i \frac{\beta_i}{\beta}\left(1 - \int_{t_0}^{t} \lambda_i e^{-\lambda_i(t-t_0)} \mathrm{d}t\right) \tag{2.50}$$

Now the evolution of the precursor families in a combined precursor can be calculated:

$$\frac{C_i(t)}{C(t)} = \frac{\frac{\beta_i}{\beta} e^{-\lambda_i t}}{\sum_j \frac{\beta_j}{\beta} e^{-\lambda_j t}} \tag{2.51}$$

When the precursor is created, $t = 0$, the fraction per precursor family is simply

$$\frac{C_i(0)}{C(0)} = \frac{\beta_i}{\beta} \tag{2.52}$$

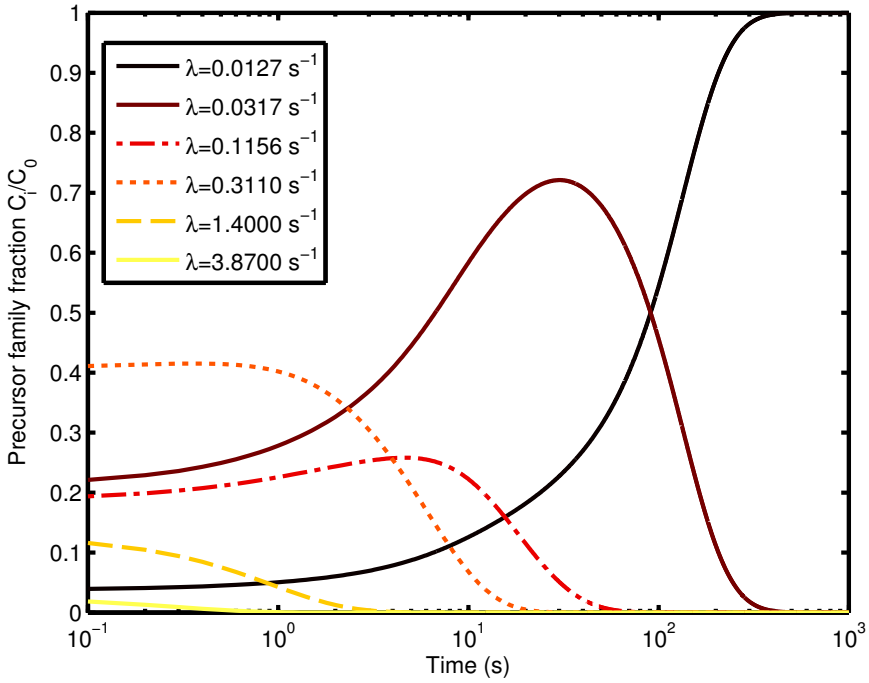


Figure 2.11: The ratio between the precursors families in a combined precursor. All families together add up to 1, but over time different precursor families are most important.

Then the fractions change overtime as shown in Fig. 2.11. The decay constants and delayed fractions are given in Table 2.3.

In a stationary case, the percentage of precursors per family remains the same. This number can be calculated combining Eqs. (2.41) and (2.44)

$$\frac{C_{i0}}{C_0} = \frac{\beta_i}{\beta}\frac{\lambda^b}{\lambda_i} \tag{2.53}$$

The difference between the precursor family distribution at the creation of a precursor and the family distribution in steady state is shown in Fig. 2.12.

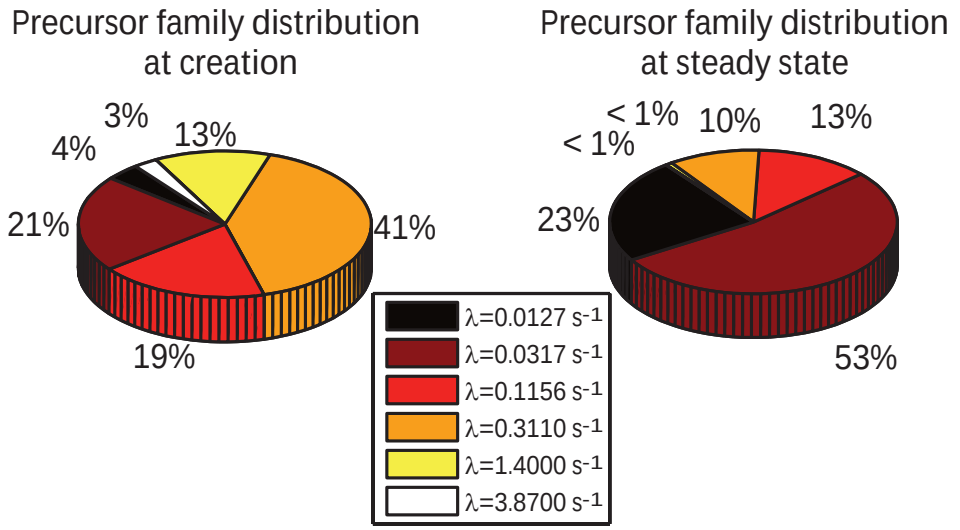


Figure 2.12: The distribution of precursor families is different for the moment of creation and at steady state.

Sampling creation time

One way to look at this problem is to realise that for a steady state the production of precursors has been constant before $t = 0$. When using forced decay all precursors that have been produced before $t = 0$ still have a contribution to the delayed neutron source. These are all precursors from $-\infty$ to 0.

Now the steady state distribution can be achieved by sampling a precursor source with a creation time t_0 uniformly between $-\infty$ and 0. This way the ratios between the different precursor families are correct:

$$C_{i,0} = \int_{-\infty}^{0} \beta_i \nu\Sigma_f \phi \, e^{-\lambda_i(0-t_0)} \mathrm{d}t_0 = \frac{\beta_i}{\lambda_i} \nu\Sigma_f \phi \tag{2.54}$$

$$C_0 = \sum_i \frac{\beta_i}{\lambda_i} \nu\Sigma_f \phi \tag{2.55}$$

combining these equations yields Eq. (2.53):

$$\frac{C_{i,0}}{C_0} = \frac{\frac{\beta_i}{\lambda_i} \nu\Sigma_f \phi}{\sum_i \frac{\beta_i}{\lambda_i} \nu\Sigma_f \phi} = \frac{\beta_i}{\beta} \frac{\lambda^b}{\lambda_i} \tag{2.56}$$

It can also be demonstrated that the precursor concentration remains constant with

constant neutron flux, when continuing from this starting point:

$$C(t)=\sum_i \frac{\beta_i}{\lambda_i}\nu\Sigma_f\phi\, e^{-\lambda_i t}+\sum_i\int_0^t \beta_i\nu\Sigma_f\phi\, e^{-\lambda_i(t-t')}\mathrm{d}t'=\frac{\beta}{\lambda^b}\nu\Sigma_f\phi \tag{2.57}$$

Now the timed weight of the precursor is altered since it has already undergone a portion of its decay probability before $t=0$. The timed weight can be calculated using Eq. (2.24). This timed weight is the weight which is used in Eq. (2.47), and therefore the total sum of all timed weights of the precursors must add up to C_0.

To achieve a uniform distribution between $-\infty$ and 0 there are two approaches. The first is to use an exponential distribution and adjust the weights of the resulting precursors, with the random number ρ uniformly between 0 and 1:

$$t = \frac{\ln(\rho)}{\lambda} \tag{2.58}$$

$$w_C = \frac{1}{\lambda}e^{-\lambda t_0} \tag{2.59}$$

A good choice for λ is the average decay constant: λ^b. Since this is the average decay constant, the sampled distribution will not differ too much from the actual distribution, making the rejection scheme more efficient.

Although this method is exact, a more practical method is to sample a starting time uniformly between $-T$ and 0, where T is a large number, typically a few times the average lifetime of the longest lived precursor family $\frac{1}{\lambda_1}$:

$$t = -\rho T \tag{2.60}$$

$$w_C = 1 \tag{2.61}$$

Next the timed weight of the precursor at $t=0$ is used for a rejection scheme:

$$w_{timed}(0)\geq \overline{w_{timed}}, \quad \text{accept}, \quad w_{timed}(0)=w_{timed}(0) \tag{2.62}$$

$$w_{timed}(0)\geq \rho\,\overline{w_{timed}}, \quad \text{accept}, \quad w_{timed}(0)=\overline{w_{timed}} \tag{2.63}$$

$$w_{timed}(0)< \rho\,\overline{w_{timed}}, \quad \text{reject}. \tag{2.64}$$

The average timed weight, $\overline{w_{timed}}$ can be chosen freely. If the precursor is rejected, a new starting time is sampled for that location.

Changing the family distribution

A more elegant method is to start precursors that are present at $t=0$ with a different family distribution. When a precursor is created the probability per family is given by:

$$P_i=\frac{\beta_i}{\beta} \tag{2.65}$$

For a precursor that is created at $t = 0$, with steady state conditions this probability is changed to:

$$P_i = \frac{\beta_i}{\beta}\frac{\lambda^b}{\lambda_i} \tag{2.66}$$

The combined precursors can take this into account nicely by altering the starting distributions of the families. The starting distributions are denoted by the fraction delayed, fd_i:

$$fd_i = \begin{cases} \frac{\beta_i}{\beta}, & \text{precursor created during simulation,} \quad (t_0 \geq 0) \\ \frac{\lambda^b}{\lambda_i}\frac{\beta_i}{\beta}, & \text{precursor created at start of simulation,} \quad (t_0 < 0) \end{cases} \tag{2.67}$$

2.5 Summary of the simulation of precursors

In this chapter the challenge of simulating precursors and neutrons in one simulation has been addressed. A precursor operates at a typical time scale of seconds and neutrons operate at a time scale of milliseconds and shorter. This difference can cause a lot of variance in the tallies, due to the fact that tallies are only scored when there are prompt neutrons being simulated. In nature there are enough neutrons to average out to steady neutron flux, but in a Monte Carlo simulation this can create large time intervals with only precursors present.

A scheme to combine neutrons and precursors into one time scale is proposed. In this scheme each precursor is forced to produce a delayed neutron in each time interval, in order to start a prompt-fission chain contributing to the tallies in that interval. In this way there are always at least as many prompt-fission chains as there are precursors in each time intervals. Although there can be moments in an interval without neutrons. Since the tallies are averaged over the interval, this is not an issue.

The statistical weight of a neutron resulting from forced decay is altered to ensure a fair game, but this decreases the weight of the resulting neutron. If the weight of this delayed neutron is too low, a Russian roulette game can be played on the precursor. The difference with a normal Russian roulette game is that the weight monitored is either the expected weight of the delayed neutron in the next time interval or the timed weight of the precursor. If this weight is below a threshold, the game is played and if the precursor survives, the main statistical weight of the precursor is increased, increasing automatically the derived weights.

Another method for population control is the combing method. This method redistributes the total weight of all particles over a fixed number of particles of equal weight. Here the total precursor concentration is conserved.

To reduce the variance even further, all precursor families are represented in one combined precursor. This way all precursor families are represented in exactly the right number.

The downside of combining the precursor families in one combined precursor is the loss of the exponential decay. This implies that the behaviour of a combined precursor changes over time and that the particle age must be taken into account, also when starting the calculation. This can be done by sampling a negative creation time for the source precursors or by giving the source precursors a steady-state distribution. This steady-state distribution reflects the distribution of the different precursor family concentrations in steady state. Precursors which are created during the simulation are given a fission distribution.

2.6 Bibliography

A. Santamarina, D. Bernard, P. Blaise, *et al.* . The JEFF-3.1.1 nuclear data library. JEFF Report 22, 2009.

N. Bohr and J. A. Wheeler. The mechanism of nuclear fission. *Physical Review*, **56**, 426, 1939.

T. E. Booth. A weight (charge) conserving importance-weighted comb for Monte Carlo. In *Proceedings of Radiation Protection and Shielding Division Topical Meeting*. N. Falmouth, MA, 1996.

H. V. Dam, T. H. J. J. van der Hagen and J. E. Hoogenboom. Nuclear reactor physics, lecture notes AP3341. Delft University of Technology, http://www.janleenkloosterman.nl/reports/ap3341.pdf, 2005.

J. J. Duderstadt and L. J. Hamilton. *Nuclear Reactor Analysis*. Wiley, New York, 1976.

G. R. Keepin. *Physics of Nuclear Kinetics*. Addison-Wesley Publishing Company, Inc, 1965.

F. Kuilman. *Implementing the combing method in the dynamic Monte Carlo*. Bachelor thesis, Delft University of Technology, 2012. PNR-131-2012-008.

D. Legrady and J. E. Hoogenboom. Scouting the feasibility of Monte Carlo reactor dynamics simulations. In *Proceedings of PHYSOR-2008 - American Nuclear Society Topical Meeting on Reactor Physics*. Interlaken, 2008.

K. O. Ott and R. J. Neuhold. *Introductory Nuclear Reactor Dynamics*. American Nuclear Society, La Grange Park, United States of America, 1985.

B. Pfeiffer, K.-L. Kratz and P. Möller. Status of delayed-neutron precursor data: half-lives and neutron emission probabilities. *Progress in Nuclear Energy*, **41**, 39, 2002.

J. Terrell. Distributions of fission neutron numbers. *Physical Review*, **108**, 783, 1957.

B. Verboomen, W. Haeck and P. Baeten. Monte Carlo calculation of the effective neutron generation time. *Annals of Nuclear Energy*, **33**, 911, 2006.

G. R. Wimett, T. F. Zeigler and R. Keepin. Delayed neutrons from fissionable isotopes of uranium, plutonium, and thorium. *Physical Review*, **107**, 1044, 1957.

3

Variance in Prompt Fission Chain Lengths

Picture: B.L. Sjenitzer

The decay of precursors is not the only source of variance in a time dependent calculation. Also the variance in the length of prompt fission chains has a significant contribution to the variance of the neutron flux and therefore to the variance in tallies. To investigate this first the physics of such branching processes is discussed in section 3.1 and modelled in sections 3.2 and 3.3. Then new variance reduction techniques are developed in section 3.4 and finally these new techniques are tested in a simple system in section 3.5.

3.1 Physics of fission chains

To study this variance caused by the prompt-fission chains, it is best to first study the physical phenomena which cause these effects. This can be done at several levels, from individual neutrons to average fission chains.

3.1.1 Microkinetics

In nuclear reactor physics it is common to look at the average neutron flux. The average neutron flux is continuous in phase space, but when considering the actual neutron flux it contains noise and it is not continuous. The neutron flux consists of individual neutrons, grouped in fission chains and studying these average fission chains is called microkinetics (Ott and Neuhold, 1985). The length of an average fission chain is finite, if a system is not super prompt critical.

The behaviour of fission chains determines the behaviour of global and more abstract quantities, e.g. the multiplication factor, the effective delayed fraction or alpha-modes. These result in fact from a super-position of the average fission chains. The chains are started by a delayed neutron, as explained in the previous chapter. This delayed neutron can cause fission, releasing new prompt neutrons. The behaviour of an average chain is depicted in Fig. 3.1. When a delayed neutron starts a chain, the neutron source is one. The neutron can then release new prompt neutrons via fission, but on average, the neutron population of the chain will decrease if the system is not super prompt critical ($k_{eff}(1-\beta)<1$). The average neutron source, started by the decay of a single precursor, will therefore decrease exponentially.

From microkinetics few macroscopic features of a transient in a nuclear reactor can be derived. For example, when the reactivity increases the population in the prompt fission chains will decrease more slowly, creating more neutrons, which is a measure for the total neutron flux. This increase in neutron flux occurs almost instantly and is called the prompt jump.

Next, these longer fission chains will produce more precursors, which will start extra fission chains. Therefore not only the number of neutrons per chain increases, but also

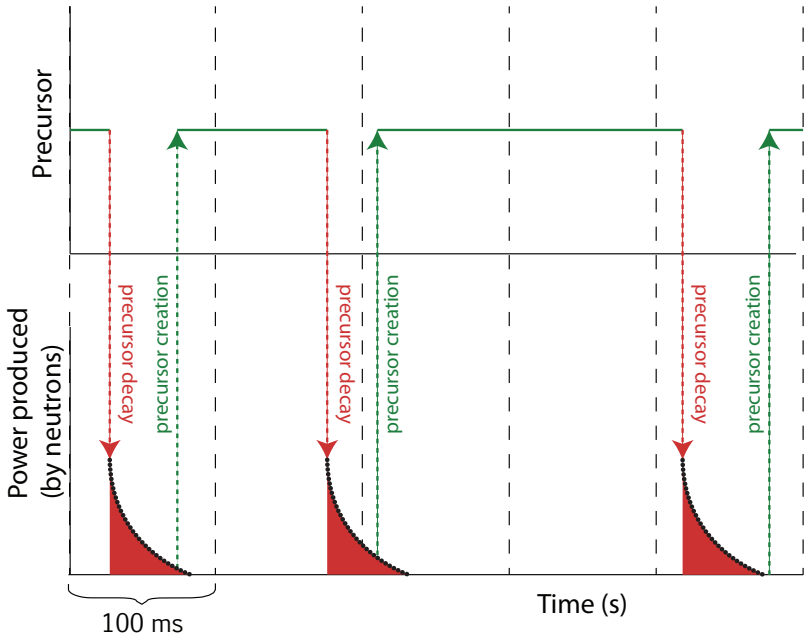


Figure 3.1: A schematic representation of prompt neutron chains in the microkinetics description. Each prompt fission chain has the same contribution to the power production.

the number of chains. This effect is not direct, but happens after the decay of the extra precursors, causing the exponential increase over longer time scales.

3.1.2 Fission-chain kinetics

Unfortunately nature does not supply us with average fission chains. For reactor analysis the model of an average fission chain may be illustrative, for Monte Carlo simulation most fission chains are not average. This will change Fig. 3.1 to Fig. 3.2. On average each chain will generate a new precursor in a critical reactor, but this is also only an average.

To reduce the variance introduced by this phenomenon it is needed to look more fundamentally to the neutron kinetics. When doing this there are some advantages, which Monte Carlo kinetics has over neutron kinetics. The number of neutrons released in a real fission event is mainly between 1 and 6 as explained in Sec. 2.1, but in Monte Carlo it will usually be the average number of neutrons rounded to the higher or lower integer, with

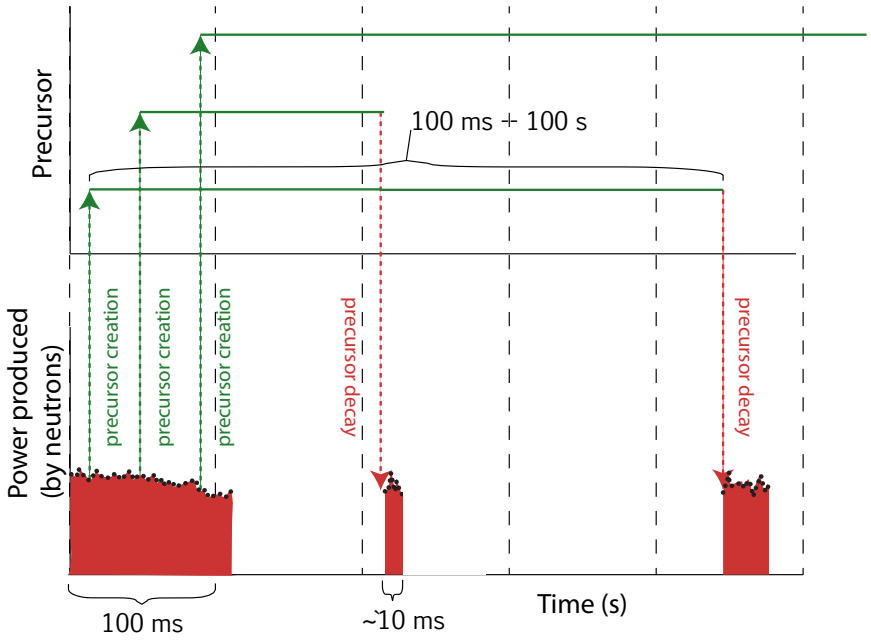


Figure 3.2: A schematic representation of the variance in the chain length of the prompt fission chain, which is a major contributor to the variance in the power production. On average each chain will produce one new precursor.

appropriate probability.

Moreover variance reduction techniques can be applied in the Monte Carlo simulation. The question of which variance reduction techniques to use for controlling the fission chains in a dynamic Monte Carlo simulation will be investigated in this chapter. In order to study only the effect of variance due to the prompt fission chains, this chapter will utilises a model with only prompt neutrons.

3.2 A simple model for chain length statistics

3.2.1 Single branch

First, a simple model is introduced to show the size of the variance caused by a branching process. More elaborate descriptions of branching processes are given by Pazsit and Pal (2008).

In this model it is assumed that the system is not super prompt critical:

$$k_p < 1 \tag{3.1}$$

Here k_p is the prompt multiplication factor:

$$k_p = k_{eff}(1-\beta) \tag{3.2}$$

This holds if

$$k_{eff} < \frac{1}{1-\beta} \tag{3.3}$$

First, lets assume that a neutron is able to produce only one new neutron during its lifetime ($\nu = 1$) and that the probability for a neutron to create a new neutron is constant. The probability to create a new neutron becomes in this case simply:

$$P_p = k_p = k_{eff}(1-\beta) \tag{3.4}$$

In such a system, the probability for a chain of prompt neutrons with a length of n neutrons is given by

$$P(n) = P_p^{(n-1)}(1-P_p) \tag{3.5}$$

The sum over possible chain lengths must yield 1:

$$\sum_{n=1}^{\infty} P(n) = (1-P_p)\sum_{n=1}^{\infty} P_p^{(n-1)} = (1-P_p)\sum_{n=0}^{\infty} P_p^{(n)} = (1-P_p)\frac{1}{1-P_p} = 1 \tag{3.6}$$

Now the average chain length can be calculated:

$$\overline{n} = \sum_{n=1}^{\infty} nP(n) = (1-P_p)\sum_{n=1}^{\infty} nP_p^{(n-1)} = \frac{1}{1-P_p} \tag{3.7}$$

For a critical system with $\beta = 0.685\%$, this average chain length is approximately 150 prompt neutrons. To calculate the variance, the average squared chain length is needed:

$$\begin{aligned} \overline{n^2} = \sum_{n=1}^{\infty} n^2 P(n) = (1-P_p)\sum_{n=1}^{\infty} n^2 P_p^{(n-1)} = \\ (1-P_p)\left(\sum_{n=0}^{\infty} n^2 P_p^{(n)} + \sum_{n=0}^{\infty} 2nP_p^{(n)} + \sum_{n=0}^{\infty} P_p^{(n)}\right) = \frac{1+P_p}{(1-P_p)^2} \end{aligned} \tag{3.8}$$

Now the variance can easily be calculated:

$$\sigma^2 = \overline{n^2} - \overline{n}^2 = \frac{1+P_p}{(1-P_p)^2} - \left(\frac{1}{1-P_p}\right)^2 = \frac{P_p}{(1-P_p)^2} \tag{3.9}$$

When P_p approaches 1, the variance becomes infinitely large, but since the chain length also becomes infinitely long, the relative uncertainty becomes:

$$\sigma_r \equiv \frac{\sigma}{\overline{n}} = \frac{\sqrt{\frac{P_p}{(1-P_p)^2}}}{\frac{1}{1-P_p}} = \sqrt{P_p} \tag{3.10}$$

This uncertainty is still large when P_p is close to 1, which is the case for many nuclear systems.

To see how this influences the efficiency of the calculation, the Figure of Merit (FoM) is defined. In this model the FoM is not straight forward, because there is no calculation time. However, the calculation time needed would be proportional to the average prompt fission chain length: $T \sim \overline{n}$. With this, the FoM can be defined as

$$\text{FoM} \equiv \frac{1}{\sigma_r^2 \overline{n}} = \frac{\overline{n}}{\sigma^2} = \frac{1-P_p}{P_p} \tag{3.11}$$

The influence of P_p on the FoM is shown in Fig. 3.3, for the curve labelled $\nu = 1$ and where P_p is equal to k_p. It shows that the efficiency of the calculation deteriorates rapidly as the system approaches prompt criticality.

3.2.2 Fission with branching

A similar model can be used in which each fission reaction yields 2 new prompt neutrons ($\nu = 2$). In this case the probability P_p that a neutron produces new prompt neutrons is

$$P_p = \frac{k_p}{2} \tag{3.12}$$

In this model the chain length can be defined in multiple ways, but for the present purposes it is most useful to define the chain length as the total number of neutrons in a chain, because this is a measure for the calculation time. The chain can only have a length of $2n+1$ neutrons, with n a natural number. The probability that a fission chain has the size of $2n+1$ neutrons is given by

$$P(2n+1) = \frac{(2n)!}{(n+1)!n!} {P_p}^n (1-P_p)^{n+1} \tag{3.13}$$

The factor $\frac{2n!}{(n+1)!n!}$ is called the Catalan number and takes into account the possibility that the same number of neutrons in a chain can be achieved by different permutations (Stanley, 1999).

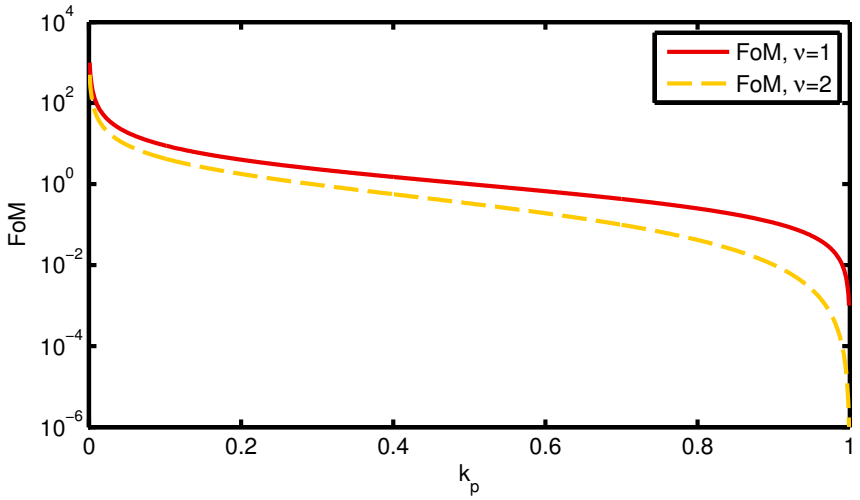


Figure 3.3: The figure of merit for the length of a prompt fission chain, as defined in Eq. (3.11). Here k_p is the prompt multiplication factor.

The total sum of all possibilities is equal to 1:

$$\sum_{n=0}^{\infty} P(2n+1) = 1 \tag{3.14}$$

With this series the average chain length and the average chain length squared can be calculated again:

$$\overline{n} = \sum_{n=0}^{\infty} (2n+1) P(2n+1) = \frac{1}{1-2P_p} \tag{3.15}$$

$$\overline{n^2} = \sum_{n=0}^{\infty} (2n+1)^2 P(2n+1) = \frac{1+2P_p-4P_p^2}{(1-2P_p)^3} \tag{3.16}$$

The resulting variance and FoM are

$$\sigma^2 = \frac{4P_p(1-P_p)}{(1-2P_p)^3} \tag{3.17}$$

$$\text{FoM} = \frac{(1-2P_p)^2}{4P_p(1-P_p)} \tag{3.18}$$

Similarly to the case where $\nu = 1$, the σ_r and the FoM can be calculated analytically. The FoM for $\nu = 2$ is also plotted in Fig. 3.3 and the plot shows that the FoM is lower if a neutron

can produce more than one new neutron; hence branching is not desired from a simulation point of view.

3.3 Moments equations

3.3.1 The model using moment equations

A more exact way of modelling the neutron multiplication is to use moment equations (Lux and Koblinger, 1991). With these equations the variance of a Monte Carlo calculation can be calculated a priori, even when variance reduction methods are used (van Wijk and Hoogenboom, 2011). In this model also the successive collisions of the particle are taken into account. The flight time between two interactions is a fixed time t_f and the total time is finite. This implies that a fixed maximum number of interactions is possible per particle in a time domain. This assumption will lead to a slight underestimation of the variance, since the flight time is actually distributed exponentially, but this effect is assumed to be small compared to the total variance.

The interactions inside a time domain are considered. They are numbered from 1 to N; the final interaction is t_N, see Fig. 3.4. The interaction at t_{N+1} is not part of the time domain of the calculation and therefore the contribution to the tally at t_{N+1} is set to 0.

Now the contribution of a particle to a tally can be calculated. Here we will take the power production as an example, but the theory is valid for any tally. The average amount of energy released in a fission reaction is given by Q_f. P_f is the probability fission occurs at an interaction and is given by

$$P_f = \frac{\Sigma_f}{\Sigma_t} \tag{3.19}$$

Then a particle at t_N will generate $P_f Q_f$ energy. A particle at t_{N-1} will produce also $P_f Q_f$, plus it has the probability P_s to scatter to the next collision at t_N and produce energy at that instance. P_s is equal to

$$P_s = \frac{\Sigma_s}{\Sigma_t} \tag{3.20}$$

The average total production for a particle starting at t_{N-1} is $P_f Q_f + P_s P_f Q_f$.

3.3.2 Score probability equations

Next the score probability function $\pi(t_n, w, s)$ is introduced. This quantity gives the probability that a neutron that starts at time t_n, with weight w, will generate a score s. With the introduction of the score probability equation, the model can easily take into account the contribution of prompt fission neutrons as well as variance reduction techniques such as implicit capture. For now the system will remain infinitely large.

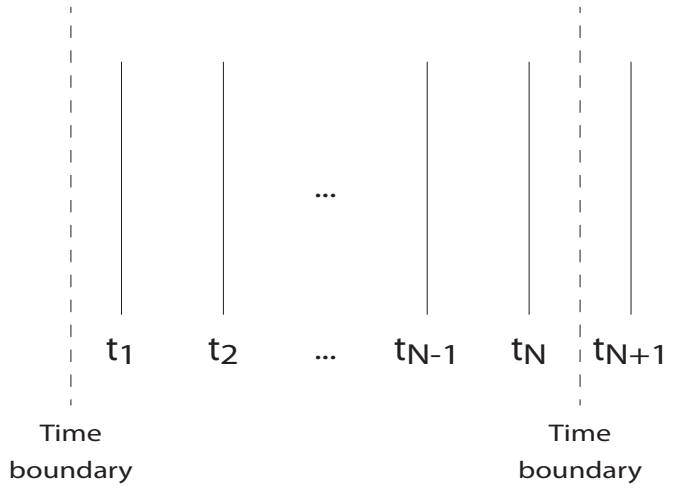


Figure 3.4: When moment equations are used, a fixed time between collisions is needed to calculate the variance of a tally a priori.

In reality at a fission reaction ν new neutrons are created and the present neutron is terminated. In the current model the probability to generate this number of new neutrons is lumped into the probability to generate one new neutron, while continuing the path of the original neutron via implicit absorption. The probability that a neutron with statistical weight w generates a new prompt neutron with weight 1 is given by $P_m = w\nu\Sigma_f/\Sigma_t$.

The scoring, in our case the energy production, at one collision is represented by a delta function, since in this model the score per collision is always exactly wP_fQ_f. When the particle scatters or produces a new particle from fission, the scores at the collisions of these particles must also be taken into account. This is done by convoluting the score at time $t = t_n$ with all possible scores at the following collisions. A convolution is defined as

$$f(s) \otimes g(s) \equiv \int_{-\infty}^{\infty} f(s-s')g(s')\mathrm{d}s' \tag{3.21}$$

At a collision in this model, in which implicit absorption and forced fission are taken into account, a neutron has two possibilities. The first possibility is that the neutron initiates no fission and the power production at $t = t_n$ is only convoluted with the power production of the scattering neutron. The second possibility is that the neutron creates a new fission neutron and scatters as well. In this case the power production is convoluted with both the scattering neutron and the fission neutron.

For clarity, the different terms in the equation for the score probability appear in colours. The score of the first collision is in red, the score of the scattering particle is in blue. The score of the fission neutron is in yellow. The scoring function is given by

$$\begin{aligned}\pi(t_n,w,s)=&(1-wP_m)\delta(s-wP_fQ_f)\otimes\pi(t_{n-1},wP_s,s)\\&+wP_m\delta(s-wP_fQ_f)\otimes\pi(t_{n-1},wP_s,s)\otimes\pi(t_{n-1},1,s)\end{aligned}\tag{3.22}$$

The first moment can be calculated by integrating over all possible scores:

$$M_1(t_n,w)=\int_{-\infty}^{\infty}s\pi(t_n,w,s)\mathrm{d}s\tag{3.23}$$

Here the following relation, introduced by Micoulaut (1976), is needed :

$$\langle s\rangle_{f_1\otimes f_2\otimes\ldots\otimes f_N}=\langle s\rangle_{f_1}+\langle s\rangle_{f_2}+\ldots+\langle s\rangle_{f_N}\tag{3.24}$$

Now Eq. (3.23) results in

$$M_1(t_n,w)=wP_fQ_f+M_1(t_{n-1},wP_s)+wP_mM_1(t_{n-1},1)\tag{3.25}$$

As for the first moment holds

$$M_1(t_n,w)=wM_1(t_n)\tag{3.26}$$

the first moment becomes

$$M_1(t_n)=P_fQ_f+(P_s+P_m)M_1(t_{n-1})\tag{3.27}$$

and the first moment resulting from a weighted particle can be calculated by multiplying this equation by the weight w.

For the second moment the calculation becomes more complicated and the following theorem by Micoulaut (1976) is needed:

$$\langle s^2\rangle_{f_1\otimes f_2\otimes\ldots\otimes f_N}=\sum_{j_i=0,1,2}\frac{2!}{j_1!j_2!\ldots j_N!}\langle s^{j_1}\rangle_{f_1}\langle s^{j_2}\rangle_{f_2}\ldots\langle s^{j_N}\rangle_{f_N}\tag{3.28}$$

where the summation is such that $\sum_i^N j_i=2$. In the case where $N=3$ this becomes:

$$\begin{aligned}\langle s^2\rangle_{f_1\otimes f_2\otimes f_3}=\int_{-\infty}^{\infty}s^2f_1\otimes f_2\otimes f_3\mathrm{d}s=&\langle s^2\rangle_{f_1}+\langle s^2\rangle_{f_2}+\langle s^2\rangle_{f_3}\\&+2\langle s\rangle_{f_1}\langle s\rangle_{f_2}+2\langle s\rangle_{f_1}\langle s\rangle_{f_3}+2\langle s\rangle_{f_2}\langle s\rangle_{f_3}\end{aligned}\tag{3.29}$$

The cross terms that appear in the second moment are given the following colours. The cross term between the first collision and the scattering neutron is in purple. The cross

terms between the fission neutron and the first collision and the scattering neutron are in green and in brown, respectively.

The second moment now becomes

$$\begin{aligned} &M_2(t_n,w)=\int_{-\infty}^{\infty} s^2\pi(t_n,w,s)\mathrm{d}s= \\ &(1-wP_m)\left(w^2P_f^2Q_f^2+M_2(t_{n-1},wP_s)+2wP_fQ_fM_1(t_{n-1},wP_s)\right)+ \\ &wP_m\left(w^2P_f^2Q_f^2+M_2(t_{n-1},wP_s)+2wP_fQ_fM_1(t_{n-1},wP_s)+\right. \\ &\left.M_2(t_{n-1},1)+2wP_fQ_fM_1(t_{n-1},1)+2M_1(t_{n-1},1)M_1(t_{n-1},wP_s)\right) \end{aligned} \tag{3.30}$$

This results in

$$\begin{aligned} M_2(t_n,w)=&w^2P_f^2Q_f^2+M_2(t_{n-1},wP_s)+wP_mM_2(t_{n-1},1)+ \\ &2w^2\left(P_fQ_fP_s+P_fQ_fP_m+P_mP_sM_1(t_{n-1})\right)M_1(t_{n-1}) \end{aligned} \tag{3.31}$$

which is in line with the derivations in Lux and Koblinger (1991) and Booth (1979)

Since the score probability function at $t=t_n$ only depends on the score at $t=t_{n+1}$ and since the score at $t=t_{N+1}$ is known to be zero, these equations can be solved backwards. The weight of a neutron is determined by the number of implicit absorptions it has had, which is equal to the number of scatter interactions, p, it has had from its start at t_{n-p} to t_n. The equation for the first moment can be solved directly:

$$M_1(n,p)=nP_s^pP_fQ_f \tag{3.32}$$

The second moment equation can be transformed into a matrix form:

$$\begin{aligned} M_2(n,p)=&P_s^{2p}P_f^2Q_f^2+M_2(n-1,p+1)+P_s^pP_mM_2(n-1,0)+ \\ &2(n-1)P_s^{2p}P_fQ_f\left(P_fQ_fP_s+P_fQ_fP_m+(n-1)P_mP_sP_fQ_f\right) \end{aligned} \tag{3.33}$$

These equations can be solved when starting at $n=N+1$ and $p=N+1$ and working backwards from there. Also note that $p\leq n_{max}$, where n_{max} is the maximum number of time steps possible.

3.3.3 Figure of Merit

The FoM in this model is defined differently from the definition in the model of section 3.2. The calculation time can be estimated more accurately by assuming it is proportional to the number of collisions simulated, c:

$$T\sim c \tag{3.34}$$

Now the FoM becomes

$$\mathrm{FoM}=\frac{1}{\sigma_r^2c} \tag{3.35}$$

t_n	Analytical			Monte Carlo		
	M_1	M_2	FoM	M_1	M_2	FoM
1	7.95×10^{-12}	6.32×10^{-23}	∞	7.95×10^{-12}	6.32×10^{-23}	10^9
2	1.59×10^{-11}	2.66×10^{-22}	7.05	1.59×10^{-11}	2.66×10^{-22}	7.05
3	2.38×10^{-11}	6.41×10^{-22}	1.53	2.38×10^{-11}	6.42×10^{-22}	1.53
4	3.18×10^{-11}	1.23×10^{-21}	0.56	3.18×10^{-11}	1.23×10^{-21}	0.56
5	3.97×10^{-11}	2.07×10^{-21}	0.26	3.98×10^{-11}	2.07×10^{-21}	0.27
6	4.77×10^{-11}	3.20×10^{-21}	0.14	4.77×10^{-11}	3.21×10^{-21}	0.15

Table 3.1: The moment equations and the resulting theoretical FoM, calculated with both the moment equations and a Monte Carlo simulation

with

$$\sigma_r = \frac{\sigma}{M_1} \tag{3.36}$$

To calculate the total number of collisions, a score probability function is used as well, $\pi_c(t_n, w, s)$, which is given by

$$\begin{aligned}\pi_c(t_n, w, s) = (1 - wP_m)\delta(s-1) \otimes \pi(t_{n-1}, wP_s, s) \\ + wP_m\delta(s-1) \otimes \pi(t_{n-1}, wP_s, s) \otimes \pi(t_{n-1}, 1, s)\end{aligned} \tag{3.37}$$

This function is similar to Eq. (3.22), but now the score is the particle's collision, which is one, convoluted with the number of collisions it makes after t_n and the collisions of its possible progeny. The total number of collisions thus becomes:

$$c(t_n, w) = 1 + c(t_{n-1}, wP_s) + wP_m\, c(t_{n-1}, 1) \tag{3.38}$$

3.3.4 First results of the moment equations model

To show the validity of these moment equations, a sample problem is devised for a short time interval. The system used in this problem has the following properties: $P_s = 0.3$, $P_m = 0.7$, $P_f = 0.25$ and $Q_f = 200 \times 1.60 \times 10^{-13}$ J. Six collisions are calculated both via the analytical route described above and via Monte Carlo simulation. The FoM calculated by the Monte Carlo calculation the theoretical FoM defined by Eq. (3.35); the real calculation time is not used. The results are shown in Table 3.1. It shows that the moment equations can predict the variance accurately and the efficiency of a time dependent Monte Carlo calculation on this simple model, a priori.

3.4 Variance reduction methods

3.4.1 Biasing fission

With the basic equations laid out before, the moment equation can now be used to find biasing schemes which improve the efficiency and stability of time-dependent calculations. High variance can initiate extremely long fission chains, making the total calculation time unpredictable.

The first place to look is biasing of the fission probability. This can be done with two different approaches. The first is by biasing the absolute fission probability. In this case the probability of creating new neutrons is independent of the neutron weight. This method is named biased fission probability. The second method does take into account the weight of the neutron and this method is named biased forced fission.

Biased fission probability

When a biased fission probability is used, the probability to create a fission neutron at a collision in the Monte Carlo simulation is set to P_m^*; the probability of creating a new fission neutron in a weighted Monte Carlo game then becomes wP_m^* and the resulting weight of the fission neutron becomes P_m/P_m^*, to ensure an unbiased result. Since both P_m and P_m^* are probabilities, they are between 0 and 1. Now the score probability function becomes

$$\begin{aligned}\pi(t_n,w,s) = (1-wP_m^*)\delta(s-wP_fQ_f)\otimes\pi(t_{n-1},wP_s,s)+\\ wP_m^*\,\delta(s-wP_fQ_f)\otimes\pi(t_{n-1},wP_s,s)\otimes\pi(t_{n-1},P_m/P_m^*,s)\end{aligned} \tag{3.39}$$

This method does not change the first moment, otherwise the result would be biased, but does change the second moment:

$$\begin{aligned}M_2(t_n,w) = \int_{-\infty}^{\infty} s^2\pi(t_n,w,s)\mathrm{d}s = \\ w^2P_f^2Q_f^2 + M_2(t_{n-1},wP_s) + wP_m^*M_2(t_{n-1},P_m/P_m^*)+\\ 2w^2\big(P_fQ_fP_s + P_fQ_fP_m + P_mP_sM_1(t_{n-1})\big)M_1(t_{n-1})\end{aligned} \tag{3.40}$$

The difference between this equation and Eq. (3.31) is in the yellow fission term, the third term. To reduce the variance, this should be smaller.

To calculate the second moment, also the second moments for different starting weights and starting times are needed. This time the weight is determined by the number of scatter interactions p and the number of fission events q, with $q = n_{max} - n - p$. The second

moments can be calculated by starting at $n = N+1$ and $p = N+1$ and then solving

$$\begin{aligned} M_2(n,p) = & P_s^{2p} P_f^2 Q_f^2 \left(P_m/P_m^*\right)^{2q} + M_2(n-1,p+1) + P_s^p P_m^* M_2(n-1,q+1) + \\ & 2(n-1) P_s^{2p} \left(P_m/P_m^*\right)^{2q} P_f Q_f \left(P_f Q_f P_s + P_f Q_f P_m + (n-1) P_m P_s P_f Q_f\right) \end{aligned} \tag{3.41}$$

Biased forced fission probability

It is also possible to bias the absolute probability of fissioning, setting the probability of creating a new fission neutron to P_n^*. In this case the weight of the new fission neutron becomes wP_m/P_n^*.

The first moment again returns an unbiased result and the matrix for the second moments can be calculated using

$$\begin{aligned} M_2(n,p) = & P_s^{2p} P_f^2 Q_f^2 \left(P_m/P_m^*\right)^{2q} + M_2(n-1,p+1,q) + P_m^* M_2(n-1,p,q+1) + \\ & 2(n-1) P_s^{2p} \left(P_m/P_m^*\right)^{2q} P_f Q_f \left(P_f Q_f P_s + P_f Q_f P_m + (n-1) P_m P_s P_f Q_f\right) \end{aligned} \tag{3.42}$$

with $p+q = n_{max} - n$.

Full forced fission

A special case of this biasing constitutes $wP_n^* = 1$, which implies the creation of a new fission neutron, with weight wP_m, at every collision. Now the score probability function becomes:

$$\pi(t_n, w, s) = \delta(s - wP_f Q_f) \otimes \pi(t_{n-1}, wP_s, s) \otimes \pi(t_{n-1}, wP_m, s) \tag{3.43}$$

To calculate the second moment, a pyramid of M_2s is needed.

$$\begin{aligned} M_2(n,p) = & P_s^{2p} P_m^{2q} P_f^2 Q_f^2 + M_2(n-1,p) + M_2(n-1,p+1) \\ & + 2(n-1) P_s^{2p} P_m^{2q} (P_f Q_f + P_m P_s P_f Q_f)(n-1) P_f Q_f \end{aligned} \tag{3.44}$$

again with $p+q = n_{max} - n$.

Resulting efficiency

To see if these techniques help to reduce the cost of a calculation, these methods are tested in a test case with a system where $P_s = 0.3$ and $P_m = 0.7$. The theoretical FoM has been plotted against the amount of biasing used, P_m^*, in Fig. 3.5.

The graph shows that using forced fission increases the efficiency of a calculation when $P_m^* > P_m$. The highest benefit is obtained when using complete forced fission, $P_m^* \approx 1$. This

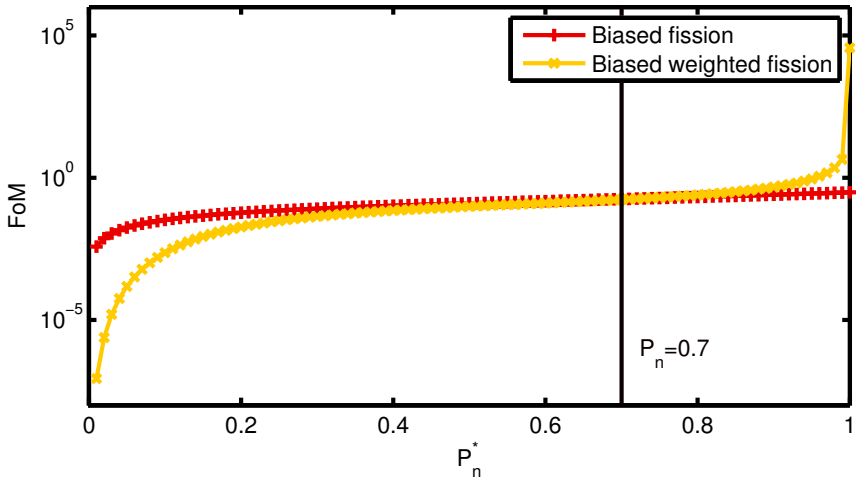


Figure 3.5: The analytical FoM for a system without leakage, when using biased fission and biased forced fission; the unbiased case is $P_m^* = 0.7$.

result is valid for this simple system with no leakage, because here there is zero variance when forced fission and implicit absorption are used. This gives an unrealistic view, because the calculation time can now approach infinity and still the FoM will be infinite. To make the system more realistic and to shorten the fission chains, leakage and Russian roulette are introduced in the next section.

3.4.2 Fission chain termination

A neutron fission chain can be ended in a few ways. It can cross the final time boundary; when implicit capture and forced fission are used, as explained before, this is the only method to end a chain. In a more realistic system, a neutron can also leak out of the system and in this case the chain is also terminated. In most Monte Carlo simulations, the chain can also end by Russian roulette. As the weight of a particle decreases, the importance of this particle reduces until it is no longer worth spending simulation time on it. The particle will then undergo a Russian roulette and can either be terminated or its weight is increased. These mechanisms are also incorporated into the model, to obtain a more realistic view on the variance.

Russian roulette

Long calculation times with implicit capture and forced fission can be reduced by Russian roulette. The moment equations can take this also into account. With Russian roulette, the particle can either survive or be killed at the end of a collision: z_1 indicates the particle's survival probability and z_0 the termination probability of the particle. These probabilities are given by

$$z_1(t_n, w) = \begin{cases} w/w_{surv}(t_n) & w \leq w_{rr} \\ 1 & w > w_{rr} \end{cases} \tag{3.45}$$

$$z_0(t_n, w) = \begin{cases} 1 - w/w_{surv}(t_n) & w \leq w_{rr} \\ 0 & w > w_{rr} \end{cases} \tag{3.46}$$

The weight of the surviving particle is w^* and it is given by

$$w^*(t_n, w) = w \qquad w > w_{rr} \tag{3.47}$$

$$w^*(t_n, w) = w_{surv}(t_n) \qquad w \leq w_{rr} \tag{3.48}$$

Russian roulette can be applied on the scattering neutron and on the fission neutron. These two roulette games are denoted by z_{1s} and z_{1f}, respectively. The score probability function now becomes

$$\begin{aligned} \pi(t_n, w, s) = & z_{0f} z_{0s} \delta(s - w P_f Q_f) + \\ & z_{0f} z_{1s} \Big(\delta(s - w P_f Q_f) \otimes \pi\big(t_{n-1}, w^*(t_{n-1}, w P_s), s\big) \Big) + \\ & z_{1f} z_{0s} \Big(\delta(s - w P_f Q_f) \otimes \pi\big(t_{n-1}, w^*(t_{n-1}, w P_m), s\big) \Big) + \\ & z_{1f} z_{1s} \Big(\delta(s - w P_f Q_f) \otimes \pi\big(t_{n-1}, w^*(t_{n-1}, w P_s), s\big) \\ & \otimes \pi\big(t_{n-1}, w^*(t_{n-1}, w P_m), s\big) \Big) \end{aligned} \tag{3.49}$$

This results in the following equation for the second moment:

$$\begin{aligned} M_2(t_n, w) = & w^2 P_f^2 Q_f^2 + \\ & z_{1f} \Big(M_2\big(t_{n-1}, w^*(t_{n-1}, w P_m)\big) + \\ & 2 w P_f Q_f M_1\big(t_{n-1}, w^*(t_{n-1}, w P_m)\big) \Big) + \\ & z_{1s} \Big(M_2\big(t_{n-1}, w^*(t_{n-1}, w P_s)\big) + \\ & 2 w P_f Q_f M_1\big(t_{n-1}, w^*(t_{n-1}, w P_s)\big) \Big) + \\ & z_{1f} z_{1s} 2 w^*(t_{n-1}, w P_s) w^*(t_{n-1}, w P_m) {M_1}^2(t_{n-1}) \end{aligned} \tag{3.50}$$

To calculate the second moments, a few more simulation weights are now possible and the M_2 matrix becomes 4 dimensional. In this equation n indicates the time step, p the

number of scattering events since the start of the particle or its last Russian roulette survival, q the number of fission events since the start or the roulette and u the starting weight, which can be 1, $w_{s,surv}$ or $w_{f,surv}$.

$$\begin{aligned} M_2(n,p,q,u) = & w^2 P_f^2 Q_f^2 + z_{1s} M_2(n-1, p1, q1, u1) + \\ & z_{1f} M_2(n-1, p2, q2, u2) + \\ & 2w P_f Q_f (z_{1s} w_s^* + z_{1f} w_f^*) M_1(n-1) + \\ & 2 z_{1f} z_{1s} w_f^* w_s^* M_1(n-1)^2 \end{aligned} \tag{3.51}$$

Here $p1$, $q1$ and $u1$ are the values in the case of a scattering event and $p2$, $q2$ and $u2$ in the case of a fission event. The weights $w_{f/s}^*$ are the weights after either fission or scattering.

These equations are used to calculate the FoM in an infinite system, again with $P_s = 0.3$ and $P_m = 0.7$. The fission and scattering Russian roulette survival weights are varied from 0 to 1 and the Russian roulette threshold is half the survival weight.

The resulting FoM is plotted in Fig. 3.6, where one horizontal axis shows the survival weight for fission and the other horizontal axis the survival weight for scattering. This plot shows it might be beneficial to have different Russian roulette thresholds for different interaction types. The results are difficult to compare with Fig. 3.5, since the addition changed fundamentally

Leakage

Finally, leakage is introduced into the system to create a system that cannot be solved with zero variance using implicit absorption and forced fission. Leakage can be simulated implicitly, but typically it is simulated explicitly in Monte Carlo calculations. The leakage is modelled in a simple way. After each collision, the particle has a probability $1 - P_{nl}$ that it will leak out of the system, with P_{nl} the average non-leakage probability. This is not an exact model of leakage, because in reality the leakage probabilities at two consecutive collisions are correlated: if the first collision was in the centre of the system, the following collision also has a low probability of being outside the system. The leakage is modelled this way, to be able to incorporate the leakage in the moment equations.

The particle can already leak before the first interaction. Therefore the score probability

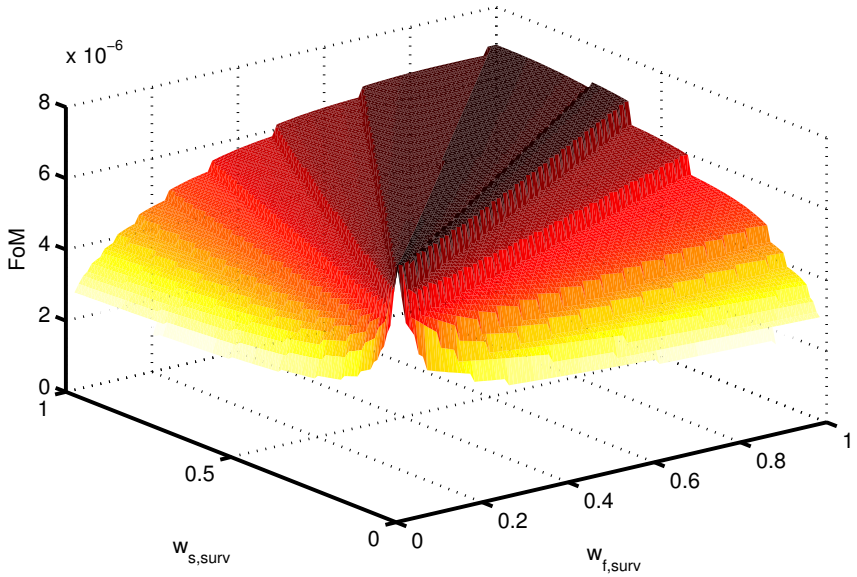


Figure 3.6: The analytical FoM at different values for the scattering and fission Russian roulette thresholds.

function becomes

$$\begin{aligned}\pi(t_n,w,s) = P_{nl}\Big(& z_{0f}z_{0s}\delta(s-wP_fQ_f)+\\ & z_{0f}z_{1s}\Big(\delta(s-wP_fQ_f)\otimes\pi\big(t_{n-1},w^*(t_{n-1},wP_s),s\big)\Big)+\\ & z_{1f}z_{0s}\Big(\delta(s-wP_fQ_f)\otimes\pi\big(t_{n-1},w^*(t_{n-1},wP_m),s\big)\Big)+\\ & z_{1f}z_{1s}\Big(\delta(s-wP_fQ_f)\otimes\pi\big(t_{n-1},w^*(t_{n-1},wP_s),s\big)\otimes\\ & \pi\big(t_{n-1},w^*(t_{n-1},wP_m),s\big)\Big)\Big)\end{aligned} \tag{3.52}$$

The first moment becomes

$$\begin{aligned}M_1 = & P_{nl}wP_fQ_f + P_{nl}z_{1s}M_1\big(t_{n-1},w^*(t_{n-1},wP_s)\big)+\\ & P_{nl}z_{1f}M_1\big(t_{n-1},w^*(t_{n-1},wP_fQ_f)\big)\end{aligned} \tag{3.53}$$

and the second moment

$$\begin{aligned} M_2(t_n,w) = P_{nl}\Big(& w^2 P_f^2 Q_f^2 + \\ & z_{1f}\Big(M_2\big(t_{n-1},w^*(t_{n-1},wP_m)\big) + 2wP_fQ_fM_1\big(t_{n-1},w^*(t_{n-1},wP_m)\big)\Big) + \\ & z_{1s}\Big(M_2\big(t_{n-1},w^*(t_{n-1},wP_s)\big) + 2wP_fQ_fM_1\big(t_{n-1},w^*(t_{n-1},wP_s)\big)\Big) + \\ & z_{1f}z_{1s}2w^*(t_{n-1},wP_s)w^*(t_{n-1},wP_m){M_1}^2(t_{n-1})\Big) \end{aligned} \tag{3.54}$$

Since these equations are similar to the ones without leakage, the calculation of the variance has a similar complexity. The M_2-matrix now becomes

$$\begin{aligned} M_2(n,p,q,u) = P_{nl}\Big(& w^2 P_f^2 Q_f^2 + z_{1s}M_2(n-1,p1,q1,u1) + \\ & z_{1f}M_2(n-1,p2,q2,u2) + 2wP_fQ_f(z_{1s}w_s^* + z_{1f}w_f^*)M_1(n-1) + \\ & z_{1f}z_{1s}2w_f^*w_s^*M_1^2(n-1)\Big) \end{aligned} \tag{3.55}$$

For the Monte Carlo simulation this case with leakage is more interesting, because the leakage is usually not simulated implicitly. In the previous cases zero variance could be achieved, if the appropriate variance reduction techniques were used. Without a relative uncertainty, it is difficult to define a useful FoM for a calculation. Also, the robustness of the variance reduction is now shown. Leakage introduces a stochastic process that is not directly targeted by one of the variance reduction techniques and therefore the robustness can be investigated.

In this test case, P_m is increased to 0.77 and P_s is increased to 0.33, ensuring the same ratio between the two probabilities as in the infinite system. The non-leakage probability is set to $P_{nl} = \frac{1}{P_m+P_s} = 0.91$. This way the system remains critical.

The results are plotted in Fig. 3.7 and it shows that the difference between the ideal scattering and fission Russian roulette thresholds has become smaller, but it is still present. Also, it shows the importance to have both the implicit absorption and forced fission Russian roulette thresholds at a similar level.

3.4.3 Tailoring the variance reduction per time step

Another variance reduction technique that can be used for a system with time evolution is to divide the system into time steps. Now the Russian roulette can be adjusted during the simulation, creating a more effective particle weight distribution in time. When the contribution of a fission chain to the power production in a certain time interval is calculated, only the interactions in that specific interval score power, but all previous interactions have

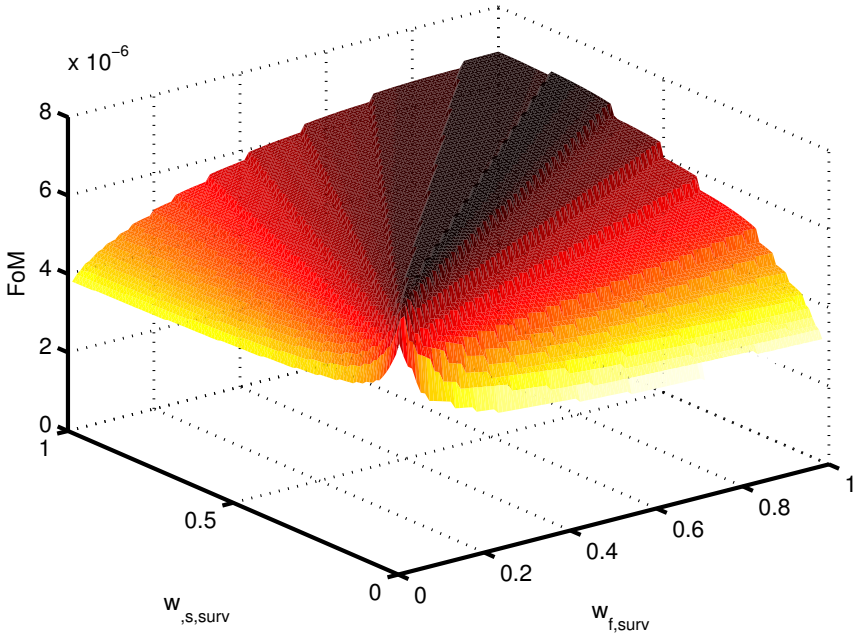


Figure 3.7: The analytical FoM at different values for the scattering and fission Russian roulette thresholds in a system with leakage.

to be considered as well to incorporate all possible branches of a fission chain that will contribute to the power tally in the time interval that is tallied.

The particles that cross a time boundary undergo a Russian roulette, which creates particles at the beginning of the next time step that have equal weights. This way the expected contribution of all particles at the beginning of a time interval is the same.

The score probability function stays the same as in Eq. (3.52), but now:

$$Q_f = 0 \qquad \text{if } t < t_b \tag{3.56}$$

Here t_b is the lower time boundary of the time interval where the power is tallied and the power production is set to zero for all the time intervals before t_b. The z-values of the Russian roulette have different values in the different time intervals and at the time boundaries. The Russian roulette threshold is set to 0.25 times the mean particle weight in the last interval, and the survival weight is set to 2 times the Russian roulette threshold. These values are very commonly used (X - 5 Monte Carlo Team, 2005). At time boundary

t_n, the threshold is equal to the survival weight and is set to $\phi(t_n)/\phi(t_1)$. Here $\phi(t_n)$ is the neutron flux at the beginning of the time interval and $\phi(t_1)$ is the flux at the beginning of the calculation. This will scales the weight of the particles to the increase in neutron flux: if the neutron flux is twice as high, the weight of the neutrons also becomes twice as high, retaining the total particle number roughly constant.

3.4.4 Branchless method

Besides biasing the fission probability, a second new method aims at reducing the variance in the power production via the chain lengths. This method uses analog scattering combined with forced fission. At a collision only one neutron will continue, either the scattered neutron or a fission neutron. The neutron has an analog probability P_s to have a scattering interaction, with an unchanged simulation weight.

If the neutron does not scatter, it terminates, producing a new fission neutron with weight:

$$w_{bls} = w\frac{\nu\Sigma_f}{\Sigma_a} \tag{3.57}$$

Here Σ_a is the absorption cross section which is equal to

$$\Sigma_a = \Sigma_t - \Sigma_s \tag{3.58}$$

This method ensures that the fission chain will not branch. However, because the weight of the neutron can increase significantly using this method, particle splitting is used in combination with this method. The neutron can be split, due to its high weight. The score probability function in this case becomes

$$\begin{aligned}\pi(T_n, w, s) = P_{nl}\Big(P_s\big(z_{0s}\delta(s - wP_fQ_f) + \\ z_{1s}\delta(s - wP_fQ_f) \otimes \pi(t_{n-1}, w, s)\big) + \\ (1-P_s)\big(\delta(s - wP_fQ_f) \otimes \prod_{j=1}^{N_{split}} \otimes z_{1f}\pi_j(t_{n-1}, w^*, s)\big)\Big)\end{aligned} \tag{3.59}$$

with

$$\prod_{j=1}^{N_{split}} \otimes z_{1f}\pi_j(t_{n-1}, w^*, s) = z_{1f}\pi_1(t_{n-1}, w^*, s) \otimes \ldots \otimes z_{1f}\pi_{N_{split}}(t_{n-1}, w^*, s) \tag{3.60}$$

and π_j is the score probability function of the j-th particle. As all particles resulting from splitting behave identically:

$$\pi_j(t_{n-1}, w^*, s) = \pi(t_{n-1}, w^*, s) \tag{3.61}$$

This represents the splitting of the particle into N_{split} particles and the convolution of all resulting particles. The splitting is used for the case where $k_\infty = \nu\Sigma_f/\Sigma_a > 1$. Then the weight of the fission neutron will be larger than the weight of the original neutron. The particle is split when its weight rises above a threshold w_{split}. For applying Russian roulette on the time boundaries the factors z_s and z_f are used again. In this case these factors are equal to 1, except if the particle crosses a time boundary. In this case the factor is

$$z_1 = \begin{cases} \frac{w_{bls}}{w_{surv}} & \text{fission at time boundary crossing and } w_{surv} \geq w_{bls} \\ \frac{w}{w_{surv}} & \text{scattering at time boundary crossing and } w_{surv} \geq w \\ 1 & \text{otherwise} \end{cases} \tag{3.62}$$

and w_{surv} the weight at the start of the new time interval. The new weight of the neutron which continues after the collision is given by

$$w* = \begin{cases} w_{surv} & \text{fission at time boundary crossing and } w_{surv} \geq w_{bls} \\ \frac{w_{bls}}{N_{split}} & \text{fission otherwise} \\ w_{surv} & \text{scattering at time boundary crossing and } w_{surv} \geq w \\ \frac{w}{N_{split}} & \text{scattering otherwise} \end{cases} \tag{3.63}$$

with

$$N_{split} = \left[\frac{w_{bls}}{w_{split}} + 1 \right] \tag{3.64}$$

The square brackets denote the integer part of the fraction.

3.5 Comparison of different variance reduction methods

It is interesting to compare these new variance reduction techniques with existing methods in production codes. However, when comparing these codes the calculation speed of the code becomes an important factor. Therefore the actual production codes are not used, instead the methods used in the codes are implemented in a simple Monte Carlo program. The variance and calculation speeds are also calculated analytically. An analytical FoM, as defined by Eq. (3.35), is calculated using these quantities to determine the efficiency of the methods used in the different production codes.

The sample problem on which all calculations in this section are performed, is a finite homogeneous system with one energy group and isotropic scattering. Only prompt neutrons are considered. The calculation is done for a 1 second time range and the power production is sampled in ten time intervals of 100 ms.

3.5.1 MCNP

The computer code MCNP5 (X - 5 Monte Carlo Team, 2005) is used in *shielding* mode with a neutron source at $t = 0$ and the scores are tallied in time bins. When a neutron enters a collision in MCNP5, implicit capture is applied first and then fission is sampled, with a probability of

$$P_{fission} = \frac{\Sigma_f}{\Sigma_f + \Sigma_s} \tag{3.65}$$

If fission takes place the number of new fission neutrons is sampled using

$$n = [\overline{\nu} + \rho] \tag{3.66}$$

where ρ is a random number uniformly distributed between 0 and 1, $\overline{\nu}$ is the average number of new neutrons in a fission reaction. The weight of the resulting neutrons is the same as that of the original neutron. The score probability function becomes in this case:

$$\begin{aligned}\pi(t_n, w, s) = P_{nl}\Big(& z_{0s}\delta(s - wP_fQ_f) + z_{1s}(1 - P_{fission}) \\ & \big(\delta(s - wP_fQ_f) \otimes \pi(t_{n-1}, w, s) + z_{1s}P_{fission}\nu^- \\ & \big(\delta(s - wP_fQ_f) \otimes \pi(t_{n-1}, w, s) \otimes \pi(t_{n-1}, w, s)\big) + \\ & z_{1s}P_{fission}\nu^+\big(\delta(s - wP_fQ_f) \otimes \pi(t_{n-1}, w, s) \\ & \otimes \pi(t_{n-1}, w, s) \otimes \pi(t_{n-1}, w, s)\big)\Big)\end{aligned} \tag{3.67}$$

Here ν^- denotes the probability that the number of fission neutrons is rounded to the lower integer and ν^+ the probability that the number is rounded to the higher integer.

The Russian roulette threshold is set to 0.25 and the survival weight to 0.5, which is standard in MCNP5.

3.5.2 Tripoli

To do this kind of calculation with the Tripoli 4.7 computer code (TRIPOLI-4 Project Team, 2010), the *fixed sources criticality* mode has to be used. Tripoli uses implicit absorption and the probability that a new fission neutron will be produced is given by

$$P_{fission} = \frac{\nu\Sigma_f}{\Sigma_t} \tag{3.68}$$

In general, only zero or one new neutron will be generated, unless $P_{fission}$ becomes larger than one. In this case the number of new neutrons will be given by

$$n = \left[\frac{\nu\Sigma_f}{\Sigma_t} + \rho\right] \tag{3.69}$$

Again n is rounded down to an integer. The weight of the fission neutrons is the same as the weight of the original neutron. In this case the score probability function becomes:

$$\begin{aligned}\pi(t_n,w,s) = P_{nl}\Big(&z_{0s}(1-P_{fission})\delta(s-wP_fQ_f)+\\ &z_{1s}(1-P_{fission})\big(\delta(s-wP_fQ_f)\otimes\pi(t_{n-1},w,s)\big)+\\ &z_{0s}P_{fission}\big(\delta(s-wP_fQ_f)\otimes\pi(t_{n-1},w,s)+\\ &z_{1s}P_{fission}\big(\delta(s-wP_fQ_f)\otimes\pi(t_{n-1},w,s)\otimes\pi(t_{n-1},w,s)\big)\Big)\end{aligned} \tag{3.70}$$

The Russian roulette threshold is 0.8 and the survival weight is 1, which are the standards for Tripoli4.

3.5.3 Forced fission and branchless method

These two methods from general purpose codes are compared to two methods that have been deduced using the method described in the previous sections. The first method uses forced fission and implicit absorption, but also uses Russian roulette and adjusts the weight windows each time step. The score probability function of this method is shown in Eq. (3.52). The Russian roulette threshold has been set to 0.25 and the survival weight to 0.5 for both fission and scattering. The other method is the branchless collision method. This method also adjusts the weight windows each time step and uses particle splitting. The score probability function is given by Eq. (3.59) and w_{split} is set to 1.

3.5.4 Improved Branchless method

The branchless method has been extended, to incorporate heterogeneous systems. The general philosophy remains the same, but it has been extended to be used in a more general case. When a neutron interacts with a purely absorbing material or a material with a small probability for fission: $(\nu_p\Sigma_f \ll \Sigma_a)$ the resulting neutron can have zero or a very low statistical weight. The particle will then be killed soon by Russian roulette, terminating the fission chain. Therefore in the improved method the resulting particle always gets the same statistical weight after each interaction. This statistical weight is given by

$$w_{bls} = w\frac{\nu_p\Sigma_f+\Sigma_s}{\Sigma_t} \tag{3.71}$$

Now the probability of having a scattering has to be modified and becomes for scattering

$$P_{scattering} = \frac{\Sigma_s}{\nu_p\Sigma_f+\Sigma_s} \tag{3.72}$$

and for fission

$$P_{fission} = \frac{\nu_p \Sigma_f}{\nu_p \Sigma_f + \Sigma_s} \tag{3.73}$$

Using this method the total scattering weight and total fissioning weight stay unbiased and the method remains useful in extreme conditions. In the case of a purely fissioning material this turns the method into forced fission method, whereas if there is no fission, the method behaves as implicit scattering.

In this case the equation for the scoring function becomes more simple. In a system with continuous energy and anisotropic scattering the distinction between fission and scattering neutrons must be made, but in our mono-energetic and isotropic case they can be treated the same.

$$\pi(t_n, w, s) = P_{nl}\left((\delta(s - w P_f Q_f) \otimes \prod_{j=1}^{N_{split}} \otimes z_1 \pi_j(t_{n-1}, w^*, s)) \right) \tag{3.74}$$

with

$$z_1 = \begin{cases} \frac{w_{bls}}{w_{surv}} & \text{at time boundary crossing and } w_{surv} \geq w_{bls} \\ 1 & \text{otherwise} \end{cases} \tag{3.75}$$

and w_{surv} the weight at the start of the new time interval. The new weight of the neutron which continues after the collision is given by

$$w* = \begin{cases} w_{surv} & \text{at time boundary crossing and } w_{surv} \geq w_{bls} \\ \frac{w_{bls}}{N_{split}} & \text{otherwise} \end{cases} \tag{3.76}$$

with

$$N_{split} = \left[\frac{w_{bls}}{w_{split}} + 1 \right] \tag{3.77}$$

For this improved branchless technique the equation for the second moment becomes

$$\begin{aligned} M_2(t_n, w) = P_{nl}\Big(& w^2 P_f^2 Q_f^2 + \\ & z_1 N_{split} \Big(M_2(t_{n-1}, w^*) + 2 w P_f M_1(n-1, w^*) + \\ & z_1 N_{split} (N_{split} - 1) M_1^2(n-1, w^*) \Big) \Big) \end{aligned} \tag{3.78}$$

3.5.5 Results

The first step is now to verify the results of the analytical calculation with the results of actual Monte Carlo calculations, by means of a system with a short timespan. This short timespan ensures that the Monte Carlo simulations will get reliable results in a reasonable

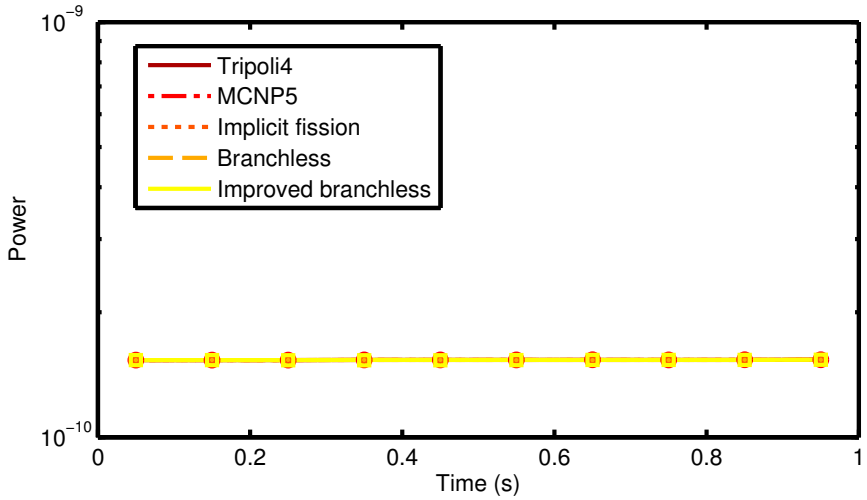


Figure 3.8: The analytically calculated variance of the different simulation schemes compared with the variance observed in the Monte Carlo simulations that use the methods as designated, but have the simplifications of the model in place. The lines indicate the analytical solution and the markers are the Monte Carlo results.

amount of time. To compare the different methods fairly the analytical FoM, defined in Eq. (3.35), is used again and the calculations are performed by a simple Monte Carlo program that can also incorporate the methods of MCNP5 and Tripoli4 in such a system.

For the test problem, the following probabilities have been used: $P_s = 0.4118$, $P_m = 0.625$, $P_f = 0,25$, $\overline{\nu} = 2.5$, $Q_f = 200 \times 1.59 \times 10^{-13}$J, $P_{nl} = k_{eff}/(P_s + P_m) = 0.96458$ and the $k_{eff} = 1.00008$. The k_{eff} is not precisely one, because this system mimics the finite block described in Chap. 2. The time of flight between two consecutive collisions is set to 45 μs.

The first calculation has its final time boundary at $t = 1$ ms. In this case the neutrons have 22 collisions before the end of the simulation. Since the time span of the calculation is short, the result can be calculated very accurately with Monte Carlo and the validity of the analytical solution can be checked. The results are plotted in Fig. 3.9. The results of the real Monte Carlo simulations are given by the markers. The results of the analytical solutions are given by the lines. The markers overlap the lines, demonstrating that the a priori calculation of the result and the variance is correct.

Next, the calculated σ_r is compared to the σ_r that is actually present in MCNP5 and Tripoli4. This time the power production is calculated over a period of 1 s. The results are shown in Fig. 3.10. This plot shows that the analytical σ_r agrees nicely with the σ_r in the

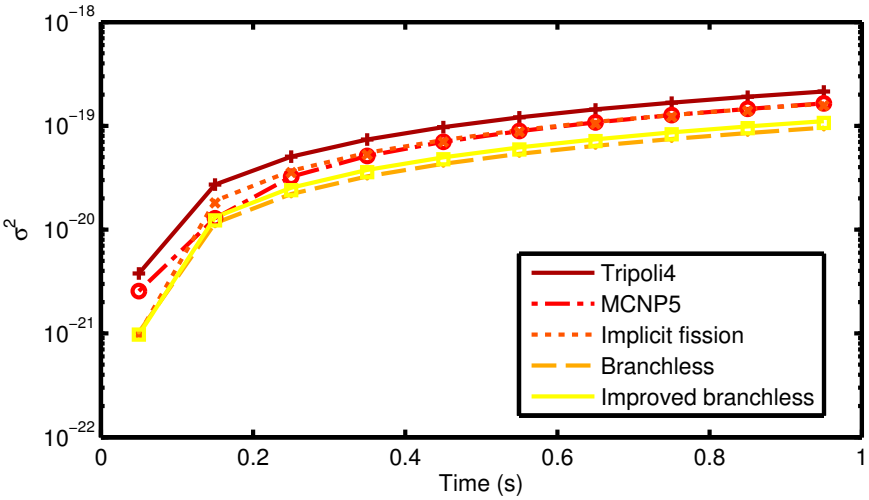


Figure 3.9: The analytically calculated variance of the different simulation schemes is compared with the variance observed in the Monte Carlo simulations that use the methods as designated, but have the simplifications of the model in place. The lines indicate the analytical solution and the markers are the Monte Carlo results.

simulations. The difference that can be noticed is due to the assumption for the leakage and the fixed time between two collisions. It can also be seen that the σ_r in this kind of calculation is high, although it should be noted that this σ_r is per source neutron.

Finally, the power production in this system is calculated for all five methods. The resulting FoMs have been plotted in Fig. 3.11. It shows that the newly developed methods have a higher efficiency than the methods currently used in general purpose Monte Carlo codes. The forced-fission method only alters the way fission is treated and is therefore only a modest change. When the branchless method is used, the scheme is quite different from current methods, but shows that it can be worth simulating fission chains differently in a multiplying system, when power (or flux) is to be calculated as a function of time. The efficiency is more than an order of magnitude better using the branchless methods. The difference between the two branchless methods is not shown in this simple case. The improved method is applicable in a more general way.

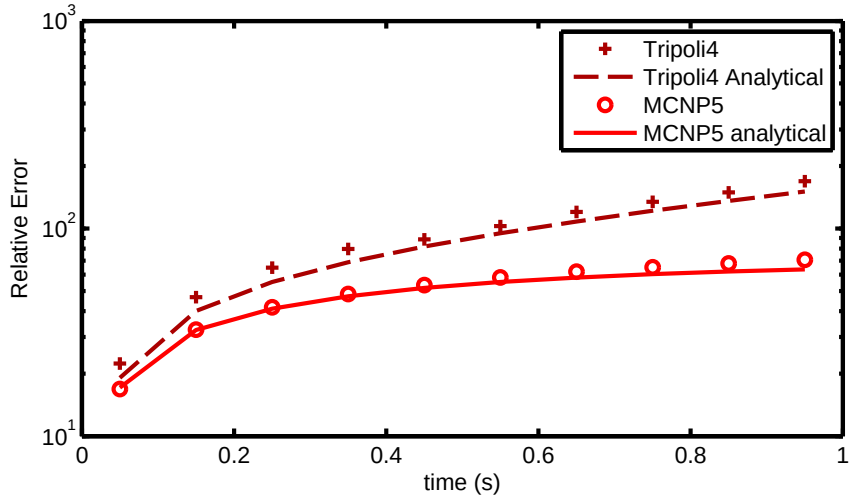


Figure 3.10: The analytically calculated σ_r of the different simulation schemes is compared with the σ_r actually observed in the simulations performed by these codes. The lines indicate the analytical solution and the markers are the Monte Carlo results.

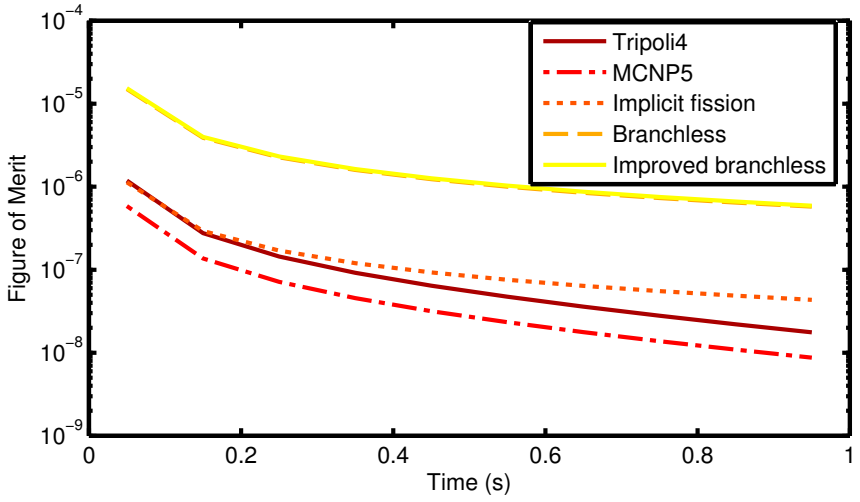


Figure 3.11: The analytically calculated FoM of the different simulation schemes. It shows that the FoM can be improved in a multiplying system with new variance reduction techniques.

3.6 Summary of improving chain length statistics

In this chapter new methods for reducing the variance have been introduced, which can be used when performing a Monte Carlo calculation in a multiplying system. These methods have been deduced using a priori variance calculations in a simplified model.

The first part of this chapter describes an analytical framework that can be used to calculate a priori the variance in a simplified model of a fixed-source Monte Carlo calculation. This method uses moment equations to calculate the first and second moment and from that the variance of the calculation. This is exact for a simple homogeneous system, with no or simplified leakage, no energy dependence and a fixed time step length. It was also shown that this model is valid for the variance in a mono-energetic, homogeneous system.

Promising new variance reduction methods have been deduced using this analytical frame. The first new method is the implementation of forced fission. It has been shown that the reduction of the variance in the prompt fission chains is maximal when the biased forced-fission probability is set to one, creating a new fission neutron at every collision. The weight of this fission neutron is adjusted to ensure a fair game. Secondly, it was shown that it can be useful to use analog scattering together with forced fission. In this case the fission chain will not branch, which is better for the efficiency of the calculation.

These new methods have been used in a simple case and compared with the variance that a general purpose Monte Carlo code would give. To ensure a fair comparison, a theoretical FoM was introduced to compare methods and not computer codes. In the test problems, the methods deduced with the a priori calculations outperform the methods in the general purpose codes MCNP5 and Tripoli4 in a multiplying system. In a test problem the branchless method had a FoM which was more than 10 times higher compared to the conventional methods. This will make it possible to use Monte Carlo in a wider spectrum of applications, since the methods can be applied in a general case.

The improved method has been successfully implemented in a purposely built Dynamic Monte Carlo code and the Tripoli4 code. The variance reduction achieved in a simulation of a more realistic geometry is shown in Chap. 4 and Chap. 5.

3.7 Bibliography

T. Booth. *Analysis of error in Monte Carlo transport calculations.* Ph.D. thesis, Los Alamos Scientific Lab., NM (USA), 1979.

I. Lux and L. Koblinger. *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations.* CRC Press, Boca Raton, United States of America, 1991.

C. A. L. Micoulaut. The n^{th} centred moment of a multiple convolution and its applications in an intercloud gas model. *Astronomy and Astrophysics,* **51**, 343, 1976.

K. O. Ott and R. J. Neuhold. *Introductory Nuclear Reactor Dynamics*. American Nuclear Society, La Grange Park, United States of America, 1985.

I. Pazsit and L. Pal. *Neutron Fluctuations, A treatise on the physics of branching processes*. Elsevier, Amsterdam, the Netherlands, 2008.

R. P. Stanley. *Enumerative combinatorics. Vol. 2*. Cambridge University Press, 1999.

TRIPOLI-4 Project Team. *TRIPOLI-4 version 7 user guide*. CEA, serma/ltsd/rt/10-4941/a edition, 2010.

A. J. van Wijk and J. E. Hoogenboom. A priori efficiency calculations for Monte Carlo applications in neutron transport. *Progress in Nuclear Science and Technology*, **2**, 722, 2011.

X - 5 Monte Carlo Team. *MCNP – A General Monte Carlo N-Particle Transport Code*. Los Alamos National Laboratory, Version 5, LA-UR-03-1987 edition, 2005.

Part III

Implementation of the Dynamic Monte Carlo Method

4

The Dynamic Monte Carlo Method: Proof of Principle

Picture: B.L. Sjenitzer

In this chapter the development of some practical Monte Carlo techniques for long-time kinetic and dynamic calculations is described. To demonstrate the principle of dynamic Monte Carlo the new techniques are developed for a mono-energetic, homogeneous system with isotropic scattering in a purpose-built Monte Carlo code, the Dynamic Monte Carlo code (DMC).

The necessary modifications to the simulation scheme are discussed in section 4.1. In a dynamic calculation the time domain is cut into time intervals and these time intervals are simulated consecutively. This is needed for the implementation of a system in which the geometry or materials change in time and for calculating thermal-hydraulic feedback with an external code. Extra attention is given to parallel computing issues, since dynamic calculations are expected to be computationally expensive. When computing in parallel it is useful to be able to reproduce the results and this requirement determines the way to generate the random numbers and restricts the parallel dynamic Monte Carlo scheme.

Not only the simulation scheme is modified, but also the tallying; in a steady-state calculation the time dependence in a Monte Carlo tally is only present implicitly, but in the case of a dynamic calculation there is an explicit time factor that needs to be handled. Also it is desirable that the tallies in the initial state of a dynamic calculation can be compared to the tallies in a criticality calculation. Furthermore, special attention must be paid to the variance estimation in a dynamic calculation. These issues are discussed in section 4.2.

Finally a numerical example is given in section 4.3 on a critical system with a positive reactivity insertion from $t = 10$ s to $t = 40$ s. After this insertion the system returns to its critical state, with the power stabilising at a new level.

4.1 Simulation scheme

Currently, Monte Carlo codes are mainly used for steady-state problems. However, in a dynamic problem the neutron flux can vary when system properties, such as temperature, density and material composition, change in time. For a Monte Carlo code to cope with such changing circumstances, the calculation scheme needs to be adapted. This will be explained in section 4.1.1.

The variance reduction techniques used in the dynamic calculation and the way to determine the thresholds of the weight windows are described in section 4.1.2.

Another issue is the determination of the initial conditions. The neutron and precursor sources must be determined, which is in fact a separate calculation performed prior to the dynamic part. This is discussed in section 4.1.3.

Finally there the parallelisation scheme applied to this calculation is discussed in section 4.1.4. In this stage of the development of a new method it is important to be able to reproduce the results of a calculation and therefore special attention must be paid to the

conservation of random number seeds.

4.1.1 Dividing the problem into time intervals

The first step is to divide the problem into separate time intervals and each time interval is simulated consecutively, as schematically represented in Fig. 4.1. The duration of the time intervals can be chosen freely and can even vary during the calculation and all precursors are forced to have a decay in every time interval.

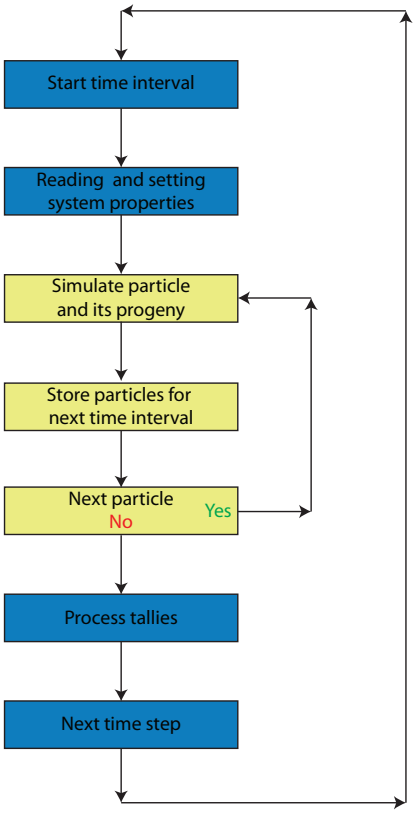


Figure 4.1: The scheme for performing a dynamic Monte Carlo calculation.

First the system properties of the new time interval are read, such as the geometrical layout of the system and the material properties. These properties can be altered in each time interval to represent external manipulation, or feedback mechanisms.

Next the particles are simulated, both precursors and neutrons. A particle is followed, until it is either killed by Russian roulette, leaked or it crosses the time boundary. A particle can create new particles due to decay, fission and splitting and these new particles are simulated directly after the current particle.

When a precursor is created, it is forced to have its first decay directly in the remainder of the time interval in which it is created. The decay time is sampled uniformly between the creation time of the precursor and the end of the time interval. The weight of the resulting particle is calculated with Eq. (2.19), but in this case Δt is the time remaining in the interval after precursor creation. The weight of the resulting neutron might be low, since the precursor is forced to have its decay in a fraction of the time interval and therefore it might be killed instantly by Russian roulette, but these particles need to be simulated to ensure an unbiased result.

Time boundary crossing

When a neutron crosses the time boundary of a time interval, it is stopped at the boundary and stored for the next time interval. The path length between two interactions is given by an exponential distribution and therefore it is allowed to simply stop the neutron at the time boundary and sample a new path length in the new time interval, using the new system properties.

A precursor which has created a delayed neutron via forced decay is always stored for the next time interval. Its creation time is stored as well, since this is needed to calculate the weight of the delayed neutron in the next interval.

Changing the dynamic system

When all particles have been simulated, the tallies are calculated. Before starting the next interval, changes can be applied to the system. This can be a mechanical change, such as the movement of a control rod or the injection of boron, but it can also be a feedback mechanism, which has to be calculated with an external code or within the Monte Carlo code itself (Griesheimer et al., 2008).

In case of thermal-hydraulic feedback the Dynamic Monte Carlo code outputs a power profile to the thermal-hydraulics code whereafter the thermal-hydraulics code calculates a new temperature profile. Next the Monte Carlo code continues with the following time interval, taking into account the cross sections at the new temperatures. This is discussed in more detail in Chap. 6.

4.1.2 Variance reduction

The variance reduction techniques deduced in Chap. 3 are applied in the program. At the collision the branchless method is used to keep the prompt neutron chains from branching unnecessarily. The weight of a particle leaving a collision is given by:

$$w_{new} = w_{old} \frac{\nu\Sigma_f + \Sigma_s}{\Sigma_t} \tag{4.1}$$

which implies a weight that increases for a critical finite system. Therefore particle splitting is used to keep the weights within reasonable limits.

Weight monitoring

At the beginning of a time interval the thresholds of the weight windows can be set. To keep the statistics in each time interval similar, the number of particles per interval is kept

roughly constant at the initial number of particles N. From this number of particles a desired average weight, W_{av}, can be calculated using the total weight of all M particles at the end of the previous interval.

$$W_{av} = \frac{1}{N}\sum_{i=1}^{M} w_i \tag{4.2}$$

This desired average weight is used to set the new Russian roulette and splitting thresholds:

$$W_x = w_x W_{av} \tag{4.3}$$

with W_x the absolute value of the threshold and w_x the relative threshold. The subscript x can be either *RR* for the Russian roulette threshold, *surv* for the survival weight, or *split* for the splitting threshold.

Population control

At the time boundary it is also possible to perform population control. This is mostly needed to prevent the accumulation of precursors. With the forced decay scheme, precursors are forced to have a decay in all time intervals. However, the importance of the precursor decreases with time, as explained in section 2.3.

For this first demonstration of the dynamic method, Russian roulette is used when storing particles for the next time interval. First the desired average weight is determined by Eq. (4.2) and then Russian roulette is played to give all particles which are stored at least the desired average weight. The probability for a neutron to be stored for the next interval is given by

$$P_{surv} = \frac{w}{W_{av}} \qquad \text{, with } w < W_{av} \tag{4.4}$$

and the weight of the surviving neutron is

$$w' = W_{av} \qquad w \leq W_{av} \tag{4.5}$$

$$w' = w \qquad w > W_{av} \tag{4.6}$$

For precursors the survival probability is given by:

$$P_{surv} = \frac{w_{d,av}}{W_{av}} \qquad \text{, with } w_{d,av} < W_{av} \tag{4.7}$$

and the weight of the surviving precursor is given by

$$w'_C = w_C \frac{W_{av}}{w_{d,av}} \qquad w_{d,av} \leq W_{av} \tag{4.8}$$

$$w'_C = w_C \qquad w_{d,av} > W_{av} \tag{4.9}$$

4.1.3 Source sampling

To sample a source distribution a criticality calculation is preformed. When a system is in a critical state the steady-state neutron flux distribution will be equal to the fundamental mode.

After the final cycle of the criticality calculation the fission source is, in the case of a homogeneous mono-energetic system, a good representation of the neutron flux distribution. To generate also a precursor source, a percentage of the source neutrons is converted to become precursors according to Eq. (2.45).

To create a proper precursor family distribution the precursors are given an age, using a uniform distribution, Eqs. (2.60) and (2.61) and a rejection scheme, Eqs. (2.62)-(2.64). After sampling the source, the actual dynamic program can start its calculation as described above.

4.1.4 Parallel calculation

To use this calculation scheme also in parallel, it is useful for research and validation purposes to keep the results independent from the number of processors used (Brown, 2008), ensuring reproducibility. To accomplish this, the random numbers used need to be the same, independent of the number of processors used.

To this end every particle is assigned a unique number. The particle generates a random number seed depending on this number. In the dynamic Monte Carlo scheme this is more difficult than in a steady state calculation, since a large number of new particles can be produced in a time interval, but a priori it is unknown what the number of new particles is. The solution is to continue the new particles with the random number sequence of the parent particle. This ensures the same random numbers are used independently of the number of processors.

At the beginning of a time interval all particles get a new unique particle number from which a random seed is produced. Then the particles are distributed over the available processors together with the particle numbers. Next, all processors will simulate particles and all their progeny; if a particle reaches the end of the time interval it will be stored. When all particles have finished, the stored particles are returned to the master processor, which will combine the stored particles in the right order and assign a new particle number to these particles and redistribute them over the processors. This is shown in Fig. 4.2.

As a particle (including its progeny) suffers many more collisions than usual in a criticality calculation, the period of the random number generator should be large enough. The random number package of the MCNP code contains such a generator, which is used in the dynamic Monte Carlo code. This random number generator has a period of 2^{46}, which is much larger than the expected number of interactions.

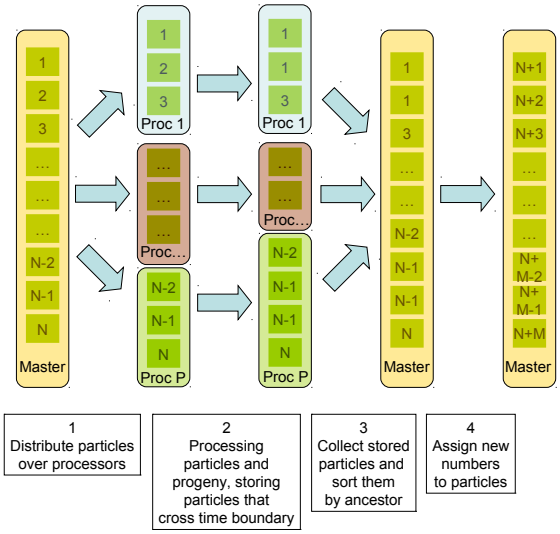


Figure 4.2: In a parallel calculation, the master processor distributes the N particles over all P processors. These processors simulate the particles, generating M new particles for the next time interval. These particles are sent back to the master processor and the master sorts them by ancestry. Then a new unique particle number is assigned to the particles for the next time interval.

The downside of this scheme is the extra communication between the master processor and its “slaves”. For development purposes, for which this first Dynamic Monte Carlo program is made, it is useful to be able to reproduce the results.

4.2 Tallies

4.2.1 Time averaged tallies in steady state

In most Monte Carlo calculations it is not the power production which is calculated, but the recoverable energy produced per neutron. This is a convenient quantity because the calculation of the power distribution in a nuclear reactor is usually done with a k_{eff} calculation. In a k_{eff} calculation the energy production is tallied per cycle. There is no time

component present and the exact number of neutrons at the start of a cycle is known. [1]

4.2.2 Tally normalisation for a dynamic calculation

Time-dependent calculations cannot apply the same normalisation as steady-state calculations. The number of neutrons in a time-dependent calculation is not exactly known, since many neutrons are produced in that specific time interval and many will cross into the next time interval. Also, the results can not be implicitly averaged over time, since the variation over time is the reason to do transient analysis. However this explicit presence of time in a transient analysis can also be considered to be an advantage: instead of applying tricks with implicit averaging, time is explicitly present.

The total power tally in volume V during a time interval from t until $t+\Delta t$ is given by:

$$P_V(t) = \frac{1}{D}\int_0^\infty \mathrm{d}E \int_t^{t+\Delta t} \mathrm{d}t' \int_V \mathrm{d}V \, \frac{Q\Sigma_f(E)}{\Sigma_t(E)}\psi(\boldsymbol{x}, E, t') \tag{4.10}$$

where D is the normalisation constant for a dynamic tally, Q is the average energy released at a fission event and $\psi(\boldsymbol{x}, E, t)$ is the interaction density. There are multiple possibilities to define the normalisation constant D and here two possibilities are discussed, which enable a comparison of a dynamic result with a steady-state result.

The first option is to normalise with the total weight of the particles, neutrons and precursors, starting the calculation, W_{tot} and the size of the time interval which is tallied:

$$D = \Delta t \times W_{tot} \tag{4.11}$$

Now Eq. 4.10 will yield a power produced per starting particle.

To enable a comparison between steady-state and dynamic calculations, the power production per neutron chain must also be calculated in a criticality calculation. The power tally in a criticality calculation is given by:

$$P_V(t) = \frac{1}{C}\int_0^\infty \mathrm{d}E \int_0^\infty \mathrm{d}t' \int_V \mathrm{d}V \, \frac{Q\Sigma_f(E)}{\Sigma_t(E)}\psi(\boldsymbol{x}, E, t') \tag{4.12}$$

This tally is integrated over all time, but to convert energy released to a power tally the actual time used by the neutrons is needed. The average lifetime of a neutron in a neutron chain is given by the effective lifetime of this neutron, ℓ_{eff}, which includes the lifetime of

[1] In fact, most tallies do not contain a time factor. For example the flux tally of MCNP is defined as particles/cm^2 per source particle (X - 5 Monte Carlo Team, 2005). The user can transform this into a time-dependent quantity by defining the source as particles per second. In this case the unit for the flux tally becomes $[\text{particles s}^{-1}\ \text{cm}^{-2}\ \text{neutron}^{-1}] = \text{flux/neutron}$ In the same way the energy production per neutron is often transformed into power produced per neutron $[\text{Js}^{-1}\text{neutron}^{-1}]$. This is allowed because most Monte Carlo problems are steady state. The results are implicitly averaged over time.

delayed neutrons and of prompt neutrons, since they are both present in neutron chains. In this case to convert the released energy in a criticality calculation to power produced, the normalisation constant C becomes:

$$C = \ell_{eff} \times W_{tot} \tag{4.13}$$

The second option is to normalise the dynamic calculation to the number of neutrons present in the system. In this case the delayed neutrons are considered as an external source. The normalisation constant D now becomes:

$$D = \Delta t \times W_{neutrons} \tag{4.14}$$

here $W_{neutrons}$ is the total weight of the neutrons starting the simulation. With this normalisation constant Eq. (4.10) will yield the power produced per starting neutron.

The energy deposition in a criticality calculation can also be converted to the power production per neutron in a dynamic calculation. This time only the prompt neutrons are considered and therefore the average lifetime of a prompt neutron, ℓ, must be used to normalise the power produced:

$$C = \ell \times W_{tot} \tag{4.15}$$

The different normalisation constants will yield a different power level, but in most practical applications the initial power production is a starting condition. However, these definitions do allow for a comparison between a dynamic and a steady-state calculation.

4.2.3 Calculating statistics

When tallying it is important to consider in what way the statistics are calculated. It is common to either calculate the variance between particles or between batches of particles (Hoogenboom, 2008). This is allowed, when the conditions of the central limit theorem are met. This theorem is a refinement of the law of large numbers and states (Laplace, 1820):

> The average of a large number of independent identically distributed random variables approximately has a normal distribution, no matter what the distribution of the random variables is.

In case of a Monte Carlo simulation of a nuclear system, the randomly distributed variables are the tallies of the neutrons and because the theorem is only valid if the variables are independent, the statistics must be calculated between the tallies scored per independent neutron. Since the neutrons are created in chains, some caution is required here. The source particles at the start of the calculation can be considered independent of each other, but after the source particles are released, the neutrons in the same chain are correlated.

For example, with forced precursor decay, a precursor creates delayed neutrons in all time intervals and therefore the power production in two time intervals are correlated. If the precursor population in interval n is higher than the true mean, then the precursor population in interval $n+1$ will have a high probability to be also above the true mean. Furthermore, if due to variance, the neutron flux is above the mean, this can cause a higher flux in the next interval as well.

To incorporate these effects in the variance estimation, the variance is calculated per source particle. To do this the identity of the original starting particle is passed along to new fission neutrons. The scores are collected per N original particles and the variance is calculated between these N scores.

4.3 Numerical Example

To demonstrate the dynamic Monte Carlo scheme, a sample problem has been set up. In this problem there is a small rectangular box of fissile material. The system has one energy group and there are six precursor families. The detailed description of the system can be found in Appendix A.1 and the system is designed in such a way that it is practically critical. The system is started at $t = 0$ s and at $t = 10$ s there is a reactivity insertion: Σ_a is decreased from 0.5882 cm^{-1} to 0.5870 cm^{-1} while keeping Σ_t constant and therefore increasing Σ_s .

The fundamental mode source distribution is calculated using 1000 cycles, starting the criticality calculation with a cosine shaped neutron source, which is calculated using standard diffusion theory. From this source the dynamic part of the calculation is started, and until the reactivity insertion the power level is nearly constant.

Then after the reactivity insertion there will be a prompt jump, followed by a neutron population which grows exponentially. Then at $t = 40$ s the system is returned to the critical state and after a negative prompt jump the system will return slowly to a new stable state, which is higher than the initial state.

The simulation is started with 10^7 particles and the time interval is 100 ms. The weight window thresholds are given by:

$$\begin{aligned} w_{RR} &= 0.25 \\ w_{surv} &= 0.5 \\ w_{split} &= 2 \end{aligned}$$

To verify the results produced by the Dynamic Monte Carlo code, the same problem has been calculated using a point-kinetics code and the results are plotted in Fig. 4.3. The results agree nicely with each other, which is expected, since the point-kinetics model is an accurate model for such a simple geometry and homogeneous cross-section change. The kinetic parameters, used in the point-kinetics, are calculated using a Monte Carlo

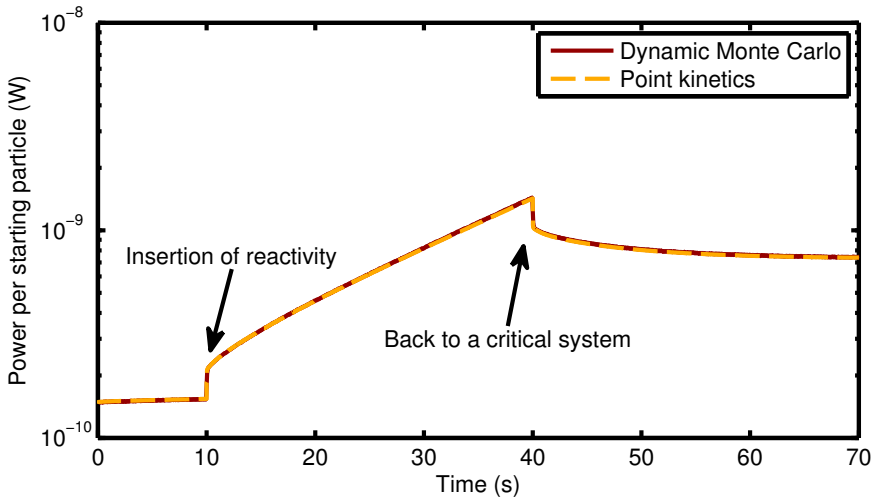


Figure 4.3: A critical system with a reactivity insertion at t=10 s. After a prompt jump the power starts to increase further, until the reactivity is set back to 0 at t=40 s. Now the system returns to a new stable state. The results dynamic Monte Carlo simulation agree very well with the point-kinetics analysis of the system.

calculation in criticality mode, both for the unperturbed and perturbed system. The generation time and the eigenvalue are sampled, while the effective delayed fraction is equal to the actual delayed fraction for a mono-energetic system.

When looking at the start of the calculation more closely, Fig. 4.4, it can be seen that the steady state power level of the dynamic calculation is not exactly equal to the power level calculated by the criticality calculation. This is due to the fact that the system is not exactly critical. The neutron and precursor source are calculated for steady state and therefore a small prompt jump can be observed at the start of the calculation, in agreement with point-kinetic theory.

The relative uncertainty for all time intervals in the calculation is less than 1% and the calculation time is approximately 100 CPU hours [2].

4.3.1 Variance reduction

It is also interesting to see how the variance reduction techniques help to increase the efficiency of the calculation. To this end, the FoM of the branchless method of Sec. 3.4.4 is

[2] A CPU hour is the one hour of calculation time on a single processor. If the calculation takes 30 minutes on two processors, it is also one CPU hour.

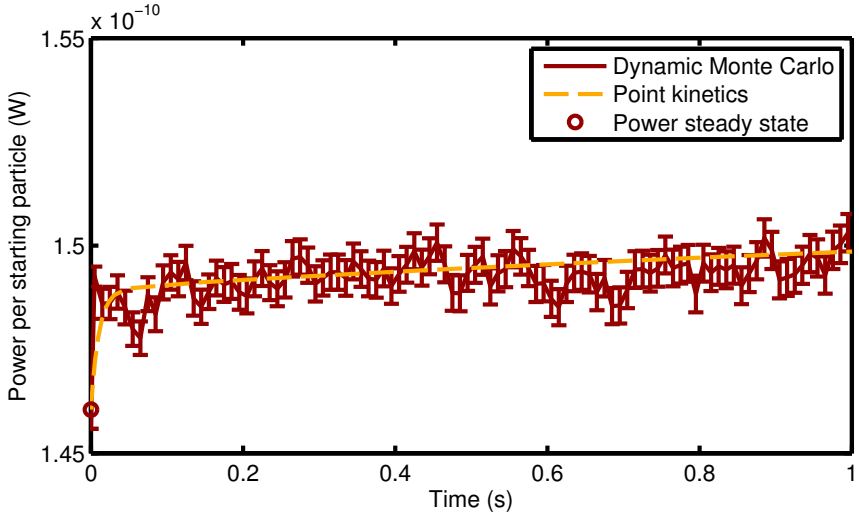


Figure 4.4: The first second of the power production in a near critical system calculated with dynamic Monte Carlo and point kinetics. A small prompt jump at the beginning of the calculation can be observed, which is due to the fact that the system is not exactly critical.

compared to the case where implicit absorption is used. With implicit absorption the new fission neutrons are selected using the method of Tripoli, Eq. (3.69). Also, forced precursor decay is compared to the analog method.

The difference in FoM can be seen in Fig. 4.5. This FoM is calculated using the total calculation time of the dynamic part and the relative uncertainty per time interval. It demonstrates the effectiveness of both variance reduction techniques.

The difference for the branchless method can be explained with the decreased variance in the prompt neutron chains. The relative standard deviation in the neutron chain length is plotted in Fig. 4.6. This demonstrates the effectiveness of the branchless method in improving the chain length statistics.

4.4 Summary of the proof of principle for the Dynamic Monte Carlo method

In this chapter the dynamic Monte Carlo method has been implemented and put to the test. For the implementation there are some issues which must be taken care of. The simulation scheme had to be adapted to incorporate a dynamic system properties. This is done by splitting the simulation domain into time intervals and simulating these time intervals

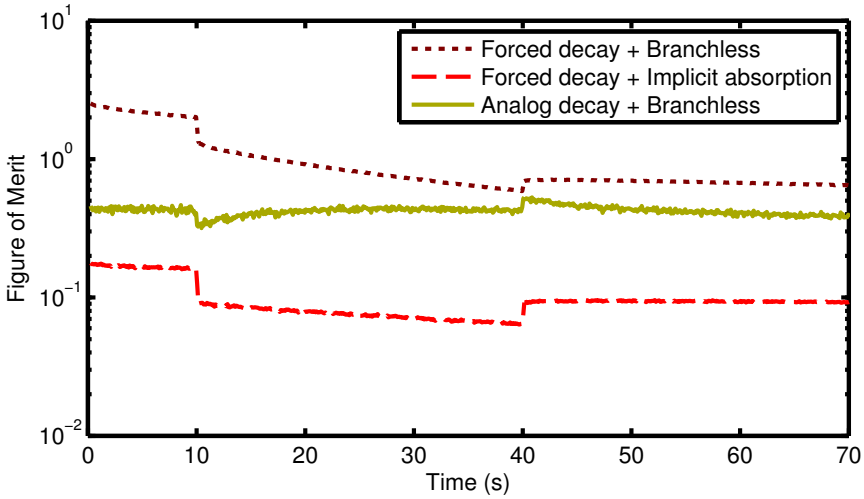


Figure 4.5: The figure of merit in each time interval of the calculation. The effect of the different variance reduction techniques is shown.

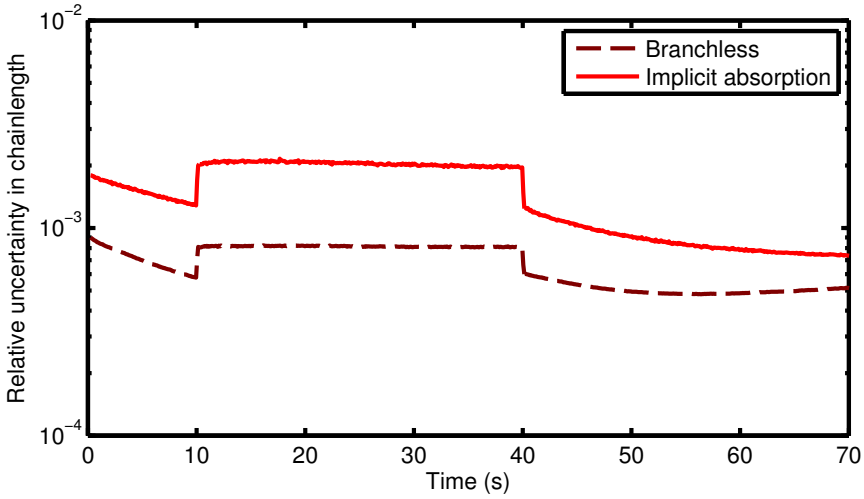


Figure 4.6: The relative standard deviation in the chain length estimation is shown. The branchless method has less variation in its chain lengths.

consecutively. However, some consideration had to be taken for neutrons crossing the time boundary.

The variance reduction techniques developed in the previous chapters have been implemented and to maintain the desired level of the weight windows, weight monitoring is performed. The thresholds of the weight windows are set proportional to the desired average neutron weight.

The geometrical source distribution is calculated using a criticality calculation. The precursor family distribution is calculated by sampling a uniformly distributed initial precursor age, together with a rejection scheme.

The tallies can be normalised either to the total particle weight at the beginning of the calculation, or to the total prompt neutron weight at the start. To calculate the variance of a tally, the statistics between the results of different source particles must be considered. This is due to the fact that particles in a time interval might be correlated, whereas the source particles are not.

Finally a parallel simulation scheme for is devised for the dynamic calculations. Since these calculations tend to be computational expensive, it is useful to be able to run the calculations in parallel. In such a scheme it is desirable to be able to reproduce the results, independent of the number of processors, in order to validate the calculation and compare results.

A sample case has been simulated. This sample case is a 3D block of homogeneous material, placed in vacuum. The neutrons are considered mono energetic and have isotropic scattering. A transient is introduced by changing the absorption cross section. The results of the Dynamic Monte Carlo method are compared with the results of a point-kinetics calculation. They agree nicely, as expected in this case.

4.5 Bibliography

F. B. Brown. Fundamentals of Monte Carlo particle transport. http://mcnp-green.lanl.gov/resources.html, 2008. LA-UR-05-4983.

D. P. Griesheimer et al. An integrated thermal hydraulic feedback method for Monte Carlo reactor calculations. In *Proceedings of PHYSOR-2008 - American Nuclear Society Topical Meeting on Reactor Physics*. Interlaken, 2008.

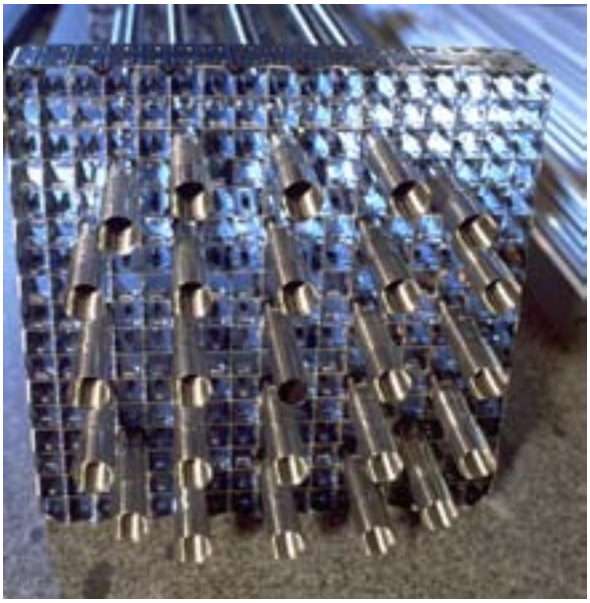
J. E. Hoogenboom. Improved estimation of the variance in Monte Carlo criticality calculations. In *Proceedings of PHYSOR-2008 - American Nuclear Society Topical Meeting on Reactor Physics*. Interlaken, 2008.

P. S. Laplace. *Théorie Analytique des Probabilités*. Imprimeur - Libraire pour les Mathématiques„ Paris, troisième edition, 1820.

X - 5 Monte Carlo Team. *MCNP – A General Monte Carlo N-Particle Transport Code*. Los Alamos National Laboratory, Version 5, LA-UR-03-1987 edition, 2005.

5

The Dynamic Monte Carlo method: Simulation of Realistic Geometry



Picture: E. Joly, Areva

In the previous chapter, the possibility of simulating a dynamic system with the Monte Carlo method has been demonstrated. However, the demonstration only showed the applicability of the dynamic scheme for simple geometry. To show general applicability, the theory is further generalised and the scheme is implemented in a general purpose Monte Carlo code.

There are many general purpose Monte Carlo code available, but two codes are the main candidates to implement the new method. The first possibility is MCNP5 (X - 5 Monte Carlo Team, 2005), which is one of the most commonly used Monte Carlo codes for nuclear reactor physics and deep knowledge and experience with this code is present in our group. The second possibility is Tripoli 4.7 (TRIPOLI-4 Project Team, 2010), which is the reactor physics Monte Carlo code of the NUclear REactor SIMulation platform, NURESIM (Chauliac et al., 2011). This is the Monte Carlo code of choice for the implementation of the dynamic Monte Carlo scheme, because it is written in a modern programming language and because of the support and new developments which are available via the NURISP project. The source of Tripoli version 4.7 was kindly made available by CEA[1], with the support of its development team.

5.1 Implementation of the dynamic method into Tripoli 4.7

To implement the dynamic scheme into Tripoli, the source code is adapted at several points, creating the Dynamic Tripoli code. In this section the focus is on the theoretical background, the general applicable methods and some specific design choices. The details of the implementation and the new keywords introduced in the input of Tripoli for the dynamic calculation can be found in App. B. The new simulation sequence chosen close to the Tripoli standard, in order to apply only the necessary changes and to utilise the intrinsic methods in Tripoli.

First, special attention is paid to the particle tracking. Although this is simply simulating the physics as is already present in the code, there are some issues which have to be implemented more carefully in a kinetic calculation, such as time boundary crossings. Then, the source sampling will be addressed, which extends the source sampling theory of the previous chapter to incorporate energy dependence. The energy-dependent source distribution can be sampled from an eigenvalue calculation. Finally the new variance reduction techniques are discussed which are implemented in the Dynamic Tripoli code. These techniques are specific for the simulation of kinetic and dynamic problems.

[1] Commissariat à l'énergie atomique et aux énergies alternatives

5.1.1 The dynamic simulation scheme for Tripoli

Tripoli has three simulation modes, criticality, fixed-sources criticality and shielding, it divides the total number of particles in a calculation into batches, which are simulated consecutively. The statistical uncertainty is estimated from the variance between the batches. When calculating in parallel, there is no communication between the different simulators. Each simulator samples its own source and starts to simulate batches. Only the results of these batches are returned to the master process, which determines if the simulation is finished.

To implement time stepping into Tripoli there are two options. The first one is transforming batches into time intervals. However, Tripoli estimates the variance of its results by calculating the variance between different batches. This is not possible when transforming the batches into time steps, since each batch will have scores in separate time bins. The variance can also be calculated between individual particle histories. Although this gives also an accurate estimation of the variance (Hoogenboom, 2008), it is not in line with the structure and philosophy of the Tripoli code and would require major modifications.

The second option is to add an extra loop in the code, which runs over all time intervals during each batch, as depicted in Fig. 5.1. This implementation has the advantage that it utilises the native variance estimation of Tripoli, which can now also take into account the variance introduced into the result by reactivity feedback. The downside of this method is that the feedback has to be calculated for every batch separately, which is more expensive and the result might not be physical valid solution in some cases, but an average of different possible states. The implication of adding feedback will be discussed in more detail in the Chap. 6.

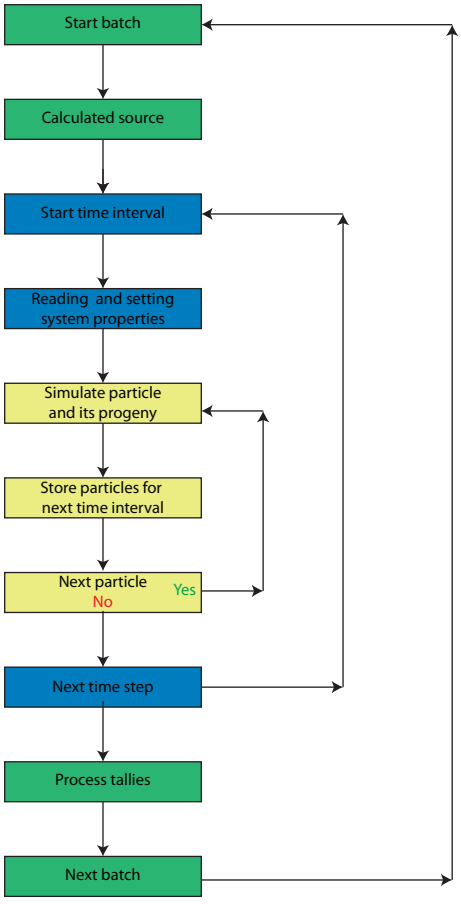


Figure 5.1: The simulation scheme for Dynamic Tripoli.

5.1.2 Time-dependent particle tracking

In a shielding calculation, a particle is followed until it is either killed, leaked or out of bounds in some other way and a particle cannot create new fission neutrons, since these fission neutrons are already present in the source distribution. When doing a criticality calculation, a particle is followed in the same way, but now fission neutrons can be produced and are stored for the next batch. In a fixed-sources criticality calculation the particle can also produce a fission neutron, but now this progeny will be followed in the same batch.

In the dynamic mode, the simulation of particles is very similar to the fixed-sources criticality mode. The only difference is in the time boundary crossing. In the dynamic mode the neutron simulation is stopped at the time boundary and then stored for the next time step.

Time boundary crossing

In a normal Tripoli calculation the crossing of a time boundary does not influence the trajectory of a particle. When using a collision estimator the score of the particle is simply added to the time bin of the collision and for a track-length estimator the part of the track before the boundary is counted in a different bin than the part of the track after this boundary. If the particle crosses the final time boundary, it is stopped after the first collision outside the time domain. An extra time bin must be added in the input to collect the tallies that are outside the time domain.

For a dynamic calculation this method is not allowed. After each time boundary the system can be changed due to feedback or changing geometry and therefore the particles must be stopped exactly at the time boundary. They can be restarted in the next time interval, with the new cross sections, which is allowed, because the path length of a neutron is given by an exponential distribution. This feature is added to the Dynamic Tripoli code.

5.1.3 Sampling of a source distribution for dynamic calculations

When in criticality mode the source of a batch is given by the fission neutrons produced in the previous batch, i.e. the fission neutrons of the previous batch are the initial state of the next batch. Only for the first batch a source distribution is given, but often this is simply a spatially flat source and a number of batches is needed to get the true source distribution. These initial batches are called inactive and the results of these batches are discarded. For a fixed-source criticality calculation or a shielding calculation, the source particles are sampled from a user-defined source distribution.

In the dynamic case it is also possible to have a user-defined source distribution, the same way as for the shielding and fixed-sources criticality modes. However, since the purpose of a dynamic calculation is to calculate the transient behaviour of a nuclear reactor, the initial

conditions are commonly a steady-state situation. These initial conditions can be sampled from a converged source of a criticality calculation, with $k_{eff} \approx 1$.

Neutron source distribution

The energy-dependent neutron flux in steady state, $\phi_0(\boldsymbol{r}, E)$, can be sampled by the Monte Carlo method using a collision estimator. The collision density $\psi_0(\boldsymbol{r}, \boldsymbol{\Omega}, E)$ is related to the neutron flux by the following relation:

$$\phi_0(\boldsymbol{r}, \boldsymbol{\Omega}, E) = \frac{\psi_0(\boldsymbol{r}, \boldsymbol{\Omega}, E)}{\Sigma_t(\boldsymbol{r}, E)} \tag{5.1}$$

Now the number of neutrons can be calculated using:

$$n_0(\boldsymbol{r}, \boldsymbol{\Omega}, E) = \frac{\phi_0(\boldsymbol{r}, \boldsymbol{\Omega}, E)}{v(E)} = \frac{\psi_0(\boldsymbol{r}, \boldsymbol{\Omega}, E)}{v(E)\Sigma_t(\boldsymbol{r}, E)} \tag{5.2}$$

Here $v(E)$ is the neutron speed of a neutron with energy E.

This distribution differs from the fission source, especially in the energy domain, because the energy of a fission neutron is a high energy, while the flux also contains moderated neutrons.

Precursor source distribution

The precursor source distribution is determined by the neutron flux. The precursor balance is given by (Bell and Glasstone, 1979):

$$\frac{\partial C_i(\boldsymbol{r}, t)}{\partial t} = \int_{4\pi} \int_0^{\infty} \beta_i(\boldsymbol{r}, E) \nu(\boldsymbol{r}, E) \Sigma_f(\boldsymbol{r}, E) \phi(\boldsymbol{r}, \boldsymbol{\Omega}, E, t) \mathrm{d}E \mathrm{d}\Omega - \lambda_i C_i(\boldsymbol{r}, t) \tag{5.3}$$

For a steady state, this precursor distribution becomes:

$$C_{i0}(\boldsymbol{r}) = \int_{4\pi} \int_0^{\infty} \frac{\beta_i(\boldsymbol{r}, E)}{\lambda_i} \nu(\boldsymbol{r}, E) \Sigma_f(\boldsymbol{r}, E) \phi_0(\boldsymbol{r}, \boldsymbol{\Omega}, E) \mathrm{d}E \mathrm{d}\Omega \tag{5.4}$$

and now using Eq. (5.1) this can be sampled using:

$$C_{i0}(\boldsymbol{r}) = \int_{4\pi} \int_0^{\infty} \frac{\beta_i(\boldsymbol{r}, E) \nu(\boldsymbol{r}, E) \Sigma_f(\boldsymbol{r}, E)}{\lambda_i \Sigma_t(\boldsymbol{r}, E)} \psi_0(\boldsymbol{r}, \boldsymbol{\Omega}, E) \mathrm{d}E \mathrm{d}\Omega \tag{5.5}$$

For all families combined:

$$\begin{aligned} C_0(\boldsymbol{r}) &= \sum_i \int_{4\pi} \int_0^\infty \frac{\beta_i(\boldsymbol{r},E)\nu(\boldsymbol{r},E)\Sigma_f(\boldsymbol{r},E)}{\lambda_i \Sigma_t(\boldsymbol{r},E)} \psi_0(\boldsymbol{r},\boldsymbol{\Omega},E)\mathrm{d}E\mathrm{d}\Omega \\ &= \int_{4\pi} \int_0^\infty \frac{\beta(\boldsymbol{r},E)\nu(\boldsymbol{r},E)\Sigma_f(\boldsymbol{r},E)}{\lambda^b(\boldsymbol{r},E)\Sigma_t(\boldsymbol{r},E)} \psi_0(\boldsymbol{r},\boldsymbol{\Omega},E)\mathrm{d}E\mathrm{d}\Omega \end{aligned} \tag{5.6}$$

with λ^b the inverse weighted average of the decay constant:

$$\lambda^b(\boldsymbol{r},E) = \frac{\beta(\boldsymbol{r},E)}{\sum_i \frac{\beta_i(\boldsymbol{r},E)}{\lambda_i}} \tag{5.7}$$

and $\beta(\boldsymbol{r},E)$ the total fraction of delayed neutrons for a fission caused by a neutron with energy E. The fraction delayed, fd, is given by Eq. (2.67). The values for β and λ are in this case space and energy dependent.

5.1.4 Dynamic particle weights

The neutron flux and the precursor density can fluctuate considerably over time. This poses a challenge for a Monte Carlo simulation; the calculation time increases linearly with the number of particles. To compensate for this effect, the thresholds of the weight windows are adjusted on the fly.

Calculation of the neutron weight windows

The thresholds of the standard weight windows of Tripoli are given by:

$$\begin{aligned} w_{RR} &= 0.8 w_r \\ w_{surv} &= w_r \\ w_{split} &= 2 w_r \end{aligned} \tag{5.8}$$

Here w_r is the required weight, which is normally set to unity.

During a dynamic calculation, the required weight is automatically adapted, by comparing the current total statistical weight W_{tot} to total weight at the start of the batch W_0:

$$w_r = \frac{W_{tot}}{W_0} \tag{5.9}$$

The total weight is the combination of the weight of the precursors and the weight of the neutrons. Which precursor weight to use for calculating the total weight is discussed hereafter.

Adjustment of the precursor weight

The possibility has been created in Tripoli to adjust the importance of a precursor relative to the neutron importance. The weight monitored is either the expected delayed neutron weight or the timed precursor weight. The setting also allows for an importance factor, I. This can be any positive real number. The precursor weight considered for the population control can be calculated using:

$$w_{imp} = \frac{w_X}{I} \tag{5.10}$$

Here the subscript X can stand for *expected* or *timed*.

5.1.5 Precursor population control

A number of different methods are available for population control. To control the neutron population, the adjustment of the weight windows suffices, since the neutron population will be adjusted after one generation, which is a small number of neutrons compared to the total number of neutrons simulated in a time interval.

The number of precursors have to be controlled more actively, since they are always stored for the next time interval. In Dynamic Tripoli they can be controlled using either Russian roulette or the combing technique at the end of a time interval.

Russian roulette

When Russian roulette is used for population control, a precursor is only stored for the next time interval after undergoing Russian roulette. The survival weight is set to the required weight. The weight used for this roulette is given by Eq. (5.10) and the roulette is played as described in Sec. 2.3.2.

Combing

When combing is used for population control, this is only done on the precursor population. This fixes the number of precursors starting an interval. When combing, w_{imp} is used as precursor weight, similar as with the Russian roulette. For the exact description of the combing method see Sec. 2.3.3.

The neutrons are not combed. If the neutrons would be combed together with the precursors, a large neutron chain could reduce the number of precursors drastically. It is not desirable that a temporary increase in neutrons changes the accuracy of the precursor sampling.

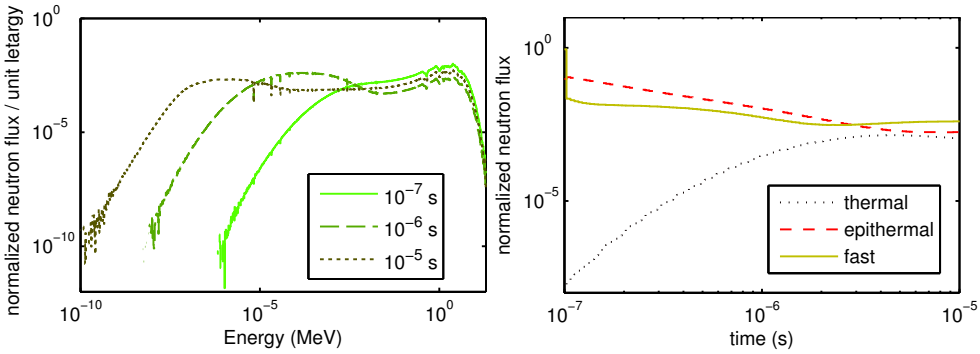


Figure 5.2: The evolution of the neutron flux is plotted in the case where the source distribution is sampled from a fission source distribution. It can be seen that the neutron flux spectrum still evolves at the start of the calculation.

5.2 Demonstrating the Dynamic Monte Carlo Method on Realistic Geometry

To demonstrate the validity of the new method, it has been applied on a few different test cases. The first case is a simple static system, but in the final test a transient due to control-rod movement is shown, with the time-dependent power calculated pin-by-pin in a fuel assembly.

5.2.1 Source confirmation

Prompt neutron source

The first test is checking the source calculated using the theory described in Sec. 5.1.3. To this end the fuel assembly described in Appendix A.2 is used and the steady-state fission source has been calculated from a criticality calculation using 150 cycles. Next, the neutron and precursor distributions are sampled as described by Eqs. (5.2) and (5.6). This method has been compared with a simulation which started directly from the fission source distribution.

In the case which started from a fission source distribution, the neutron flux is not constant from the beginning as shown in Fig. 5.2. It the decrease of fast neutrons over time and the increase of thermal neutrons until they reach an equilibrium, which can be expected from fission neutrons. Since the lifetime of neutrons is fairly short, the energy-dependent neutron flux reaches a steady state within 10 μm.

When the simulation is started from a neutron distribution which is sampled using

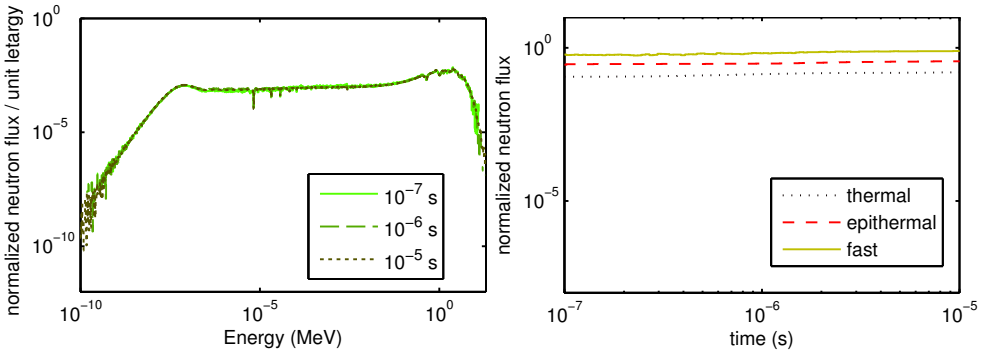


Figure 5.3: The evolution of the neutron flux is plotted in the case where the source distribution is sampled from a steady-state neutron flux. It can be seen that the neutron flux spectrum remains constant.

Eq. (5.2), the flux is directly at its steady state as shown in Fig. 5.3. A small increase in the neutron flux over time can be seen for all three energy groups, this is due to the fact that the system is not exactly critical, but nearly critical; all three energy groups increase equally fast.

Precursors source

A similar argument holds for precursors. It is important to sample the precursor distribution from the steady-state distribution. Both the spatial precursor distribution, as indicated in Eq. (5.6), and distribution between the different precursor families, Eq. (2.67), must be sampled for steady state.

In Fig. 5.4 a calculation is started with precursors which all have a distribution according to the fission *fd*, Eq. (2.17). The time scales are a lot longer here than for the neutrons and it is clearly shown that it is not a steady-state situation. The second line in Fig. 5.4 shows a calculation where the precursor source is correctly sampled and a nice steady system is shown here.

5.2.2 Demonstration in a simple geometry

The next test for Dynamic Tripoli 4.7 is a comparison with the results of the Dynamic Monte Carlo code of the previous chapter. Calculations using the same geometry have been performed and when possible they have been compared to the results of the previous chapter. Also some tests, which were not possible with the Dynamic Monte Carlo code are performed. These tests are compared with other transient analysis tools.

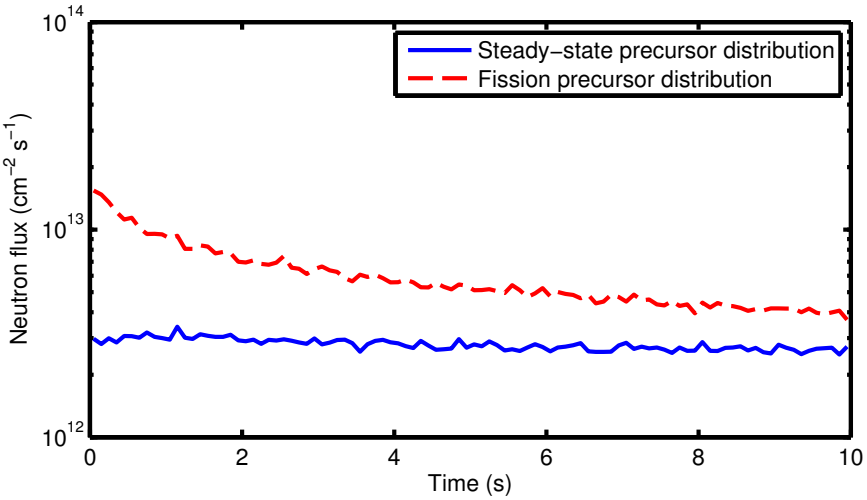


Figure 5.4: The neutron flux when the calculation is started with the steady-state precursor family distribution and with the fission distribution (β_i/β). With steady-state distribution the flux remains constant, but with the fission distribution there will be relatively more short lived precursors present and these short lived precursors will decay more quickly than average, which will not result in a steady state.

Steady state

The first test for the new Dynamic Tripoli code is on the same simple system as used in Chap. 4 which is in a near-critical state, $k_{eff} \approx 1$. The system is homogeneous and mono energetic with $k_{eff} = 1.00012 \pm 0.00005$ as calculated with the unmodified version of Tripoli 4.7. The details of this system can be found in Appendix A.1

Tripoli will calculate the time evolution of this system over 70 s. To be able to calculate such a system, the scheme must be stable. If the system is not stable it will diverge towards the end, with a continuously increasing variance.

The calculated time evolution of the neutron flux is plotted in Fig. 5.5 and as clearly shown the neutron flux in this system is nearly constant from the start. Also the results are stable over time. There is some variance, but this does not influence the stability of the total simulation. The neutron flux slowly increases, as expected for a slightly supercritical system.

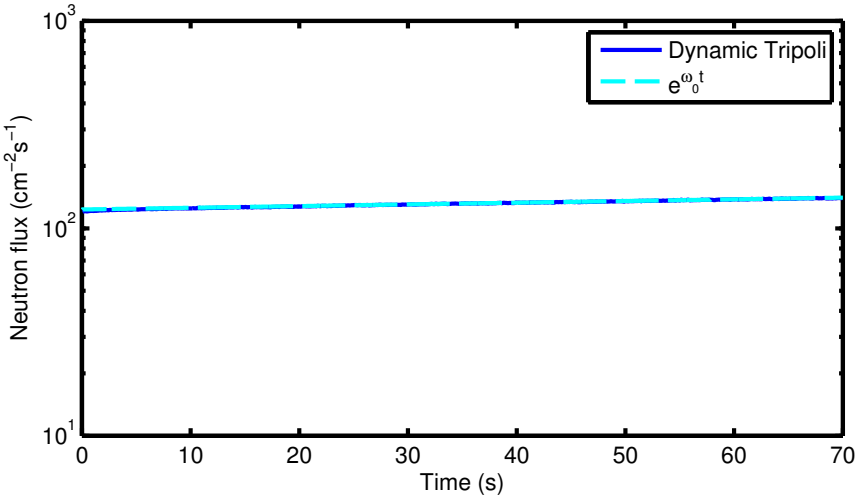


Figure 5.5: The time evolution of the total neutron flux in a stable system. The results are compared to the inverse time constant ω_0 of the system.

Subcritical system

Next, the new method is tested on a subcritical system. First the system is made subcritical, by reducing its size. The source is a steady-state neutron-precursor source at $t = 0$. This way the calculation can be compared with a point-kinetics calculation. The parameters for the point-kinetic calculation are calculated using an unmodified version of Tripoli 4.7.

In the first calculation the time mesh is an equidistant mesh with time intervals of 0.1 s. The results are shown in Fig. 5.6. The result compares nicely with the results of the point-kinetics calculation. This is to be expected for such a homogeneous system.

To look a bit closer to the prompt jump, the same calculation has been done, but with a time mesh with time intervals which grow exponentially in size. The results are shown in Fig. 5.7. Again both calculations give the same answer and it can be seen that the method yields the correct answer, both for small timescales and large timescales. This is a demonstration that the particle tracking, described in Sec. 5.1.2 is working correctly. An error in time boundary crossing would showup especially in the range with small time intervals.

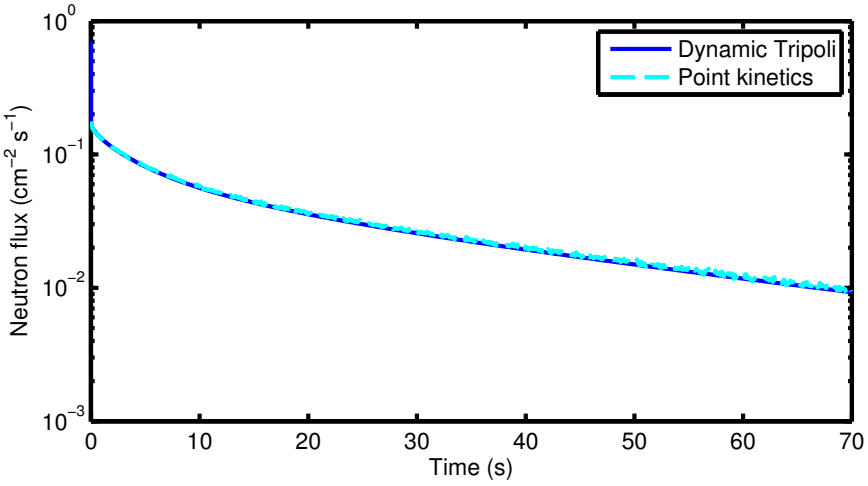


Figure 5.6: The evolution of the neutron flux in a subcritical system, calculated with a point-kinetics calculation and with Dynamic Tripoli, plotted on a linear time scale.

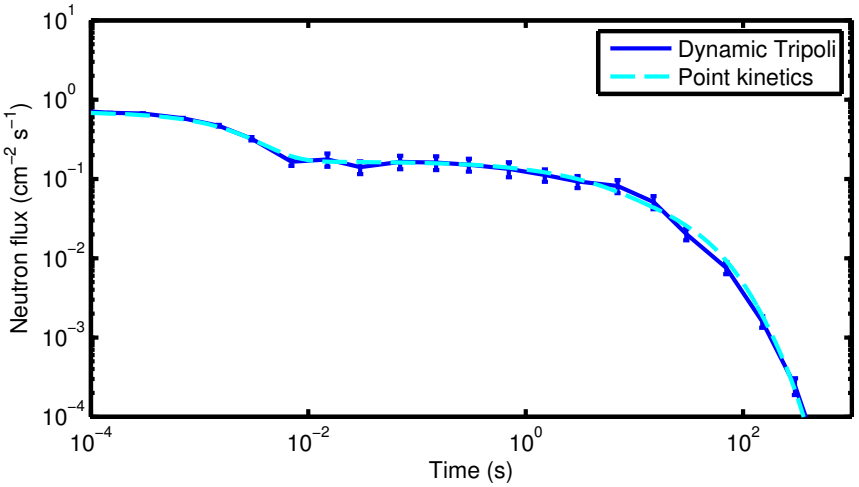


Figure 5.7: The evolution of the neutron flux in a subcritical system, calculated with a point-kinetics calculation and with Dynamic Tripoli, plotted on a logarithmic time scale and calculated with increasing time-bin size.

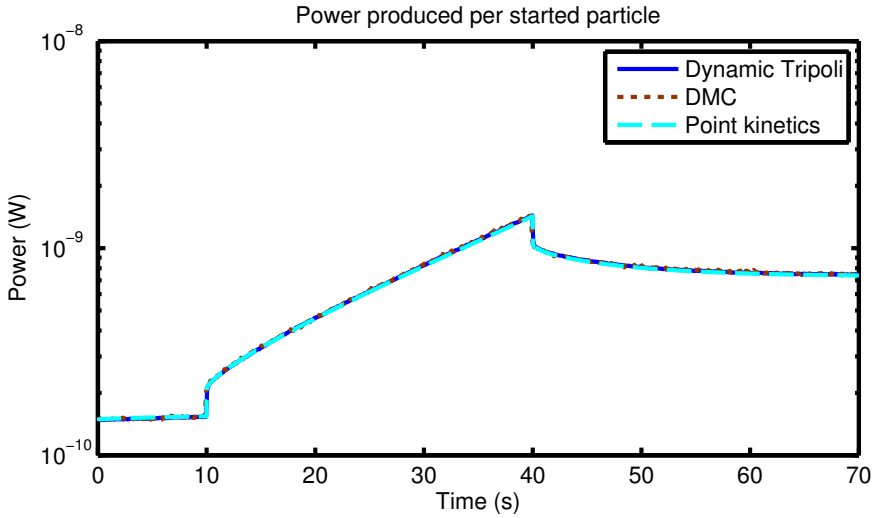


Figure 5.8: A critical system has a reactivity insertion at t=10 s. After a prompt jump the power starts to increase further, until the reactivity returned to 0 at t=40 s, returning the system to a new stable state. The results of all three methods agree well with one another.

5.2.3 Dynamic system calculation

The next test case for Dynamic Tripoli, is the calculation on a changing system. Again the cube of Chap. 4 is used with the properties of Appendix A.1. Reactivity is inserted at $t = 10$ s and the system is returned to its former state at $t = 40$ s. The results are plotted in Fig. 5.8 and all calculation methods agree with each other within statistics. The results were achieved in 288 CPU hours running at 2.3 GHz.

It can be seen from this plot that it is possible for Dynamic Tripoli to also incorporate a dynamic environment. It can be clearly seen that there is first a positive reactivity step and then a negative reactivity step. In this case it is the absorption cross section of the entire system that changes, but it could also be a new temperature for the materials or the change of a material, to simulate the movement of a control rod.

5.2.4 Demonstration of the Continues Energy Model

To demonstrate the dynamic Monte Carlo scheme in an energy-dependent setting, while keeping it simple enough to compare it with the point-kinetics model, the cuboid of the previous sections is made of pure ^{235}U. The density has been adjusted to make a critical system and the reactivity insertion is generated by an increase in density. Although there

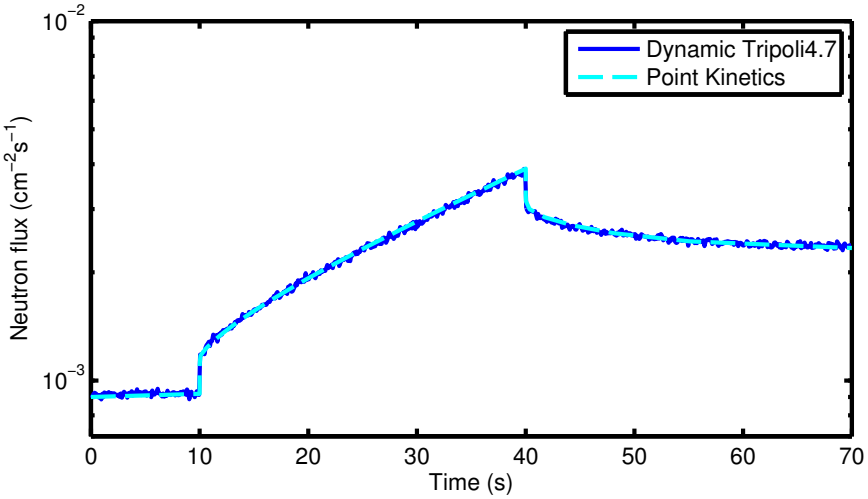


Figure 5.9: The evolution of the neutron flux in a critical cuboid of ^{235}U with a reactivity insertion from 10 to 40 s, calculated with a point-kinetics calculation and with Dynamic Tripoli, with continuous energy treatment.

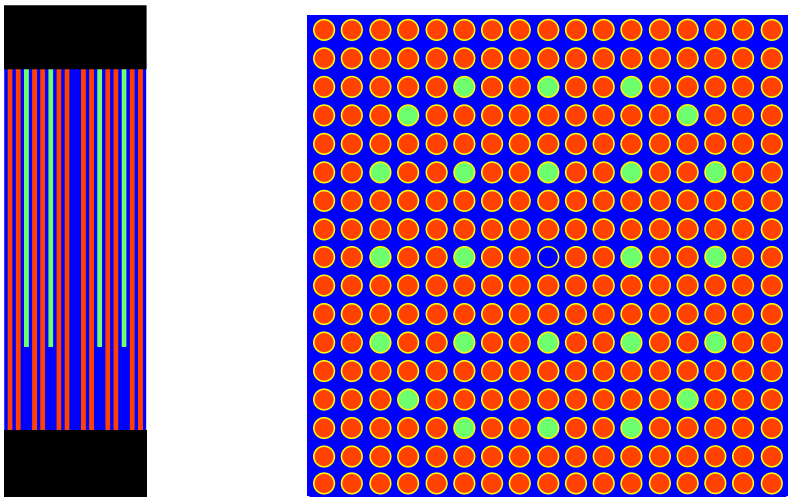
will not be a lot of moderation, the energy dependence of the cross sections will play an important role in this problem. The density has been set to 4.4925 atoms per 10^{-22} cm^3 and increased to 4.500 atoms per 10^{-22} cm^3.

The results of this calculation are shown in Fig. 5.9 and the results of a point-kinetics calculation for the same problem are also given in this plot. It demonstrates the possibility of solving a continuous energy transient problem with the dynamic Monte Carlo method.

5.2.5 Transient in a Fuel Assembly

After these successful tests with simplified geometry, the dynamic Monte Carlo method has been applied to a complex system. With the dynamic method implemented in the source code of Tripoli 4.7, the versatility of this general-purpose code can be applied to simulate more complex systems. In general, one of the features of the Monte Carlo method is that it can easily scale up to more complex systems.

Now the calculations will be performed on a fuel assembly with reflective boundary conditions and it has uranium based fuel. The layout of the fuel assembly is given in Fig. 5.10, which is modelled in full detail. The geometry is based on the transient benchmark of Kozlowski and Downar (2003), but there is no feedback present. The system is made



(a) Vertical cross section of fuel assembly, x- and z-axis are not at the same scale.

(b) Horizontal cross section of fuel assembly

Figure 5.10: Layout of a fuel assembly: the fuel is coloured red, the control rods are in green, the borated water is in blue, the top and bottom reflectors are in black and the cladding is in yellow.

close to critical, by adjusting the control rod height accordingly. The nuclear data used is from the JEFF3.1.1 evaluation (A. Santamarina, D. Bernard, P. Blaise, *et al.* , 2009). The exact geometry is given in App. A.2.

The transient considered here is initiated by the movement of the control rods. From $t = 3$ s to $t = 3.5$ s the control rods are slowly, with a fixed velocity withdrawn 1 cm. This creates a positive transient and after 7.5 s there is a scram, fully inserting all control rods.

The results of this calculation are given in Fig. 5.11, which shows an initially stable system. Then, during the control-rod movement, the neutron flux starts to increase, due to the increase in prompt-neutron chain lengths. The reactivity insertion due to the control rod movement is quite large for a small displacement, which is due to the reflective boundary conditions, creating an infinite lattice with fuel assemblies with control rod movement.

The results are compared to a quasi-static calculation; when the system changes, all kinetic parameters are recalculated with a separate run of the unmodified version of Tripoli.

The variance during the transient is higher than during the steady state. This is caused by the increased reactivity. The value of $k_{eff,prompt}$ is closer to unity generating significantly

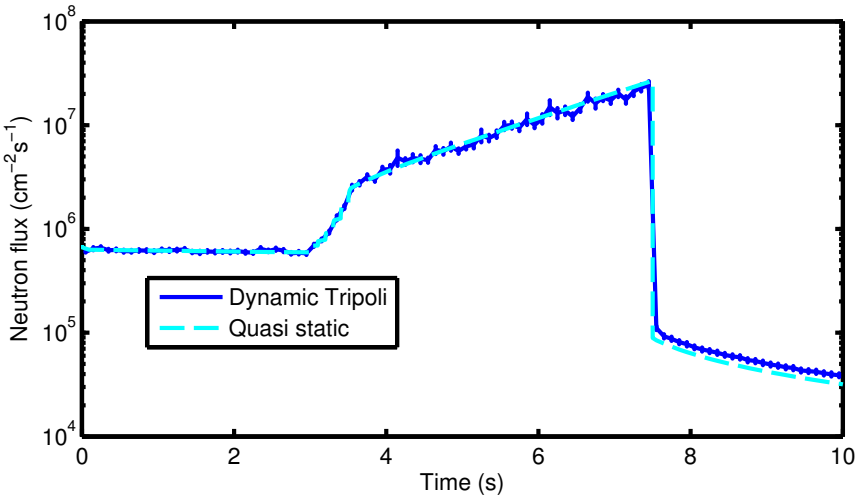


Figure 5.11: The evolution of the neutron flux in a fuel assembly with a reactivity insertion from 3.0 to 3.5 s and a scram after 7.5 s, calculated with the Dynamic Monte Carlo method. During the scram a deviation from quasi-static calculation can be seen.

more variance, even with the branchless method implemented.

After the scram, the power produced by the delayed neutron source can be seen. Although the relative uncertainty between the different time intervals seems to be lower than during the transient, it is in fact of the same order of magnitude. The variance between the time bins is smaller, because the correlation between the time bins is larger in a subcritical system. In the subcritical system the power is mainly dependent on the delayed neutron source, which is similar for all time intervals. This creates a large correlation between the intervals. On the other hand, during the transient the prompt neutrons are more important for the total power. The lifetime of the prompt neutrons is smaller than the duration of a time interval, making the time intervals statistically more independent.

5.2.6 Pin-by-Pin Sampling

Finally the same system is used, but now pin-by-pin power sampling and with a different transient. This time three control rods in the top left corner of the fuel assembly are ejected at $t = 3$ s. This creates an asymmetric flux profile which is very different from the flux profile during normal operation. Then after 7.5 s the system is scrammed, with the remaining control rods. The Monte Carlo simulation of this system used 8000 CPU hours.

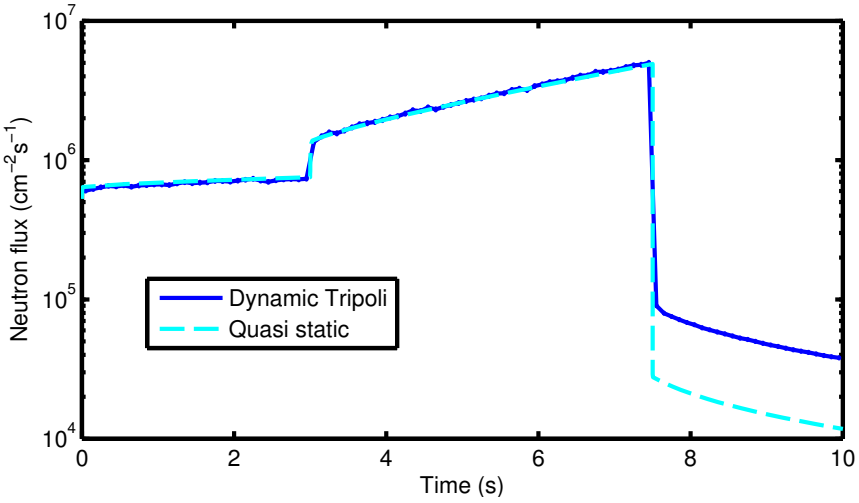


Figure 5.12: The evolution of the neutron flux in a fuel assembly with a transient causing a distorted and asymmetric neutron flux profile, calculated with the Dynamic Monte Carlo method and point kinetics. The deviation from point kinetics is larger than in the previous case.

The total power production is plotted in Fig. 5.12, which shows a quasi-static method which agrees well with Monte Carlo simulation until the final scram. Here the two methods start to deviate, which can be attributed to the major distortion of the flux profile during the transient. This distortion, combined with the insertion of the control rods, causing a negative reactivity of many dollars, makes the quasi-static method not applicable.

The flux profile at the start of the calculation, during the transient, and at the end of the calculation are shown in Fig. 5.13. At the start of the calculation, the flux profile is symmetric, as expected. The flux near the edge is slightly higher than in the middle, because these edges are at a relatively large distance from the control rods. Differences are approximately 1-2 %.

Then the ejection of the three control rods tilts the flux profile, and a higher flux is seen in the top left corner. The differences between centre and the top left corner is now around 5 %.

Finally the remaining control rods are fully inserted. The flux profile is now even more asymmetric: differences are up to 20 %. Also, when the control rods are inserted the variance between the different fuel pins fluctuates more than in the steady-state situation. This might be explained by the fact that the neutrons can travel less with the rods fully inser-

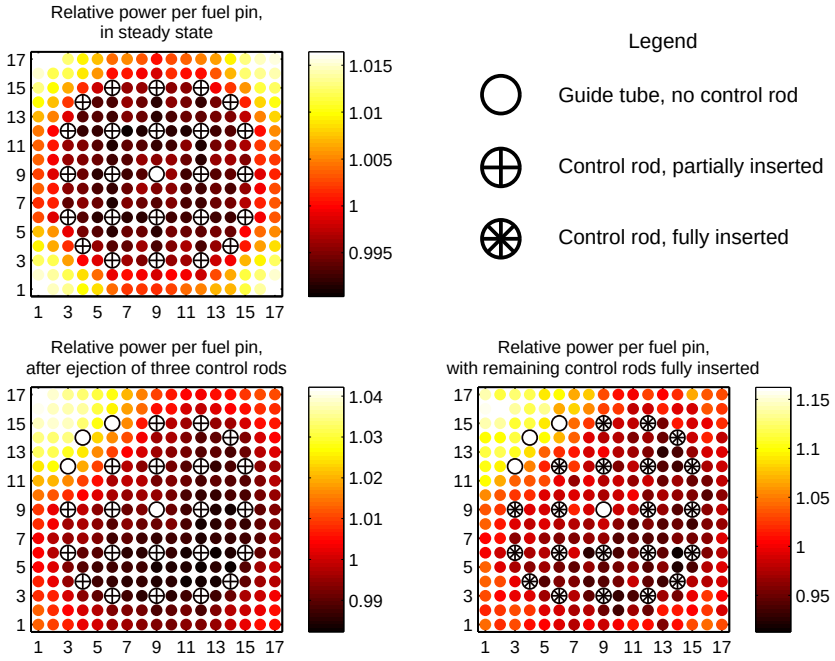


Figure 5.13: The evolution of the neutron flux profile in a fuel assembly with the ejection of three top left control rods. The first figure shows the initial conditions, the second figure the spatial profile during the transient and the final figure shows the profile after the scram with the remaining control rods, as calculated with the Dynamic Monte Carlo method. Note: the relative fluxes in the three figure have a different scale.

ted. This will reduce the transfer of neutrons between the fuel pins giving less correlation between them.

5.2.7 Variance reduction

To inspect the effect of the various variance reduction techniques the calculation on the fuel assembly with the reflective boundary conditions has been performed with different settings. All following tests have been performed on this problem.

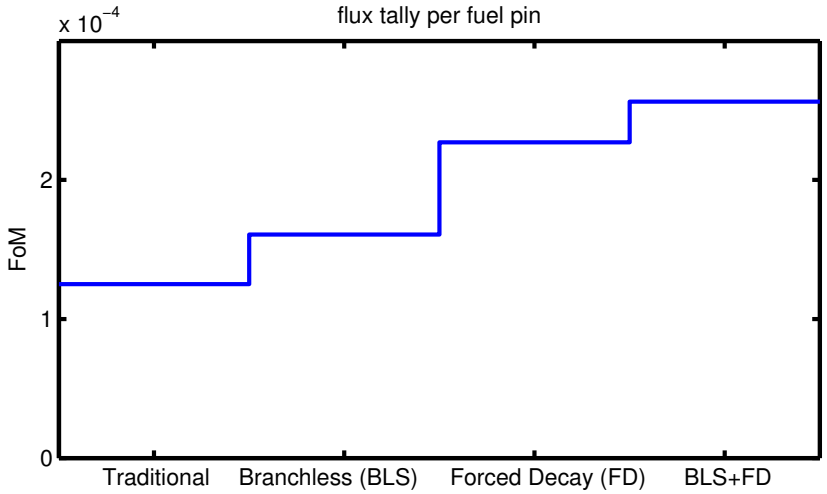


Figure 5.14: The Average Figure of Merit when several variance reduction techniques are applied.

Branchless method and forced precursor decay

The first variance reduction techniques to be investigated are the branchless method and the forced precursor decay. These methods have proven to be applicable in a simple example, but here they are tested in a realistic setting, with continuous energy and in a complex heterogeneous system, which makes it more difficult for a variance reduction technique to improve the results than in the tests in previous chapters. All methods give an unbiased result and the gain in FoM of the two methods separately and combined are given in Fig. 5.14.

Here it can be seen that both the branchless transport and the forced precursor decay are improving the efficiency of the calculation significantly. When the two methods are combined, the efficiency increases even more. The efficiency gain is not as high as with the DMC-code of the previous chapter, which can be attributed to the overhead costs of a general purpose code such as Tripoli and to the heterogeneous system.

Setting the precursor importance

The ratio between the weight of the precursors and the weight of the prompt neutrons has been varied to determine the best value for the statistical weight of the precursors.

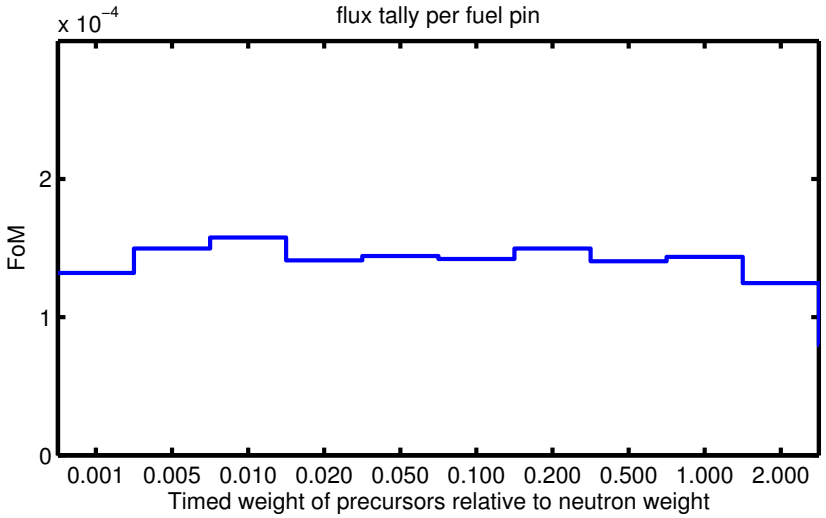


Figure 5.15: The average Figure of Merit For several different weights of the precursors.

The differences in efficiency for the different weights are not very large. The timed weight of 0.01 seems to perform best. For this system, this timed weight is equivalent to a delayed neutron with weight of approximately 1.

Weight windows

To demonstrate the importance of the moving weight windows, some calculations have been performed with the same number of neutron histories, but with a varying batch size. With smaller batch sizes the weight windows are determined less accurately; it can be seen in Fig. 5.16 that properly set weight windows are required to get a good efficiency in the calculation and a minimum number of particles per batch is needed to properly determine these weight windows.

5.3 Summary on the development of Dynamic Tripoli

In this chapter the dynamic Monte Carlo method has been further developed for general applicability. It is now possible to sample an energy-dependent source distribution from a criticality calculation. The scheme has been adapted to fit in the general purpose Monte Carlo code Tripoli. Also issues concerning population control and variance reduction have been addressed.

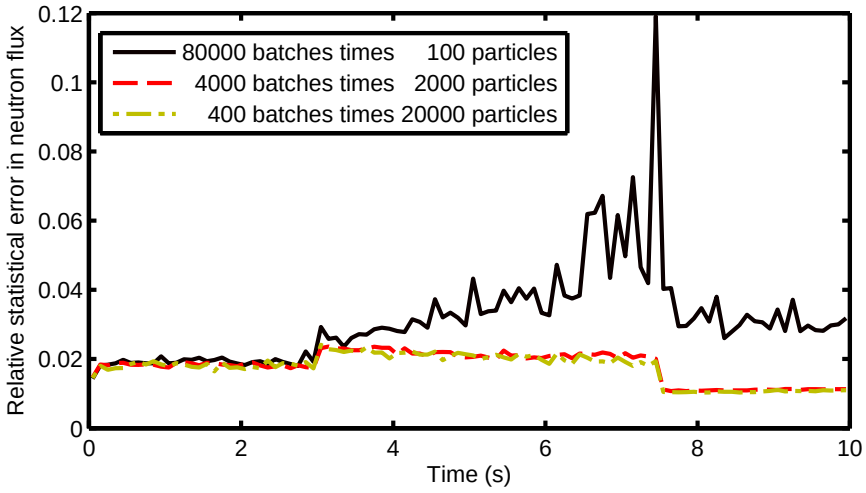


Figure 5.16: The relative error using different batch sizes. Each calculation has the same amount of starting neutron chains. The variance can become higher, when the number of particles per batch is too low. This is due to poor estimation of moving weight windows.

Next, the method has been demonstrated on various cases. First the method has been tested on simple cases to verify the results. Next the method has been tested on more complicated systems: a fuel assembly with a transient, where the results do differ from the results of a quasi-static calculation.

Also the efficiency of the calculation has been tested with the different variance reduction options. The improvement in Figure of Merit is less than in the previous chapter with the DMC program in simplified conditions. This is commonly seen, when implementing variance reduction in a general purpose Monte Carlo code (Christoforou and Hoogenboom, 2011; Christoforou, 2010). The actual simulation time is in such a code a smaller fraction of the total calculation time than in a purpose-built Monte Carlo code.

5.4 Bibliography

A. Santamarina, D. Bernard, P. Blaise, *et al.* . The JEFF-3.1.1 nuclear data library. JEFF Report 22, 2009.

G. Bell and S. Glasstone. *Nuclear reactor theory.* Van Nostrand Reinhold Company, 1979.

C. Chauliac et al. NURESIM - A European simulation platform for nuclear reactor safety:

Multi-scale and multi-physics calculations, sensitivity and uncertainty analysis. *Nuclear Engineering and Design*, **241**, 3416, 2011.

S. Christoforou. *A Zero-variance Based Scheme for Monte Carlo Criticality Simulations*. Ph.D. thesis, Delft University of Technology, 2010.

S. Christoforou and J. E. Hoogenboom. A zero-variance-based scheme for Monte Carlo criticality calculations. *Nuclear Science and Engineering*, **167**, 91, 2011.

J. E. Hoogenboom. Improved estimation of the variance in Monte Carlo criticality calculations. In *Proceedings of PHYSOR-2008 - American Nuclear Society Topical Meeting on Reactor Physics*. Interlaken, 2008.

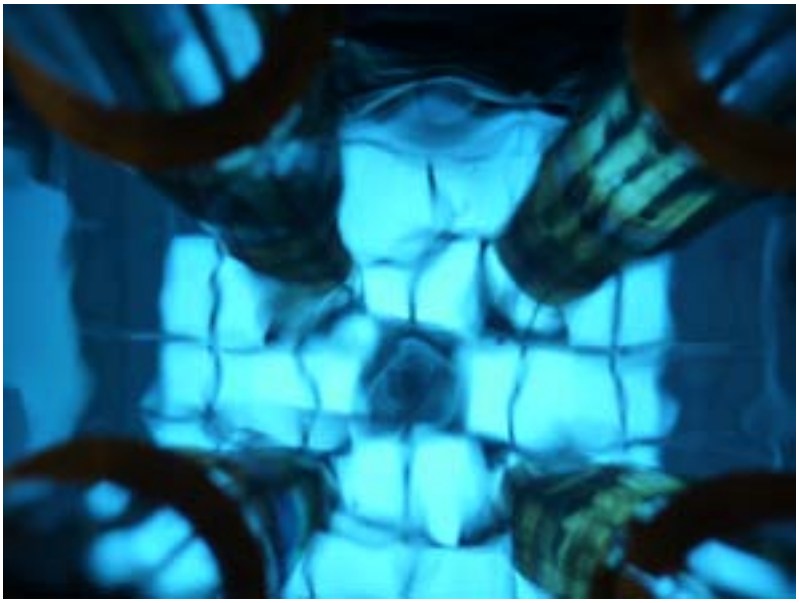
T. Kozlowski and T. J. Downar. OECD/NEA and U.S. NRC PWR MOX/UO2 core transient benchmark, 2003.

TRIPOLI-4 Project Team. *TRIPOLI-4 version 7 user guide*. CEA, serma/ltsd/rt/10-4941/a edition, 2010.

X - 5 Monte Carlo Team. *MCNP – A General Monte Carlo N-Particle Transport Code*. Los Alamos National Laboratory, Version 5, LA-UR-03-1987 edition, 2005.

6

Coupling of Dynamic Monte Carlo with Thermal-Hydraulic Feedback



Picture: RID-TUDelft

6.1 Introduction

One of the most important aspects for transient analysis is the thermal-hydraulic feedback. This feedback mechanism is essential for the safe operation of nuclear reactors. For example, when the temperature in a light water reactor increases, the density of the moderator decreases and neutrons will be less moderated, causing, in general, a reactivity decrease. When the reactivity becomes negative, the power produced will be reduced, which lowers the temperature of the reactor. There are many of these feedback mechanisms, some positive and some negative. When designing a nuclear reactor, it is crucial to take these effects into account, since they determine the time dependent behaviour of a reactor and therefore the maximum temperatures reached during a transient.

To incorporate these feedback mechanisms into the calculation, it is common to couple a thermal-hydraulics code to a neutronics code. A thermal-hydraulics code calculates the density profile of the coolant and the temperature profile in the reactor using heat-transfer models and the new material properties are then used for the neutronics calculation. An elaborate description of the methods currently applied can be found in D'Auria et al. (2004).

6.1.1 Deterministic coupled transient calculations

With the ever increasing computing power, many new developments coupling a deterministic neutronics solver to a thermal-hydraulics solver can be found. The neutronics solver can be a nodal code (Vedovi et al., 2004), or more advanced, use the method of characteristics (Hursin et al., 2011) and these codes can be either internally or externally coupled to the thermal-hydraulics code. The advantage of external coupling is the limited adjustments needed to the codes; they can stay autonomous. With internal coupling on the other hand, the two codes are merged into one code, which is usually faster and more accurate, but this requires more adjustments and the merged code must be validated separately. The external coupling is becoming a standard calculation technique for the analysis of transients in a light water reactor (Peltonen and Kozlowski, 2011), but is also done for less common reactor types such as the high temperature reactor (Boer et al., 2010) or the molten salt reactor (Kópházi et al., 2009).

A downside of coupling a deterministic neutronics solver with a thermal-hydraulics solver, is the limited accuracy of the deterministic method. When a deterministic neutronics calculation is done, there is always a number of approximations applied, such as discretisation in time, space, energy and angle. Also, more fundamental approximations might be needed, such as homogenisation or application of the diffusion theory and it is difficult to estimate the error in such calculations. One way of addressing this problem is by comparing the results with a stand-alone Monte Carlo calculation (Broeders et al., 2003).

6.1.2 Coupled Monte Carlo thermal-hydraulics calculations

A recent development in the field of coupled calculations is the coupling between a Monte Carlo neutronic calculation and thermal-hydraulic analysis. Presently this can be achieved only for steady-state calculations, with a fixed power level. Therefore, only the thermal-hydraulic effects that influence the power profile in a reactor are taken into account and the feedback mechanisms that influence the total power are neglected; only the static effect on the reactivity can be analysed, neglecting the dynamic nature of the feedback.

The first occurrence in the literature of the coupling of a Monte Carlo neutronics code with a thermal-hydraulics code was in 2003, when MCNP4C was coupled with SIMMER-III by Mori et al. (2003), but in this case there is no iterative process. The thermal-hydraulics code SIMMER is run once to calculate the temperature profile for MCNP.

Next, in 2004 Bakanov et al. (2004) coupled TDMCC with STAR-CD, but there is little description of the method. A demonstration has been given of a coupled calculation of a fuel assembly, but there is no discussion on the accuracy of the calculation. Joo et al. (2004) used MCCARD to verify the DeCART transport code, both coupled with a simplified feedback script. Here the results of a coupled analysis of a mini core seem to match relatively well between MCCARD and DeCART.

Next, coupling methods have become part of research tools, with the coupling of stand-alone codes, such as MCNP and STAFAS. Waata et al. (2005a) implemented this method to analyse the HPLWR (Waata et al., 2005b).

Meanwhile, the developments continued, focusing on an increased efficiency of the neutronics calculation, with Tippayakul et al. (2008) who used a nodal code to improve fission source convergence, Sanchez and Al-Hamry (2009) worked on the improvement of the convergence of the coupled solution and the mapping between the two codes is further optimised by Seker et al. (2007a). A different direction of research is to develop partly internal coupling (Leppänen et al., 2012) or complete internal coupling investigated Griesheimer et al. (2008). In these works (some) thermal-hydraulics models are integrated in the Monte Carlo code.

The most recent developments focus on increasing the flexibility of the coupling scheme, making it generally applicable (Hoogenboom et al., 2011; Ivanov et al., 2011) and extension of the coupled calculations to whole-core applications (Kotlyar et al., 2011; Vazquez et al., 2012). The hybrid deterministic/stochastic method which increases the efficiency of the Monte Carlo calculation is further developed by Espel et al. (2013).

6.1.3 Coupled Dynamic Monte Carlo

In this chapter a novel method is created to perform transient calculations including feedback, using only a Monte Carlo approach for the neutronics part of the calculation. The

Monte Carlo code is coupled to a sub-channel code, which is a fast thermal-hydraulics code, but not a high-fidelity code. Although it might seem more logical to couple a computational-expensive high-fidelity neutronics code to a high-fidelity thermal-hydraulics code, the efficiency of the sub-channel code is more useful during the development of a new method. The theory can be extended easily to any kind of external code, including a high-fidelity computational fluid dynamics (CFD) code (Seker et al., 2007b).

6.2 Theory

The coupling of a Monte Carlo neutronics code to a thermal-hydraulics code has a number of key challenges. In this section, these challenges are described and possible solutions are discussed. The first challenge is the coupling scheme itself. First, the initial conditions must be calculated and a convergence criterion must be defined for the initial conditions. Then, for the transient part, there are a few possible coupling schemes to choose from. These schemes must be adapted to enable the use of Monte Carlo results. This is discussed in Sec. 6.2.1.

The next challenge is a typical Monte Carlo one: variance estimation. When performing a coupled calculation it should be acknowledged that part of the calculation is not stochastic. The implications are discussed in Sec. 6.2.3. Finally, the use of temperature dependent cross sections is discussed in Sec. 6.2.4.

6.2.1 Coupling scheme

In nuclear reactor modelling, it is common to use an operator-splitting technique to solve a multi-physics problem. Although this approach does not take into account the non-linearities which are present in a typical coupled reactor physics problem (Ragusa and Mahadevan, 2009), it is a logical start for the novel coupling of a Monte Carlo method for the neutron transport with a deterministic method for the thermal-hydraulics analysis. When the possibility of coupling Monte Carlo with thermal-hydraulic feedback is demonstrated, a more advanced coupling schemes can be investigated.

There are in general three ways of coupling the thermal-hydraulics calculation and the neutronics calculation, implicit, semi-implicit and explicit. With the implicit scheme, the coupled codes are iterated per time interval until the combined codes have converged. Then the scheme will continue to the next time interval, as depicted in Fig. 6.1. This is the most accurate scheme, but for the implementation major alterations have to be made to the existing solvers to allow for the iteration steps. Also, there is a lot of data exchange between the coupled codes and therefore this scheme is usually implemented using internal coupling. A first attempt was made by Mahadevan et al. (2011) to use a Jacobian-free Newton-Krylov method to perform implicitly coupled calculations with the code system KARMA. On the other hand Watson and Ivanov (2012) incorporate implicit

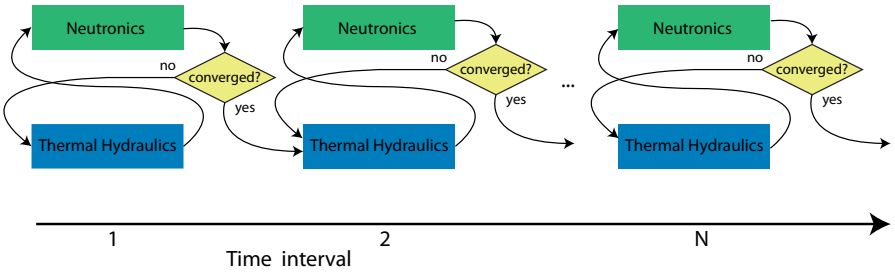


Figure 6.1: Scheme for an implicit coupling of the neutronics solver and the thermal-hydraulics solver. In this case, each time interval has to converge before continuing to the next interval.

coupling in the TRACE/PARCS code system by explicitly forming the full Jacobian matrix and solving for a global residual.

For the semi-implicit method, some of the data of the previous time interval is used for the calculation of the current time interval and some data of the current time interval. For example, when the TRAC-PF1/NEM code calculates the fluxes and the power production, it uses coolant temperatures and densities of the current time interval, but fuel rod temperatures of the previous time interval (Ivanov and Avramova, 2007).

In most cases the explicit coupling scheme is used, since this is the easiest to implement and validated codes can be used. Here the temperature and density profiles are calculated with the thermal-hydraulics code and with these profiles the power distribution is calculated. This power distribution is then used in the thermal-hydraulics code. This scheme is depicted is Fig. 6.2 and this is also the scheme used in this work, because of its calculation speed and the possibility to couple externally.

Coupling coefficients

In a normal coupled calculation using two deterministic codes, the explicit coupling scheme can be depicted as follows:

$$T^{n+1} = T^n + \Delta t \frac{\mathrm{d}T}{\mathrm{d}t}(P^n) \tag{6.1}$$

$$P^{n+1} = P^n + \Delta t \frac{\mathrm{d}P}{\mathrm{d}t}(T^{n+1}) \tag{6.2}$$

Here P is the power profile resulting from the neutronics calculation and T is the temperature and density profile resulting from the thermal-hydraulics calculation. The superscript

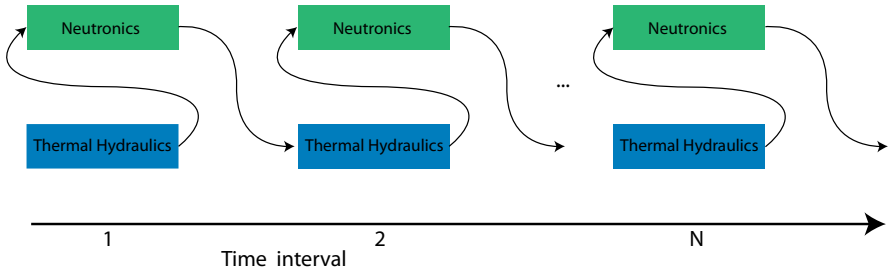


Figure 6.2: Scheme for an explicit coupling of the neutronics solver and the thermal-hydraulics solver. In this case, first the thermal-hydraulic conditions are calculated and then the power distribution.

n indicates that the values are to be taken at the start of time interval n.

When doing a Monte Carlo calculation, it is important to realise that the code will not calculate the power in a single point in time. In fact, the output of a Monte Carlo calculation is the total energy E released during the time interval:

$$E^n = \int_t^{t+\Delta t} P(T^n, t)\mathrm{d}t \tag{6.3}$$

From the total energy released during the time interval, an average power can be calculated over the time interval. This average can then be extrapolated to a power level at the boundary of the next time interval. When assuming a linear increase this becomes:

$$P^{n+1} = \frac{2E^n}{\Delta t} - P^n \tag{6.4}$$

This power distribution can be used for the thermal-hydraulics calculation.

6.2.2 Practical implementation of the coupling scheme

A coupled Monte Carlo calculation consists of two parts: the calculation of the initial conditions and the actual transient. To calculate the initial conditions, a coupled steady-state calculation is used, which is based on the method developed by Hoogenboom et al. (2011). This method has been extended to calculate the critical conditions, which is in the case of a pressurised water reactor the critical boron concentration, but for other types of reactors this could be adjusted to calculate for example the critical control rod position. Also the fission source convergence has been improved.

When using this method, the steady-state total-power level is set externally and then the temperature and density profiles are calculated with the thermal-hydraulics code. With these profiles, a Monte Carlo calculation is started, calculating a new power profile and k_{eff}. After this calculation, three modification are made to the input of the Monte Carlo calculation. First, the new temperature and density profiles are calculated using the thermal-hydraulics code and the Monte Carlo input is updated with these new temperatures and densities.

Secondly, the new boron concentration, C_{boron}, is calculated by a linear extrapolation of the concentration from the two previous iterations. As explained before, for pressurised water reactors a critical boron concentration is calculated, but the method can also be used to calculate a different critical condition, e.g. a critical control rod height. For iteration number n this is done using:

$$C_{n,boron} = C_{n-1,boron} - \frac{k_{eff\,n-1} - 1}{k_{eff\,n-1} - k_{eff\,n-2}} \left(C_{n-1,boron} - C_{n-2,boron}\right) \tag{6.5}$$

For the first two cycles an initial guess is needed.

Thirdly, the accuracy is increased if the following condition is met:

$$\left|k_{eff} - 1\right| < 2\sigma_{k_{eff}} \tag{6.6}$$

with $\sigma_{k_{eff}}$ the standard deviation of k_{eff}. In this case the number of active batches is increased by a factor of two. This allows for the first iteration steps to use less calculation time and only increase accuracy when needed. There are more advanced schemes under development to optimise this iteration process (Dufek and Gudowski, 2006), improving not only the way the accuracy is increased, but also the way the temperatures and densities are sampled. This novel method might be extended to improve the estimation of the correct boron concentration. However, for this demonstration, a simple scheme suffices.

Also, the fission sources of the last cycle are stored to use for the next iteration, which reduces the number of inactive cycles needed to reach a converged source. Since the difference in fission source profile between two iterations is not very large, the final fission source of the previous iteration is a good initial guess for the present iteration.

In the second part of the calculation, the actual transient is simulated, using the temperature and density profiles calculated in the first part as initial conditions. First, a dynamic neutron and precursor source is sampled using a classical steady-state calculation as explained in section 5.1.3. Then, the first time interval is calculated using the dynamic Monte Carlo method. The calculated power profile is then transferred to the thermal-hydraulics code, which calculates the temperature and density profiles for the new time interval. These are returned to the Monte Carlo code to continue with the next time interval. Both the Monte Carlo calculation and the thermal-hydraulics calculation perform a single transient calculation per batch, waiting at the end of a time interval to exchange information with the other code.

6.2.3 Variance estimation

The mean result in a Monte Carlo calculation is a stochastic variable, which has a normal distribution according to the central limit theorem and usually the variance is estimated between neutrons. However, in a coupled calculation neutrons are no longer uncorrelated, making the central limit theorem no longer applicable.

This can be overcome by performing the thermal-hydraulics calculation for each batch separately. In this case the variance can be estimated between the different, statistically independent, batches. Also, the statistical uncertainty introduced by the coupling can be estimated, but it does not take into account any bias in the thermal-hydraulics code. A downside is that the feedback is calculated on smaller sample sizes and that might prove unreliable.

Also, if the thermal-hydraulic feedback has a strong non-linear behaviour, such as a step-like response function, this cannot be taken into account. Only the average response of all possible outcomes is tallied. An example where such effects are prominent is the simulation of instabilities in a supercritical water reactor (T'Joen and Rohde, 2012). Here this coupling strategy could yield an non-physical result: an average of the power oscillations. In such a case, it could be more useful to estimate to maximum power per batch or to study the distribution of batch results.

For coupled burnup calculations, Dumonteil and Diop (2011) have demonstrated that the result of a coupled calculation might be biased, when the estimator is not properly corrected for the non-linearity of the coupled equations. For the Bateman equations it is possible to find an analytical solution for the correction factor, but when coupling Monte Carlo to a thermal-hydraulics code, this will prove difficult. This is not only a problem for transient analysis, but also when doing steady-state calculations. Since the calculations in this chapter do not include regimes with two-phase flow, it is assumed that the mean and variance are unbiased.

6.2.4 Generation of temperature dependent cross sections

One of the major challenges for a coupled Monte Carlo calculation is the temperature dependence of cross sections (Brown et al., 2008). Although it is theoretical possible to generate cross sections at any temperature, it is not feasible to do this for a full-core coupled calculation. In such a calculation there are many nuclides and each nuclide is present at a large range of temperatures; each zone has a different temperature. Therefore, it is common to evaluate the cross sections at 50 K intervals and use for each nuclide two cross-section sets at these preselected temperatures, mixing two versions of the same nuclide to get the correct temperature (Espel et al., 2013; Hoogenboom et al., 2011). This is called stochastic mixing, but it is also known as stochastic interpolation, pseudo materials or temperature interpolation.

Currently there are more advanced methods being developed, such as on-the-fly-Doppler broadening which is implemented in MCNP (Yesilyurt et al., 2012) and MONK (Armishaw et al., 2011) and the explicit treatment of thermal motion in Serpent (Viitanen and Leppänen, 2012) and Tripoli (Zoia et al., 2013). Also direct interpolation of the cross sections is possible, yielding promising results (Sjenitzer, 2009). Although these methods are potentially more accurate, they are still experimental and not widely available. Also, they take more calculation time than the mixing method. Therefore, stochastic mixing is used in this work.

Since the cross-section dependence on temperature is predominant with $\sqrt{T}$ the first mixing schemes used a $\sqrt{T}$ interpolation (Van Der Marck et al., 2005). Later, Donnelly (2011) showed that the linear interpolation scheme yields similar results, if not slightly more accurate. Therefore the linear-interpolation scheme will be used in the current work.

6.3 Practical implementation and demonstration

The development of the coupling between the Monte Carlo neutronics calculation and the thermal-hydraulics calculation has been done in three steps. First, a simplified model is created to study the feasibility of coupling Monte Carlo with macroscopic feedback mechanisms and to investigate the impact of the variance on the stability of the calculation.

Next, a more realistic case is simulated, coupling the power profile in a single fuel pin with a realistic thermal-hydraulics model. This calculation is performed on a seconds to minutes time scale, investigating the feasibility of doing a realistic coupled calculation and studying the effect of variance on the thermal-hydraulics code. Finally, an analysis on a mini-core benchmark has been performed to demonstrate the method in a realistic geometry and to verify the results achieved.

6.3.1 Second-order feedback model

Problem setup

For the first investigation of the coupling, the simplified system described in App. A.1 is used. This implies mono-energetic neutrons, a homogeneous medium and a rectangular geometry. The feedback is incorporated by making the total absorption cross section a function of the power to mimic Doppler broadening, as suggested by Legrady and Hoogenboom (2008):

$$\Delta\Sigma_a = \frac{(P_0 - P_{50ms})^2}{4P_0^2}\Delta\Sigma_{a,var} \tag{6.7}$$

Here P_0 is a reference power level, from which the power deviation is calculated and $\Delta\Sigma_{a,var}$ is a scaling variable change in absorption cross section. P_{50ms} is the average power

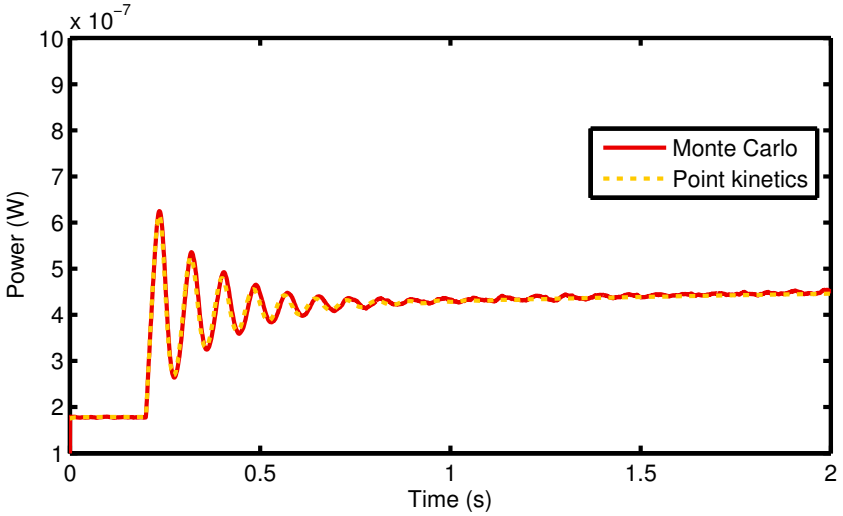


Figure 6.3: The power production in a simple system with feedback on a short time scale. After 0.2 s a reactivity insertion is introduced. The oscillating behaviour on a short time scale is correctly predicted by the Monte Carlo method and the point-kinetics method.

produced in the last 50 ms. It is averaged over the last 50 ms, to incorporate the time required for the the deposited energy to disperse via thermal conduction.

To initiate the transient the total absorption cross section is reduced from 0.5882 cm^{-1} to 0.5840 cm^{-1} at $t = 0.2$ s. $\Delta\Sigma_{a,var}$ is set to 0.0042 cm^{-1}, which is equal to the reactivity insertion and the time interval size is set to 1 ms. The reactivity insertion is equal 1 \$.

To validate these results the calculation has also been performed with a point-kinetics calculation, with the following feedback incorporated, assuming a linear relation between the reactivity and the total absorption cross section:

$$\Delta\rho = \frac{(P_0 - P_{50ms})^2}{4P_0^2}\Delta\rho_{var} \tag{6.8}$$

with $\Delta\rho_{var} = -1$ \$.

First, the transient is simulated and the results are compared with a point-kinetics calculation. Then, the influence of a stochastic feedback on the final result is tested.

Results

The power evolution during the transient is plotted in Fig. 6.3. The reactivity insertion can be clearly seen and after a short time the negative feedback effect can be observed.

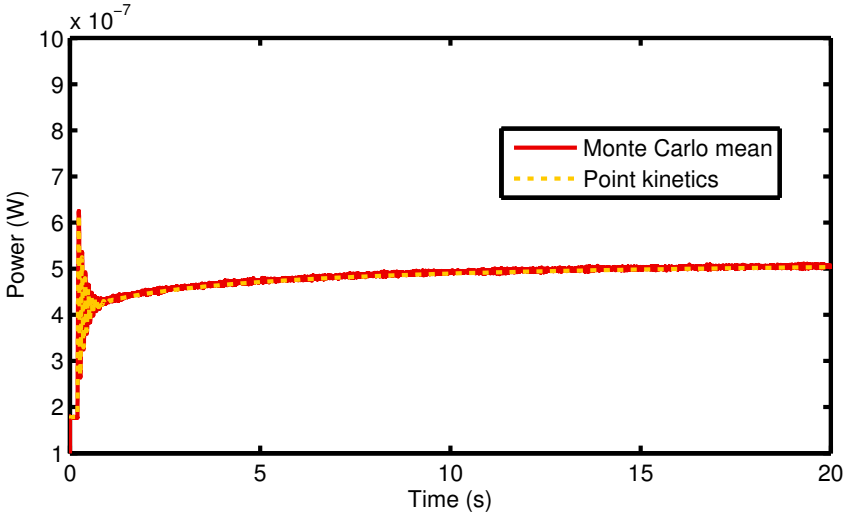


Figure 6.4: The power production in a simple system with feedback on a large time scale. After 0.2 s a reactivity insertion is introduced. On a large time scale the Monte Carlo method and the point-kinetics method converge to the same power level.

The results of the point-kinetics calculation agree nicely with the Monte Carlo calculation. When looking at the long term behaviour in Fig. 6.4 it can be seen that the power level converges to a new steady state level at $P_{stst} = 3.0P_0$, as it should.

There is a small discrepancy in oscillation period between the Monte Carlo calculation and the point-kinetics calculation, which uses a very small time interval, updating the feedback coefficient frequently. This discrepancy is due to the coupling scheme used, which is an explicit scheme and in such a scheme there is always an error due to the time interval size. This is not a Monte Carlo specific problem, but an issue for all coupled calculations.

For the final test, the importance of an accurate power tally is investigated. The variance in the power creates a variance in the feedback, which influences the power. To investigate the necessity of an accurate power tally, the feedback is calculated with a varying number of neutron histories. The calculation is performed with 10^7 particles, which is divided into 10^4 batches times 10^3 particles, 10^3 batches times 10^4 particles, 10^2 batches times 10^5 particles and a reference solution with a single batch using all 10^7 particles.

It can be seen from Fig. 6.5 that most batch sizes yield the correct result. However, in the case with only 10^3 particles per batch, the results become biased. It should be noted though, that in this case the average standard deviation in the power tally is 20 %, which is very high. For the case with 10^4 particles per batch the uncertainty drops to 6 %, for 10^5

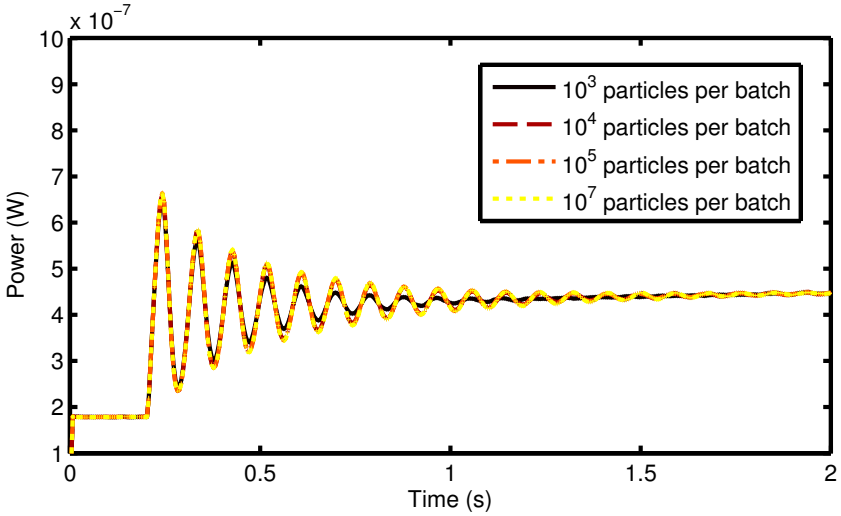


Figure 6.5: The power production in a simple system with the feedback coefficients calculated per batch. The results are mostly independent of the batch size. However, when the uncertainty in the power is larger than 20%, as in the case with 10^3 particles per batch, the results become biased.

starting particles this standard deviation is 2 % and in the reference case with all 10^7 in one batch the standard deviation is 0.2 %. The total variance for the total calculation is the same for all simulations.

It demonstrates that even with relatively high variance, the results are still correct, but caution must be taken. In this example, a statistical uncertainty smaller than 20% is needed to ensure an unbiased power tally. Although this is only a simple test, with a second-order feedback mechanism, the example does give an indication that such a coupled scheme is valid and robust, when the feedback does not have any discontinuities.

6.3.2 Pin cell calculation

Problem setup

In the next step in developing a coupled code system, the dynamic version of Tripoli, developed in Chap. 5 is used together with the thermal-hydraulics code SubChanFlow (Imke and Sanchez, 2012). In this example, a coupled calculation is performed on a single fuel pin, which is surrounded by water. The enrichment of the fuel varies axially with higher enriched fuel at the top. The boundary conditions in the x-y plane are periodic and there is

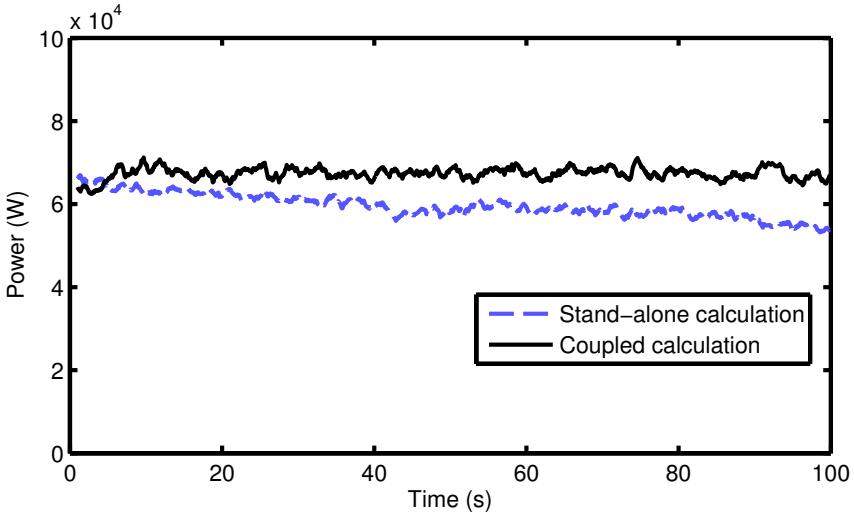


Figure 6.6: A zero-transient calculation on an infinite array of fuel pins. The stand-alone calculation slowly decreases in power, but the feedback mechanisms in the coupled calculation keep the power at steady state.

a vacuum boundary in the z-direction. The complete geometry description can be found in App. A.3. The pin cell has been divided into 20 axial regions to capture the axial power profile and the effective temperature of the fuel rod, as explained by Rowlands (1962), can be approximated from the surface temperature of the rod, T_s and the centre temperature, T_c:

$$T_{eff} = \frac{1}{3} T_s + \frac{2}{3} T_c \tag{6.9}$$

First, a steady-state situation is calculated using the method described in section 6.2.2, yielding the correct initial temperature and density profiles, with a critical boron concentration. Next two transients are simulated. The first transient is a so called zero-transient, where a time-dependent calculation is performed on a critical system, without changing the initial conditions. The second transient which is investigated is initiated by a pump trip, which does create fluctuations in the power level.

Results

After calculating the initial conditions, the first time-dependent calculation is performed, which is a steady-state system. This is also called a zero transient. The result is shown in Fig. 6.6, where the coupled calculation is compared with a stand-alone calculation. It can be seen that the initial conditions are not precisely steady state, since the power in the

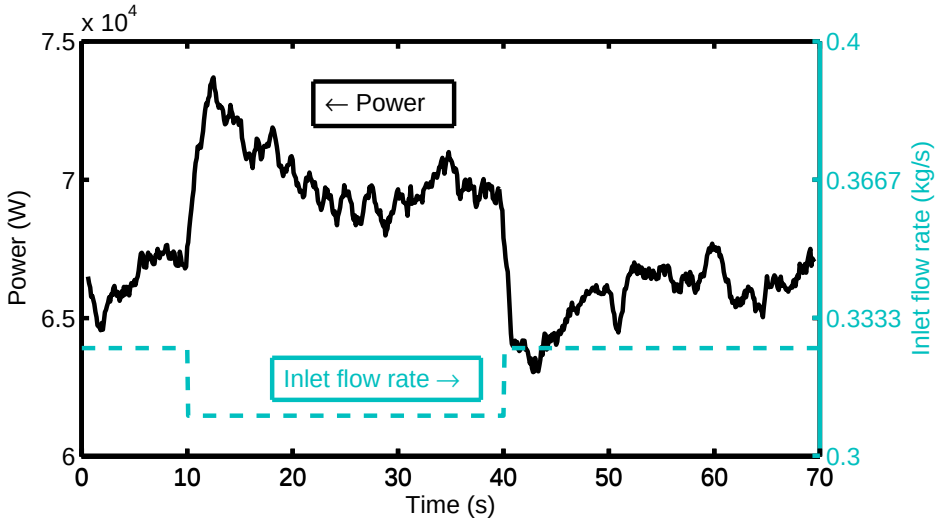


Figure 6.7: The evolution of the total power in the fuel pin with a pump trip from 10 s until 40 s. The power excursion due to the reduced pump flow can be observed clearly.

stand-alone calculation starts to decrease slowly over time. In the coupled case however, the feedback mechanisms in the system keep the system at a steady state. The differences at the start of the calculations are within the uncertainty observed in the calculation. This simulation demonstrates the stability of a realistic coupled calculation.

Next a pump trip is initiated. In this example, the coolant flow will decrease at $t = 10$ s. In Fig. 6.7 it can be seen that this initiates an increase in power production. The power production peaks at $t = 12.5$ s and then decreases to a new steady state level. Then at $t = 40$ s the pump is returned to its initial state, with the power level also returning to its original level after a few oscillations.

The axial power profile is tallied in 20 zones and the profiles at different stages during the transient are plotted in Fig. 6.8. Here it can be seen that the main contribution to the total power increase is given by the highly enriched regions, as expected. The flux profile at the start and end of the transient are nearly identical.

It can be seen from Fig. 6.9 that the transient is indeed driven by the pumping power. Due to the reduced coolant flow, the coolant heats up first, thereby reducing the coolant density leading to an increase of the fuel temperature. During the phases with constant coolant flow, oscillations can be observed where neither the coolant nor the fuel temperature seems to be leading. When the pump flow is returned to nominal, again the coolant properties change first.

The size of the power jump can be justified when the feedback coefficients are investig-

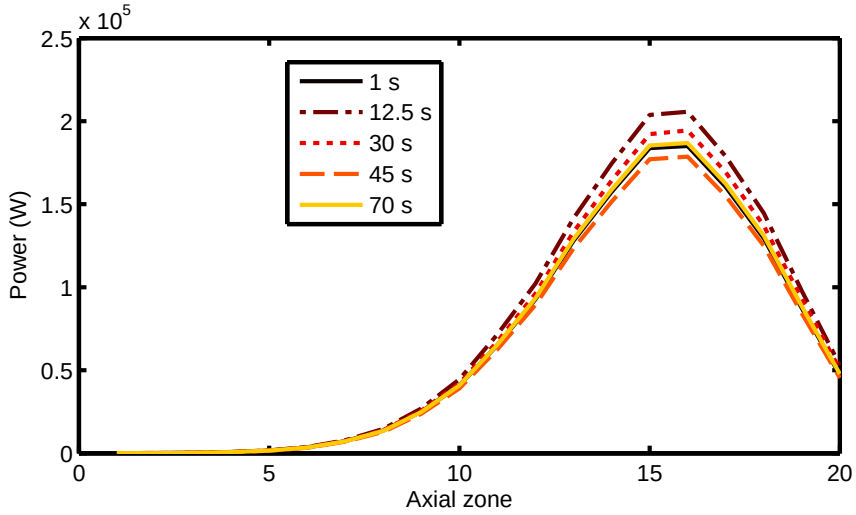


Figure 6.8: The axial power profile in the fuel pin at different times during the transient. At 1 s the system is still at steady state, at 12.5 s is the power peak, at 30 s is the new steady state, at 45 s is just after the negative reactivity insertion and at 70 s is the final state, which is similar to the initial state.

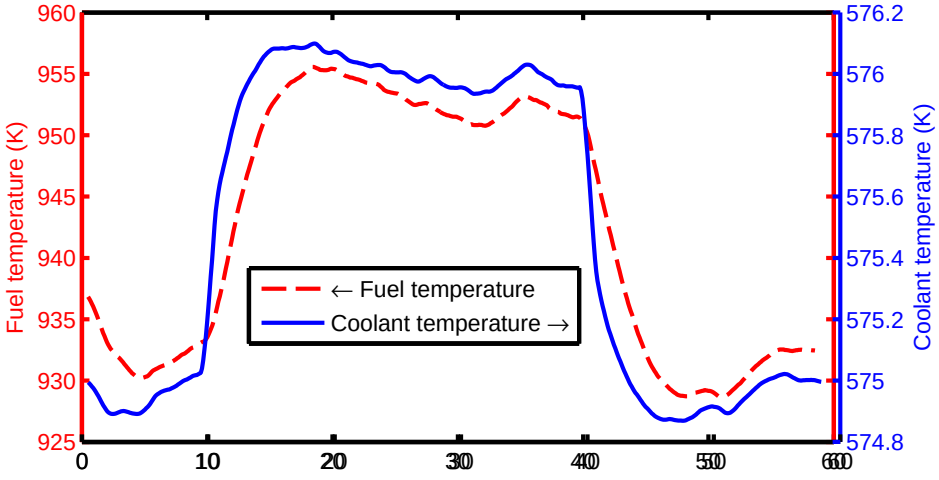


Figure 6.9: The axial temperature profiles of the fuel and coolant are plotted. It can be seen that the transient is initiated by the change in coolant temperature and therefore the coolant density.

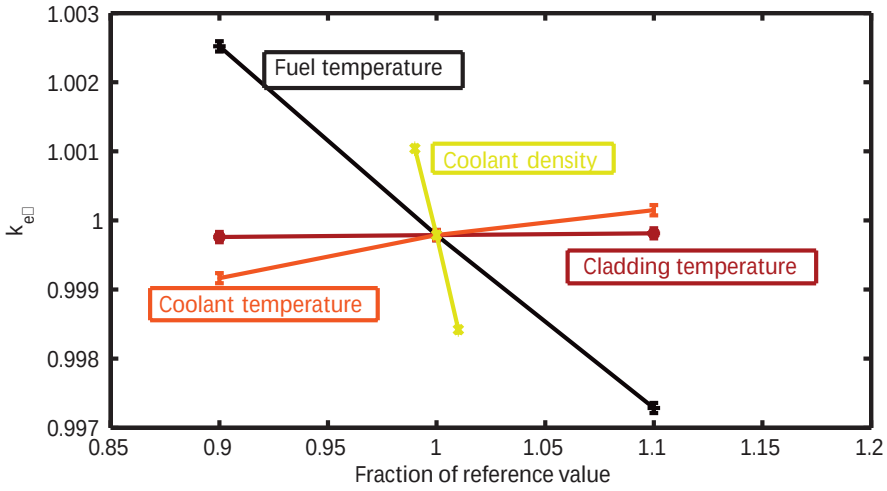


Figure 6.10: Feedback coefficients for a pin cell calculated using steady-state calculations.

ated. In Fig. 6.10 the feedback coefficients of the material properties are given and it can be seen that after the reduction of pump flow they cancel each other. First, the reduced density and increased temperature of the coolant increase the reactivity and then the Doppler effect in the fuel temperature compensate this increased reactivity.

When the feedback coefficients are calculated from Fig. 6.10 using linear regression, reactivity contributions of the different material properties can be visualised, Fig. 6.11. Although this is only an approximate calculation, using average temperatures, it can be seen how the increase in fuel temperature compensates for the reduced coolant density.

Although the results are still somewhat noisy, these examples do demonstrate the feasibility of performing a coupled calculation with Monte Carlo neutronics calculation and a thermal-hydraulics calculation. It is possible to do a calculation on a seconds scale and the effects of the feedback mechanisms can be clearly observed. However, it is still unknown if the results are valid; to verify the novel method, a comparison is needed.

6.3.3 NURISP benchmark

Problem setup

For the final test, a benchmark problem defined in the NURISP rapport D3.1.2.2 (Kliem et al., 2011) is analysed, which consists of a steady-state calculation and a transient in a mini-core. As part of the transient calculation, a second steady-state calculation is needed

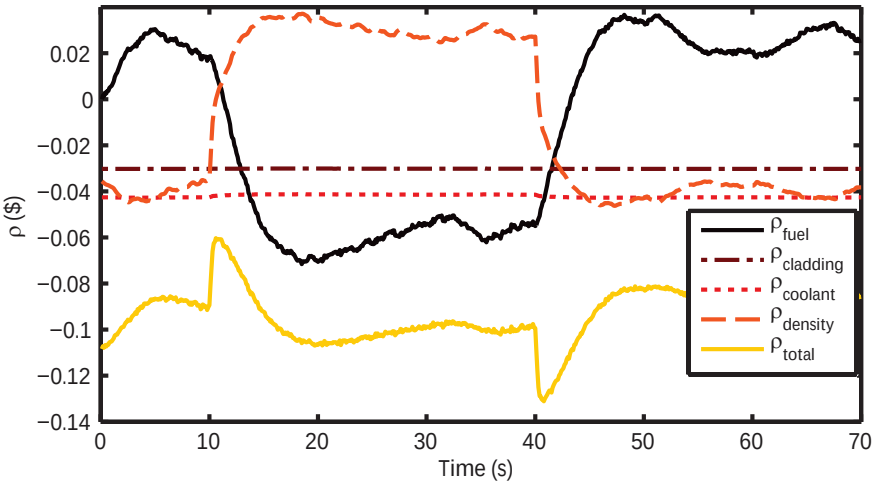


Figure 6.11: The evolution of the reactivity, calculated using average temperatures and denisty It can be observed how the effect of the density change and the fuel temperature cancel each other.

to determine the critical boron concentration and from this critical state an unprotected rod-ejection accident is simulated. The fact that the transient is unprotected indicates that no external measures are being taken after the control-rod ejection. The benchmark is based on the transient benchmark of Kozlowski and Downar (2003).

The exact geometry of the system can be found in App. A.4. The mini-core consists of nine numbered fuel assemblies surrounded by a water reflector, as depicted in Fig. 6.12, with each fuel assembly consisting of 17x17 rod positions. The eight outer assemblies are MOX based and the central assembly has an UO_2 based fuel. The control rods are located in the central fuel assembly and there is vacuum at the boundaries. The transient starts at a power of 1 W and from that moment the control rods are ejected within 0.1 s, creating a strong reactivity insertion. Due to the low starting power there is little feedback expected in the beginning, but at the end of the calculation the power is expected to be high enough to induce feedback effects.

To model the geometry in detail, including all separate fuel pins, cladding, burnable-poison coatings etc., the dynamic version of Tripoli is extended to incorporate ROOT geometry. ROOT is a framework for data processing, which also has an extensive geometry package (Brun and Rademakers, 1997) and Tripoli 4.7 can handle ROOT-geometry as input, but some alterations had to be made, to incorporate geometry updates throughout the calculation in the dynamic version. With the ROOT geometry it is relatively easy to create complex geometries.

First, a test calculation is performed to see if the geometry is implemented correctly. The test calculation consists of a steady-state calculation at hot full power, with the control rods completely withdrawn. The results can be compared with the results calculated using DYN3D coupled with FLICA (Gomez-Torres et al., 2012; Jiménez et al., 2012).

Results

First, the hot full power calculation of the benchmark is performed, with a power level is fixed at 100 MW; the resulting axial power profile in the central fuel assembly is plotted in Fig. 6.13. It shows that the power distribution calculated with both methods agree reasonably well. In Figs. 6.14 and 6.15 the temperature and density of the coolant is plotted and also these quantities agree fairly well; similar differences can be found between coupled steady-state calculations both using Monte Carlo, but with different codes (Hoogenboom et al., 2011). The k_{eff} calculated with DYN3D-FLICA is 1.00828, and with Tripoli4-SubChanFlow this value is 1.01800.

The differences between the two methods might be attributed to the two different thermal-hydraulics solvers used. SubChanFlow is a quasi 3D subchannel code, solving the mass, momentum and energy equations based on the three-equation approach. On the other hand, FLICA is a 3D two-phase flow code (Toumi et al., 2000), where the two-phase mixture is modelled using a set of four balance equations (mass, momentum and energy of mixture and mass of vapour). It is perceivable that the two methods yield a slightly different

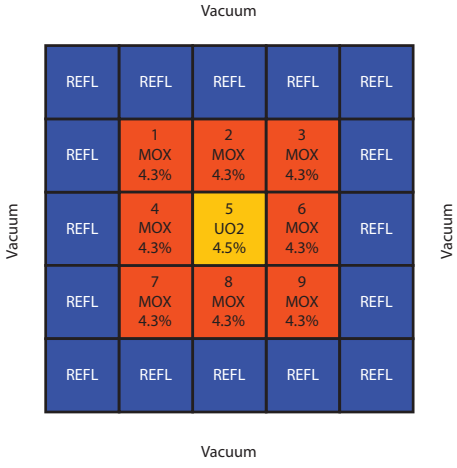


Figure 6.12: The layout of the fuel assemblies in the mini core. To distinguish between the different fuel assemblies they have been assigned a number.

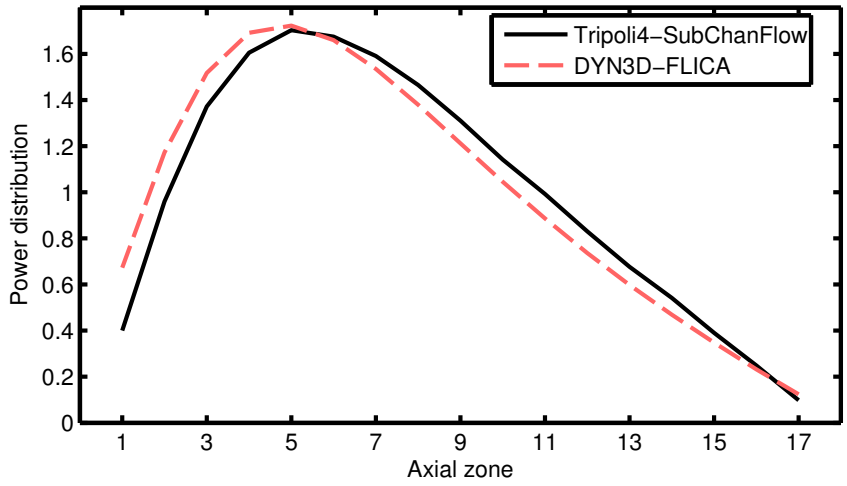


Figure 6.13: The power profile in the central fuel assembly at hot full power. A small discrepancy can be seen between the two different code systems.

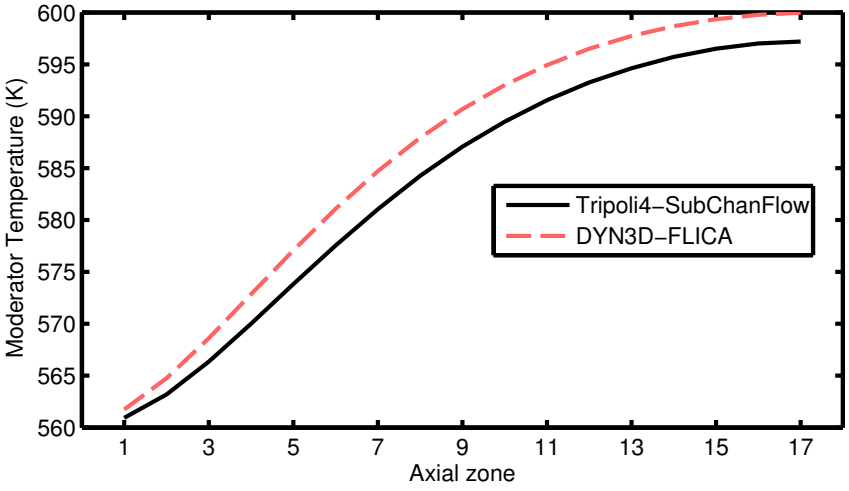


Figure 6.14: The temperature profile of the moderator in the central fuel assembly at hot full power. A small discrepancy can be seen between the two different code systems.

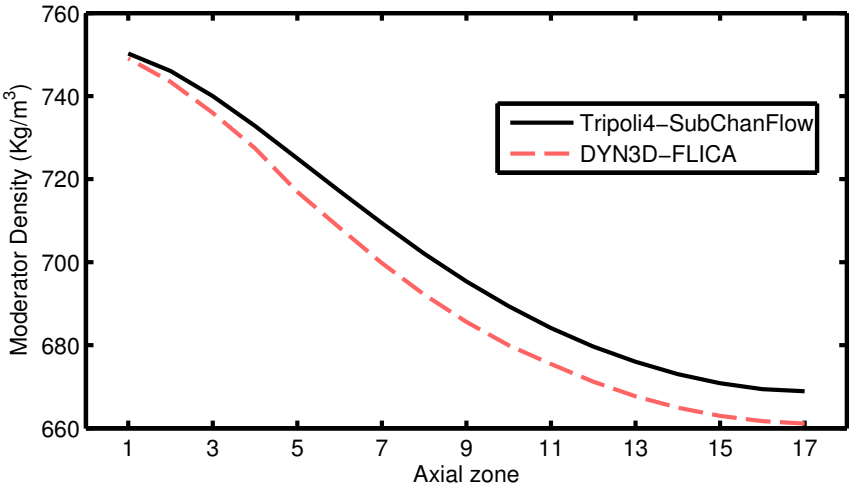


Figure 6.15: The density profile of the moderator in the central fuel assembly at hot full power. A small discrepancy can be seen between the two different code systems.

result.

After verifying the coupled-code system with the steady-state calculation, the transient analysis can be performed. The first step of the transient calculation is determining the initial conditions. This is achieved by calculating the critical boron concentration using iterative steady-state calculation at a power of 1 W, as explained in Sec. 6.2.2. Due to the low power, the thermal-hydraulic calculations are straight forward, so once the critical boron concentration is found the actual transient can be started.

For the transient calculation, the control rods are partially inserted in the central fuel assembly. At the start of the calculation the control rods are ejected during 0.1 s, creating a large reactivity insertion. The power production in the mini core is then simulated for 0.2 seconds. The results can be found in Fig. 6.16.

It can be seen that the total power level increases more than 10 decades and the highest power level is reached before the control rods are fully ejected. It can also be seen that during the power increase the two calculations have very similar results. It is only when the thermal-hydraulic effects become dominant that the two different methods start to deviate. The deviation may have the same origin as in the steady-state calculation, but the effects of the applied time grid can also be influence the width of the peak (Zerkak et al., 2011).

In Fig. 6.17 the evolution of the power profile in the central fuel assembly can be seen. The location where the maximum power is produced, starts at the bottom, due to the control rods which are inserted from the top. Then, as the control rods are withdrawn,

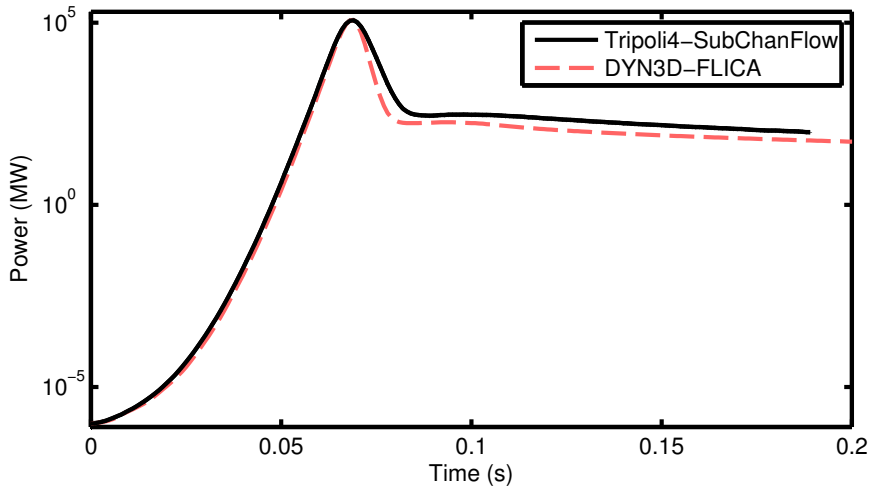


Figure 6.16: The total power during the transient reaches a peak before the control rods are fully ejected. The feedback mechanisms then have a larger reactivity effect than the further withdrawal of the control rods. The two calculation techniques agree well during the power increase, but show a small difference towards the end.

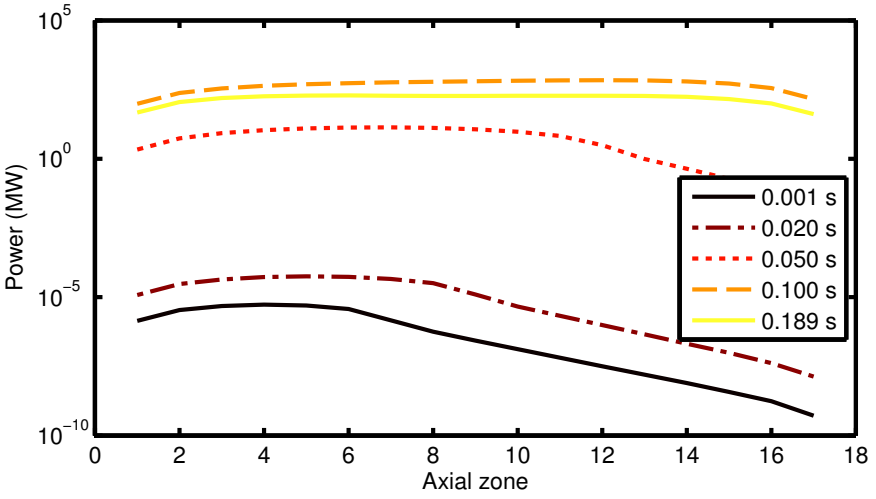


Figure 6.17: The power profile evolution in the central fuel assembly. It can be noted that the power peak moves upwards.

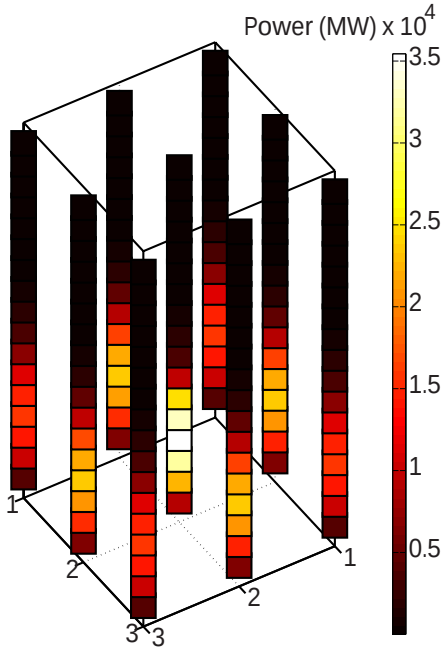


Figure 6.18: The normalised power profile after 0.001 s. Each bar depicts a fuel assembly and the 3D power profile can be seen. In the beginning of the transient the power peak is located near the bottom due to the control rods. It can also be seen that most of the power is produced in the central fuel assembly.

the maximum starts to move upwards. A 3D-visualisation of the power profile per fuel assembly at the start of the transient is depicted in Fig. 6.18. Here it is shown that most power is produced in the central fuel assembly, even with control rods inserted.

In Fig. 6.19 and Fig. 6.20 the fuel temperature and moderator density evolutions can be seen, respectively. It demonstrates that the fuel temperature responds more direct to the transient than the coolant density. Also, the response per fuel assembly is plotted, showing the impact of the control rod movement in the central fuel assembly on the outer fuel assemblies.

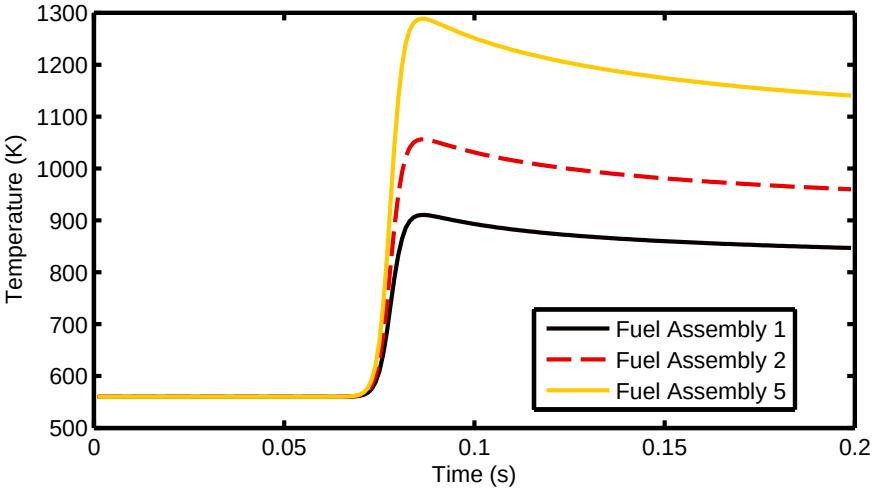


Figure 6.19: The time evolution of the average temperature in the fuel pins. It can be noted that the pins heat up quickly, but the cooling is more gradual. The central fuel assembly (5) reaches the highest temperature and the fuel assemblies at the corners (1) remain the coolest. The assembly numbers can be found in Fig. 6.12.

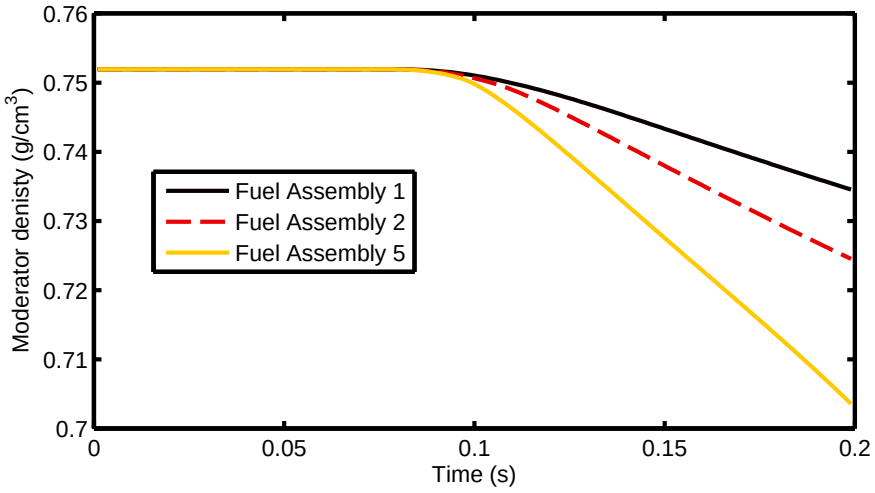


Figure 6.20: The time evolution of the average moderator density. When comparing this with Fig. 6.19, it can be noted that the change in moderator density is much slower than the heating of the fuel pins. The assembly numbers can be found in Fig. 6.12.

6.4 Summary of the coupled calculation

In this chapter a new method for coupled transient analysis is developed, which uses a stochastic neutron solver. The stochastic nature of the neutronics calculation requires special attention in the coupling script. One point of interest is the fact that a stochastic method rarely provides a solution in a point in time, but usually only a solution in an interval. Also the statistical uncertainty of the stochastic method must be taken into account.

Some numerical examples are given to demonstrate the possibility of a fully dynamic simulation using Monte Carlo for the neutronic part of the calculation. First, some simplified problems are simulated to demonstrate the robustness of a coupled calculation with respect to the stochastic nature of the Monte Carlo solution. It shows that some care must be taken, especially when the variance in a single batch becomes very large, but in most cases the couplings scheme is stable.

The following examples are more realistic: coupling SubChanFlow with Dynamic Tripoli explicitly. The first example is a pin cell on an infinite lattice and the transient is initiated with changing pump flow. With this simulation it is demonstrated that it is feasible to do a dynamic calculation on the seconds scale.

The final calculation is the benchmark calculation developed for the NURISP project. Here a stronger transient is calculated on a much shorter time scale. The results can be compared with the results obtained with a code system which couples a diffusion code with a thermal-hydraulics code. The results of the two methods compare nicely, although the final powers do deviate. The difference is 40 %, which is reasonable for a power increase with a factor of 10^{10}. A comparison with the coupled COBAYA-SubChanFlow code system Calleja et al. (2012) would be interesting to further investigate the effect of the thermal-hydraulics code to this final discrepancy.

All though there are still issues to be further investigated, it is demonstrated that it is possible to perform a coupled Monte Carlo/thermal-hydraulics calculation, which allows the exact calculation of the neutronics during a transient in a nuclear reactor. This is very valuable for validation calculations and for the development of new and possibly exotic reactor types with complex geometries.

6.5 Bibliography

M. Armishaw et al. The ANSWERS code MONK-a new approach to scoring, tracking, modelling and visualisation. In *Proceedings of 9th International Conference on Nuclear Criticality Safety (ICNC 2011), Edinburgh, UK*, 2011.

V. Bakanov et al. TDMCC Monte-Carlo package coupled with STAR-CD thermal-hydraulics code. In *Transactions of the American Nuclear Society*, volume 91, 250–251, 2004.

B. Boer et al. Validation of the DALTON-THERMIX code system with transient analyses of the HTR-10 and application to the PBMR. *Nuclear Technology*, **170**, 306, 2010.

C. Broeders et al. Validation of coupled neutron physics and thermal-hydraulics analysis for HPLWR. In *Proceedings of ICAPP03, Cordoba, Spain*, 2003.

F. Brown et al. Advanced Monte Carlo for reactor physics core analysis. In *Workshop at PHYSOR-2008, Interlaken, Switzerland*, 14–19, 2008.

R. Brun and F. Rademakers. ROOT - an object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, **389**, 81, 1997.

M. Calleja et al. A coupled neutronic/thermal-hydraulic scheme between COBAYA3 and SUBCHANFLOW within the NURESIM simulation platform. In *Proceedings of International Conference on the Physics of Reactors 2012, PHYSOR 2012: Advances in Reactor Physics*, volume 3, 2196–2210, 2012.

F. D'Auria et al. Neutronics/thermal-hydraulics coupling in LWR technology, vol. 1. Technical report, OECD/NEA, 2004.

J. Donnelly. Interpolation of temperature-dependent nuclide data in MCNP. *Nuclear Science and Engineering*, **168**, 180, 2011.

J. Dufek and W. Gudowski. Stochastic approximation for Monte Carlo calculation of steady-state conditions in thermal reactors. *Nuclear science and engineering*, **152**, 274, 2006.

E. Dumonteil and C. Diop. Biases and statistical errors in Monte Carlo burnup calculations: An unbiased stochastic scheme to solve Boltzmann/Bateman coupled equations. *Nuclear Science and Engineering*, **167**, 165, 2011.

F. P. Espel et al. New developments of the MCNP/CTF/NEM/NJOY code system – Monte Carlo based coupled code for high accuracy modeling. *Annals of Nuclear Energy*, **51**, 18, 2013.

A. Gomez-Torres et al. DYNSUB: A high fidelity coupled code system for the evaluation of local safety parameters - part I: Development, implementation and verification. *Annals of Nuclear Energy*, **48**, 123, 2012.

D. Griesheimer et al. An integrated thermal hydraulic feedback method for Monte Carlo reactor calculations. In *Proceedings of PHYSOR-2008 - American Nuclear Society Topical Meeting on Reactor Physics*, 2008.

J. E. Hoogenboom et al. A flexible coupling scheme for Monte Carlo and thermal-hydraulics codes. In *Proceedings of M&C conference*. Rio de Janeiro, 2011.

M. Hursin, T. J. Downar and B. Kochunas. Analysis of the core power response during a PWR rod ejection transient using the PARCS nodal code and the DeCART MOC code. *Nuclear Science and Engineering*, **170**, 151, 2011.

U. Imke and V. Sanchez. Validation of the subchannel code SUBCHANFLOW using the NUPEC PWR tests (PSBT). *Science and Technology of Nuclear Installations*, 2012.

A. Ivanov, V. Sanchez and U. Imke. Development of a coupling scheme between MCNP5 and SUBCHANFLOW for the pin- and fuel assembly-wise simulation of LWR and innovative reactors. In *Proceedings of M&C conference*. Rio de Janeiro, 2011.

K. Ivanov and M. Avramova. Challenges in coupled thermal-hydraulics and neutronics simulations for LWR safety analysis. *Annals of Nuclear Energy*, **34**, 501, 2007.

J. Jiménez, A. Gomez-Torres and V. Sanchez. Solutions to exercises of boron dilution benchmark by DYN3D/FLICA4 coupled codes at pin level. Technical Report D3.1.2.3c, NURISP, 2012.

H. Joo et al. Consistent comparison of Monte Carlo and whole-core transport solutions for cores with thermal feedback. In *Proceedings of the PHYSOR 2004: The Physics of Fuel Cycles and Advanced Nuclear Systems - Global Developments*, 35–47, 2004.

S. Kliem et al. Definition of a PWR boron dilution benchmark. Technical Report D3.1.2.2, NURISP, 2011.

J. Kópházi, D. Lathouwers and J. Kloosterman. Development of a three-dimensional time-dependent calculation scheme for molten salt reactors and validation of the measurement data of the molten salt reactor experiment. *Nuclear Science and Engineering*, **163**, 118, 2009.

D. Kotlyar et al. Coupled neutronic thermo-hydraulic analysis of full PWR core with Monte-Carlo based BGCore system. *Nuclear Engineering and Design*, **241**, 3777, 2011.

T. Kozlowski and T. J. Downar. OECD/NEA and U.S. NRC PWR MOX/UO2 core transient benchmark, 2003.

D. Legrady and J. E. Hoogenboom. Scouting the feasibility of Monte Carlo reactor dynamics simulations. In *Proceedings of PHYSOR-2008 - American Nuclear Society Topical Meeting on Reactor Physics*. Interlaken, 2008.

J. Leppänen, T. Viitanen and V. Valtavirta. Multi-physics coupling scheme in the Serpent 2 Monte Carlo code. *Transactions of the American Nuclear Society*, **107**, 1165, 2012.

V. Mahadevan, J. Ragusa and V. Mousseau. A verification exercise in multiphysics simulations for coupled reactor physics calculations. *Progress in Nuclear Energy*, 2011.

M. Mori et al. Monte-Carlo/Simmer-III reactivity coefficients calculations for the supercritical water fast reactor. In *Proceedings of Global 2003: Atoms for Prosperity: Updating Eisenhowers Global Vision for Nuclear Energy*, 1754–1762, 2003.

J. Peltonen and T. Kozlowski. Development of effective algorithm for coupled thermal-hydraulic-neutron-kinetics analysis of reactivity transient. *Nuclear Technology*, **176**, 195, 2011.

J. C. Ragusa and V. S. Mahadevan. Consistent and accurate schemes for coupled neutronics thermal-hydraulics reactor analysis. *Nuclear Engineering and Design*, **239**, 566, 2009.

G. Rowlands. Resonance absorption and non-uniform temperature distributions. *J. Nuclear Energy, Pts. A & B. Reactor Sci. and Technol.*, **16**, 1962.

V. Sanchez and A. Al-Hamry. Development of a coupling scheme between MCNP and COBRA-TF for the prediction of the pin power of a PWR fuel assembly. In *Proceedings of American Nuclear Society - International Conference on Mathematics, Computational Methods and Reactor Physics 2009, M and C 2009*, volume 5, 2009.

V. Seker, J. Thomas and T. Downar. Reactor physics simulations with coupled Monte Carlo calculation and computational fluid dynamics. In *Proceedings of 13th International Conference on Emerging Nuclear Energy Systems 2007, ICENES 2007*, volume 1, 345–349, 2007a.

V. Seker, J. Thomas and T. Downar. Reactor simulation with coupled Monte Carlo and computational fluid dynamics. In *Proceedings of Joint International Topical Meeting on Mathematics and Computations and Supercomputing in Nuclear Applications, M and C + SNA 2007*, 2007b.

B. L. Sjenitzer. Temperature dependent Monte Carlo simulation: Thermalization. Technical report, Delft University of Technology / Commissariat à l'Énergie atomique, 2009.

C. Tippayakul et al. Investigations on Monte Carlo based coupled core calculations. In *Proceedings of Societe Francaise d'Energie Nucleaire - International Congress on Advances in Nuclear Power Plants - ICAPP 2007, "The Nuclear Renaissance at Work"*, volume 2, 797–806, 2008.

C. T'Joen and M. Rohde. Experimental study of the coupled thermo-hydraulic-neutronic stability of a natural circulation HPLWR. *Nuclear Engineering and Design*, **242**, 221, 2012.

I. Toumi et al. FLICA-4: A three-dimensional two-phase flow computer code with advanced numerical methods for nuclear applications. *Nuclear Engineering and Design*, **200**, 139, 2000.

S. Van Der Marck, R. Meulekamp and A. Hogenbirk. New temperature interpolation in MCNP. In *Proceedings of M&C 2005 Conference, Avignon, France*, 2005.

M. Vazquez et al. Coupled neutronics thermal-hydraulics analysis using Monte Carlo and sub-channel codes. *Nuclear Engineering and Design*, **250**, 403, 2012.

J. Vedovi et al. Analysis of Ringhals I stability benchmark with TRACE/PARCS: Steady-state initialization, invited. *Transactions of the American Nuclear Society*, **90**, 560, 2004.

T. Viitanen and J. Leppänen. Explicit treatment of thermal motion in continuous-energy Monte Carlo tracking routines. *Nuclear Science and Engineering*, **171**, 165, 2012.

C. Waata et al. Coupling of MCNP with sub-channel code for analysis of HPLWR fuel assembly. In *Proceedings of the 11th International Topical Meeting on Nuclear Reactor Thermal-Hydraulics (NURETH-11)*, 2005a.

C. Waata et al. Results of a coupled neutronics and thermal-hydraulics analysis of a HPLWR fuel assembly. In *Proceedings of the American Nuclear Society - International Congress on Advances in Nuclear Power Plants 2005, ICAPP'05*, volume 1, 300–306, 2005b.

J. Watson and K. Ivanov. Demonstration of implicit coupling of TRACE/PARCS using simplified one-dimensional problems. *Nuclear technology*, **180**, 174, 2012.

G. Yesilyurt, W. Martin and F. Brown. On-the-fly doppler broadening for Monte Carlo codes. *Nuclear Science and Engineering*, **171**, 239, 2012.

O. Zerkak et al. Revisiting temporal accuracy in Neutronics/TH code coupling using the NURESIM LWR simulation platform. In *The 14th International Topical Meeting on Nuclear Reactor Thermalhydraulics (NURETH-14)*, 2011.

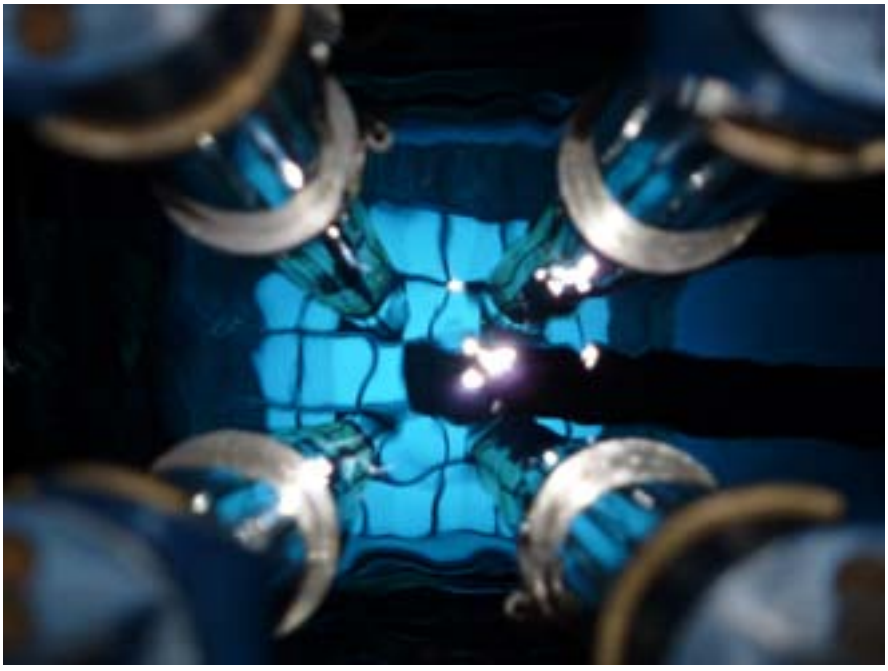
A. Zoia et al. Doppler broadening of neutron elastic scattering kernel in tripoli-4®. *Annals of Nuclear Energy*, **54**, 218, 2013.

Part IV

Conclusions and Recommendations

7

CONCLUSION AND RECOMMENDATIONS



Picture: RID-TUDelft

In this thesis a novel Monte Carlo method for long-time kinetic and dynamic calculations has been developed and demonstrated. One of the major challenges in this method was simulating particles which operate at different time scales, ranging in many orders of magnitude. Precursors operate in the seconds to minutes scale, whereas prompt neutrons and prompt neutron chains exist in the range of microseconds.

Another major challenge was the simulation of these prompt-neutron chains, since their lengths have a large naturally occurring variance. Also, the feedback mechanisms had to be incorporated into the method to be able to calculate realistic transients in power reactors.

In the last part of this thesis, a successful demonstration has been presented, performing a fully dynamic stochastic analysis of a transient in a realistic and complex geometry, modelled in full detail and simulated with continuous energy.

7.1 New Monte Carlo techniques

Precursor sampling

The technique of forced precursor decay has made it possible to sample precursors in such a way that they will induce enough prompt-neutron chains for reliable statistics in all time intervals. All precursors are forced to start a chain in each time interval, which roughly ensures equally accurate statistics per time interval. Most accurate results are achieved when combining all precursor families into a single particle, although caution must be taken when simulating such a combined particle, because the decay probability is now time dependent.

Simulation of a prompt-neutron chain

The naturally occurring variance in prompt-neutron chains can be reduced by applying the newly developed branchless method. With this method an interaction will always yield a single neutron, which can be either a scattering neutron or a fission neutron. This way, from each collision one particle will continue, ensuring a prompt-neutron chain, which does not branch into multiple chains and therefore reduces the variance in the tallies.

Dynamic simulation scheme

The new dynamic simulation scheme needs a few alterations, from the traditional simulation schemes, since it does not organise a simulation by particle, but by time interval. If a particle in a time interval crosses the time boundary, it will be stopped and stored until the scheme finishes all particles and continues with the next time interval. Between these intervals the system properties can be updated.

The initial condition of a dynamic calculation can be sampled from a criticality calculation. If a converged source is achieved, initial neutron and precursor distributions can be sampled.

Thermal-hydraulics coupling

To incorporate feedback effects, the dynamic scheme can be coupled to a thermal-hydraulics code. This enables the possibility to perform a fully dynamic analysis of a nuclear system. When coupling a stochastic method with a deterministic method some caution is needed. A stochastic method does not yield standard coupling coefficients, but an integral estimate in an interval. Also the variance estimation and mean result must be inspected closely, since the neutron histories have become correlated.

7.2 Demonstration of the dynamic scheme for Monte Carlo

The possibilities of these new techniques and schemes have been demonstrated successfully in a number of examples, each focussing on a different technique.

Validating new Monte Carlo techniques

With a purpose-built Monte Carlo program the feasibility has been demonstrated of performing a kinetic analysis applying only a stochastic method. The Dynamic Monte Carlo code yields the same results as a point-kinetics calculation, which is considered to be accurate in this case.

Also the two new Monte Carlo techniques, the new precursor sampling technique and the branchless method, have proven to be effective. The efficiency gained with each technique has been measured and the FoM increased significantly. The new techniques not only increased the FoM in the test cases, which were specifically designed to demonstrate the variance reducing techniques, but they also increased the FoM by a factor of ten in the general transient problem.

Realistic demonstration problem

The implementation of the dynamic method in the general purpose Monte Carlo code Tripoli has shown its the general applicability. Demonstration calculations have shown that the quasi-static calculations no longer fully agree with the results of the dynamic Monte Carlo calculation, in the complex geometry of a fuel assembly, with the ejection of a number of control rods. This is to be expected with such a complicated transient.

In such a complex geometry variance reduction techniques usually are less effective than in simple test cases, but it can still seen that the newly developed techniques increase the efficiency of the total calculation by a factor of two. The results of the Dynamic Tripoli code demonstrate not only the general applicability of the new methods, but also the practical feasibility of a transient analysis in realistic complex geometry.

Fully dynamic problem

The possibility of coupling Monte Carlo with thermal-hydraulics has been demonstrated in three steps. First a rudimentary feedback mechanism implemented in the basic Monte Carlo code demonstrated the feasibility to perform a Monte Carlo calculation with feedback. Then a realistic, but small system was simulated, demonstrating the possibility of coupling a real thermal-hydraulics solver to the Monte Carlo solver, analysing a transient of more than a minute.

The analysis of a NURISP benchmark problem demonstrated that the results of dynamic Monte Carlo method agree fairly well with the results of an advanced coupled diffusion calculation with pin power reconstruction. It shows that it is possible to perform realistic transient analysis using Monte Carlo, even when the system is super-prompt-critical. More research is needed to investigate the source of the discrepancies and to further validate the dynamic Monte Carlo code.

General conclusions

The method presented in this thesis is a novel and unique method. Never before was it possible to calculate a transient problem with the exact and detailed modelling of the geometry and with continuous energy. This accuracy makes the method very suitable as a validation tool for other computational methods.

The calculations performed so far, all agree nicely with existing deterministic methods, which might imply that the Monte Carlo method is a more expensive way to calculate problems which can already be solved. However, these problems have been selected to be solvable by deterministic methods, creating a convincing example of the validity of the new Monte Carlo code.

The main advantage of the Monte Carlo method is that no system-specific approximations are used and therefore it is generally applicable. It can solve not only these transient problems, but also transient problems in new reactor types and unique research reactors. Due to the high computational cost, the main application will be to act as a validation tool for the computationally less expensive deterministic methods.

7.3 Future work

Validation

The calculations shown in this thesis demonstrate the feasibility of performing a coupled Monte Carlo neutronics/thermal-hydraulics calculation, even in a realistic geometry. However, it has not been possible yet to validate these calculations completely, even though the results compare well with the DYN3D-FLICA results. A future challenge lies in the validation of the dynamic Monte Carlo method and its implementation in the Tripoli code. The method should be benchmarked against real-life, well-documented experiments.

Furthermore, it is necessary to further investigate the effects of the variance on the coupling mechanism. It would be ideal if a theoretical description can be found to determine the validity of the estimation of the mean and the variance.

Also, the method of tallying the produced power must be further investigated. In this thesis the variance is calculated between batches of neutrons and the thermal-hydraulic conditions are calculated per batch. To validate the results, it is advised to investigate the results of a tally, which should have a normal distribution. If this is not the case, the central limit theorem is not valid.

If the results do not have a normal distribution, which could be the case for a reactor with flashing channels, it might prove meaningful to tally also a different quantity next to the power profile. For example, the maximum power peak reached during the transient disregarding the exact location of the peak could be tallied or the time it takes to reach a power peak. The regular power tally is of course still needed for the coupling with the thermal-hydraulics code.

Further development

Some parts of the method can be improved. Since this work is the first attempt to couple a stochastic neutronics code for transient analysis to a thermal-hydraulics code, simple schemes and techniques have been used for the coupling. After this demonstration, the next step is to further improve the simulation scheme and Monte Carlo techniques.

The neutronics part of the calculation can be improved with better use of weight windows. In this work the weight windows have only been adjusted per time interval. The efficiency of the calculation can be improved when proper weight windows are applied, preferably in an automated fashion. This way the weight windows can be optimised not only in the temporal variation, but also in the other dimensions of phase space.

Another neutronics improvement can be found in the generation of temperature-dependent cross sections. Multiple novel techniques are already successfully applied for accurate temperature modelling in steady-state Monte Carlo thermal-hydraulics coupling and one

of these novel ways can also make the dynamic simulation more accurate.

A kinetic-specific improvement can be found in the statistics of the prompt-neutron chain-length, which still is a large contributor to the total variance. An approach could be to use material cross sections, instead of first selecting a nuclide from the material and then using the nuclide cross sections. This will average the interaction response, which would decrease the variance in chain lengths.

The geometry, temperatures and densities can be made time dependent to further improve the coupling. In the current implementation the system properties are piecewise constant, with a step at the time boundaries, but this is not compulsory for the method. The time dependence during a time interval can be implemented for externally induced changes to the system, such as the movement of a control rod, but also for results coming from the coupling, such as the temperature profile. This last feature is especially useful when using a (semi-) implicit coupling scheme.

Furthermore, the thermal-hydraulics part of the calculation can be improved. In this work a sub-channel code has been used for the thermal-hydraulics, but there are higher-fidelity computational-fluid-dynamics codes available. During the development of the coupling scheme, the efficiency of the sub-channel code is useful, but it is more logical to couple a high-fidelity method like Monte Carlo, with a high-fidelity thermal-hydraulics code.

Finally, the coupling scheme can be further improved. For example, an implicit scheme can be used for the coupling. This can be achieved by storing a copy of the particles at the start of a time interval and use these copies to start iterating per time interval. Also the data transfer becomes large when extending the method to full core calculations, with separate temperatures for all axial and radial zones of the fuel pins. Therefore integrating neutronics and thermal-hydraulics further can be an interesting option, enabling internal coupling and automated geometry mapping between the two codes.

New possibilities

The simulation of precursor particles also opens new possibilities. For example, precursor transport can be easily added to the method. In molten salt reactors the transport of precursors has a big influence on the behaviour of the reactor and therefore it would be a good asset to be able to simulate this in a Monte Carlo calculation.

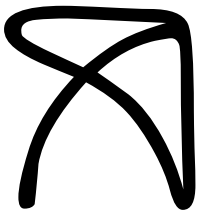
Also the simulation of different precursor nuclides instead of complete precursor families should be developed. In this case more information is needed about the yield of specific precursors, but the method can easily handle an increase in precursor types. This would remove another approximation from the method, but it will only improve the accuracy if the nuclear data of the separate precursor nuclides is accurate, which might prove challenging.

Application of dynamic Monte Carlo

All in all, it has been shown that it is feasible to perform dynamic Monte Carlo analyses on nuclear systems. The main task is now to make this method generally available, so it can be tested on many different applications for further refinement.

Part V

Appendices



GEOMETRY DESCRIPTIONS

A.1 Homogeneous system

The homogeneous system consists of a box of artificial fissile material. The box is 24 by 10 by 12 cm and is placed in vacuum, see Fig. A.1. The material properties are given in Table A.1, which also shows that the neutrons in the system have a single speed, which is set to this value in order to achieve a reasonable generation time. There are six precursor families; the yields and properties of the precursors families are given in Table A.2, which are taken from Dam et al. (2005). The scattering is isotropical and the scattering cross section is chosen such that the system is near critical: the reactivity is 0.01 \$. The reactivity insertion is realised by changing the absorption cross section from Σ_{a1} to Σ_{a2}, while keeping the total cross section constant.

Figure A.1: The homogeneous system is a small cuboid, with mono-energetic neutrons and isotropic scattering. The cuboid is placed in vacuum.

Table A.1: Material properties of the test problem. The box is homogeneous and the problem is mono energetic

Material properties
$\Sigma_t = 1.0000\ \text{cm}^{-1}$
$\Sigma_f = 0.2500\ \text{cm}^{-1}$
$\Sigma_{a1} = 0.5882\ \text{cm}^{-1}$
$\Sigma_{a2} = 0.5870\ \text{cm}^{-1}$
$\nu = 2.500$
$\beta = 0.006850$
$v = 2.2000 \times 10^4\ \text{cm}^{-1}$

Table A.2: The precursors are divided into six families. Here the fractions and decay constants per precursor family i are given. Also the total delayed fraction and inversely weighted average decay constant are shown.

Family	λ (s^{-1})	β
1	0.0127	0.000260
2	0.0317	0.001459
3	0.1156	0.001288
4	0.3110	0.002788
5	1.4000	0.000877
6	3.8700	0.000178
av/tot	0.0784	0.006850

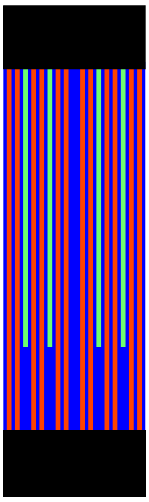
A.2 Fuel assembly

The fuel assembly consists out of 17 by 17 fuel pins, from which 24 pins are replaced by control rods and the central pin is replaced by a water filled guide tube for measurement equipment. The exact layout of the fuel assembly is given in Fig. A.2 and the boundary conditions in the x-y plane are reflective.

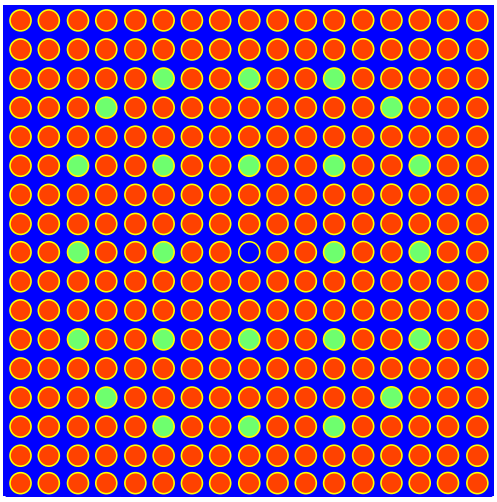
The fuel consists out of uranium-dioxide and the control rods are boron based. The coolant is borated water and also the cladding and gap regions are modelled, as depicted in Fig. A.3. The material densities are given in Tab. A.3 The nuclear data used is JEFF3.1.1A. Santamarina, D. Bernard, P. Blaise, *et al.* (2009).

The top and bottom each have a reflector and at the edge of the reflector vacuum is assumed. The width of the system is 21.42 cm and the height of the fuel pins is 366 cm. the top reflector is 40 cm thick and the bottom reflector 46 cm.

The model is based on the transient benchmark of Kozlowski and Downar (2003), but some simplifications are used and there is no feedback present. The system is made close to critical, by adjusting the control rod height accordingly.



(a) Vertical cross section of fuel assembly, x- and z-axis are not at the same scale.



(b) Horizontal cross section of fuel assembly

Figure A.2: Layout of a fuel assembly; the fuel is colored red, the control rods are in green, the borated water is in blue, the top and bottom reflectors are in black and the cladding is in yellow.

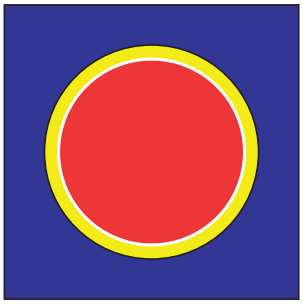


Figure A.3: The layout of the fuel rod (left) and control rod (right) in the fuel assembly. The centre region of the control rod can be either filled with the control rod or with coolant.

fuel		gap		cladding		control rod	
^{235}U	0.001237	^{16}O	0.000063	^{90}Zr	0.032543	C	0.033261
^{238}U	0.003673			^{91}Zr	0.007097	^{10}B	0.115219
^{16}O	0.075926			^{92}Zr	0.010848	^{11}B	0.026072
				^{94}Zr	0.010993		
				^{96}Zr	0.001771		

coolant		top		bottom	
H in H_2O	0.081857	H in H_2O	0.010424	H in H_2O	0.120848
^{16}O	0.041308	^{16}O	0.005212	^{16}O	0.060424
^{10}B	0.000048	^{10}B	0.000003	^{10}B	0.000039
^{11}B	0.000182	^{11}B	0.000014	^{11}B	0.000162
		^{54}FE	0.002452	^{54}FE	0.004879
		^{56}FE	0.038495	^{56}FE	0.076589
		^{57}FE	0.000889	^{57}FE	0.001769
		^{58}FE	0.000118	^{58}FE	0.000235

Table A.3: Material properties for the various materials in the fuel assembly. The nuclide densities are given in mol/cm^3

A.3 Pin cell

The pin-cell geometry consists of a single pin with reflective boundaries in the x-y plane and vacuum at the top and bottom. The layout and dimension are given in Fig. A.4, where also the mapping from one code to the other is depicted. In the thermal-hydraulics model the gap between the fuel and the cladding is modelled, but for the Monte Carlo solution this gap is neglected. The fuel is made of UO_2, but the enrichment is varied in the axial direction. This enrichment is plotted in Fig. A.5. In this case the system is made critical by adjusting the boron concentration in the coolant, which will yield a relative high boron concentration, since there is no leakage in the x-y plane and no other neutron absorbers.

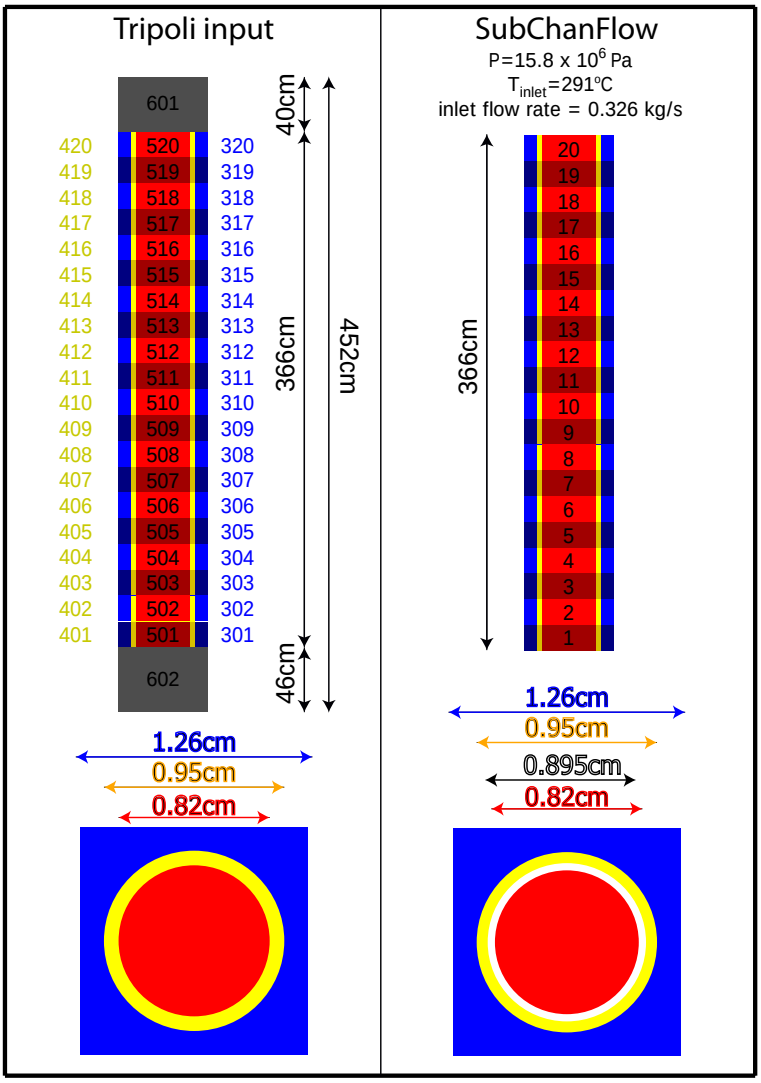


Figure A.4: The system for the numerical example is a small homogeneous cuboid, with mono-energetic neutrons and isotropic scattering. The cuboid is placed in vacuum.

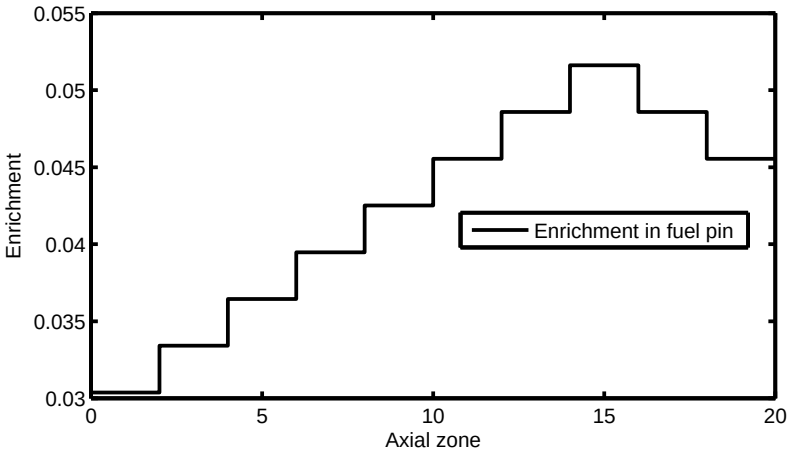


Figure A.5: The system for the numerical example is a small homogeneous cuboid, with mono-energetic neutrons and isotropic scattering. The cuboid is placed in vacuum.

A.4 Mini core

The mini-core is taken from the benchmark described in Kliem et al. (2011), which is based upon the OECD benchmark for MOX/UO2 core transients (Kozlowski and Downar, 2003). In the system there are two types of fuel assemblies: one with MOX fuel and one with UO_2 fuel. The mini core consists of nine fuel assemblies, from which eight contain MOX and the central fuel assembly contains UO_2. The whole set up is surrounded by water reflectors, as depicted in Fig. A.6. The assembly has 17×17 rod positions, which contains different kind of rods, as depicted in Figs. A.7 and A.8. The width of a fuel assembly is 21.42 cm, the height is 366 cm and the pitch between the rods is 1.26 cm. The control rods are only present in the UO_2 fuel assembly.

The geometry of the different types of rods can be found in Fig. A.9 together with Tab. A.5 and the dimensions can be found in Tab. A.6 and this layout is modelled in full detail. The nuclide densities in the materials can be found in Tab. A.7 and cross section are taken from the JEFF3.1.1 evaluations (A. Santamarina, D. Bernard, P. Blaise, *et al.* , 2009). Other boundary conditions can be found in Tab. A.4. The exact nuclide densities in the coolant will change, depending on the results of the thermal-hydraulics calculation.

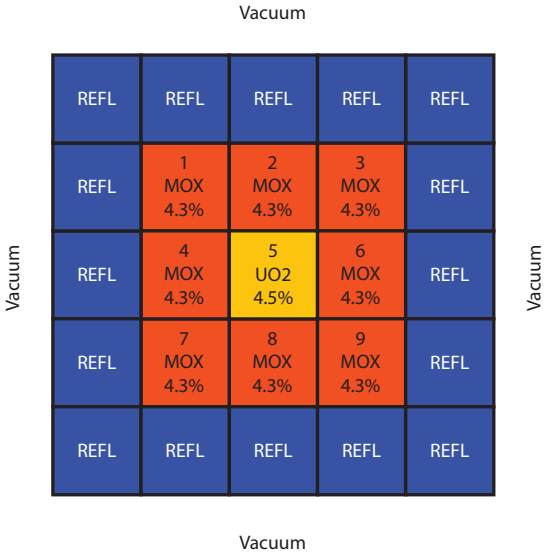


Figure A.6: The layout of the fuel assemblies in the mini core.

Table A.4: System properties for hot full power (HFP) and hot zero power (HZP)

Boundary conditions HFP	
Core Power	100MW
Mass flow rate (core)	739.08 kg/s
Mass flow rate (FA)	82.12 kg/s
Core Outlet pressure	15.40 Mpa
Coolant inlet temperature	560 K
Boron concentration	200 ppm (nuclide density)
Insertion depth control rods	0.0 cm
Boundary conditions HZP	
Core Power	1 W
Mass flow rate (core)	739.08 kg/s
Mass flow rate (FA)	82.12 kg/s
Core Outlet pressure	15.40 Mpa
Coolant inlet temperature	560 K
k_{eff} ($t = 0$ s)	1.0000
Insertion depth control rods	232.433 cm
Scenario	CR ejection, linear speed fully out at 0.1 s

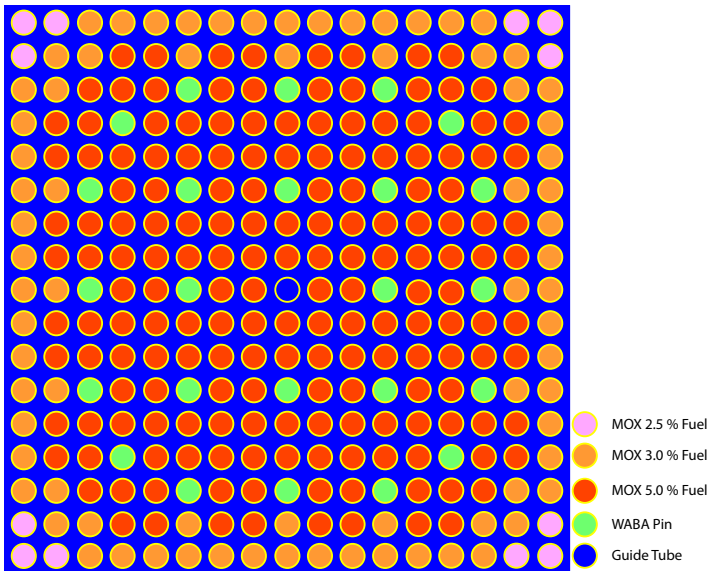


Figure A.7: The rod configuration of the MOX containing fuel assembly.

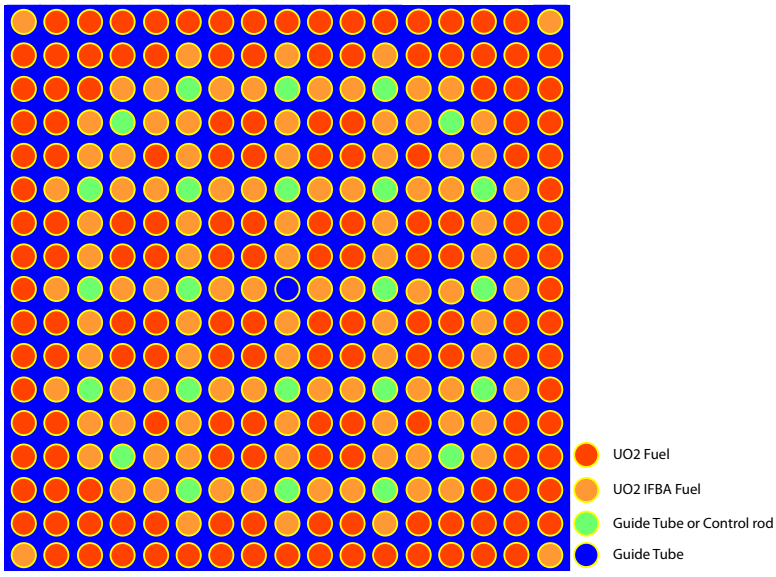


Figure A.8: The rod configuration of the UO_2 containing fuel assembly.

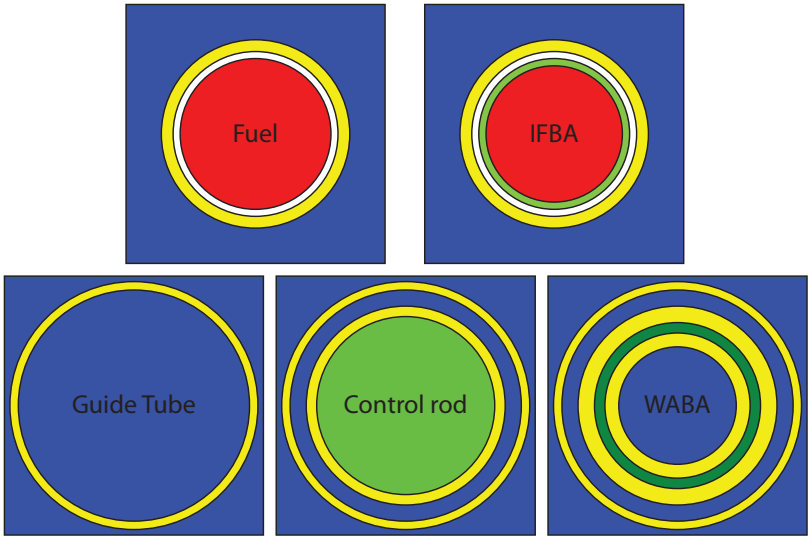


Figure A.9: The layout of the rods in the mini core.

Table A.5: Rod materials

	Fuel	IFBA	GT	CR	WABA
r0-r1	Fuel	Fuel	Water	Cr	Water
r1-r2	Gap	Ifba	Clad	Clad	Clad
r2-r3	Clad	Gap		Water	Waba
r3-r4		Clad		Clad	Clad
r4-r5					Water
r5-r6					Clad

Table A.6: Rod dimensions in cm

	Fuel	IFBA	GT	CR	WABA
r1	0.3951	0.3951	0.5624	0.4331	0.2858
r2	0.4010	0.3991	0.6032	0.4839	0.3531
r3	0.4583	0.4010		0.5624	0.4039
r4		0.4583		0.6032	0.4839
r5					0.5624
r6					0.6032

Table A.7: Materials used in the mini core

UO_2	
^{235}U	6.27×10^{-4}
^{238}U	1.31×10^{-2}
^{16}O	2.75×10^{-2}

MOX 2.5 %	
^{234}U	2.77×10^{-7}
^{235}U	2.76×10^{-5}
^{236}U	1.37×10^{-7}
^{238}U	1.36×10^{-2}
^{239}Pu	3.26×10^{-4}
^{240}Pu	2.04×10^{-5}
^{241}Pu	1.38×10^{-6}
^{242}Pu	3.44×10^{-7}
^{16}O	2.80×10^{-2}

MOX 3.0 %	
^{234}U	2.76×10^{-7}
^{235}U	2.75×10^{-5}
^{236}U	1.37×10^{-7}
^{238}U	1.35×10^{-2}
^{239}Pu	3.91×10^{-4}
^{240}Pu	2.45×10^{-5}
^{241}Pu	1.66×10^{-6}
^{242}Pu	4.12×10^{-7}
^{16}O	2.80×10^{-2}

MOX 5.0 %	
^{234}U	2.70×10^{-7}
^{235}U	2.69×10^{-5}
^{236}U	1.34×10^{-7}
^{238}U	1.33×10^{-2}
^{239}Pu	6.52×10^{-4}
^{240}Pu	4.09×10^{-5}
^{241}Pu	2.76×10^{-6}
^{242}Pu	6.87×10^{-7}
^{16}O	2.80×10^{-2}

Waba	
C	2.34×10^{-3}
^{10}B	2.01×10^{-3}
^{11}B	7.36×10^{-3}
^{27}Al	2.28×10^{-2}
^{16}O	3.42×10^{-2}

Control rod	
C	1.21×10^{-2}
^{10}B	1.04×10^{-2}
^{11}B	3.80×10^{-2}

Ifba	
^{90}Zr	2.80×10^{-3}
^{91}Zr	6.10×10^{-4}
^{92}Zr	9.32×10^{-4}
^{94}Zr	9.44×10^{-4}
^{96}Zr	1.52×10^{-4}
^{10}B	2.33×10^{-3}
^{11}B	8.53×10^{-3}

Gap	
^{16}O	2.27×10^{-5}

Cladding			
^{90}Zr	1.31×10^{-2}	^{54}Fe	2.96×10^{-6}
^{91}Zr	2.85×10^{-3}	^{56}Fe	4.65×10^{-5}
^{92}Zr	4.36×10^{-3}	^{57}Fe	1.07×10^{-6}
^{94}Zr	4.41×10^{-3}	^{58}Fe	1.43×10^{-7}
^{96}Zr	7.11×10^{-4}	^{50}Cr	1.97×10^{-6}
^{112}Sn	2.89×10^{-6}	^{52}Cr	3.80×10^{-5}
^{114}Sn	1.97×10^{-6}	^{53}Cr	4.31×10^{-6}
^{115}Sn	1.01×10^{-6}	^{54}Cr	1.07×10^{-6}
^{116}Sn	4.33×10^{-5}	^{14}N	8.39×10^{-5}
^{117}Sn	2.29×10^{-5}	^{15}N	3.10×10^{-7}
^{118}Sn	7.22×10^{-5}	^{122}Sn	1.38×10^{-5}
^{119}Sn	2.56×10^{-5}	^{124}Sn	1.73×10^{-5}
^{120}Sn	9.71×10^{-5}		

Coolant (560 K, 200 ppm boron)	
^{1}H in H_2O	3.03×10^{-2}
^{16}O	1.51×10^{-2}
^{10}B	1.95×10^{-6}
^{11}B	7.14×10^{-6}

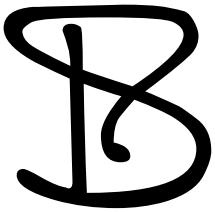
A.5 Bibliography

A. Santamarina, D. Bernard, P. Blaise, *et al.* . The JEFF-3.1.1 nuclear data library. JEFF Report 22, 2009.

H. V. Dam, T. H. J. J. van der Hagen and J. E. Hoogenboom. Nuclear reactor physics, lecture notes AP3341. Delft University of Technology, http://www.janleenkloosterman.nl/reports/ap3341.pdf, 2005.

S. Kliem et al. Definition of a PWR boron dilution benchmark. Technical Report D3.1.2.2, NURISP, 2011.

T. Kozlowski and T. J. Downar. OECD/NEA and U.S. NRC PWR MOX/UO2 core transient benchmark, 2003.



FROM STATIC TO DYNAMIC TRIPOLI: A REFERENCE GUIDE

This appendix is a technical description of the modifications made to the Tripoli 4.7 code, which to enable the calculation of dynamic behaviour of nuclear reactors.

The first section is a users guide, which explains how to apply the new functionalities. The remainder of the appendix is more a developers guide, which describes the exact implementation of some specific dynamic functionalities such as the new precursor class, the dynamic simulation mode and the variance reduction techniques for dynamic calculations. Also the implementation of the feedback is discussed and finally there are some general improvement which are not specific for dynamic calculations.

B.1 User Guide

There are a couple of new keywords and switches in Dynamic Tripoli, which are all optional. The keywords can be added to the input file and a switch is added when executing the code.

DYNAMIC	There is a new simulation mode, next to *CRITICALITY*, *FIXED_SOURCES_CRITICALITY* and *SHIELDING* there is now the *DYNAMIC* mode. This mode is similar to the fixed sources criticality mode, but instead of it calculates time intervals consecutively, A detailed description can be found in section B.3. One batch will contain all time intervals and a time mesh is compulsory. Also, the branchless transport method will be automatically selected.
PRECURSOR	There is a new particle type called precursor. Since it produces only neutrons, it has the same limits as neutrons. Its only function is to produce delayed neutrons.
PRECURSOR_DECAY	This can be either *biased*, which is default, or *unbiased*, which will set natural precursor decay. With biased decay a precursor is forced to produce a delayed neutron in all time intervals.
PRECURSOR_IMPORTANCE	The precursor importance needs two inputs, first is either *TIMED* or *EXPECTED*. *TIMED* calculates the importance from the timed weight of a precursor. This is the statistical weight which represents the statistical weight of the precursor at the current time. *EXPECTED* calculated the importance relative to the expected weight of the delayed neutron which will be produced by the precursor in the next time interval. The next input is a real. This is the importance of the precursor. The weight of the precursor is $w_{timed} = \frac{w_{av}}{importance}$
SOURCE	There are two new sources. The PRECURSOR-type source will generate a source of only precursors and the NEUTRON_PRECURSOR-type source will calculate a steady-state distribution of neutrons and precursors.
DCS <N>	This Dynamic Criticality Source keyword instructs Tripoli to perform <N> criticality cycles before sampling a source. From the last cycle of the criticality calculation the dynamic source will be sampled. The DCS cycles have the forward transport mode, instead of the branchless mode.
TRANSPORT_MODE BRANCHLESS	This is a new transport mode, which is automatically selected in the dynamic mode. If this is not desired, the keyword TRANSPORT_MODE FORWARD can be used for normal transport. In a fixed sources calculation this mode can also be used.

NO_COMBING	This will switch off the combing technique for population control in a dynamic calculation. Russian roulette will be played on the precursors instead.
FEEDBACK	If this keyword is present in the input file the code will look for the feedback script and run it to get a new input for every time interval. This option will also look for a tally named feedback and output this tally in a power file, with extension <datafilename>.power.
-F	This switch is applied when calling Tripoli, it is not in the input file. It is used in this manner: -F <scriptname> It is used to indicate the script which couples Tripoli to a code which calculates feedback effects. The script will be called in the following way: <scriptname> <powerfilename> <filename of new inputfile> <directory in which feedback script is run>
SIMULATION_TIME	The simulation time will be tallied. The simulation time per time interval will be given in the output file with uncertainty.
PRINT_HISTORIES <N_1><N_2><N_3>	This is an event logger with maximal N_1 lines, N_2 lines per particle and the N_3 initial lines skipped

B.2 Precursor class

The precursor class simulates a precursor and it inherits from the particle class, but it also has a few precursor specific attributes: it contains decay constants, delayed groups and a creation time.

If the precursor biasing is set to unbiased, the follow() routine of a precursor will generate a delayed neutron a time t where t is distributed according to the decay constants. If the biasing is set to biased, a delayed neutron will be created in every time interval, with a modified weight to ensure an unbiased game.

The precursors can be generated with two constructors, each generating another family distribution: the steady-state and the fission distribution. The steady-state distribution is given by $\frac{\lambda^b \beta_i}{\lambda_i \beta}$ and the fission distribution is given by $\frac{\beta_i}{\beta}$. The third constructor is a simple copy constructor.

B.3 Time stepping

When the Dynamic mode is selected, the simulation will progress per time interval, which requires a time mesh. All particles in a time interval are processed including its progeny, before continuing to the next interval. If a neutron transverses the time boundary it will be stopped there and stored for the next time interval.

A precursor in the unbiased case will create a neutron which will be stored until its time interval is reached. In the biased case the precursor will generate a new delayed neutron at an uniformly distributed time in the interval and then the precursor particle is saved for the next interval.

After the time interval the required weight is adjusted to keep approximately the same number of neutrons throughout the entire calculation. Also a new geometry file can be read. This gives the opportunity to alter the layout of the system (moving control rods, swelling, expanding, changing temperatures, boron insertion etc.)

B.4 Source

For the source distribution there are a few new options. There is the possibility to have a source of precursors and there is the possibility to have a combined source of precursors and neutrons. When the precursor source is selected all source particles will be precursors, with a steady-state family distribution. If the neutron_precursor option is selected the fraction of prompt neutrons will be given by:

$$\frac{1}{1+\frac{\beta}{\lambda^b}\upsilon\nu\Sigma_f}$$

where all properties are dependent on energy and location $(E,\mathbf{x})$

It is also possible to use the DCS option and in this case the code will first perform <N> criticality cycles. This calculation is performed only with neutrons and it will calculate a eigenfunction for the neutron flux. From this steady-state neutron flux a initial distribution will be sampled, with either neutrons, precursors or both, according to the steady-state distribution for both neutrons and precursors, which are sampled using, respectively:

$$n_0(\boldsymbol{r},\boldsymbol{\Omega},E)=\frac{\phi_0(\boldsymbol{r},\boldsymbol{\Omega},E)}{\upsilon(E)}=\frac{\psi_0(\boldsymbol{r},\boldsymbol{\Omega},E)}{\upsilon(E)\Sigma_t(\boldsymbol{r},E)}$$

$$C_0(\boldsymbol{r})=\int_{4\pi}\int_0^\infty\frac{\beta(\boldsymbol{r},E)\nu(\boldsymbol{r},E)\Sigma_f(\boldsymbol{r},E)}{\lambda^b(\boldsymbol{r},E)\Sigma_t(\boldsymbol{r},E)}\psi_0(\boldsymbol{r},\boldsymbol{\Omega},E)\mathrm{d}E\mathrm{d}\Omega$$

B.5 Variance reduction

For dynamic calculations a new variance reduction technique has been developed, which is called the branchless method (BLS). It will keep prompt-neutron chains from branching unnecessarily. At every interaction a single particle will emerge with statistical weight $\frac{\Sigma_s+\nu\Sigma_f}{\Sigma_t}$ and this can be either a fission neutron, with probability $\frac{\nu\Sigma_f}{\Sigma_s+\nu\Sigma_f}$ or a scattering neutron, with probability $\frac{\Sigma_s}{\Sigma_s+\nu\Sigma_f}$.

B.6 Feedback

To perform a calculation with feedback, two steps must be taken. First an extra output file is generated by adding the keyword FEEDBACK to the input file. This will signal Tripoli to create a power output file, which is named the same as the input file with the suffix ".power". In this file the power profile is written for every time interval. The score for the power profile has to be defined by the user and is marked with the name FEEDBACK.

Secondly the option -F <feedback_script> must be given to Tripoli. This script will be used to calculate the feedback, it has a few tasks:

- Read the power output file
- Create input for feedback code
- Run feedback code
- Read output of feedback code
- Create input for Tripoli

These scripts have to be tailor made, allowing the user to choose his/her favourite thermal-hydraulics code.

B.7 General improvements

Time limit	When crossing a time boundary a neutron is now stopped at this boundary. This will ensure that there are no scores outside the time domain. Also the fatal error for particles scoring outside a time mesh has been replaced with a warning.
Warning for deleting particles	A warning message has been added when a particle is removed, because it is out of bounds.

Logging function	A logging function has been added which prints all interaction of neutrons and precursors.
Extended mesh with time mesh	Added the possibility of a time mesh to the extended mesh.

B.8 Memory improvements

Removed memory leak by:

geutil.c	free ge_surf_tab_info.ge_surf
	free ge_volu_tab_info.ge_volu
	free ge_combi_tab_info.ge_combi_type
t4dispatch.cc	update_read_data t4material_list.free_all()
t4pctdoublediffentry.cc	re_init_repartition_function
t4main.cc	delete t4_njoy_dictio
njpctnucleus	delete xsfile
	delete njoy_interface.dictio.entry
t4geomTripoli.cc	added constructor re_initialise
t4matlist.cc	added free_all, which deletes: compositions,materials, nuclei, volunumlist, fissile_volume_list, cell_list, needed_nuclei_id
t4baseunit.cc	delete[] unitstring
t4pctnmultndiff.cc	delete total_discrete_diff_section
t4compolist	delete materials
t4nucleus	delete nucleus_name
	delete atom
t4interdensity.cc	re_init_repartition_function
t4njoy	added destructor
njoyio.cc	delete buffer_for_xdr
t4read.cc	composition_list=0
	delete ponderation_list
	delete colllocus_list
t4xsabsorption.cc	delete total_discrete_diff_section

Summary

In this thesis a new method for the analysis of power transients in a nuclear reactor is developed, which is more accurate than the present state-of-the-art methods. Transient analysis is important tool when designing nuclear reactors, since they predict the behaviour of a reactor during changing conditions, such as a control-rod movement, induced by an operator, or an accident scenario.

The current methods for transient analysis apply deterministic solvers to calculate the neutronic response, but not only do these deterministic solvers always need to discretise the problem, they also often apply more fundamental approximations such as homogenization or diffusion. Therefore a stochastic method is needed, which only has a statistical uncertainty.

A challenge for applying the Monte Carlo method on transient analysis, is the different types of particles which must be simulated in a single simulation. It is common for a neutronics calculation to simulate the prompt neutrons, however the simulation of precursors is new. The main challenge when simulating precursors lies in their long lifetime, which is seconds, whereas the prompt neutrons have a lifetime of microseconds. This has been solved by dividing the transient problem into time intervals and forcing the precursors to produce a delayed neutron is all time intervals. This will ensure prompt-neutron chains in all intervals.

The prompt neutrons will form prompt-neutron chains, which poses another challenge for the Monte Carlo method. The chain length of these prompt neutrons varies a lot and the can also split into many branches, which averages out in a real power reactor, but is difficult to simulate on a computer. To improve the chain-length statistics, a new variance reduction technique is developed, which dictates that a single neutron will emerge from each interaction. This emerging neutron can be the result of a scattering event or a fission event.

Finally, a dynamic simulation scheme, which can run in parallel, has been devised, simulating all time intervals consecutively. Also, a method for sampling the initial conditions is created and this scheme is implemented in a purpose-built Monte Carlo code and in a general-purpose Monte Carlo code. The two codes have been tested in several cases and they behave as expected, agreeing with deterministic method where expected, but deviating when the deterministic methods are no longer valid.

In order to simulate realistic transients in a power reactor, feedback has to be taken into account. To achieve this, the dynamic Monte Carlo method has been coupled to a thermal-hydraulics code, using an explicit scheme. The results of this coupled simulation are compared to a state-of-the-art coupled diffusion calculation in a NURISP-benchmark calculation and the results agree well, except for a small deviation towards the end. The exact source of this deviation should be further investigated, but the difference is small, when realising that there is already a deviation in the steady-state results.

The results demonstrate the feasibility of performing a fully dynamic transient analysis, using only Monte Carlo for the neutronics part of the calculation. Transients can now be calculated, with detailed modelling of any complex geometry and with continuous energy, which is especially useful for newly developed reactor types and one-of-a-kind research reactors.

Samenvatting

Dit proefschrift beschrijft een nieuwe rekenmethode voor het analyseren van vermogenstransities in een nucleaire reactor, die nauwkeuriger is dan de beste huidige methodes. Het analyseren van transities in een nucleaire reactor is belangrijk tijdens het ontwerp, omdat deze het gedrag van een reactor beschrijven wanneer de toestand van de reactor veranderd, bijvoorbeeld wanneer een reactoroperator een regelstaaf laat bewegen of in het geval van een ongeval.

De huidige rekentechnieken voor transitieanalyse maken gebruik van deterministische methodes om de neutronica te beschrijven, maar het nadeel van deze technieken is niet alleen het gebruik van discretisatie, maar er worden vaak nog fundamentelere benaderingen gemaakt, zoals homogenisatie en diffusie theorie. Daarom is het nuttig als er een stochastische methode beschikbaar is, die slechts een statistische onzekerheid heeft.

Een van de uitdagingen voor het toepassen van de Monte Carlo methode op transitieberekeningen is dat de verschillende tijdschalen waarin de verschillende deeltjes opereren in één berekening moeten worden gesimuleerd. Het is gebruikelijk om prompte neutronen in een neutronica berekening te simuleren, maar de simulatie van moederkernen is nieuw. De grootste uitdaging voor het simuleren van moederkernen is hun lange levensduur, die in de secondes loopt, terwijl prompte neutronen een levensduur hebben van microsecondes. Deze tijdschalen zijn verenigd door het transitieprobleem op te delen in tijdsintervallen en iedere moederkern wordt geforceerd om in ieder tijdsinterval een nakomend neutron te genereren. Hierdoor worden er in ieder tijdsinterval ongeveer evenveel prompte neutronenkettingen geïnitieerd, wat op zijn beurt weer leidt tot gelijkmatige statistiek voor alle intervallen.

De prompte neutronen zullen kettingen vormen, hetwelk ook een uitdaging vormt voor de Monte Carlo methode. De kettinglengte van de prompte neutronen varieert sterk, met vele mogelijke vertakkingen. Dit alles middelt in een echte vermogensreactor uit, maar is moeilijk te simuleren op een computer. Om de statistiek in de kettinglengte te verbeteren is er een nieuwe variantiereductietechniek ontwikkeld, die voorschrijft dat er één enkel neutron uit een interactie voortkomt. Dit is danwel het botsingsneutron danwel een splijtingsneutron.

Tot slot is er een dynamisch simulatie schema ontwikkeld, wat ook parallel kan rekenen en wat alle tijdsintervallen achtereenvolgens simuleert. Dit schema is, samen met een methode om de initiële deeltjes te selecteren, geïmplementeerd in een speciaal gecreëerd dynamisch Monte Carlo programma en in een algemeen Monte Carlo programma. De beide programma's zijn aan vele testen onderworpen en ze gedroegen zich zoals verwacht: soms hetzelfde resultaat als de deterministische methodes, maar een ander resultaat op het moment dat de deterministische benadering niet meer van toepassing is.

Bij een realistische transitie in een vermogensreactor is terugkoppeling van groot belang. Om dit te kunnen simuleren is de dynamische Monte Carlo methode gekoppeld aan een

thermo-hydraulicacode, met behulp van een expliciet koppelingsschema. De resultaten van deze gekoppelde simulatie zijn vergeleken met de resultaten van een state-of-the-art gekoppelde diffusieberekening gedaan op een NURISP-benchmark en de resultaten komen grotendeels overeen. Slechts naar het einde van de berekening toe is er een kleine deviatie tussen de resultaten. De precieze oorzaak van deze deviatie moet verder worden onderzocht, maar als men realiseert dat er al grote verschillen zijn in de stationaire berekening, is dit slechts een kleine deviatie.

De resultaten demonstreren dat het mogelijk is om een volledig dynamische transitieberekening uit te voeren, waarbij alleen de Monte Carlo methode wordt gebruikt voor het neutronica gedeelte van de berekening. De gevolgen van transities kunnen nu accuraat worden berekend, met een gedetailleerde beschrijving van elke complexe geometrie en met gebruik van continue energie voor de neutronen. Deze eigenschappen zijn speciaal nuttig voor nieuw te ontwikkelen reactortypes en unieke onderzoeksreactoren.

Acknowledgements

Here I would like to thank everyone who has contributed in the creation of this thesis. First of all I would like to thank my promotor Tim van de Hagen and copromotor Eduard Hoogenboom and section head Jan Leen Kloosterman. With the three of them they have guided me through my four years of PhD. Especially the close collaboration with Eduard has been invaluable. I have really appreciated our weekly meetings and enjoyed our trips for the NURISP project meetings and the conferences. They helped me a great deal and contributed to the timely finish of this thesis.

Also I would like to thank the people at the Tripoli development team at CEA. You have always been a great help to me, when I was implementing the new method in the Tripoli code (François-Xavier) and when I was trying to run the code (Emeric). Thanks for the hospitality at Saclay (Cheikh, Jean-Christophe, Fausto) and thanks for the fast assistance when I was not in France.

Another thanks goes to the people at KIT for the help with my coupling efforts. Not only did they share their tips and tricks for coupling two codes with me (Aleksandar, Manuel), and kindly shared the SubChanFlow code (Uwe), they also helped a great deal with the validation of the results of the coupled calculation (Javier, Victor).

Also I like to acknowledge the SARA computing centre for allowing me to run some calculations on the LISA cluster. Also I like to thank all groups of the HPC11 cluster for allowing me to use their processors via the guest queue. Thanks a lot, without those I would still be calculating.

Big thanks also goes to all the colleagues from PNR. The nice atmosphere in the group was for me an important argument to start this PhD and it did not disappoint me. Thanks for all to nice coffee breaks, drinks in “het koepeltje”, drinks late at night, excursions and random talks. Special thanks go to Stavros for warning me, Karoly for exchanging complaints, Frank for the football and political talks, Stuart for “learning” me English, Luca for my introduction to the European network of Italians in nuclear, Zoltan for the little nerdy scripts, Gert Jan for being another ancient-one, Christophe for the always surprising stories, Norbert for the rowing stories, Jurriaan for the rare smoke, Wim ha collega, Jozsef for all my questions regarding real reactors, Danny for the help with computing, Martin for thermal-hydraulic questions, Dick for the help with the ict, Peter for the cycling talks, August for the cluster talks, Dimitrios for the drinks, Ming for the different view and Ine for making sure everything went smooth and making the group such a good place to work.

Also I was happy to welcome four bachelor students into the PNR Monte Carlo team. Mark, Aldo, Fedde and Tim, it was a pleasure to guide you through the worlds of Nuclear, Monte Carlo simulations and computational physics.

Finally I am like to thank my supportive family, you’ve been always interested and supportive, which is the best encouragement one can ask for. Most of all I would like to thank Anne, for all her mental support, encouragements and love.

List of Publications

B. L. Sjenitzer. Temperature dependent Monte Carlo simulation: Thermalization. Technical report, Delft University of Technology / Commissariat à l'Énergie atomique, 2009.

B. L. Sjenitzer and J. E. Hoogenboom. A Monte Carlo method for calculation on the dynamic behaviour of nuclear reactors. In *Proceedings of International Conference SNA + MC*, Tokyo, 2010.

B. L. Sjenitzer and J. E. Hoogenboom. Implementation of the dynamic Monte Carlo method for transient analysis in the general purpose code Tripoli. In *Proceedings of M&C conference*, Rio de Janeiro, 2011a.

B. L. Sjenitzer and J. E. Hoogenboom. Variance reduction for fixed-source Monte Carlo calculations in multiplying systems by improving chain-length statistics. *Annals of Nuclear Energy*, 38:2195–2203, 2011b.

B. L. Sjenitzer and J. E. Hoogenboom. Possibilities and efficiency of long-time kinetic and dynamic Monte Carlo calculations. Technical Report D1.1.4, NURISP, 2011c.

B. L. Sjenitzer and J. E. Hoogenboom. General purpose dynamic Monte Carlo with continuous energy for transient analysis. In *Proceedings of International Conference on the Physics of Reactors 2012, PHYSOR 2012: Advances in Reactor Physics*, volume 1, pages 812–825, 2012.

B. L. Sjenitzer and J. E. Hoogenboom. Dynamic Monte Carlo method for nuclear reactor kinetics calculations. *Nuclear Science and Engineering*, Accepted for publication.

J. E. Hoogenboom and B. L. Sjenitzer. Extentions of the MCNP5 and TRIPOLI4 Monte Carlo code for transient analysis. In *Submitted to International Conference SNA + MC*, Paris, 2013.

B. L. Sjenitzer, J. E. Hoogenboom, Victor Sanchez-Espinoza, and Javier Jiménez Escalante. Coupled transient analyses using Monte Carlo. *Nuclear Engineering and Design*, To be submitted.

Curriculum Vitae

Personal Details

Name:	Bart Sjenitzer
Date of birth:	19-04-1981
E-mail:	Bart@Sjenitzer.nl
Nationality:	Dutch

Employment and Education

2009–2013	**PhD researcher** at Delft University of Technology *Tasks* include: – The development and implementation of the new Monte Carlo method – Implementation of the new method in a commercial computer program – Coupling different computer codes – Teaching classes for Master and Bachelor students – Supervising thesis projects of students – Writing papers for scientific journals and EU-reports – Presenting at international conferences – Collaborating in a multi-disciplinary, international team – High performance computing on the national supercomputer SARA – Combining Stochastic and deterministic methods
2001–2009	**Master and Bachelor Degree in Applied Physics**, Delft University of Technology Specialty: Computer modelling of neutron transport Master's thesis: "Variance Reduction using the Correcton method in Monte Carlo Simulation" Bachelor's thesis: Neutron moderation in a rotating disc
2004–2008	**International athlete (rowing)** with official status of Dutch Olympic Committee *Tasks included*: – 14 Workouts per week – Developing and planning yearly training schedules – Team work under stressful conditions – Performing under pressure – International races at the highest level
2000–2001	**Electrical Engineering**, Delft University of Technology
1999–2000	Work and travel in Australia
1993–1999	**St. Ignatius Gymnasium**, high school, graduated in 2 extra subjects

Awards and Achievements

2010	Best Student Paper Award Award for best student paper and presentation at the International conference "Supercomputing on Nuclear Application and Monte Carlo" in Tokyo
2004–2007	Rowing: Competitor at the World Championships (2006, 2007) Competitor at the World Cup Final (2006, 2007) Three time Dutch National Champion (2005, 2006, 2007) Silver medal at FISU World Championships (2004) Finalist at Henley Royal Regatta (2004)

Extracurricular activities

2012	**IYNC Track Chair: Reactor Physics Track** The International Youth Nuclear Congress is organized once every two years, It is a large gathering of young nuclear professionals and the responsibilities of a track chair is to create an international review team and evaluate the submissions for the reactor physics track.
2009–2012	**Core group Dutch Young Generation** The Dutch Young Generation (DYG) is the young generation of the Netherlands Nuclear Society (NNS). The DYG organizes technical visits, lectures etc. to increase the exchange of knowledge and experience among the members. The DYG is part of the European Nuclear Society Young Generation Network (ENS-YGN).
2010	**Chair Dutch Young Generation** The chair of the DYG, has to organize the meetings of the DYG and manage the contacts with the NNS, European network and the public.
Additional training	Radiological Health Physics level 3 Media training Scientific writing A summer school on high fidelity modeling of nuclear reactors A Summer school on the EPR