Department of Applied Physics

Development and applications of multi-physics capabilities in a continuous energy Monte Carlo neutron transport code

Ville Valtavirta

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 66/2017
VTT SCIENCE 150

Development and applications of multi-physics capabilities in a continuous energy Monte Carlo neutron transport code

Ville Valtavirta

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall E of the school on 19. 5. 2017 at 13:00.

Aalto University
School of Science
Department of Applied Physics

Supervising professor
Prof. Filip Tuomisto, Aalto University, Finland

Thesis advisor
Adj. Prof. Jaakko Leppänen, VTT Technical Research Centre of Finland Ltd

Preliminary examiners
Dr. Kevin T. Clarno, Oak Ridge National Laboratory, United States of America
Assist. Prof. Dan Kotlyar, Georgia Institute of Technology, United States of America

Opponent
Dr. Mark DeHart, Idaho National Laboratory, United States of America

Aalto University publication series
DOCTORAL DISSERTATIONS 66/2017

VTT SCIENCE 150

ISBN 978-952-60-7377-4 (printed)
ISBN 978-952-60-7376-7 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-7376-7

ISBN 978-951-38-8530-4 (printed)
ISBN 978-951-38-8529-8 (pdf)
ISSN-L 2242-119X
ISSN 2242-119X (printed)
ISSN 2242-1203 (pdf)
http://urn.fi/URN:ISBN:978-951-38-8529-8

Unigrafia Oy
Helsinki 2017

Finland

NORDIC ECOLABEL
441 697
Printed matter

A? Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Ville Valtavirta

Name of the doctoral dissertation
Development and applications of multi-physics capabilities in a continuous energy Monte Carlo neutron transport code

Publisher School of Science

Unit Department of Applied Physics

Series Aalto University publication series DOCTORAL DISSERTATIONS 66/2017

Field of research Engineering Physics

Manuscript submitted 28 November 2015 **Date of the defence** 19 May 2017

Permission to publish granted (date) 8 March 2017 **Language** English

□ **Monograph** ☒ **Article dissertation** □ **Essay dissertation**

Abstract

The accurate modeling of nuclear reactors is essential to the safe and economic operation of current and future reactor types. Due to the physical feedback effects between the neutron distribution in the reactor core and the fuel and coolant material temperatures and densities, all of the three fields need to be solved simultaneously in order to obtain a solution for the behavior of the nuclear reactor. This coupled problem has traditionally been solved using a two-stage approach for the neutron transport. In this approach, the neutron interaction properties of different parts of the reactor core are first averaged using accurate neutron transport methods in a series of small scale simulations. These averaged quantities are then used in a simplified neutron transport model to obtain the full-core solution in a reasonable time allowing for multiple iterations between the solvers of the different physical fields.

In recent years, advances in methodology as well as in computational power have made it possible to apply the accurate neutron transport methods directly to the full core problem, enabling modeling of the important feedback effects in more detail than has been possible with the two-stage approach. Moving to direct modeling of the coupled problem with the accurate neutron transport methods initially developed for the lattice calculations in the first part of the two-stage approach requires various changes to the neutron transport methods.

In this thesis, capabilities required to solve the neutron transport part of the coupled problem are developed and implemented in the continuous energy Monte Carlo neutron transport code Serpent 2. Coupled calculation schemes were developed and implemented both for internally and externally coupled calculations for three different simulation types: Steady state calculations, burnup calculations and time-dependent transient calculations.

The new coupled calculation capabilities were applied to the effective fuel temperature approximation, in which the complex fuel temperature distribution in a fuel rod or a fuel assembly is replaced with a single effective temperature for the neutron transport calculations. The new capabilities made it possible to estimate the effects of this approximation by providing an accurate reference solution using realistic temperature distributions provided by an internally or externally coupled fuel temperature solver.

Keywords Neutron transport, Monte Carlo, multi-physics, coupled calculation, reactor analysis

ISBN (printed) 978-952-60-7377-4 **ISBN (pdf)** 978-952-60-7376-7

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2017

Pages 172 **urn** http://urn.fi/URN:ISBN:978-952-60-7376-7

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Ville Valtavirta

Väitöskirjan nimi
Multifysiikkamenetelmien kehitys ja sovellukset jatkuvaenergisessä Monte Carlo neutronikuljetuskoodissa

Julkaisija Perustieteiden korkeakoulu

Yksikkö Teknillisen fysiikan laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 66/2017

Tutkimusala Teknillinen fysiikka

Käsikirjoituksen pvm 28.11.2015 **Väitöspäivä** 19.05.2017

Julkaisuluvan myöntämispäivä 08.03.2017 **Kieli** Englanti

□ **Monografia** ☒ **Artikkeliväitöskirja** □ **Esseeväitöskirja**

Tiivistelmä

Tiivistelmä
Neutronien käyttäytymisen ja reaktorisydämen materiaalien lämpötilan ja tiheyden välillä olevat fysikaaliset takaisinkytkennät tekevät reaktorimallinnuksesta kytketyn ongelman: reaktorisydämen tehojakauman ratkaisemiseksi täytyy samanaikaisesti ratkaista myös jäähdytteen ja polttoaineen lämpötila- ja tiheysjakaumat. Menetelmiltä vaaditun lyhyen laskenta-ajan vuoksi neutronien käyttäytyminen ratkaistaan tyypillisesti käyttäen kaksivaiheista laskentaketjua. Ketjun ensimmäisessä osassa reaktorisydämen eri alueille lasketaan tarkoilla menetelmillä joukko neutronien vuorovaikutuksia kuvaavia keskiarvoistettuja parametreja. Toisessa vaiheessa näitä keskiarvoistettuja suureita käytetään yksinkertaistetussa mutta nopeassa koko reaktorisydämen kattavassa laskentamallissa.

Viime vuosina laskentamenetelmien ja käytettävissä olevan laskentatehon kehitys on tehnyt mahdolliseksi soveltaa ensimmäisen vaiheen tarkkoja neutronilaskentamenetelmiä suoraan kokosydänongelmaan. Näitä menetelmiä ei kuitenkaan ole alun perin kehitetty kytketyn ongelman ratkaisemiseen, mikä vaatii uusien toiminnallisuuksien kehittämistä ja toteuttamista.

Tässä väitöskirjassa kehitettiin kytketyn ongelman ratkaisemiseen vaadittavat laskentamenetelmät Serpent 2 nimiseen jatkuvaenergiseen Monte Carlo neutronikuljetuskoodiin. Työn valmistuttua Serpent 2 pystyy ratkaisemaan kytketyn ongelman neutroniikkaosuuden osana laskentajärjestelmää, jossa toiset Serpentiin sisäisesti tai ulkoisesti kytketyt ratkaisijat vastaavat lämpötila ja tiheyskenttien ratkaisusta. Kytkettyyn laskentaan kehitetyt menetelmät soveltuvat ajasta riippumattomien laskujen lisäksi palamalaskentaan ja aikariippuvaan transienttilaskentaan.

Väitöskirjassa kehitettyjä laskentamenetelmiä käytettiin esimerkkisovelluksena ydinpolttoaineen efektiivisen lämpötilan tutkimiseen. Nipputason neutronikuljetuslaskuissa tyypillisesti käytettävä yksinkertaistus on korvata ydinpolttoaineen monimutkainen lämpötilajakauma yhdellä efektiivisellä lämpötilalla, joka valitaan siten että mallinnuksesta saatavat tulokset vastaisivat mahdollisimman hyvin oikealla lämpötilaprofiililla saatavia tuloksia. Uudet menetelmät mahdollistivat tarkan vertailuratkaisun laskemisen oikeaa lämpötilaprofiilia käyttäen ja efektiivisten lämpötilamallien vaikutusten arvioinnin.

Avainsanat Neutronikuljetus, Monte Carlo, multifysiikka, kytketyt ongelmat, reaktorimallinnus

ISBN (painettu) 978-952-60-7377-4 **ISBN (pdf)** 978-952-60-7376-7

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2017

Sivumäärä 172 **urn** http://urn.fi/URN:ISBN:978-952-60-7376-7

Preface

I want to express my appreciation for everything that my instructor Dr. Jaakko Leppänen has provided me with. Not only did he offer me a stimulating topic for my postgraduate research and funding to follow through with it, he has also both created a Monte Carlo code that is extremely nice to work with and been a key factor in bringing together a group of young motivated researchers, who also happen to be extremely nice to work with.

On this note, I would like to thank all of my colleagues at VTT who have served as my surrogate family for some 36.25 or, more recently, 37.5 hours a week. I want to extend special thanks to the people who have accompanied me on, and whom I have met in, various conference and Serpent UGM travels. Many of the most memorable moments of my PhD candidacy are from those trips.

For the support I have received at VTT, I also want to thank my team leader Petri Kotiluoto and the former Head of Research Area Timo Vanttola.

I want to thank four specific doctors of technology, Jaana Vapaavuori, Maria Pusa, Tuomas Viitanen and Eric Dorval, whom I have had the pleasure to work with and whose journeys towards their PhD's have inspired me on my own.

I am grateful for my two professors from the Department of Applied Physics in Aalto University: Professor Emeritus Rainer Salomaa, who encouraged me to pursue a PhD and Professor Filip Tuomisto who has acted as my supervisor during these years.

I was delighted to hear that Dr. Mark DeHart from Idaho National Laboratory agreed to serve as my opponent in the defense of this thesis and I thank him for accepting this task. I also want to express my gratitude for my two pre-examiners Dr. Kevin T. Clarno from Oak Ridge National Laboratory and Dr. Dan Kotlyar from the Georgia Institute of Technology. Their comments and suggestions were of great value in the final phase of preparing my dissertation.

My parents have had an enormous role in feeding my thirst for knowledge all through my childhood and education. I fondly remember our frequent visits to the local library that we often conducted by bike. The access to a vast amount of interesting books and the encouragement I received for reading has surely shaped me from an early age.

I am grateful to my wife Anna-Maija, whose musings on how it is preferable to do difficult things and feel like one gets a bit smarter every day than to do easy things and feel like one gets a bit more dumb each day bring me solace whenever I'm trying to tackle advanced topics myself.

Finally, I want to thank our dog Susu who helped me in the final phase of my thesis work by taking a vacation and living in Vaasa with my parents, so that I could focus on the writing process.

This work has been funded by the Finnish Academy project NUMPS, the Finnish Research Programme on Nuclear Power Plant Safety SAFIR and the Fortum Foundation.

List of publications

This thesis is based on the following original publications which are referred to in the text as **I**–**VI**. The publications are reproduced with kind permission from the publishers.

I V. VALTAVIRTA, T. VIITANEN and J. LEPPÄNEN, ”Internal neutronics–temperature coupling in Serpent 2”, *Nuclear Science and Engineering*, **177**, pp. 193-202 (2014).

II V. VALTAVIRTA, V. TULKKI, J. LEPPÄNEN and T. VIITANEN, ”The universal fuel performance code interface in Serpent 2”, *In proc.* TopFuel 2013, Charlotte, NC, Sept. 15–19, 2013, (2013).

III V. VALTAVIRTA, T. IKONEN, T. VIITANEN and J. LEPPÄNEN, ”Simulating fast transients with fuel behavior feedback using the Serpent 2 Monte Carlo code”, *In proc.* PHYSOR 2014, Kyoto, Japan, Sept. 28–Oct. 3, 2014, (2014).

IV V. VALTAVIRTA, M. HESSAN and J. LEPPÄNEN, ”Delayed neutron emission model for time dependent simulations with the Serpent 2 Monte Carlo code – first results”, *In proc.* PHYSOR 2016, Sun Valley, ID, May 1–5, 2016, (2016).

V V. VALTAVIRTA, J. LEPPÄNEN and T. VIITANEN, ”Coupled neutronics–fuel behavior calculations in steady state using the Serpent 2 Monte Carlo code”, *Annals of Nuclear Energy*, **100, Part 2**, pp. 50–64 (2017).

VI V. VALTAVIRTA and J. LEPPÄNEN, ”Estimating the effects of homogenized fuel temperature in group constant generation using Serpent 2”, *Annals of Nuclear Energy*, **105**, pp. 79–94 (2017).

Author's contribution

Publication I: Internal neutronics–temperature coupling in Serpent 2

The author designed and implemented the internal analytic fuel temperature solver and coupled calculation scheme described in the article, chose the test problem, conducted all the simulations, analyzed the results and was the main author of the article.

Publication II: The universal fuel performance code interface in Serpent 2

The author designed and implemented the general fuel performance code interface described in the article as well as the coupled calculation routines needed for the external coupling, chose the test case, conducted the simulations, analyzed the results and was the main author of the article.

Publication III: Simulating fast transients with fuel behavior feedback using the Serpent 2 Monte Carlo code

The author designed and implemented the internal coupling with the FINIX fuel behavior module as well as the solution procedure for the coupled time-dependent simulation described in the article. The author also chose the time-dependent test problem, conducted the simulation, analyzed the results and was the main author of the article.

Publication IV: Delayed neutron emission model for time dependent simulations with the Serpent 2 Monte Carlo code – first results

The author developed the mesh-based precursor tracking in collaboration with the third author of the article and implemented the methodology described in the article for source generation, precursor tracking and conducting time-dependent simulations with delayed neutrons. The author chose the test case, conducted the simulations, analyzed the results and was the main author of the article.

Publication V: Coupled neutronics–fuel behavior calculations in steady state using the Serpent 2 Monte Carlo code

The author designed and implemented the coupled calculation schemes for external and internal coupling described in the article. The author chose the test case, conducted the simulations, analyzed the results and was the main author of the article.

Publication VI: Estimating the Effects of Homogenized Fuel Temperature in Group Constant Generation using Serpent 2

The author implemented the Stochastic Implicit Euler depletion scheme with thermal feedback into Serpent. The author also designed and implemented the coupled calculation routines Serpent required for the externally coupled burnup calculation. The author chose the test case, conducted the simulations, analyzed the results and was the main author of the article.

Contents

List of symbols and abbreviations

Symbols:

E	: Neutron energy.
v	: Neutron speed, scalar variable.
$\vec{v}$	: Neutron velocity, vector variable
$\hat{\Omega}$	: Direction of neutron movement: $\vec{v} = \hat{\Omega} v$
m_n	: Neutron mass.
ϕ	: Scalar neutron flux
σ_r^n	: Microscopic cross section for reaction r with material or nuclide n.
Σ_r^n	: Macroscopic cross section for reaction r with material or nuclide n.
$\dot{r}_x$	: Reaction rate density for reaction x.
$\dot{R}_x$	: Reaction rate for reaction x.
$k_{\text{eff.}}$	: Effective multiplication factor
s	: Neutron track length between two interactions.

Abbreviations:

BWR	: Boiling water reactor
CDF	: Cumulative distribution function
IQS	: Improved Quasi-Static
JEFF	: Joint European Fission and Fusion File
JFNK	: Jacobian-free Newton–Krylov
LEU	: Low enriched uranium
LFR	: Lead-cooled fast reactor
LWR	: Light water reactor
MC	: Monte Carlo
MOX	: Mixed oxide
PDF	: Probability density function
PWR	: Pressurized water reactor
SFR	: Sodium-cooled fast reactor
SIE	: Stochastic implicit Euler
TMS	: Target Motion Sampling
VVER	: A PWR design originally developed in the Soviet Union

1. Introduction

Power production in nuclear fission reactors is based on a stable neutron chain reaction, in which the power and neutrons are supplied by the neutron-induced fission of heavy elements, typically uranium-235. The combination of nuclear power's low lifetime greenhouse gas emissions for electricity generation (see Chapter 7 in [1]) and the strong load-following capabilities of both current and future reactor types [2] ensures that nuclear power can be used in conjunction with intermittent energy sources such as wind and solar power to tackle the continuing problem of increasing global greenhouse gas emissions.

In order to operate nuclear reactors in a safe and economic manner, it is imperative to understand accurately the behavior of the reactor, both in normal operating conditions and in accident scenarios. As it is both difficult and expensive to conduct experiments with full-sized reactors, such experiments are complemented with experiments executed in test reactors as well as with numerical simulations. Since power generation is based on the neutron chain reaction, accurate modeling of the neutron distribution in the reactor core is of great importance.

Safe reactor design relies on using the natural laws of neutron interactions with matter to design the reactor in such a way that physical laws ensure the existence of a *negative power feedback* in the reactor. This means that an increase in the reactor power should bring about feedback effects that oppose the increase in reactor power and vice versa. In thermal reactors, the negative power feedback effect is based on designs in which an increase in the fuel or coolant temperature will affect the power level in a decreasing manner.

The strong coupling between the power distribution and the material temperatures in the reactor core makes the reactor safe to operate, but difficult to model: For example, solving the power distribution in the reactor using a neutron transport tool requires a solution for the fuel and coolant temperature distributions as input data. However, the temperature distributions can only be solved if the power distribution is already known. This is the two-way coupled multi-physics problem in nuclear reactor modeling. Accounting for the feedback effects in the neutron transport solution transforms the neutron transport problem from a straightforward linear problem into a complex coupled non-linear one.

Traditionally, the coupled problem has been too costly to be solved using high-fidelity neutron transport methods. Instead, a two-level calculation approach has been used, in which the neutron interaction properties of a fuel assembly are first evaluated by an accurate lattice transport code for several different state points. These properties are then parametrized as the function of different thermal-hydraulic states and used in a coarser full-core simulator that only sees the neutron interactions through the pre-generated data.

Although the daily work of multi-physics calculations will continue to be conducted with simplified neutronics models in the future, application of the accurate continuous energy Monte Carlo neutron transport codes to multi-physics problems has increased in popularity during the past decade (see, e.g., [3], [4], [5], [6], [7]). Despite their high computational cost the Monte Carlo codes hold several advantages over simpler nodal solvers: The same code can naturally model different reactor and fuel types, no major approximations are made in the neutron physics models and the best possible interaction physics data can be used in the calculations. These features are important especially in new reactor concepts, in which the accuracy of the traditional two-level approach cannot be established by comparisons to experimental data.

Most Monte Carlo neutron transport codes have not originally been developed with multi-physics problems in mind, which means that the increased interest in the topic has necessitated the implementation of various new models in Monte Carlo codes. The models mainly focus on treatment of the complex temperature and density fields that appear in the multi-physics problem and on operation of the code in a coupled manner as a part of a larger multi-physics framework consisting of several solvers for the different physics fields.

The main topic of this thesis is the development and implementation of several new multi-physics and coupled calculation capabilities for the continuous energy Monte Carlo code Serpent 2 [8]. The implemented capabilities were demonstrated in various coupled calculation scenarios involving the coupling between the neutron flux and the fuel temperature. This work is described in Publications **I**–**VI** and summarized shortly in Sections 7 and 8. The physical and theoretical background of the multi-physics problem in the context of Monte Carlo neutron transport is briefly described in Sections 2–6.

Although the demonstration calculations in Publications **I**–**VI** focus on the neutronics–fuel temperature coupling, the capabilities developed in this thesis are by no means limited to the fuel behavior feedback.

2. Basics of neutron interactions with matter

This section will present and define some basic concepts that are required in order to understand the Monte Carlo neutron transport described in Section 3 as well as the description of the operation of thermal nuclear reactors given in Section 4.

2.1 Cross sections

Free neutrons travel in straight lines[1] at a velocity v determined by their kinetic energy E. These straight neutron tracks can be interrupted by interactions with matter. There are several possible interactions between neutrons and atomic nuclei, some of which will be introduced in Section 2.3. The probability for a neutron to interact with a specific nuclide via a specific reaction type depends on the nuclide, the reaction type and the neutron energy. These interaction probabilities are referred to as nuclear *cross sections*. The next sections will briefly describe the concept of macroscopic and microscopic nuclear cross sections.

2.1.1 Macroscopic cross sections

The probability for a neutron to interact with matter is strongly dependent on the neutron's energy and the composition of the material. The material total macroscopic cross section $\Sigma_t(E)$ describes the probability of a neutron with energy E to interact with the material when it travels a differential track-length ds in the material:

$$\frac{dP(E)}{ds} = \Sigma_t(E). \tag{1}$$

Based on this definition, the unit of macroscopic cross section is 1/cm. The macroscopic total cross section can be divided into components: As the total interaction probability must be the sum of the interaction probabilities with each nuclide in the material composition, we can divide the material total macroscopic cross section into parts based on the target nuclide. For example, the total macroscopic cross section for light water H_2O will be the sum of the macroscopic cross sections of ^{1}H and the oxygen isotopes:

$$\Sigma_t^{\mathrm{H_2O}} = \Sigma_t^{^1\mathrm{H}} + \Sigma_t^{^{16}\mathrm{O}} + \Sigma_t^{^{17}\mathrm{O}} + \Sigma_t^{^{18}\mathrm{O}}. \tag{2}$$

The quantity $\Sigma_t^{^1\mathrm{H}}$ then represents the interaction probability with hydrogen-1 on the differential track-length ds. The nuclide-wise interaction probabilities can be further divided into interaction probabilities for the different reaction types:

$$\Sigma_t^{^1\mathrm{H}} = \Sigma_\mathrm{s}^{^1\mathrm{H}} + \Sigma_\mathrm{c}^{^1\mathrm{H}}, \tag{3}$$

i.e. the probability for the neutron to interact with ^{1}H in any way on the differential track-length is the sum of the probabilities for a scattering reaction and a capture reaction with ^{1}H on the track-length. The material total macroscopic cross section is

[1] The effect of gravity on neutron trajectories can be ignored in reactor applications.

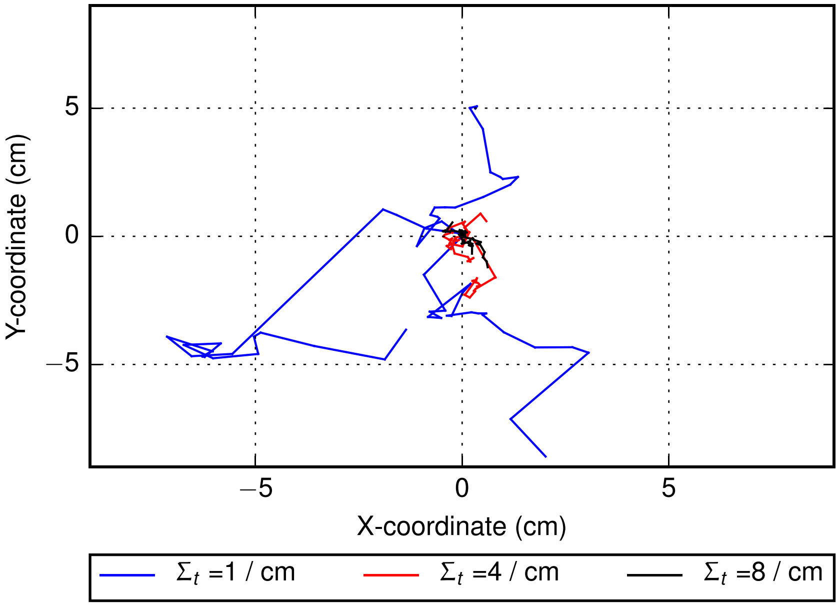


Figure 1. 2D projections of 3D particle tracks in a medium consisting of a pure isotropic scatterer with three different material total macroscopic cross sections. Increasing macroscopic total cross section leads, on average, to shorter track-lengths between scattering interactions. Three particle tracks were sampled for each Σ_t starting from (0,0) with 20 interactions for each particle track. Due to the projection of the tracks to the XY-plane, the segments traveling significant distances in the Z-direction are shortened.

then the sum of the nuclide total macroscopic cross sections, which themselves are the sums of the different reaction cross sections for the nuclide:

$$\Sigma_t = \sum^{\text{nuclides}} \left[\sum^{\text{reactions}} \Sigma_r^n \right]. \tag{4}$$

As the material total macroscopic cross section describes the interaction probability over an infinitesimal track-length, Eq. 1 can be used to derive an expression for the probability density function for the track-length (or path-length) between two interactions in an infinite homogeneous medium. The derivation can be found in most reactor physics textbooks such as [9, 10], and the resulting probability density function (PDF) has the simple form:

$$\mathbb{P}(E, s) = \Sigma_t(E) e^{-s\Sigma_t(E)}, \tag{5}$$

where s is the track-length between two interactions. This PDF can be used to obtain the expected value for the track-length, which is called the *mean-free-path* of neutrons of energy E in the material:

$$I(E) = \int_0^\infty s\mathbb{P}(E,s)\mathrm{d}s = \int_0^\infty s\Sigma_t(E)e^{-s\Sigma_t(E)}\mathrm{d}s, \tag{6}$$

where the integral can be evaluated using integration by parts to yield

$$I(E) = \frac{1}{\Sigma_t(E)}. \tag{7}$$

The expected value for the track-length at energy E is thus the inverse of the material macroscopic total cross section at energy E. The effect of Σ_t on the track-length distribution is illustrated in Fig. 1, which shows track-lengths sampled for particles in three purely scattering materials with different macroscopic cross sections. Larger Σ_t corresponds on average to shorter track-lengths.

The macroscopic cross section (interaction probability) of each nuclide is a macroscopic property in the sense that it scales linearly with the number density of the nuclide. The interaction probability over an infinitesimal track-length in helium gas at 2 mol/dm^3 is twofold that in helium gas at 1 mol/dm^3. The total macroscopic cross section of a certain nuclide is thus not a physical constant. In order to compare the interaction probabilities of different nuclides, it is useful to divide the macroscopic cross section by the nuclide density to obtain the *microscopic cross section* of the nuclide, which is independent of the material nuclide density and is an intrinsic property of the nuclide.

2.1.2 Microscopic cross sections

The microscopic cross section (denoted with a lowercase sigma σ) is a quantity associated with a certain nuclide (not material) and nuclear reaction type that reflects the likelihood of a neutron to undergo that reaction with that nuclide. The microscopic cross section can be obtained by dividing the macroscopic cross section of the nuclide with the number density of the nuclide for a certain material:

$$\sigma_t^n = \frac{\Sigma_t^n}{N^n} \Leftrightarrow \Sigma_t^n = \sigma_t^n N^n \tag{8}$$

This relation also applies to the individual reaction cross sections, not only to the nuclide total cross section. Therefore the microscopic cross sections can also be divided into the different reaction modes. For example, for ^{1}H it stands similar to Eq. 3 that

$$\sigma_t^{^1\mathrm{H}} = \sigma_\mathrm{s}^{^1\mathrm{H}} + \sigma_\mathrm{c}^{^1\mathrm{H}}. \tag{9}$$

The unit of microscopic cross sections is that of area, but due to the small values of the microscopic cross sections the unit barn (b) is typically used:

$$[\sigma] = 1\,\mathrm{b} = 1 \times 10^{-24}\,\mathrm{cm}^2. \tag{10}$$

2.1.3 Energy dependence of cross sections

The microscopic cross sections, and by extension their macroscopic counterparts, are heavily dependent on the incoming neutron energy. Figure 2 shows the microscopic cross sections of some important reaction modes in nuclear reactors. Due

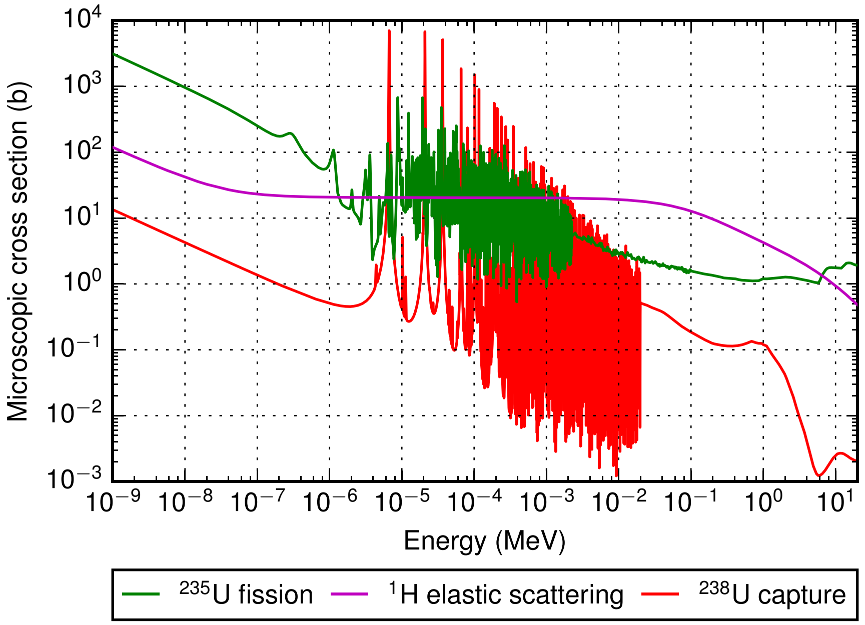


Figure 2. Example of microscopic cross sections of some important reaction modes with two uranium isotopes in nuclear fuel. Data from JEFF-3.1.1 at 300 K temperature.

to the wide variation in both the neutron energy and the interaction probabilities typically seen in reactor applications, it is customary to use logarithmic scales for both axes in cross section plots. Three different energy regions can be established from the plot: The region in the left side of the figure, where all of the three cross sections have a smooth shape, is called the *thermal energy region*. The thermal energy region extends from low energies to approximately 1 eV. At higher energies, we can see high interaction probability peaks in the cross sections at specific energies. At these high probability *resonance energies*, the energy obtained by the compound nucleus produced in the interaction corresponds to the difference of two excitation states of the nucleus, which greatly increases the probability of the interaction resulting in the *resonance peaks* seen in the cross section data. The energy region, where the peaks can be seen in the cross sections, is called the *region of resolved resonances*. At even higher energies, it becomes difficult to resolve the resonance peaks from each other using experimental techniques and only the average smoothed interaction probability can be measured. This high energy region is called the *region of unresolved resonances*.

2.1.4 Temperature dependence of cross sections

As the interaction probability between a neutron and a target nucleus depends on the incoming neutron energy, the interaction probabilities (cross sections) are also temperature dependent. The origin of this temperature dependence is related to the thermal motion of the atoms in the medium that the neutron is travelling in: In a material at a temperature of zero Kelvin, the constituent atoms would experience no thermal motion, i.e. they would be completely stationary targets for any incoming neutrons. In such a case a neutron with a kinetic energy E_0 would experience an interaction probability of $\sigma^{0\,\mathrm{K}}(E_0)$, where $\sigma^{0\,\mathrm{K}}$ now refers to the interaction probability, when the target nucleus is part of a material at zero Kelvin or when the target nucleus is at rest.

At a higher temperature, the constituent atoms of the material are not at rest. Instead, they are subject to random thermal motion based on the Maxwellian distribution at the temperature of the material. Therefore, when a neutron has a kinetic energy of E_0 in the laboratory frame, the energy of the incoming neutron will be slightly different, $E_0 + \Delta E(T)$, in the rest frame of the target nucleus. The additional term $\Delta E(T)$ is a stochastic term that can be positive or negative depending on the velocity and movement direction of the target nuclide. As the interaction probability in the target at rest frame follows the zero Kelvin cross section, it is easy to see that for an interaction between the neutron and a single target nucleus, the interaction probability at an elevated temperature of T will be

$$\sigma^{T}(E_{\mathrm{Lab.}}) = \sigma^{0\,\mathrm{K}}(E_{\mathrm{T\ at\ R}}) = \sigma^{0\,\mathrm{K}}(E_{\mathrm{Lab.}} + \Delta E(T)),$$

where $\Delta E(T)$ is again the stochastic term resulting from the Maxwellian based thermal movement of the target neutron and the lower indices ”Lab.” and ”T at R” indicate energies at the laboratory frame and the target at rest frame, respectively.

This temperature effect is seen at all energies, but it is easiest to understand near the resonance energies of the target nucleus. In Figure 3 this effect is illustrated for a neutron interacting with ^{238}U, when the neutron has the laboratory frame energy corresponding to a certain resonance energy of the target nuclide E_0. If the

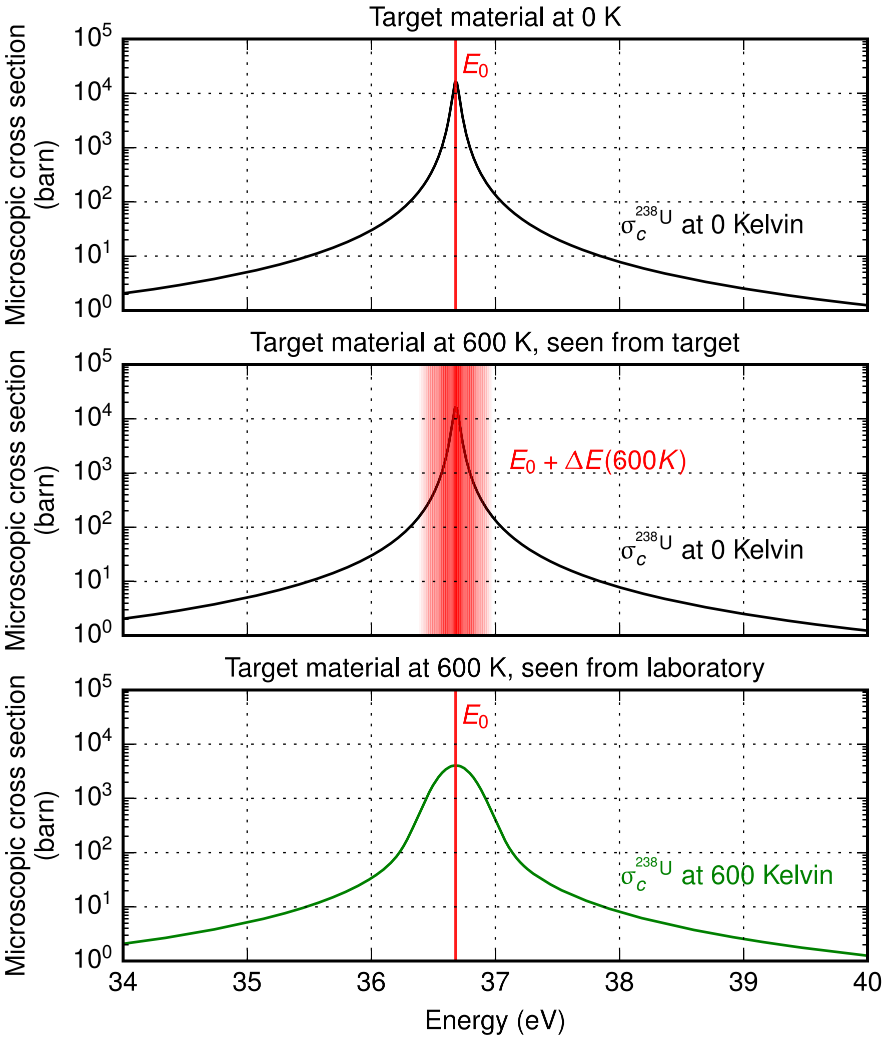


Figure 3. Temperature effect on the microscopic capture cross section σ_c of uranium 238. **Top:** A neutron interacting with uranium 238 at zero Kelvin with energy E_0 that corresponds to the position of the resonance peak has a high probability to undergo radiative capture. **Middle:** If the target nucleus belongs to a material at an elevated temperature (here 600 K), the target nucleus is subject to random thermal motion, which leads to a stochastic modification ΔE(600 K) to the incoming neutron energy in the target at rest frame. **Bottom:** The temperature effect is seen in the laboratory frame as a reduction of the interaction probability at the resonance peak energy and an increase in the interaction probability around the peak energy.

target nuclide is at rest, the interaction probability follows the zero Kelvin cross section for the target nuclide and there is a high probability for the neutron to undergo radiative capture. However, if the target nuclide is in a material at an elevated temperature, the target nuclide is subject to thermal motion following the Maxwellian velocity distribution at the temperature of the material. In the target at rest frame, this is seen as a stochastic modification in the energy of the incoming neutron. The result of this stochastic term is that the energy of the neutron does not necessarily correspond to the resonance energy in the target at rest frame, which, on average, lowers the capture probability. In the laboratory frame this effect of thermal motion of the target is seen simply as a decrease in the capture probability at E_0. At the same time, neutrons that have an energy close to, but not equal to, E_0 in the laboratory frame may have an energy of E_0 in the target at rest frame, which is seen as an increase in the interaction probability at energies around the resonance.

This temperature effect, called the *Doppler broadening* of the resonance peaks, plays a significant role in the control of nuclear reactors and is revisited in Section 4.3.1, when the fuel temperature feedback in thermal reactors is discussed.

2.2 Neutron flux

The objective of reactor physics calculations is to calculate derived results, such as the fission power distribution or atomic transmutation speeds in a certain target, based on the neutron distribution in a nuclear reactor or some other neutronically relevant system. In order to provide the derived quantities, the neutron distribution in the system must first be obtained. Although the energy-dependent neutron density $n(E)$ (neutrons per cm^3) might seem to be a straightforward quantity with which to measure the neutron distribution, the scalar neutron flux

$$\phi(E) = n(E)v(E) \tag{11}$$

is typically a more useful quantity. This is due to the fact that the neutron distribution cannot be measured directly. Rather, reaction rates (reactions per second) or reaction rates integrated over time (total reactions after irradiation of time T) must be measured, and the mathematical expressions for neutron-induced reaction rates contain the product of neutron density $n(E)$ and the neutron speed $v(E)$: The macroscopic cross sections for different reactions give the probability for a neutron with a certain energy to interact on an infinitesimal track-length (Eq. 1). Equation 1 can be rewritten to give the interaction probability during an infinitesimal time increment:

$$\frac{dP}{dt} = \Sigma_t \frac{ds}{dt} = \Sigma_t v, \tag{12}$$

where $v = \frac{ds}{dt}$ is simply the neutron speed. If we now have a constant neutron density of $n(E)$ neutrons per unit volume at energy E moving with a speed $v(E)$ in a material with a material total macroscopic cross section of $\Sigma_t(E)$, we have a reaction rate density of

$$\dot{r}_t = n(E)\frac{dP(E)}{dt} = n(E)\Sigma_t(E)v(E) = \Sigma_t(E)\phi(E) \tag{13}$$

reactions per unit volume per unit time. Physical instruments can only measure reaction rates in a finite volume caused by neutrons at a finite energy range, which

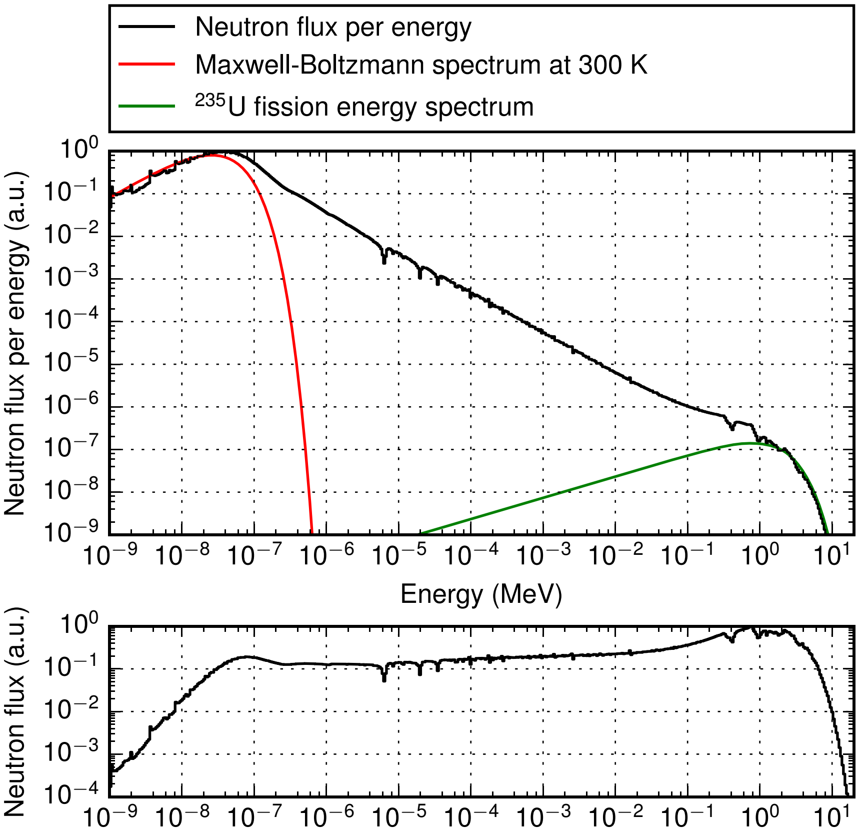


Figure 4. Top: The neutron flux energy spectrum in an LWR-like system shows the typical spectrum shape for thermal reactors. Maxwell-Boltzmann distribution at the moderator temperature is plotted in red, whereas the fission spectrum of ^{235}U is plotted in green. The tallied flux spectrum is normalized to a maximum of 1.0 and the other plotted spectra have been manually scaled to fit the tails of the tallied distribution. **Bottom:** The flux spectrum in the top figure integrated over 500 intervals of equal logarithmic energy width.

can be expressed as a double integral

$$\dot{R}_t = \int_V \int_E \Sigma_t(E)\phi(E)\mathrm{d}E\mathrm{d}V. \tag{14}$$

The previous expression is the total number of all reactions, but typically a specific reaction r of a specific nuclide n is measured

$$\dot{R}_r = \int_V \int_E \Sigma_r^n(E)\phi(E)\mathrm{d}E\mathrm{d}V. \tag{15}$$

Figure 4 shows the scalar flux as a function of energy in a thermal reactor geometry system. The neutron flux was tallied in 500 energy bins with equal logarithmic width using Serpent 2.1.27. For the top figure, the values of each bin were divided by the energy width of the bin while for the bottom figure the values were plotted undivided. Both representations have their merits and are commonly used. The shape of the flux energy spectrum in this figure is typical for thermal reactors. The physical background of the shape of the neutron energy spectrum in thermal systems is discussed in more detail in Section 4.

2.3 Interaction types

Free neutrons can interact with matter through various different interaction types (also referred to as reaction types or reaction modes). This section describes the basics of three interaction types: Scattering reactions, capture reactions and neutron-induced fission reactions. The latter two can be classified as absorption reactions, in which the incident neutron is consumed in the reaction, whereas in scattering reactions the neutron is not consumed, but its energy and direction of movement are altered.

Based on whether the neutron enters the target nucleus or not, the reaction modes can be divided into *potential scattering* and various *compound nucleus reactions*. In the former reaction type, the neutron and the nucleus interact through elastic scattering without the absorption of the neutron into the nucleus, whereas in the latter reaction type the neutron is absorbed into the nucleus, forming a compound nucleus at an excited energy state, which will then decay in one of various ways resulting in one of several different reaction types.

2.3.1 Scattering reactions

Scattering reactions are simple collisions between the neutron and the target nucleus. By definition, at least a single neutron is emitted in the scattering reaction. Based on the scattering kinematics, neutron scattering is further divided into *elastic* and *inelastic* scattering. A third class of reactions, namely (n,xn) reactions, is also introduced here as a group of scattering reactions, although they could be considered to be a reaction type of their own.

Potential scattering is always an elastic scattering reaction, i.e. the initial kinetic energy of the neutron and the atomic nucleus is divided between the neutron and the nucleus. The other possible scattering reactions are elastic compound scattering, inelastic (compound) scattering and (compound) (n,xn) reactions. In elastic compound scattering, the nucleus absorbs the neutron, producing a compound nucleus, which then sheds all of the gained energy by emitting a single neutron and returning to its initial energy state. A single neutron is also emitted in inelastic scattering, but the target nucleus remains at an excited state. In (n,xn) reactions some additional particles are emitted alongside the outgoing neutron. The x in the (n,xn) represents these additional particles and the group of (n,xn) reactions can be divided into multiple reaction types such as (n,2n) and (n,3n), etc., in which multiple neutrons are emitted from the compound nucleus or charged particle reactions such as (n,pn) or (n,αn), in which a proton or an alpha-particle, respectively, is emitted in addition to the neutron.

Inelastic scattering reactions are *threshold reactions*, which means that they have a minimum energy (*threshold energy*) that will be required from the neutron to induce the reaction. The threshold energies of inelastic scattering typically start from tens of kiloelectronvolts and depend on the energy required to excite the compound nucleus. Elastic scattering reactions, on the other hand, are also common at low energies. The (n,xn) reactions are also threshold reactions as energy is required to dislodge additional particles from the target nucleus.

2.3.2 Capture

Capture reactions are a group of compound nucleus reactions, in which the incident neutron is absorbed by the target nucleus in the process. The excited state of the compound nucleus decays through the emission of a charged particle or a photon, leaving the incident neutron as a part of the target nucleus. Based on the exiting particle(s), the capture reactions can be approximately divided into radiative capture (n,γ), where a photon is emitted, and charged particle reactions such as (n,p) and (n,α), in which a proton and an alpha particle are emitted, respectively.

The capture cross sections of several important nuclides are plotted in Fig. 5. In nuclear reactors, the capture of neutrons by different nuclides serves multiple roles: Many neutrons are lost to radiative capture by ^{238}U during the slowing down of the neutrons in thermal reactors. The light hydrogen (^{1}H) in LWRs also captures some neutrons in the coolant/moderator, leading to a need to isotope-enrich the uranium used in the fuel with respect to the fissile ^{235}U isotope. Another approach to the capture by ^{1}H is to use heavy water (D_2O) as the neutron moderator due to the much lower neutron capture probability by deuterium (^{2}H). Nuclides with a large neutron capture cross section are also used intentionally to control the reactivity of the reactor in control elements (silver-indium-cadmium alloy, boron carbide and borated steel), as burnable absorbers co-mixed with the fuel (gadolinium) or as soluble absorbers mixed with the moderator (boric acid) or in burnable fixed control elements (borosilicate glass).

It should be noted that whereas the radiative capture of neutrons by ^{238}U reduces the number of neutrons able to induce fission reactions, it also leads to the production of ^{239}Pu via the following reaction:

$$^{1}_{0}\mathrm{n} + ^{238}_{92}\mathrm{U} \rightarrow [^{239}_{92}\mathrm{U}^{*}] \rightarrow ^{239}_{92}\mathrm{U} + \gamma$$

$$^{239}_{92}\mathrm{U} \xrightarrow[23.45\,\mathrm{min}]{\beta^{-}} {}^{239}_{93}\mathrm{Np} \xrightarrow[2.356\,\mathrm{d}]{\beta^{-}} {}^{239}_{94}\mathrm{Pu}$$

Plutonium-239 is an important fissile nuclide, and a significant amount of the energy production in the later parts of fuel life is due to ^{239}Pu fissions.

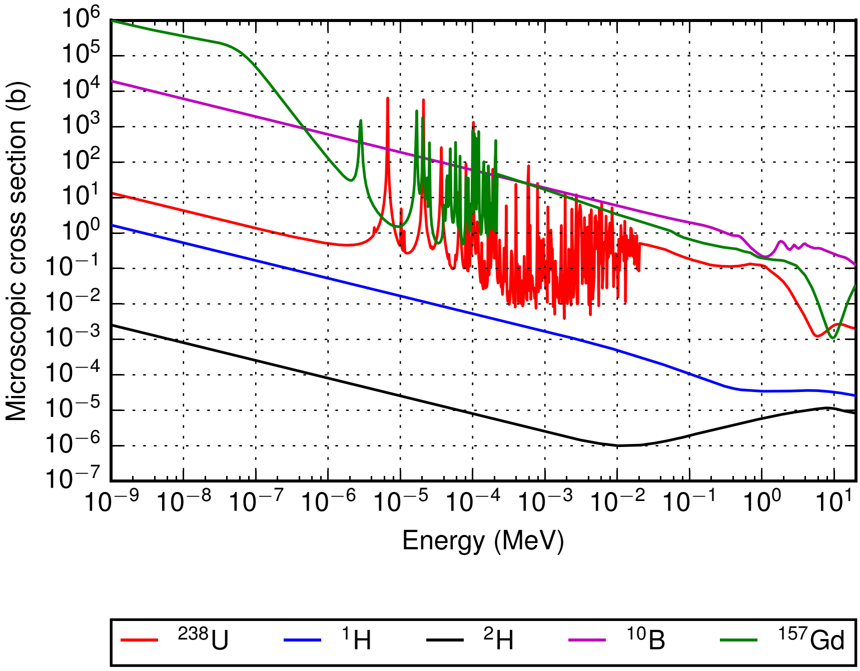


Figure 5. Microscopic capture cross sections of several important nuclides from the point of view of reactor physics. The straight $1/\sqrt{E}$ base shape of the curves is broken by resonances. The cross sections of heavier nuclides exhibit more resonances due to a greater number of possible excitation states for the compound nucleus formed in the neutron absorption.

Table 1. Quantities related to the fission of some interesting fissile and fertile nuclides used in nuclear fuel at two incident neutron energies. Data from JEFF-3.1.1.

		Fissions per absorption σ_f/σ_a		Fission energy release Q (MeV)		Fission neutron release $\bar{\nu}$	
Incident energy:		0.3 eV	3 MeV	0.3 eV	3 MeV	0.3 eV	3 MeV
^{232}Th	(fertile)	0.00	0.82	200	202	2.17	2.29
^{233}U	(fissile)	0.92	0.99	198	198	2.49	2.83
^{235}U	(fissile)	0.81	0.98	202	203	2.44	2.77
^{238}U	(fertile)	0.00	0.98	209	209	2.41	2.79
^{239}Pu	(fissile)	0.60	0.99	206	206	2.85	3.30
^{241}Pu	(fissile)	0.66	0.98	210	210	2.93	3.38

2.3.3 Neutron-induced fission

In neutron-induced fission reactions, the compound nucleus formed by the neutron absorption has sufficient energy to split into typically two smaller nuclei. In the fission reaction of heavy elements, a significant amount of binding energy is released (see Table 1). This released energy is the basis of power production in nuclear fission reactors.

The fissions also release free neutrons at a relatively high energy (in the MeV range). These instantaneously released fission neutrons are called *prompt neutrons* due to their prompt emission. The produced fission fragments have a high neutron-to-proton ratio for their mass number and many of them decay to a more favorable neutron-proton configuration via beta decay. Some of the beta decay events for certain fission fragments can release sufficient energy to knock out a neutron from the atom nucleus. These neutrons, emitted due to the beta-decay of the fission fragments on a broad timescale of milliseconds to minutes after the fission event, are called *delayed neutrons* due to their delayed emission. Although the fraction of delayed neutrons of all neutrons released due to a fission reaction is small, they are very important for the control of the nuclear reactor due to their delayed emission. There is a large number of potential delayed neutron precursor nuclides ([11] lists over 150 nuclides in the fission product range that models predict to emit delayed neutrons) and they are typically grouped to a small number of groups based on their decay constants in the evaluated nuclear data libraries such as ENDF/B and JEFF.

In typical nuclear fuel nuclides, the number of neutrons released in a fission reaction and the subsequent radioactive decay of the fission products is significantly greater than one, the number of neutrons lost in the fission reaction. Table 1 also shows the average number of neutrons produced due to the fission of several fuel isotopes at two different incident neutron energies. At thermal energies, approximately 1.4 – 1.9 neutrons are gained in addition to offsetting the single neutron lost in the fission. The number of produced prompt fission neutrons increases as a function of incident neutron energy due to the addition of energy to the compound nucleus formed by the neutron absorption. This release of fission neutrons makes it possible to produce a stable neutron chain reaction, which is essential for the steady state operation of fission reactors.

All nuclides with an atomic number higher than 90 can be made to fission if the kinetic energy of the incident neutron is sufficiently high. Although all nuclides that can be made to fission are typically called *fissionable*, there is a subset of fissionable nuclides that fission with a high probability regardless of incident neutron energy. These *fissile* nuclides are important for producing a fission chain reaction in nuclear reactors. This can be seen in Table 1: The probability of fission by absorption of 0.3 eV neutrons is very low for ^{232}Th and ^{238}U. Based on this, these two nuclides are not fissile. However, the radiative capture of neutrons by ^{232}Th will lead to the production of fissile ^{233}U and the radiative capture by ^{238}U will produce fissile ^{239}Pu. Nuclides that can be transmuted to fissile nuclides through neutron capture are called *fertile*.

To complete the nomenclature concerning nuclides that can fission, it should be noted that some nuclides, such as ^{240}Pu, are not fissile, i.e. they cannot reliably be fissioned with low energy neutrons, but can support a fast neutron chain reaction. Kelly and Clayton proposed *fissible* as a separate term for nuclides capable of supporting a neutron chain reaction with high-energy neutrons ($>\approx$1 MeV) but not with neutrons at the thermal energy range [12].

2.4 Multiplication factor and reactivity

Balancing the neutron production from fission reactions with the neutron loss to capture reactions and neutron leakage is the basis of neutron-economical design of nuclear reactors: Some of the neutrons released in fission reactions leak out of the reactor or are lost in capture reactions with the materials in the reaction. Other neutrons manage to induce new fission reactions, producing the next generation of fission neutrons. The average number of fission neutrons produced by a single fission neutron in a system is called the *effective multiplication factor* of the system and denoted with $k_{\text{eff.}}$. The effective multiplication factor is a useful quantity, as it is easy to see that a multiplication factor less than unity corresponds to a diminishing chain reaction (the neutron population decreases, the system is *sub-critical*) and a multiplication factor greater than unity corresponds to a growing chain reaction (the neutron population increases, the system is *supercritical*). A multiplication factor of exactly 1 corresponds to a stable chain reaction (the neutron population remains constant, the system is *critical*).

As the multiplication factor in reactor applications is typically close to unity (the power level of the reactor is stable or changing only slowly), it is often easier to work with the *reactivity* of the system

$$\rho = \frac{k_{\text{eff.}} - 1}{k_{\text{eff.}}}, \tag{16}$$

which is typically given in pcm (per cent mille)

$$1\,\text{pcm} = 1 \times 10^{-5} = 0.001\,\%. \tag{17}$$

Based on the multiplication factor it is easy to deduce that the reactivity of a critical system is zero. Negative reactivities correspond to a sub-critical system, whereas positive reactivities correspond to a super-critical system.

3. Monte Carlo neutron tracking

Whereas deterministic neutron transport solvers are based on solving the Boltzmann transport equation for neutrons using several simplifications[2] that are required to obtain a solution, the Monte Carlo approach completely bypasses the transport equation. Instead of modeling the neutron flux as a distributed quantity, individual neutron histories are modeled as a random walk process and the behavior of the neutron population at large is gleaned through statistical estimates obtained for a large number of individual neutron histories.

3.1 Neutron transport as a random walk

The Monte Carlo methods approach the neutron transport problem by simulating the interactions of individual neutrons with high detail from their birth to the eventual absorption or leakage of the neutron. The advantage of the Monte Carlo method is that, even in extremely complex systems, the computational process for modeling the individual neutron histories can be divided into a series of relatively simple steps:

0. Start with a source neutron with a certain position $\vec{r}_0$ and velocity $\vec{v}_0 = v_0\hat{\Omega}_0$.
1. Sample track-length to the next interaction: s.
2. Move the neutron to the interaction site.
3. Sample the interaction that the neutron will experience.
4. Model the reaction
5. If the reaction was an absorption, model the next history. Otherwise continue from 1 with the current neutron position and velocity.

Statistical estimates for various variables such as reaction rates, neutron production or loss rates, neutron kinetic parameters and many others can be collected during the simulation.

The physics enters the random walk through modeling the interactions based on the best possible experimental and theoretical models and basing the random sampling of neutron track-lengths and interactions on physical probabilities — the macroscopic and microscopic cross sections. The following sections will describe the individual parts of the random walk in greater depth.

3.2 Sampling track-lengths

The track-length from one interaction to the next in an infinite homogeneous medium follows the probability density function given in Eq. 5 and can be sampled using the cumulative inverse sampling procedure, in which first the cumulative distribution function is calculated

$$P_{\mathrm{CDF}}(E,s) = \int_0^s \mathbb{P}(E,s)\mathrm{d}s = \int_0^s \Sigma_t(E)e^{-s\Sigma_t(E)}\mathrm{d}s = 1 - e^{-s\Sigma_t(E)}, \tag{18}$$

[2] Typical simplifications are spatial discretization and homogenization, energy discretization and condensation, angular discretization of neutron travel directions and time discretization.

whereafter the inverse of the cumulative distribution function is obtained. Denoting the value of the CDF with ξ, we can derive

$$\begin{aligned}
\xi &= 1 - e^{-s(E,\xi)\Sigma_t(E)} \\
\log\left[1-\xi\right] &= \log\left[e^{-s(E,\xi)\Sigma_t(E)}\right] \\
\log\left[1-\xi\right] &= -s(E,\xi)\Sigma_t(E) \\
s(E,\xi) &= -\frac{\log\left[1-\xi\right]}{\Sigma_t(E)} \\
s(E,\xi) &= -\frac{\log\left[\chi\right]}{\Sigma_t(E)}. \qquad (19)
\end{aligned}$$

Now sampling random numbers from the unit interval $\chi \in [0, 1)$ we can sample track-lengths that follow the PDF in Eq. 5. It is important to note that the derivation of the PDF assumes an infinite homogeneous medium, in which the macroscopic total cross section is independent of location. In reality, Σ_t also depends on the spatial location and will be written as $\Sigma_t(E, \vec{r})$. This means that situations, in which the sampled track length extends over a material boundary require special consideration. There are two approaches for the non-constant Σ_t which are briefly described here.

In the surface tracking algorithm, also sometimes referred to as ray-tracing [13], the track-lengths that extend out of the initial material region are stopped at the first material boundary and a new track-length is then sampled in the upcoming material.

In the Woodcock delta tracking algorithm [14], rejection sampling is used instead. The derivation of the Woodcock delta tracking algorithm may be conceptually easiest to perform via the introduction of virtual interactions[3]: The virtual interactions that have an associated macroscopic cross section of $\Sigma_{\text{virt.}}$ are interactions that do not change the neutron's movement direction or speed in any way. The idea is then to first find out the maximum material total cross section at each incoming neutron energy, also called the *majorant cross section*

$$\Sigma_{\text{maj.},t}(E) = \max \Sigma_t(E, \vec{r}), \qquad (20)$$

where the maximum is taken over the whole geometry. The next step is to "pad" each material total cross section with a different amount of the virtual cross section so that the modified material total cross section is the same throughout the geometry, e.g.

$$\Sigma_t^*(E, \vec{r}) = \Sigma_{i,t}(E, \vec{r}) + \Sigma_{i,\text{virt.}}(E, \vec{r}) = \Sigma_{\text{maj.},t}(E), \qquad (21)$$

where $\Sigma_t^*(E, \vec{r})$ is the padded macroscopic cross section at $\vec{r}$ and $\Sigma_{i,\text{virt.}}(E, \vec{r})$ is the virtual cross section added to the total cross section at $\vec{r}$. After the addition of the virtual cross sections the whole geometry has the same cross section

$$\Sigma_t^*(E, \vec{r}) = \Sigma_{\text{maj.},t}(E), \qquad \text{for all } \vec{r}, \qquad (22)$$

which is independent of the spatial coordinates and can be used to sample the track-lengths. The track-lengths sampled using the majorant cross section will be, on average, shorter than if the local material total cross section would be used. However, the physicality of the simulation is preserved by rejecting a fraction of

[3] The other possibility would be through the concept of rejection sampling.

the sampled interaction points as virtual interactions (see Section 3.4) that do not affect the movement of the neutron in any way. This will conserve the track-length distribution between two non-virtual interactions [15].

3.3 Moving the neutron

In order to move the neutron to the sampled interaction site, only the neutron position has to be updated. If the initial position and direction of the neutron were $\vec{r}_0$ and $\hat{\Omega}_0$, respectively, and the sampled track-length was s, the new neutron position would simply be

$$\vec{r}_1 = \vec{r}_0 + s\hat{\Omega}_0. \tag{23}$$

In time-dependent simulations, the neutron time can also be updated. If the time at the beginning of the track-length was t_0, the new time is simply the previous time incremented with the track-length divided by the neutron speed[4]:

$$t_1 = t_0 + \frac{s}{v_0}. \tag{24}$$

3.4 Interaction sampling

When the neutron has been moved to its interaction site $\vec{r}_1$ the interaction type must be sampled. If the track-length was sampled using delta tracking, it is first necessary to establish whether the interaction will only be a virtual reaction. The probability of a virtual reaction is simply the ratio of the virtual cross section to the padded total cross section

$$P_{\text{virt.}} = \frac{\Sigma_{\text{virt.}}(E, \vec{r}_1)}{\Sigma_t^*(E, \vec{r}_1)} = \frac{\Sigma_{\text{virt.}}(E, \vec{r}_1)}{\Sigma_{\text{maj.},t}(E)}. \tag{25}$$

If a virtual reaction is sampled, the new track-length can be sampled immediately.

If the reaction was not virtual or the track-length was sampled using surface tracking, the target nuclide and reaction mode can be sampled. It should be noted that the macroscopic total cross section is a sum of the macroscopic total cross sections of the different nuclides in the material (Eq. 2) and the cross section of each nuclide is a sum of the individual reaction mode cross sections for that nuclide (Eq. 3). If an interaction has been sampled to occur, the probability for the interaction to be with nuclide n with reaction type r is then the reaction r macroscopic total cross section for nuclide n divided by the material macroscopic total cross section:

$$P(\text{target} = n, \text{reaction} = r, E) = \frac{\Sigma_r^n(E, \vec{r})}{\Sigma_t(E, \vec{r})}. \tag{26}$$

The target and reaction can be sampled simply by sampling a random number from the unit interval ξ and then summing the individual reaction cross sections divided by the macroscopic material total cross section until the sum exceeds ξ (see Algorithm 1).

[4] To be precise, relativistic effects generally have to be accounted for when modeling fast neutrons ($E \approx 10$ MeV and higher).

Algorithm 1. Algorithm for sampling target nuclide and interaction type in Monte Carlo neutron tracking.

```
$\xi = rand(0, 1)$
$a = 0$
**for** $n$ in $0, 1, ..., N_n$:
    **for** $r$ in $0, 1, ..., N_{r,n}$:
        $a = a + \Sigma_r^n(E)/\Sigma_t(E)$
        **if** $a > \xi$: break
    **end for**
    **if** $a > \xi$: break
**end for**
Reaction target nuclide $n$ reaction type $r$.
```

3.5 Modeling interactions

After the interaction has been sampled, the reaction is modeled based on the interaction type. The interested reader is directed to the physics section of the OpenMC Monte Carlo code documentation [16], but some simplified examples of interaction modeling in the context of the random walk are given here:

- Scattering reactions: Update neutron energy and direction based on elastic or inelastic scattering kinematics[5].
- Capture reactions: Terminate the neutron track.
- Fission reactions: Terminate the neutron track and sample a number of fission neutrons. For each fission neutron sample, whether the neutron is prompt or delayed. For delayed neutrons sample the emission time based on the delayed neutron group structure of the fissioning isotope obtained from the nuclear data library. Finally, sample the emission direction and emission energy for each produced neutron.

Sampling in the interaction modeling is performed either based on analytic descriptions of reaction kinematics (e.g. scattering reactions) or on tabulated data (e.g. delayed neutron fractions and fission spectra).

3.6 Tallying results

The results obtained from a Monte Carlo simulation are statistical estimates. There are several ways to calculate these estimates: The simplest way is to use *analog estimators*, which simply means that each time an event of interest is sampled, a score is added to a tally. The final score of the tally at the end of the simulation then represents the probability of that event of interest. For example, we could tally the number of interactions that occur with ^{1}H as the target nucleus simply by adding 1

[5] Separate treatments can be applied to high and low energy regions and for bound scatterers.

to a tally each time ^{1}H is sampled as the target nuclide in the interaction sampling. The probability of adding 1 to the tally would then simply be

$$P(\text{add } 1) = P(\text{target} =^1\text{H}) = \frac{\Sigma_t^{^1\text{H}}}{\Sigma_t}. \tag{27}$$

The disadvantage of analog estimators is the fact that the tally is only incremented when the event of interest is actually sampled. If the probability of the event is low, there might only be a few scores in the tally, which makes the uncertainty associated with the result high. A better way would then be to use an *implicit estimator*, for example the *collision estimator*, in which instead of adding 1 to the tally with the probability of the event, we add the probability of the event to the tally with a probability of 1. In the hydrogen example, we would simply add

$$\frac{\Sigma_t^{^1\text{H}}}{\Sigma_t} \tag{28}$$

to the tally at each interaction, whether or not the sampled target nuclide is actually ^{1}H. The use of implicit estimators results in adding smaller values to the tally more frequently, and via an increased number of samples better statistics are obtained. The alternative to the collision estimator is to use the *track-length estimator*, which will add the probability of the event of interest over the previous track-length s rather than at the current collision:

$$\frac{\Sigma_t^{^1\text{H}}}{\Sigma_t} s. \tag{29}$$

If the track-length extends over multiple material regions, as can occur with the use of delta-tracking, the cross sections in Eq. 29 are not constant over the track-length. In such cases, the track-length must be divided into multiple segments s_i, with each segment lying in a single material region with constant cross section $\Sigma_{r,i}^n$. The track-length estimate can then be summed up through the whole track-length with

$$\sum_i \frac{\Sigma_{t,i}^{^1\text{H}}}{\Sigma_{t,i}} s_i. \tag{30}$$

The track-length estimator makes it easy to obtain statistical estimates in materials that have a small Σ_t and thus a low interaction probability. The collision estimator is scored each time an interaction point is sampled and by default interaction points are rarely sampled into materials with a low Σ_t. The efficiency of the collision estimator in low Σ_t materials can be enhanced by increasing the small material total cross sections via the use of the virtual cross section $\Sigma_{\text{virt.}}$, as the collision estimator can be scored even if the interaction ends up being a virtual one.

4. Operation of thermal reactors

As described in the previous sections, the operation of nuclear reactors is based on the chain reaction of fissions caused by neutrons. The fission neutrons are born with a relatively high kinetic energy (see green curve in Fig. 4). There are two main approaches to reactor design, based on how these fission neutrons are led to induce the next generation of fissions. *Fast reactors* are based on fissions induced by the high-energy (fast) fission neutrons with the least number of other interactions between the birth of the neutrons and the consumption of the neutrons in a new fission event. *Thermal reactors* are based on slowing the high-energy neutrons to low kinetic energies via collisions with the atoms of a *moderator* material, before allowing the neutrons to induce a new fission.

Although multi-physics problems exist both in fast and thermal systems, the applications in this thesis were from thermal systems[6]. For this reason, only the thermal neutron chain reaction and the main temperature feedback mechanisms in a thermal reactor are described in the following.

4.1 Thermal neutron chain reaction

The advantage of slowing down the fission neutrons to thermal energies can be seen in Fig. 6. In order to have a non-decreasing chain reaction, each absorption of a neutron to the fuel must, on average, produce at least one neutron. With the enrichment of the fuel being at or below 5 wt.% this is only possible for neutrons at energies lower than 2–3 eV or higher than 1 MeV. However, as the top part of the figure shows, the absorption probability at high energies is rather low, of the order of 1×10^{-2}/cm. This translates to a mean track-length of 100 cm before an absorption reaction occurs. The probability for the neutron to scatter from a fuel or coolant nuclide before being absorbed to the fuel is high, and each scattering reaction will make the neutron lose energy and move it farther from the high-energy region.

The thermal reactor design embraces the energy loss in scatterings and aims to slow the neutrons down to thermal energies around 0.025 eV, at which both the number of neutrons produced per absorption in the fuel and the absorption probability are high. The potential downside of slowing the neutrons down stems from the fact that should the neutrons return to the fuel material in the resonance energy range, there is a high probability of their being lost in resonance absorption without the production of new fission neutrons.

The thermal neutron chain reaction then relies on ensuring first that a large percentage of the fission neutrons reach the moderator material, and second that the neutrons shed their kinetic energy by scattering in the moderator quickly, with a low probability of returning to the fuel at intermediate energies. The energy loss of the neutrons in scattering reactions is inversely proportional to the mass of the target nucleus. The maximum energy loss (slowing down) is thus achieved in moderator materials consisting of light elements. For this reason, some of the most popular moderator materials include light water (H_2O), heavy water (D_2O) and graphite (C).

[6] The multi-physics capabilities developed in this thesis are applicable to both thermal and fast systems.

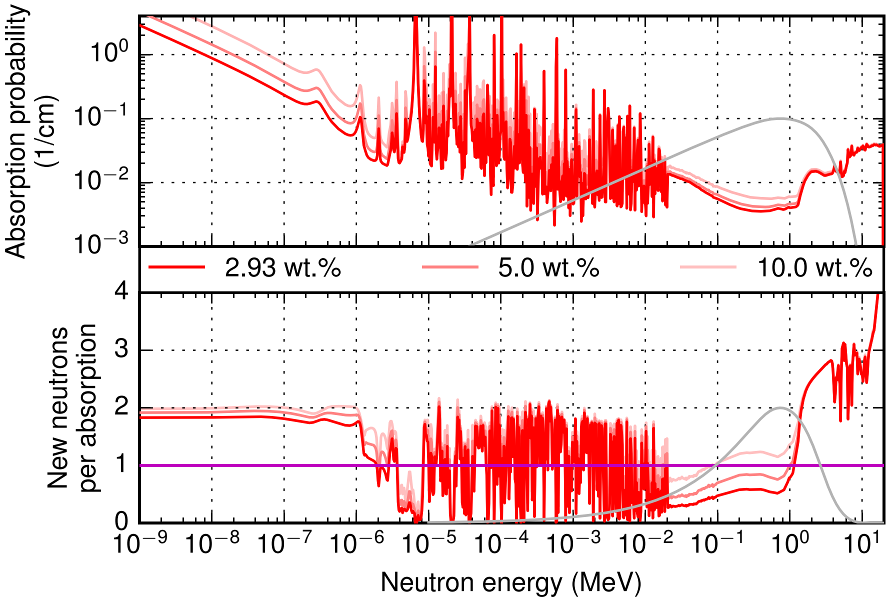


Figure 6. Energy-dependent neutron interaction properties of LEU fuel at three different enrichments. Energy spectrum of fission neutrons is plotted in both figures in grey with arbitrary units. **Top**: The absorption probability per unit distance traveled in fuel. **Bottom**: The average number of neutrons produced by absorption of a single neutron to the fuel.

4.2 Heat transfer in thermal reactors

Figure 7 shows a schematic picture of the different parts of a nuclear reactor relevant for this section. Most of the heat released in the fission reactions is deposited in the fuel material, which is typically contained in long narrow fuel rods (pellets inside a cladding). In the fuel rods the heat conducts through the pellet, the gas gap and the cladding to the cladding outer surface. From the cladding surface the heat is transferred to the coolant. Some of the energy released in the fission reactions is also directly deposited to the coolant, neutron absorbers such as control rods and structural materials by neutrons and photons. The coolant material transfers the heat out of the core and will be either cooled down in heat exchangers providing heat to a secondary coolant circuit (most reactor types), or led directly to a steam turbine (boiling water reactors).

In order to safely operate at constant power there is a simple requirement for the heat balance of the core: The amount of heat produced in the core in unit time must equal the amount of heat transferred out of the core in unit time. Specifically, this can be divided into two important balance requirements:

- The power deposited to the fuel must be equal to the power transferred from

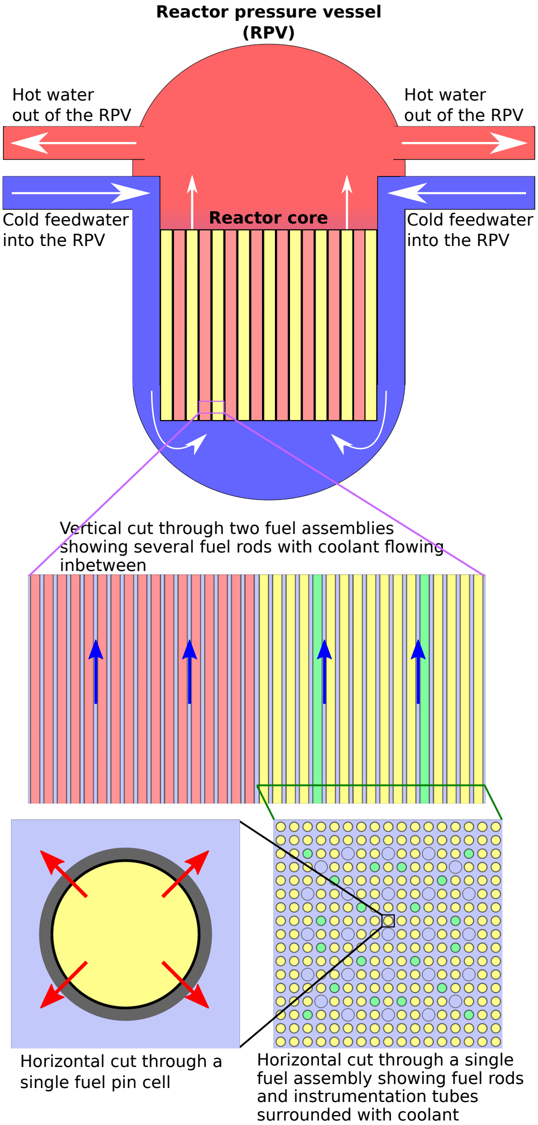

Reactor pressure vessel (RPV)
Hot water out of the RPV
Hot water out of the RPV
Cold feedwater into the RPV
Reactor core
Cold feedwater into the RPV
Vertical cut through two fuel assemblies showing several fuel rods with coolant flowing inbetween
Horizontal cut through a single fuel pin cell
Horizontal cut through a single fuel assembly showing fuel rods and instrumentation tubes surrounded with coolant

Figure 7. Schematic picture of heat transfer in a light water reactor. The heat deposited in the fuel is transferred horizontally to the cladding surface and coolant (red arrows). The coolant transports the heat by flowing vertically through the core (blue arrows). Cold water enters the reactor core from the bottom and hot water is extracted from the top part of the pressure vessel.

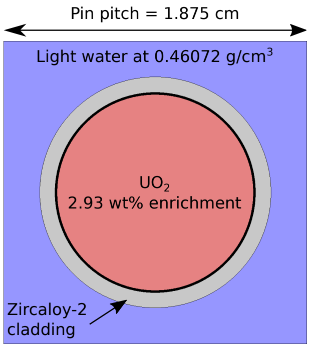


Figure 8. Unit cell of the infinite lattice geometry used to illustrate the thermal feedback effects in an LWR core. Rod and lattice geometry as well as material compositions are from the Peach Bottom 2 benchmark in [17].

the fuel rod to the coolant.

- The power transferred from the fuel rods to the coolant and the power deposited directly to the coolant must be equal to the power transferred out of the core by the coolant.

A failure to fulfil the first requirement will result in an increase in fuel temperature, which, if left uncontrolled, can lead to shattering of the fuel pellet or melting of the fuel. Failure to fulfil the second requirement will lead to an increase in coolant temperature, which can result in reduced heat transfer from fuel rods to the coolant and thus an increase in the fuel temperature.

4.3 The multi-physics coupling in a thermal reactor core

In order to decrease the possibility of sudden adverse increases in the reactor power level, nuclear reactor design makes use of several natural laws to ensure that the power feedback in the reactor is negative in the operating conditions, i.e. that an increase in the power will bring about feedback effects that limit the increase in power. As these feedback effects are based on the physics of the neutron interactions, they are automatic and require no human intervention. The power feedback of a well-designed nuclear reactor is a prime example of passive safety. The three main feedback effects utilized in light water reactor (LWR) design, namely fuel temperature, moderator temperature and moderator density feedback are introduced in the following sections.

Some results tallied from example criticality source (k-eigenvalue) calculations are shown in order to illustrate the different thermal feedback effects in an LWR-like system. The calculation geometry is an infinite 2D lattice of low-enriched uranium fuel rods. The unit cell of the calculation geometry is shown in Fig. 8. The fuel outer radius was 0.605 79 cm, the cladding outer radius was 0.715 01 cm and the cladding thickness was 0.093 98 cm. This simple geometry is infinite both horizontally and axially, which while obviously not realistic, is not very far from the perception of neutrons that are born in the middle of a large LWR core.

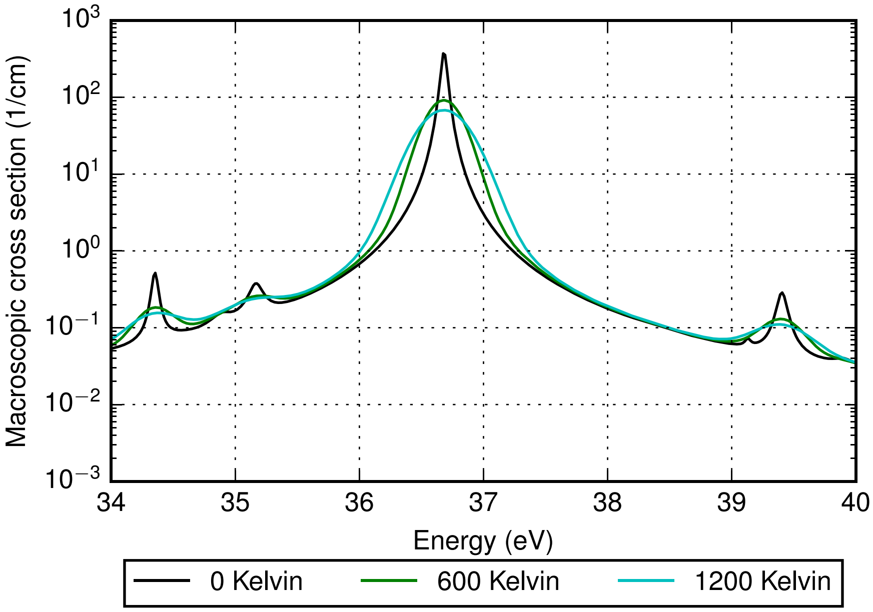


Figure 9. Effect of Doppler broadening on macroscopic capture cross section of low enriched UO_2 fuel at energies close to the ^{238}U resonance at 36.7 eV. Three different fuel temperatures. Data from JEFF-3.1.1.

4.3.1 Fuel temperature feedback

The fuel temperature feedback in LWR reactors is based on the temperature effects on the cross sections of the fuel nuclides, or more specifically, on the Doppler broadening of the resonances in the radiative capture cross section of ^{238}U. An increase in the fission power brings about an increase in the fuel temperature, which affects the neutron interaction probability in the fuel: The probability of neutrons in the resonance energy range being consumed in radiative capture reactions with ^{238}U increases more than the probability of neutrons producing new neutrons in neutron-induced fission reactions. This increases the loss of neutrons in the system and decreases the reactivity. Conversely, a decrease in fuel temperature will decrease the probability of neutrons undergoing radiative capture, which introduces positive reactivity into the system.

Figure 9 shows the Doppler broadening effect on the macroscopic capture cross section of 2.93 wt.% enriched UO_2 fuel with a density of 10.42 g/cm^3 near the 36.7 eV resonance of ^{238}U. The effects of the Doppler broadening are easy to see: The resonance peaks in the cross section become flatter and wider. Figure 10 shows the effect of this Doppler broadening on the neutron flux in the fuel material of the example geometry at the same energy range[7]. For the high resonance, the

[7] For this plot the calculation was normalized to fission power.

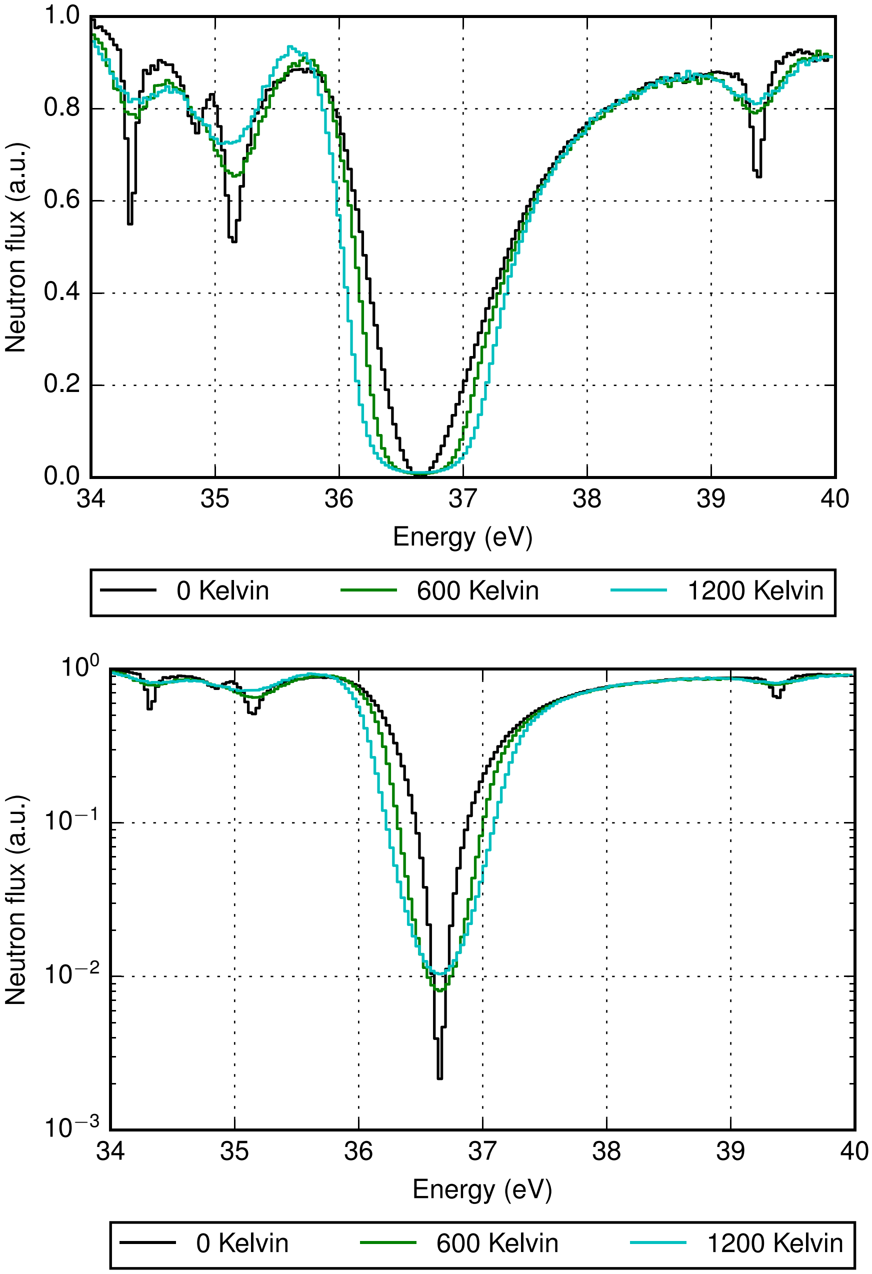


Figure 10. Effect of Doppler broadening on neutron flux in fuel in the example geometry at energies close to the ^{238}U resonance at 36.7 eV. Flux spectrum with three different fuel temperatures calculated with Serpent 2. **Top:** Linear y-axis. **Bottom:** Logarithmic y-axis.

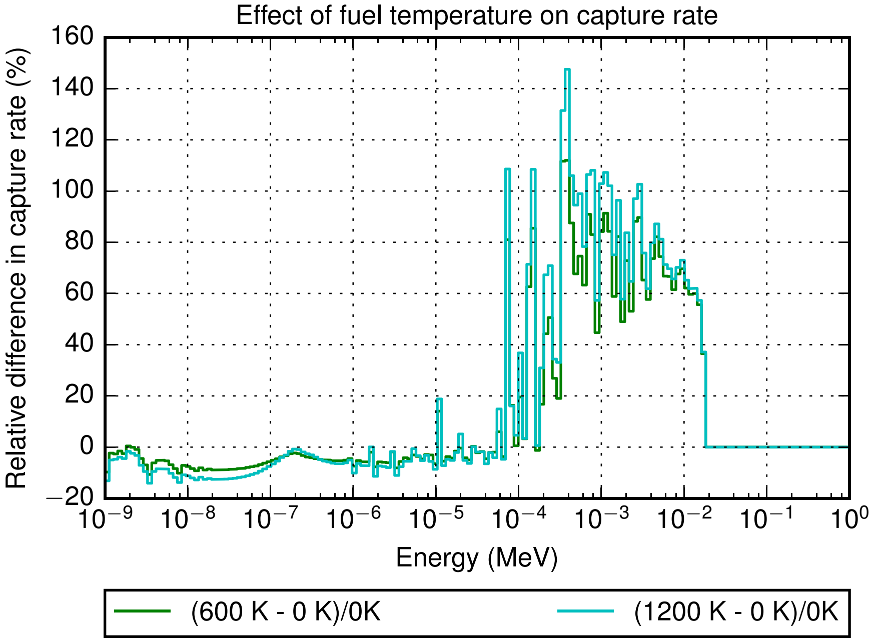


Figure 11. The neutron capture rate in the example geometry was tallied into 200 bins of equal logarithmic width using three different fuel temperatures. This figure shows the relative difference in the neutron capture rate at different energies between the simulations with an elevated fuel temperature and the simulation with a 0 K fuel temperature. Eigenvalue simulation with flux normalization to 1 neutron per second source rate.

flattening of the peaks does not offset the widening of the peak. Neutrons are lost from a wider energy range around the resonance, which increases the total number of neutrons lost to the resonance capture.

An increase in neutron loss in the resonance energy range decreases the number of neutrons that manage to thermalize to the high fission probability thermal energy region and provides a negative reactivity feedback to the system. This is illustrated in Figures 11–13. The flux in these simulations is normalized to a fixed source rate (1 fission neutron per second), which means that the neutron flux at high energies is equal between the simulations. Figure 11 shows the relative difference in the neutron capture rate between simulations with an elevated fuel temperature (600 or 1200 K) and a fuel temperature of 0 K. At the energy range of the unresolved resonances (energies greater than approximately 2×10^{-2} MeV), the capture rate is similar for all fuel temperatures. At the region of resolved resonances, the capture rates are much higher in the systems with elevated fuel temperature, although the differences are smaller at the low energy end of the region. The effect on the flux energy distribution is shown in Figure 12. The increased neutron capture in the

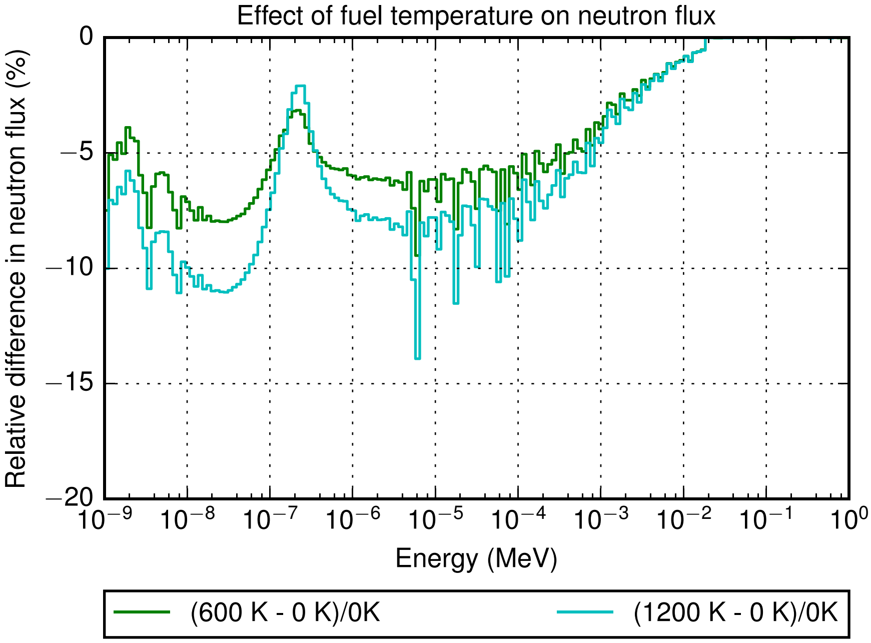


Figure 12. The neutron flux in the example geometry was tallied into 200 bins of equal logarithmic width using three different fuel temperatures. This figure shows the relative difference in the neutron flux at different energies between the simulations with an elevated fuel temperature and the simulation with a 0 K fuel temperature. Eigenvalue simulation with flux normalization to 1 neutron per second source rate.

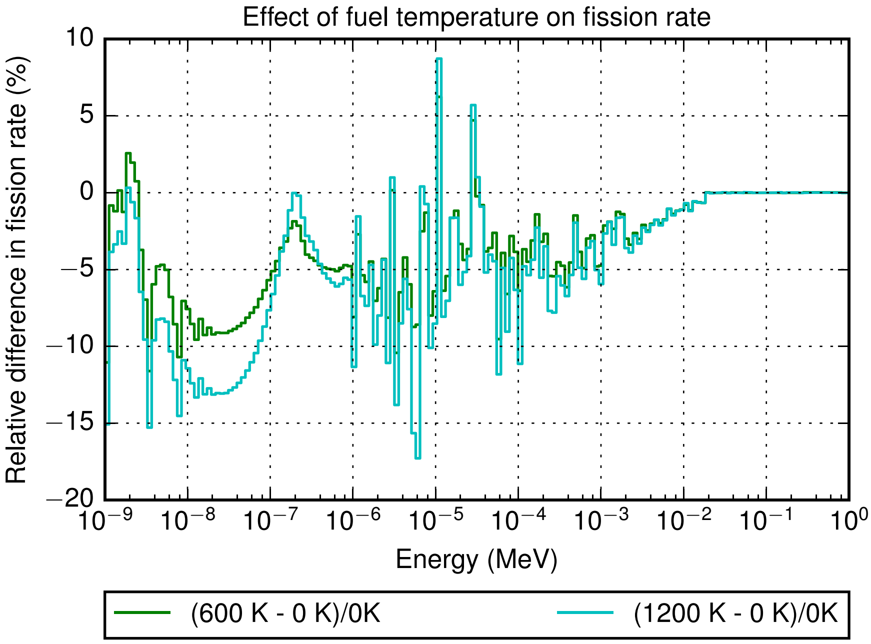


Figure 13. The fission rate in the example geometry was tallied into 200 bins of equal logarithmic width using three different fuel temperatures. This figure shows the relative difference in the fission rateat different energies between the simulations with an elevated fuel temperature and the simulation with a 0 K fuel temperature. Eigenvalue simulation with flux normalization to 1 neutron per second source rate.

Table 2. Tallied capture rate, neutron flux and fission rate in the thermal ($<$ 0.625 eV) and fast ($>$ 0.625 eV) energy ranges in the example geometry using three different fuel temperatures. The flux was normalized to 1 fission neutron per second source rate.

	Capture rate (1/s)		Relative difference compared to 0 K	
	Thermal	Fast	Thermal	Fast
0 K	0.161	0.306	–	–
600 K	0.151	0.345	-6.23 %	12.56 %
1200 K	0.148	0.356	-8.05 %	16.42 %

	Neutron flux ($1/(\mathrm{cm}^2\,\mathrm{s})$)		Relative difference compared to 0 K	
	Thermal	Fast	Thermal	Fast
0 K	7.43	55.9	–	–
600 K	7.01	54.9	-5.58 %	-1.72 %
1200 K	6.92	54.7	-6.79 %	-2.15 %

	Fission rate (1/s)		Relative difference compared to 0 K	
	Thermal	Fast	Thermal	Fast
0 K	0.392	0.143	–	–
600 K	0.366	0.139	-6.42 %	-2.35 %
1200 K	0.359	0.138	-8.43 %	-3.02 %

systems with elevated fuel temperature leads to a steady decrease in the neutron flux at the energy region of resolved resonances. Some low energy effects can be observed around 2×10^{-7} MeV that are explained by the fact that low energy neutrons scattering from fuel at an elevated temperature may gain energy, whereas low energy neutrons neutrons scattering from 0 K fuel material will always lose energy (see the next section for details). The effect on fission rates is shown in Figure 13: The relative differences in the fission rate follow the basic form of the relative differences in the neutron flux with some additional detail in the resolved resonance range due to widening of the resonance peaks in the fission cross section of ^{235}U.

The changes in the reaction rates and neutron flux are summarized in Table 2 for the thermal and fast energy regions. It is easy to see that the increasing fuel temperature leads to an increase in the fast capture rate and a decrease in the thermal capture rate as well as thermal and fast fission rates.

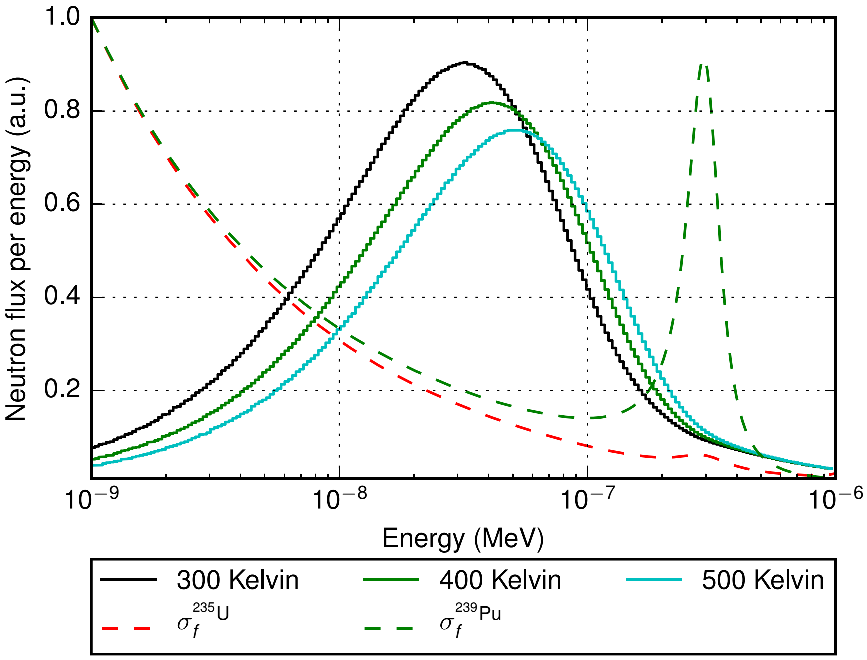


Figure 14. Shifting of the thermal peak with increasing moderator temperature. The shapes of ^{235}U and ^{239}Pu fission cross sections (scaled to 1.0 at 1×10^{-9} MeV) are plotted in the same figure with red and green dashed lines, respectively, using a linear scale for the y-axis.

4.3.2 Moderator temperature feedback

In light water reactors the water in the reactor core serves a dual purpose being both the coolant material and the moderator material. Since an increase in fission power will lead to an increase in the coolant material temperature, the moderating properties of the material will change at the same time as a result of the increased power.

The moderator temperature feedback effect in light water reactors means that an increase in the moderator temperature decreases the system reactivity. This purely temperature-related effect originates from the fact that as neutrons lose their energy in collisions with moderator atoms, they move towards a thermal equilibrium with the moderator material.

In an infinite moderator material without neutron loss or production, the neutron energy distribution would approach the Maxwell-Boltzmann distribution at the moderator material temperature. Although the reactor environment contains both neutron loss and production, which makes the situation more complex, the effect of this thermal equilibrium between low-energy neutrons and the moderator atoms can be seen in the fact that the energy-dependent neutron flux in a thermal reactor is at its highest at neutron energies close to the maximum of the Maxwell-Boltzmann distribution at the moderator temperature (see Fig. 4).

Fig. 14 shows the energy spectrum for the thermal energy range in the example geometry with different moderator temperatures. The movement of the thermal peak to higher energies with increasing moderator temperature can clearly be seen. Inspection of the fission cross section of ^{235}U, which is plotted in the same figure, shows that as the thermal peak moves to higher energies, the fission probability at the thermal peak also decreases leading to a decrease in system reactivity. It should be noted, however, that in the case of the fissile ^{239}Pu isotope, the displacement of the thermal peak to higher energies increases the overlap between the thermal peak and the 0.296 eV resonance peak in the ^{239}Pu fission cross section leading to an increase in the reactivity. For this reason, the moderator temperature feedback may be positive in systems in which the importance of ^{239}Pu fission is significant compared to the importance of ^{235}U fission.

4.3.3 Moderator density feedback

The moderator density feedback is independent of the moderator temperature feedback, although changes in the moderator density may be due to heating up of the material, causing both feedback effects to occur at the same time. Other notable reasons for changes in the moderator density without an associated change in the moderator temperature include sudden changes in the system pressure or boiling of the coolant, which is important in boiling water reactors.

As the density of the moderator decreases, the macroscopic cross sections for the moderator nuclides decrease (see Eq. 8). This results in longer mean-free-paths for the neutrons traveling in the moderator (see Eq. 7), which increases the probability of the neutrons returning to the fuel before they have sufficiently thermalized, leading to a decrease in the system reactivity.

It should be noted that if the moderator material contains a large amount of neutron-capturing nuclides (such as soluble boric acid), a decrease in the moderator material density will also lead to a decrease in the neutron capture rate and may lead to a net positive moderator density feedback effect. This is mainly a possibility in storage and cold-zero-power conditions, in which a large amount of soluble neutron absorber is added to keep the system significantly subcritical.

4.3.4 Thermal mechanical feedbacks

The heating of the different parts of the nuclear reactor such as the fuel pellets, fuel rod cladding and assembly grid spacers brings about geometry changes resulting from thermal expansion. The uneven heating and different thermal expansion coefficients of the materials present in the reactor core results in uneven thermal expansion, which can have major effects on, for example the heat transfer in the fuel rods: During full power operation, the temperature of the nuclear fuel is hundreds of Kelvin higher than the temperature of the rod cladding. This results in the thermal expansion of the fuel pellet being significantly larger than that of the cladding, which leads to a decrease in the width of the gas gap in the fuel rod and an increase in the heat transfer between the fuel pellet and cladding.

The thermal expansion of the cladding can lead to a decrease in the flow area of the coolant (shown in blue in Figure 8) if the thermal expansion of the assembly spacing grids that control the pin pitch cannot counteract the expansion of the cladding. The decrease in the coolant flow area leads to a reduction in the neutron moderation in the fuel lattice, which will lead to a decrease in the reactivity.

4.4 Summary: Multi-physics coupling in thermal reactors

Figure 15 shows a schematic representation of the coupling between different physical fields in a thermal reactor core: The neutron flux distribution determines the power distribution in the reactor core, which acts as a source term for the thermal–hydraulic and thermal–mechanical phenomena in the core producing the temperature and density fields that in part affect the neutron flux distribution.

The cumulative effect of the feedback effects introduced in the previous sections means that the total power feedback effect is negative in all currently operating LWRs in their operating conditions. The implications of this are the following: If a critical reactor operating at the nominal power level is slightly perturbed by the addition of a small amount of positive reactivity, e.g. through a small retraction of the reactor control elements, the power level will initially start to increase slowly. However, as the power increases and the temperatures of the fuel and the coolant materials increase, the feedback effects will decrease the reactivity of the core and, eventually, fully counteract the added reactivity. At this point, the reactor will have reached a new stable power level slightly higher than the initial one.

Conversely, the addition of a small amount of negative reactivity will eventually lead to the stabilization of the system to a new power level slightly lower than the initial one.

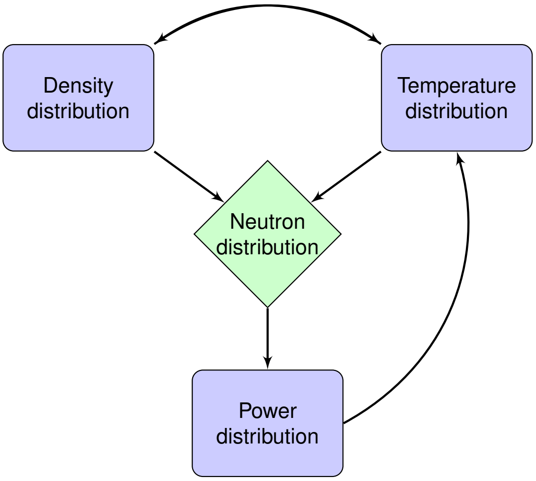


Figure 15. Schematic representation of the multi-physics coupling in a thermal reactor core.

5. Computational methods for multi-physics calculations

The feedback effects presented in the previous section indicate that there is a two-way coupling between the neutron flux and material temperature and density distributions. The result is that if an accurate realistic solution is required for one of the fields it has to be obtained to all of them. For example, one might want to obtain an accurate estimate of the maximum fuel temperature in the geometry presented in Fig. 8 at a specific linear power level and bulk coolant temperature. In order to calculate the temperature distribution inside the fuel rod, an accurate estimate for the fission power distribution inside the fuel rod is required, but as the fuel temperature affects the interaction probabilities in the fuel material, an accurate estimate of the fuel temperature distribution is required in order to calculate the fission power distribution.

This section presents the mathematical formalism of the multi-physics problem and the most common solution techniques used with deterministic solvers. The argumentation borrows heavily from the description of the multi-physics problem in the theory manual of LIME [18], the Lightweight Integrating Multi-physics Environment developed at Sandia National Laboratories.

5.1 Mathematical description of the multi-physics problem

In deterministic applications, a single physics can be described using a system of equations

$$\begin{aligned} f_1(\dot{u}_1, \dot{u}_2, \ldots, \dot{u}_{nu}, u_1, u_2, \ldots, u_{nu}, p_1, p_2, \ldots, p_{np}, t) &= 0 \\ f_2(\dot{u}_1, \dot{u}_2, \ldots, \dot{u}_{nu}, u_1, u_2, \ldots, u_{nu}, p_1, p_2, \ldots, p_{np}, t) &= 0 \\ &\vdots \\ f_{nu}(\dot{u}_1, \dot{u}_2, \ldots, \dot{u}_{nu}, u_1, u_2, \ldots, u_{nu}, p_1, p_2, \ldots, p_{np}, t) &= 0, \end{aligned} \tag{31}$$

where u are the state variables to be solved, $\dot{u}$ are their time derivatives, p are fixed parameters coming from other sources and t is the time coordinate. For example, u could be material temperatures at different locations, p could be heat production as well as material properties at different locations and f would be a set of energy conservation equations used to solve for the unknown variables. We shall denote an array of equations of the previous form with

$$\boldsymbol{f}(\dot{\boldsymbol{u}}, \boldsymbol{u}, \boldsymbol{p}, t) = 0, \tag{32}$$

where each of the bold symbols indicates an array (or a vector) of elements. For steady state applications, the previous equation becomes simpler:

$$\boldsymbol{f}(\boldsymbol{u}, \boldsymbol{p}) = 0. \tag{33}$$

We can now define the multi-physics problem. Let us say that

$$\boldsymbol{f}_1(\dot{\boldsymbol{T}}, \boldsymbol{T}, \boldsymbol{p}_1, t) = 0, \tag{34}$$

does indeed represent a system of energy conservation equations in some spatial domain, written for the material temperature distribution $\boldsymbol{T}$ and its time derivative. The parameters $\boldsymbol{p}_1$ are the heat conductivity distribution $\boldsymbol{\lambda}$, the specific heat capacity distribution $\boldsymbol{c_v}$, the power density distribution $\boldsymbol{q}$ and possibly some additional parameters $\boldsymbol{\mathcal{P}}_1$:

$$\boldsymbol{p}_1 = \{\boldsymbol{\lambda}, \boldsymbol{c_v}, \boldsymbol{q}, \boldsymbol{\mathcal{P}}_1\} .$$

If we assume that the parameters are independent of the $\boldsymbol{T}$ we can solve Eq. 34 as a single physics problem. However, as material thermal properties can be significantly dependent on temperature, we can write separate single physics equations for them:

$$\begin{aligned} \boldsymbol{f}_2(\boldsymbol{\lambda}, \boldsymbol{p}_2) &= 0 \\ \boldsymbol{f}_3(\boldsymbol{c_v}, \boldsymbol{p}_3) &= 0 \end{aligned}$$

For this example, we can reasonably assume that the material properties are mostly dependent on the instantaneous temperature distribution and some additional model parameters:

$$\begin{aligned} \boldsymbol{p}_2 &= \{\boldsymbol{T}, \boldsymbol{\mathcal{P}}_2\} \\ \boldsymbol{p}_3 &= \{\boldsymbol{T}, \boldsymbol{\mathcal{P}}_3\} . \end{aligned}$$

The multi-physics problem then is to acknowledge the interdependence of these parameters by dividing the parameters $\boldsymbol{p}_i$ into the coupled parameters $\boldsymbol{z}_i$, that depend on the solution of the other physics problems, and the independent parameters $\boldsymbol{p}_i$, that are truly independent from the set of different physics problems that is being solved. Each single physics equation system (Eq. 32) is then transformed to the multi-physics form:

$$\boldsymbol{f}(\dot{\boldsymbol{u}}, \boldsymbol{u}, \boldsymbol{z}, \boldsymbol{p}, t) = 0. \tag{35}$$

Keeping to the multi-physics notation, our multi-physics problem for the heat conduction can be now expressed as

$$\begin{aligned} \boldsymbol{f}_1(\dot{\boldsymbol{T}}, \boldsymbol{T}, \{\boldsymbol{\lambda}, \boldsymbol{c_v}\}, \{\boldsymbol{q}, \boldsymbol{p}_1\}, t) &= 0 \\ \boldsymbol{f}_2(\boldsymbol{\lambda}, \boldsymbol{T}, \boldsymbol{p}_2) &= 0 \\ \boldsymbol{f}_3(\boldsymbol{c_v}, \boldsymbol{T}, \boldsymbol{p}_3) &= 0 \end{aligned} \tag{36}$$

In general then, a multi-physics problem can be expressed (as in Eq. 36) using a set of N physics equation systems

$$\begin{aligned} \boldsymbol{f}_1(\dot{\boldsymbol{u}}_1, \boldsymbol{u}_1, \boldsymbol{z}_1, \boldsymbol{p}_1, t) &= 0 \\ \boldsymbol{f}_2(\dot{\boldsymbol{u}}_2, \boldsymbol{u}_2, \boldsymbol{z}_2, \boldsymbol{p}_2, t) &= 0 \\ &\vdots \\ \boldsymbol{f}_N(\dot{\boldsymbol{u}}_N, \boldsymbol{u}_N, \boldsymbol{z}_N, \boldsymbol{p}_N, t) &= 0 \end{aligned} \tag{37}$$

to be solved in a self-consistent manner. We shall denote the group of all $\boldsymbol{u}_i$ with $\boldsymbol{U}$, and the group of all $\boldsymbol{p}_i$ with $\boldsymbol{P}$. As the $\boldsymbol{z}_i$ are functions of some $\boldsymbol{u}_k$ and some $\boldsymbol{p}_k$ we can denote the previous set of equation systems in a slightly simpler form

$$\begin{aligned} \boldsymbol{g}_1(\dot{\boldsymbol{U}}, \boldsymbol{U}, \boldsymbol{P}, t) &= 0 \\ \boldsymbol{g}_2(\dot{\boldsymbol{U}}, \boldsymbol{U}, \boldsymbol{P}, t) &= 0 \\ &\vdots \\ \boldsymbol{g}_N(\dot{\boldsymbol{U}}, \boldsymbol{U}, \boldsymbol{P}, t) &= 0 \end{aligned} \tag{38}$$

which is then written simply as

$$\boldsymbol{F}(\dot{\boldsymbol{U}}, \boldsymbol{U}, \boldsymbol{P}, t) = 0, \tag{39}$$

where $\boldsymbol{F}$ represents the set of physics equations $\{\boldsymbol{g}_i\}$. In real applications, time discretization is applied to Eq. 39 and the system of equations to solve for a specific time step is

$$\boldsymbol{F}(\boldsymbol{U}, \boldsymbol{P}) = 0. \tag{40}$$

In the next sections we shall not write the constant parameter vector $\boldsymbol{P}$ as a part of the previous expression in order to simplify the upcoming equations, but the constant parameter vector will still be included in the equation system.

5.2 Fixed point iteration

One of the simplest approaches to multi-physics coupling is a fixed-point iteration, where the different solvers are executed consecutively keeping the coupled physics constant. This solution method uses a zeroth order approximation for the coupled physics in each of the solvers, but the consecutive execution of the solvers results in the convergence of the coupled solution.

Algorithm 2. Fixed point iteration in a single-physics application.

```
Start with initial guesses $\boldsymbol{u}^{(0)}$.
$k = 0$
while not converged:
    $\boldsymbol{u}^{(k+1)} = G(\boldsymbol{u}^{(k)})$
    $k = k + 1$
end while
```

The solution of a single-physics problem using fixed-point iteration is described in Algorithm 2: The solution scheme simply relies on obtaining a new iterate $\boldsymbol{u}^{(k+1)}$ from the previous iterate $\boldsymbol{u}^{(k)}$ using an operator G. The operator G is application specific, but can be as simple as evaluating the set of physical constraint equations $\boldsymbol{f}$. The fixed point iteration can be applied to multi-physics problems in various specific ways, but one of the simplest possibilities is detailed in Algorithm 3: In this variant, the individual operators G_i are often separate solvers for the separate physical equation systems $\boldsymbol{f}_i$. The iteration itself then only consists of executing the different solvers in a consecutive manner and updating each of the solution fields $\boldsymbol{u}_i$ based on G_i.

Algorithm 3. One approach to the use of fixed point iteration in multi-physics.

```
Start with initial guesses $\boldsymbol{u}_1^{(0)}, \boldsymbol{u}_2^{(0)}, \dots \boldsymbol{u}_N^{(0)}$.
$k = 0$
**while** not converged:
    $\boldsymbol{u}_1^{(k+1)} = G_1(\boldsymbol{u}_1^{(k)}, \boldsymbol{u}_2^{(k)}, \dots, \boldsymbol{u}_N^{(k)})$
    $\boldsymbol{u}_2^{(k+1)} = G_2(\boldsymbol{u}_1^{(k+1)}, \boldsymbol{u}_2^{(k)}, \dots, \boldsymbol{u}_N^{(k)})$
⋮
    $\boldsymbol{u}_N^{(k+1)} = G_N(\boldsymbol{u}_1^{(k+1)}, \boldsymbol{u}_2^{(k+1)}, \dots, \boldsymbol{u}_N^{(k)})$
    $k = k + 1$
**end while**
```

The fixed point iteration is an easy way to couple pre-existing solvers, as the solvers G_i only need to provide the new solution for their individual physics $\boldsymbol{u}_i^{(k)}$ considering the other fields to be constant parameters. One of the main drawbacks of the fixed point iteration is the slow convergence rate of the algorithm [18].

5.3 Newton's method

Algorithms based on Newton's method linearize the non-linear equation system (Eq. 32) using a first order multivariate Taylor's expansion around the current solution point $\boldsymbol{U}^{(k)}$:

$$\boldsymbol{F}(\boldsymbol{U}) \approx \boldsymbol{F}(\boldsymbol{U}^{(k)}) + \boldsymbol{F}'(\boldsymbol{U}^{(\boldsymbol{k})})(\boldsymbol{U} - \boldsymbol{U}^{(k)}), \tag{41}$$

where the derivative of $\boldsymbol{F}$ is simply its Jacobian at the point

$$\boldsymbol{F}'(\boldsymbol{U}^{(k)}) = \boldsymbol{J}(\boldsymbol{U}^{(k)}) = \begin{bmatrix} J_{1,1} & \cdots & J_{1,N} \\ \vdots & \ddots & \vdots \\ J_{N,1} & \cdots & J_{N,N} \end{bmatrix}_{\boldsymbol{U}=\boldsymbol{U}^{(k)}} \tag{42}$$

shown here in the block-matrix format, where the blocks $J_{i,j}$ represent the sensitivities of the constraint equations from physics i ($\boldsymbol{f}_i$) to the state variables from physics j ($\boldsymbol{u}_j$):

$$J_{i,j} = \begin{bmatrix} \frac{\partial f_{i,1}}{\partial u_{j,1}} & \cdots & \frac{\partial f_{i,1}}{\partial u_{j,N_{j,1}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{i,N_{i,1}}}{\partial u_1} & \cdots & \frac{\partial F_{N_{i,1}}}{\partial u_{j,N_{j,1}}} \end{bmatrix}_{\boldsymbol{u}_j=\boldsymbol{u}_j^{(k)}} . \tag{43}$$

The next estimate for the solution of Eq. 40 can be obtained by solving for the next iterate from the linear model:

$$\boldsymbol{F}(\boldsymbol{U}^{(k)}) + \boldsymbol{J}(\boldsymbol{U}^{(k)})(\boldsymbol{U}^{k+1} - \boldsymbol{U}^{(k)}) = 0 \tag{44}$$

$$\boldsymbol{J}(\boldsymbol{U}^{(k)})\boldsymbol{x} = -\boldsymbol{F}(\boldsymbol{u}^{(k)}), \tag{45}$$

from which the next step in the solution space $\boldsymbol{x} = \boldsymbol{U}^{k+1} - \boldsymbol{U}^{(k)}$ can be solved using suitable linear algebra tools.

Algorithm 4. Use of Newton's method in multi-physics.

```
    Start with an initial guesses $\boldsymbol{U}^{(0)}$.
    $k = 0$
1:  while not converged:
2:      $\boldsymbol{x} \leftarrow \boldsymbol{J}(\boldsymbol{U}^{(k)})\boldsymbol{x} = -\boldsymbol{F}(\boldsymbol{U}^{(k)})$
3:      $\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} + \boldsymbol{x}$
4:      $k = k + 1$
5:  end while
```

The application of Newton's method for multi-physics problems is outlined in Algorithm 4. The derivative terms of the Jacobian can be calculated analytically or by finite difference approximation, e.g. by using forward difference:

$$\frac{\partial f_i(\boldsymbol{U}^{(k)})}{\partial u_j} = \frac{f_i(\boldsymbol{U}^{(k)} + \delta\hat{\boldsymbol{e}}_j) - f_i(\boldsymbol{U}^{(k)})}{\delta}, \tag{46}$$

where $\hat{e}_j$ is the unit vector in the direction of u_j.

One downside of the standard Newton's method is the need to construct the full Jacobian matrix. If the dimension of the original problem is large, calculating and storing the matrix can be expensive and prone to error. However, as the next section shows, solution schemes exist that apply Newton's method without the need to construct and store the full Jacobian.

5.4 Jacobian-free Newton–Krylov

Knoll and Keyes [19] provided a thorough overview of the Jacobian-free Newton–Krylov (JFNK) methods, but a short overview is given here as JFNK is a popular approach to multi-physics problems in the case of deterministic solvers. The idea behind JFNK is to solve step 2 of Newton's method (Algorithm 4) using a Krylov-subspace method, in which the Krylov-subspace is spanned by

$$\boldsymbol{K}_j = \text{span}\left(\boldsymbol{r}_0, \boldsymbol{J}\boldsymbol{r}_0, \boldsymbol{J}^2\boldsymbol{r}_0, \ldots, \boldsymbol{J}^{j-1}\boldsymbol{r}_0\right), \tag{47}$$

where

$$\boldsymbol{r}_0 = -\boldsymbol{F}(\boldsymbol{U}^{(k)}) - \boldsymbol{J}\boldsymbol{x}_\mathbf{0} \tag{48}$$

is the residual of Eq. 45 with an initial guess $\boldsymbol{x}_0$. The main attraction of the Krylov-methods in the context of the multivariate Newton's method is the fact that they can be used to solve linear equations of the form

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{49}$$

without the need to explicitly form the matrix $\boldsymbol{A}$. Only a number of matrix-vector products need to be formed for $\boldsymbol{A}$. In the case of Newton's method, the matrix $\boldsymbol{A}$ is the Jacobian in Eq. 45, which means that only a number of vector products of the form $\boldsymbol{J}\boldsymbol{v}_k$ have to be calculated for the Jacobian. These are essentially directional derivatives of the initial equation system, and can be either evaluated analytically or approximated with finite differences:

$$\boldsymbol{J}\boldsymbol{v}_k = \frac{\boldsymbol{F}(\boldsymbol{U} + \delta\boldsymbol{v}_k) - \boldsymbol{F}(\boldsymbol{U})}{\delta}. \tag{50}$$

In this way, the JFNK methods can utilize Newton's method for solving complex multi-physics problems without the need to calculate and store the full Jacobian.

6. Multi-physics with Monte Carlo particle tracking

Coupling the Monte Carlo neutron tracking to other solvers in a multi-physics framework is challenging, since much of the formalism presented in Section 5 does not directly apply to Monte Carlo neutronics; the Monte Carlo neutron tracking does not solve any equation system written for the neutron flux distribution to obtain the solution. Instead the Monte Carlo neutron transport solution can be seen as a general operator $\mathbb{M}$, which takes in the dependent parameters $\boldsymbol{z}_M$, for example material temperature and density distributions, as well as a group of independent parameters $\boldsymbol{p}_M$ and provides a vector of results:

$$\hat{\boldsymbol{u}}_M = \mathbb{M}(\boldsymbol{z}_M, \boldsymbol{p}_M) \tag{51}$$

Here the operation $\mathbb{M}(\boldsymbol{z}_M, \boldsymbol{p}_M)$ refers to executing the Monte Carlo simulation (in k-eigenvalue or fixed source mode) and the result vector $\hat{\boldsymbol{u}}_M$ could generally contain statistical estimates for various result variables such as power distribution or neutron flux distribution tallied in a variety of spatial cells, multiplication factor of the system etc. In the following discussion, however, we shall only include the power distribution in $\hat{\boldsymbol{u}}_M$.

Applying Newton's method to a problem requires evaluating the residual of the multi-physics equation system (right hand side of Eq. 45), which requires evaluating the residuals for the partial single-physics problems. Although there has been success in the development of residual Monte Carlo methods for specific applications (see e.g. [20], [21] and[22]), these methods have yet to be applied to the continuous energy Monte Carlo neutron transport methods used in this work.

A second challenge for Monte Carlo multi-physics is the fact that the power distribution $\hat{\boldsymbol{u}}_M$ obtained from the Monte Carlo simulation is only a statistical estimate for the actual result $\boldsymbol{u}_M$ and contains an additional stochastic error term ε

$$\hat{\boldsymbol{u}}_M = \boldsymbol{u}_M + \varepsilon. \tag{52}$$

As there is no functional dependence between the coupled fields and the result distribution, there is no possibility to evaluate the derivatives required for the Jacobian in the Newton's method analytically and due to the stochastic error term present in the Monte Carlo solution, the evaluation of finite differences requires a large number of neutron histories in order to obtain reliable estimates for the sensitivities. Due to the computational cost of the Monte Carlo transport solution, evaluating thousands or millions of sensitivities for a single coupled-calculation iteration is not a realistic possibility if each sensitivity has to be evaluated with a separate simulation. The use of Newton's method with continuous energy Monte Carlo neutron transport can be challenging[8].

Monte Carlo neutronics can be used as a part of a fixed point iteration scheme (Algorithm 3). The Monte Carlo operator $\mathbb{M}$ will then approximate one of the G_i operators for the fixed point iteration with a stochastic operator $\hat{G}_M(\boldsymbol{U}) = G_M(\boldsymbol{U}) + \varepsilon(\boldsymbol{U})$. The application of this operator on the current solution vector $\boldsymbol{U}$ (containing, e.g. the material temperatures and densities calculated based on the previous power

[8] Newton's method or JFNK can, however, be used for the other fields of coupled physics even when the power distribution is supplied by Monte Carlo neutronics.

distribution) requires executing the Monte Carlo simulation and will yield the next stochastic estimate for the power distribution $\hat{\boldsymbol{u}}_M$:

$$\hat{\boldsymbol{u}}_M^{(k+1)} = \hat{G}_M(\boldsymbol{U}^{(k)}), \tag{53}$$

where the right hand size is evaluated by running the Monte Carlo simulation.

In fixed point iteration the conventional practice is to estimate the convergence of the coupled problem by monitoring the differences in fields between two subsequent iterations[9]

$$\left\| \boldsymbol{u}_i^{(k+1)} - \boldsymbol{u}_i^{(k)} \right\| = \left\| G_i(\boldsymbol{u}_i^{(k)}) - \boldsymbol{u}_i^{(k)} \right\|, \tag{54}$$

which should tend towards zero as the iteration tends towards the correct solution. For the Monte Carlo neutron transport solution, the difference will be limited by the stochastic error terms

$$\left\| \hat{\boldsymbol{u}}_M^{(k+1)} - \hat{\boldsymbol{u}}_M^{(k)} \right\| = \left\| \hat{G}_M(\hat{\boldsymbol{u}}_M^{(k)}) - \hat{\boldsymbol{u}}_M^{(k)} \right\| = \left\| G_M(\boldsymbol{u}_M^{(k)} + \epsilon^{(k)}) + \epsilon^{(k+1)} - (\boldsymbol{u}_M^{(k)} + \epsilon^{(k)}) \right\|, \tag{55}$$

which can be decreased by using a large number of source particles in the simulation but cannot be completely eliminated. The stochastic approximation-based solution techniques presented in the next section attempt to find a way around these difficulties.

6.1 Stochastic approximation-based solution techniques

In the stochastic approximation approach, the coupled problem can be reformulated around the Monte Carlo transport operator described in Eq. 53.

The problem is to find the converged solution for the power distribution ($\boldsymbol{u}_M$) so that

$$G_M(\boldsymbol{u}_1^{(k+1)}, \boldsymbol{u}_2^{(k+1)}, \dots, \boldsymbol{u}_M^{(k)}, \dots, \boldsymbol{u}_N^{(k)}) = \boldsymbol{u}_M^{(k+1)} = \boldsymbol{u}_M^{(k)}, \tag{56}$$

i.e. the calculation (G_M) of the new power distribution using the coupled fields ($\boldsymbol{u}_i$, where $i \neq M$) calculated using the previous power distribution $\boldsymbol{u}_M^{(k)}$ yields a power distribution equal to the one obtained on the previous step.

The power distribution $\boldsymbol{u}_M^{(k)}$ will also be the solution for the minimization problem

$$\min f(\boldsymbol{u}_M^{(k)}) = \min \left\| G_M(\boldsymbol{u}_1^{(k+1)}, \boldsymbol{u}_2^{(k+1)}, \dots, \boldsymbol{u}_M^{(k)}, \dots, \boldsymbol{u}_N^{(k)}) - \boldsymbol{u}_M^{(k)} \right\|_2, \tag{57}$$

which minimizes the ℓ^2 norm of the residual field of Eq. 56. The problem is that the operator G_M can be accessed only through the stochastic estimate $\hat{G}_M$. Luckily, the Robbins–Monro algorithm [23] for stochastic approximation was developed for exactly such problems.

Dufek and Gudowski [24] showed that the stochastic approximation can be applied to the calculation of the multi-physics problem with Monte Carlo neutronics in steady state. They showed that with slight approximations based on the localized and negative feedback effects between the neutron flux and the coupled fields, the Robbins-Monro algorithm for the Monte Carlo multi-physics can be worked to

$$\boldsymbol{u}_M^{(k+1)} = \boldsymbol{u}_M^{(k)} - a_k d \left(\boldsymbol{u}_M^{(k)} - \hat{G}_M(\boldsymbol{u}_1^{(k+1)}, \boldsymbol{u}_2^{(k+1)}, \dots, \boldsymbol{u}_M^{(k)}, \dots, \boldsymbol{u}_N^{(k)}) \right), \tag{58}$$

[9] It should be noted that this conventional approach has a risk of providing a false prediction of convergence.

where a_k is the step size at step k and d is an under-relaxation factor. There are several ways to choose the step size a_k: Robbins and Monro used

$$a_k = \frac{C}{k}, \tag{59}$$

where C is a constant. Dufek and Gudowski suggested that in the Monte Carlo context, a good choice for the step size would be

$$a_k = \frac{s_k}{\sum_i^k s_i}, \tag{60}$$

where s_i is the simulated active neutron population at step i. The step size for step k would then be the number of active histories simulated on step k divided by the sum of active histories simulated on previous steps up to and including k. The operation of stochastic approximation methods in this context can then be summarized in Algorithm 5.

Algorithm 5. Use of stochastic approximation in multi-physics as a part of a fixed point iteration scheme. $\boldsymbol{u}_M$ is the solution of the Monte Carlo neutron transport simulation. The execution of the MC simulation is denoted by the stochastic operator $\hat{G}_M$.

```
Start with initial guesses $\boldsymbol{u}_1^{(0)}, \boldsymbol{u}_2^{(0)}, \ldots, \boldsymbol{u}_M^{(0)}, \ldots, \boldsymbol{u}_N^{(0)}$.
$k = 0$
while not converged:
    $\boldsymbol{u}_1^{(k+1)} = G_1(\boldsymbol{u}_1^{(k)}, \boldsymbol{u}_2^{(k)}, \boldsymbol{u}_M^{(k)}, \ldots, \boldsymbol{u}_N^{(k)})$
    $\boldsymbol{u}_2^{(k+1)} = G_2(\boldsymbol{u}_1^{(k+1)}, \boldsymbol{u}_2^{(k)}, \boldsymbol{u}_M^{(k)}, \ldots, \boldsymbol{u}_N^{(k)})$
    ⋮
    $\boldsymbol{x}^{(k+1)} = \hat{G}_M(\boldsymbol{u}_1^{(k+1)}, \boldsymbol{u}_2^{(k+1)}, \ldots, \boldsymbol{u}_M^{(k)}, \ldots, \boldsymbol{u}_N^{(k)})$
    $\boldsymbol{u}_M^{(k+1)} = \boldsymbol{u}_M^{(k)} - a_k d(\boldsymbol{u}_m^{(k)} - \boldsymbol{x}^{(k)})$
    ⋮
    $\boldsymbol{u}_N^{(k+1)} = G_N(\boldsymbol{u}_1^{(k)}, \boldsymbol{u}_2^{(k)}, \ldots, \boldsymbol{u}_M^{(k+1)}, \boldsymbol{u}_N^{(k)})$
    $k = k + 1$
end while
```

6.2 Changes required to MC algorithms

Hitherto, we have assumed the existence of a Monte Carlo neutron transport operator $\mathbb{M}$ that is capable of providing a high-fidelity neutron transport solution $\boldsymbol{u}_M$ based on the coupled fields provided by the other solvers $\boldsymbol{U}$. Monte Carlo neutron transport relies on basing the random sampling in the simulation on the neutron cross sections $\sigma(E)$ and $\Sigma(E)$ that were largely considered to be constant for each material region when discussing the Monte Carlo neutron transport in Section 3. However, Section 4.3 established that the interaction probabilities (the cross sections) are strongly dependent on material temperature and density:

$$\sigma = \sigma(E, T)$$
$$\Sigma = \Sigma(E, T, \rho) = \sigma(E, T)N(\rho)$$

In multi-physics applications the temperature and density fields may be supplied to the Monte Carlo code by other solvers as a high-fidelity distribution. The Monte Carlo code should then be able to simulate the random walk of neutrons efficiently even if the distributions are rather complex. There are several parts of the Monte Carlo neutron transport process that must be adapted for multi-physics applications as described in the following sections.

6.2.1 Thermal effects on cross sections

Due to the temperature effects on the microscopic cross sections, the Monte Carlo code needs a way to evaluate the microscopic cross sections at the temperatures that the neutrons may encounter in the calculation model. The cross section temperature treatment approaches can be divided into pre-processing and on-the-fly methods: Pre-processing methods, as their name suggest, generate the cross section data for each required temperature before the start of the calculation using, for example, the NJOY code [25]. All of the cross section data is then loaded into the program memory for rapid access during the simulation. On-the-fly methods, on the other hand, operate with a reduced number of *base* cross section libraries at specified *base* temperatures and account for the temperature effect between the simulation temperature and the base temperature at runtime (on-the-fly).

Pre-processing the cross sections is only feasible in systems with a limited number of temperatures as each new temperature requires pre-processing and storing an additional set of cross sections. As the number of temperatures increases, so does the memory consumption. Another drawback for the pre-processing approach is the fact that, when the temperatures in the simulation change during the multi-physics solution process, the cross section data has to be pre-processed again, thus increasing the computational effort.

Multi-physics simulations using high-fidelity (or continuous) temperature fields are only feasible with the use of on-the-fly temperature treatment methods. The first on-the-fly approaches did not even try to evaluate the cross sections at the correct temperature. Instead they relied on generating temperature-dependent cross sections for a set of temperatures and then either

- simply using the closest temperature, **or**,
- interpolating between the bracketing temperatures, **or**,
- using *stochastic mixing* to ”interpolate” between the bracketed temperatures.

The first possibility is not generally applicable to multi-physics problems, in which accurate results are needed over wide temperature ranges but it was applied as an approximation in [4] and [5]. The accuracy of the second approach was studied by Trumbull in [26] and found to be rather poor. The stochastic mixing is possibly the first widely used and feasible approach to reconciling the temperature effects with a reasonable number of pre-generated cross section libraries. In the stochastic mixing approach, materials between the base temperatures are modeled as mixtures of materials at the base temperatures. For example, water at 506 K might be approximated with a mixture of water at 500 K and water at 510 K. Although the approach is non-physical, it has produced reasonable results in a number of test cases and many researchers applied it as an approximation due to its simplicity (for example in [3, 5, 6, 27]) before the arrival of the more advanced on-the-fly techniques.

The second group of on-the-fly temperature treatment techniques endeavors to solve the temperature-dependent cross section at the target temperature using techniques similar to those used in NJOY, but during the neutron transport simulation. Two such methods are the approach proposed by Li [28] and the method implemented to the BINGO collision processor in the Monte Carlo code MONK [29]. For a discussion of the differences in the performance of these methods the reader is directed to a recent article by Romano and Trumbull [30].

In recent years several advanced approaches to the problem have been developed, including the following three:

- Use a single set of base cross sections and generate a series expansion for the temperature dependence and quickly calculate the temperature-dependent cross sections on-the-fly [31]. The method is used in MCNP6.
- Store a single set of base cross sections in multipole representation rather than pointwise tabulated format and calculate the analytical Doppler broadening on-the-fly [32]. This approach has been implemented in the OpenMC Monte Carlo code.
- Use a single set of base cross sections and take the temperature effect into account on-the-fly using the physical Target Motion Sampling temperature treatment technique [33]. This approach is used in the Serpent 2 Monte Carlo code.

The calculations in this work used the Target Motion Sampling (TMS) temperature treatment technique previously implemented in Serpent.

6.2.2 Effects on track length sampling

As was described in Section 3.2: track-lengths for neutrons can be sampled using Eq. 19, and the equation only applies if the cross section in the denominator is constant over the sampled track-length. Material boundaries are handled either by stopping the track-lengths at material interfaces and sampling a new track-length in the upcoming material (surface tracking), or by using a common majorant cross section for the track-length sampling in all materials and using the concept of virtual collisions to conserve the physics of the sampling (delta-tracking).

As the material temperature affects the material cross sections it is evident that the cross section in the denominator of Eq. 19 will depend on the material temperature, and even the different temperature regions of one material should be treated as different materials in the track-length sampling. With high-fidelity temperature distributions (see Fig. 16), the efficiency of surface tracking may become very poor if no modifications are made. The same is true in the case of high fidelity density distributions.

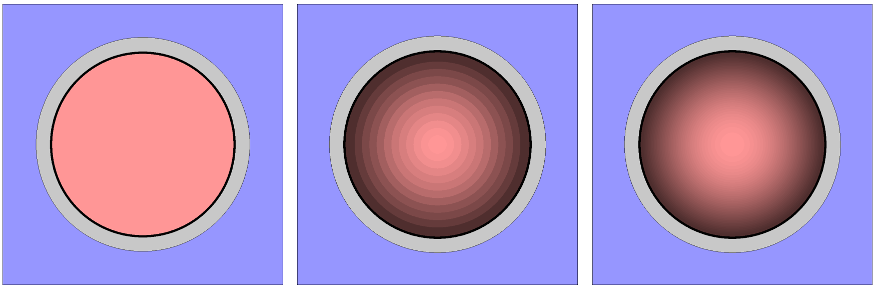


Figure 16. Pin-cell geometry with fuel temperature shown as shading. **Left**: Radially constant fuel temperature. **Center**: Radial temperature step profile. **Right**: Radially continuous temperature profile.

The simple solution for sampling track-lengths in materials with very refined or even continuous temperature and density distributions is to utilize rejection sampling in the track-length sampling [34, 35]. This approach uses a material-wise majorant cross section $\Sigma_{\text{maj.},*,t}(E)$ that covers the range of temperature and density variation for the material. These material-wise majorants are then used in track-length sampling inside the material region and the physics of the system are conserved by rejecting some of the sampled collision points. As this rejection sampling is used *inside* each material region, it can be used with both the surface tracking and Woodcock delta-tracking methods that handle the variation of the cross sections *between* the material regions.

6.2.3 Scoring tallies

There will be no changes required to the scoring of analog estimators. For collision estimators, the cross sections in the denominator of the tallied value (Eq. 28) will be temperature dependent and the nominator will be the multi-physics majorant for the local material (introduced in section 6.2.2) that is used for the track-length sampling inside the material:

$$\frac{\Sigma_t^{^1\text{H}}(E, T, \vec{r})}{\Sigma_{\text{maj.},*,t}(E)}. \tag{61}$$

The temperature-based cross section must be evaluated on-the-fly using the temperature at the interaction site. The tallied value for the track-length estimator is changed in the same way

$$\sum_i^{\text{materials}} \sum_j^{\text{temperatures}} \frac{\Sigma_{t,i,j}^{^1\text{H}}(E, T, \vec{r})}{\Sigma_{\text{maj.},*,i,t}(E)} s_i \tag{62}$$

and the previous track-length must be segmented not only based on the material region as in Eq. 30, but also based on the material temperature. The macroscopic cross sections of the material are constant over the track-length only if both the material and the material temperature are constant over the track-length. Track-length estimators can be used with arbitrarily refined, but not continuous, temperature and density distributions although their performance may be compromised if a very large number of temperature/density region boundaries is crossed by the mean track-lengths of neutrons in the system.

6.2.4 Temperature and density distribution handling

The traditional approaches for defining the temperatures and densities in the Monte Carlo geometry have been to bind the temperature distribution to materials and the density distribution to either geometry cells or materials. This ensures that each geometry cell has a constant microscopic and macroscopic cross sections.

However, with the use of collision estimators the temperatures and densities are only required by the code at the interaction sites, which gives considerable freedom for the functional forms that can be used to represent the temperature and density distributions: Any function that translates coordinates into temperature and density information can be used. With track-length estimators the functions have the additional constraint of needing to yield step profiles, when evaluated over neutron tracks.

The traditional material and geometry cell-based temperature distributions can obviously be used in the multi-physics simulations, but the temperature and density information can also be separated from the underlying geometry. This means that, if the multi-physics interface of the Monte Carlo code is designed appropriately, changes in the temperature or density meshing do not necessitate changes in the Monte Carlo geometry model. The changes in the meshing might be simply changes in the mesh size (Fig. 17 left to center)) or even changes in the external solver from a structured mesh-based solver to a solver providing the fields using point fields instead of structured meshes (17 right). The possibility for the neutronics model to be agnostic of the coupled solver and to use the supplied distributions as-is, without cumbersome grid matching, is a major advantage of this kind of multi-physics approach.

It takes no additional effort to model temperature and density distributions that are continuous in space and time. The work in this thesis has utilized this fact via the use of radially continuous temperature distributions given either in functional form (Publication **I**) or using linear interpolation with respect to space (Publications **III**, **V** and **VI**) and time (Publication **III**). Continuous temperature representations have also been used with the OpenMC Monte Carlo code in combination with functional expansion tallies [36].

6.2.5 Providing external solvers with power distributions

As a part of multi-physics calculation sequence, the Monte Carlo code is required to provide a power distribution to the other codes in the calculation sequence. Providing an accurate heat deposition distribution for all of the associated materials is a complex task as heat is deposited, not only in fissions, but also in neutron and photon reactions with fuel, moderator, control rod and structural materials. The direct heating in moderator, control rod and structural materials is an important source term for thermal-hydraulic calculations. For accurate representation, the prompt energy deposition by neutrons and photons needs to be tallied, for example using the kinetic energy release per unit mass (KERMA) coefficients for each material of in-

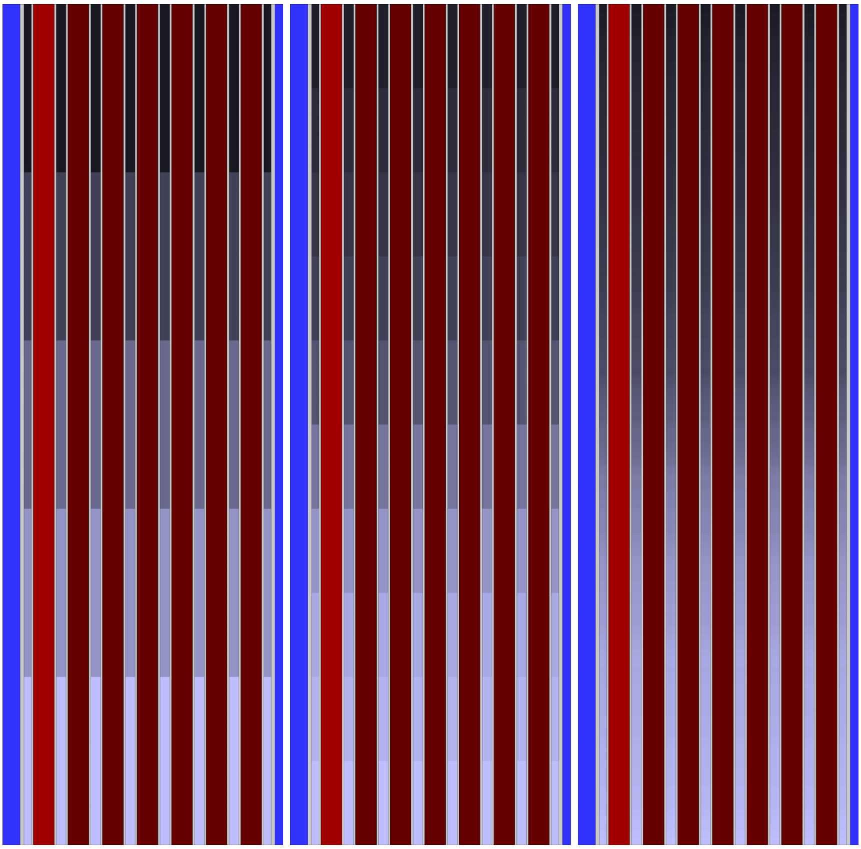


Figure 17. Serpent 2 YZ-geometry plots with different coolant density distributions provided by an external solver. Darker color indicates lower density. A well designed multi-physics approach means that no changes are required in the neutron transport geometry model if e.g. the meshing of the external solver changes (left to center) or if the external solver is swapped for one that provides the data as a point-field instead of on a structured mesh (right).

terest. This will require both a neutron and a photon transport calculation. The information on the prompt heating then needs to be combined with the calculation of the decay heat produced by decaying fission products.

The calculations conducted in this thesis were coupled calculations with fuel behavior solvers, for which only the heat generation in the fuel material was provided. The heat generation in the fuel was approximated by using a fixed value for local heat deposition due to one fission (κ), which for ^{235}U was 202.27 eV, the default value used by Serpent. For other fissionable nuclides the κ was scaled based on their fission Q-value

$$\kappa_n = \frac{Q_n}{Q_{\mathrm{U5}}} \kappa_{\mathrm{U5}}, \tag{63}$$

where the lower index U5 refers to ^{235}U. This approximation does not consider the accurate direct heat deposition of neutrons, photons or beta-particles, which is a potential topic for future improvement.

6.3 Application to time-dependent problems

An inherent advantage of the use of Monte Carlo codes in multi-physics simulations is the fact that Monte Carlo codes can easily be made to be inherently four dimensional. As the particle is moved in space between two interactions it can also be moved in time. By keeping track of the ”internal clock” of each neutron during the random walk, the time-dependent behavior of the particle population can be directly modeled without the need to solve the time-dependent behavior as a time-integral based on solutions at discrete time points.

As described by Bart Sjenitzer in his PhD thesis on the development of a dynamic Monte Carlo method [37], practical time-dependent multi-physics calculations with Monte Carlo neutronics typically require two additional implementations to the Monte Carlo code: First, a possibility should exist to divide the simulated time period into time intervals with a possibility for particle population control and data exchange with coupled codes at the time interval boundaries. Second, a model for tracking the delayed neutron precursors should be utilized instead of generating delayed neutrons directly from fissions.

A dynamic simulation mode based on the external source simulation mode with population control at time interval boundaries was implemented in Serpent in [38]. The mode was reworked to allow for time-dependent coupled calculations and demonstrated using the internally coupled fuel behavior module FINIX in a fast transient scenario without delayed neutron emission in Publication **III**. Delayed neutron precursor tracking and emission routines were later implemented and described in Publication **IV**. The implemented delayed neutron routines used an automatic group structure based on the cross section libraries chosen by user and the fissionable nuclides present in the problem.

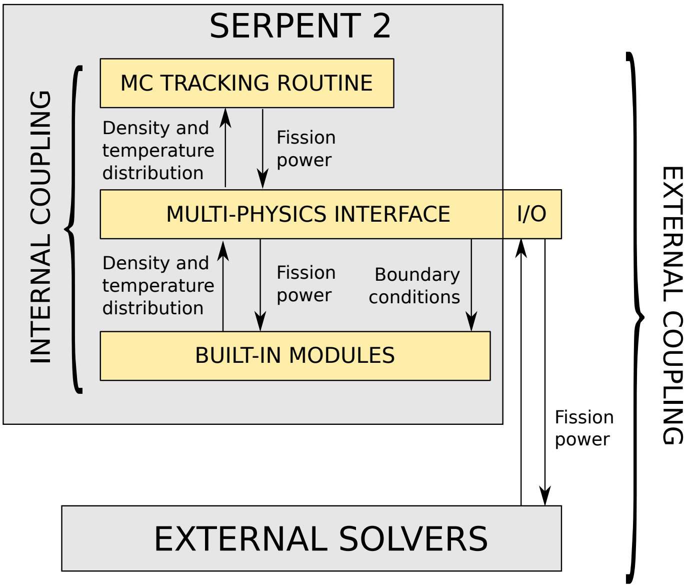


Figure 18. The two level coupling scheme in Serpent 2. Built-in solver modules can be used to obtain fast solutions for fuel and coolant behavior and high-fidelity solutions from state-of-the-art external solvers can be imported via the universal multi-physics interface.

7. Development of coupled calculation capabilities for Serpent 2

This section discusses the development and implementation of the coupled calculation capabilities of Serpent 2 in this thesis, whereas the next section (Section 8) will discuss the application of these capabilities to the problem of effective fuel temperatures. Although the development of the capabilities and their application did go hand-in-hand during the thesis work, a clearer picture of each topic can be given by focusing on one of the topics at a time.

7.1 Design choices

The goal of this thesis was to design the coupled calculation capabilities of Serpent 2 [8], a continuous energy Monte Carlo burnup code, in a sustainable manner. Sustainable, in the sense that the design choices and implementations would allow multi-physics simulations for a variety of simulation types in a code agnostic sense, while being easily extendable for future applications. The main simulation types that were included in the goals of the coupled calculation development were:

1. Steady state calculations using the criticality source simulation (k-eigenvalue) mode.

2. Burnup calculations.
3. Time-dependent simulations.

Relatively early in the development of Serpent 2, the design choice was made [39] to utilize a two-level coupling scheme (see Fig. 18), in which internal light-weight solvers could be used to obtain rapid solutions for fuel and coolant behavior and external state-of-the-art solvers could be coupled to Serpent via a universal code-agnostic multi-physics interface to provide truly high-fidelity solutions for the coupled fields. The approach taken in the implementation of the two-level coupling scheme in this work was to have the external and internal coupling to operate as similarly to each other as possible, i.e. to use the same sub-routines and data-structures whenever possible. Developing, implementing and maintaining a single set of sub-routines was the preferred approach from the point of view of code development.

The two level coupling scheme also eased the testing of the internal and external interface routines as they could be used for cross-verification as the two were applied to the same problem. The testing of the coupled calculation routines was mainly conducted via a set of test problems designed to test separate parts of the implementations (unit testing). These test problems were executed each time a version of Serpent was publicly distributed.

In 2012, when this thesis work began, Serpent had the physics models for handling the complex temperature and density distributions required in the multi-physics simulations, but the coupled calculation routines were still in their infancy. The first multi-physics interface types were included in Serpent 2.1.3, released in March 2012, for bringing in temperature and density data using point-fields or regular meshes, but no real coupled calculations were executed at the time. The first approach to an internally coupled calculation (Publication **I**) was developed in parallel with the first interface types.

7.2 First approach to internal coupling and steady state calculations

In Publication **I** the recently implemented TMS on-the-fly temperature treatment was applied to an actual coupled problem by implementing an internal analytic radial fuel temperature solver for UO_2 fuel rods and using the solver in a coupled calculation of a two-dimensional BWR assembly geometry at full power. Serpent tallied the radial fission power distribution in each fuel rod, whereas the implemented solver gave a solution for the time-independent radial heat equation in the cylindrical geometry of the fuel rod. The solver considered the temperature dependence of the heat conductivity of each material as well as the radial changes in the fuel rod geometry due to thermal expansion and pellet relocation. The possibility to apply the changes in the radial geometry of the fuel rod to the Monte Carlo geometry model was a notably novel approach, which makes it possible to estimate the effects that these changes have on the coupled solution.

This first coupled calculation implementation used a simple fixed point iteration (Algorithm 3 on page 50) until a convergence was reached between neutronics and fuel behavior. The coupled calculation iteration was executed after the simulation of each neutron batch. The fuel behavior was solved as a separate fixed point iteration between the temperature solution, the temperature-dependent material properties and the temperature-dependent radial changes in fuel rod geometry.

The internal fuel temperature solver proved to be successful in solving the coupled problem between fuel temperature and neutronics, although it was later replaced with the newly developed FINIX fuel behavior module for multi-physics applications [40, 41].

7.3 First approach to external coupling and burnup calculations

The calculation routines for external coupling were developed next in the context of externally coupled burnup calculations with the ENIGMA [42] fuel performance code in Publication **II**. For this purpose, several new capabilities were developed and implemented into Serpent, including

- The universal fuel performance code interface, also known as the fuel behavior interface and used in Publications **II**, **V** and **VI**. Many of the fuel performance code interface routines were also used by the internal FINIX fuel behavior module in Publication **III**.
- Radial coordinate transformations inside fuel rods to simulate modifications to the radial geometry in the fuel rods in the neutron tracking.
- POSIX-signalling capabilities and the ability to function as a part of an iterative coupled calculation scheme, without the need to restart Serpent for each iteration.

The fuel behavior interface allowed Serpent to read in the radial temperature and strain distributions for fuel rods at different axial elevations and overlay the distributions on the geometry model. On the other hand, the fuel behavior interface would automatically tally the radial fission power distribution at different axial elevations and write the power distribution to a file to be read by the external code. Two options were implemented for the functional form of the radial temperature profile: a step profile and linear interpolation between point-wise values.

The pre-processing of the problem geometry, material compositions and cross sections required for the neutron transport simulation takes a considerable time, especially in burnup calculations with a large number of transport nuclides. In order to increase the computational efficiency, several routines were implemented which made it possible to wait for the solution for coupled fields and read in updated distributions without the need to terminate and restart Serpent in between. In publication **II**, these newly developed capabilities were demonstrated in a coupled burnup calculation of a short VVER-440 type fuel assembly with burnable absorber rods. The coupling was extremely simple, as was also denoted in the publication, consisting simply of a single fixed-point iteration between the two solvers for each burnup step. As the two solutions were not iterated to convergence, the time-coupling can be described as a loose coupling.

7.4 Capabilities for time-dependent simulations

After the initial implementations of coupled calculation routines for steady state and burnup calculations, Publication **III** tackled the coupled time-dependent problem using the internally coupled FINIX fuel behavior module. A time-dependent simulation mode had previously been implemented in Serpent [38], but in Publication **III** it was reworked to support coupled calculations. The new transient simulation mode was applied to a coupled simulation of a super-prompt critical transient in a 2D PWR pincell geometry. The coupled solution scheme was slightly more advanced than in the earlier publications: A fixed point iteration was used between the two solvers, with stochastic approximation-based solution relaxation applied to the power distribution solved by Serpent. Each time interval was iterated until the maximum difference in the end-of-interval fuel temperature distribution between two subsequent iterations was below 3 K. Due to this iteration, the time-coupling can be described as a tight coupling.

During the neutron transport for a certain time interval, the time-dependence of the temperature distribution was accounted for by linearly interpolating the temperature between the beginning-of-interval and end-of-interval temperature distributions based on the interaction time. The fuel behavior simulation used the integral power over the time interval divided by the time interval length as a constant power level throughout the interval, which did not account for the time-dependence of the power during the interval but did conserve the energy deposition.

A major shortcoming of the physics models in the demonstration simulation in Publication **III** was the lack of delayed neutron emission during the transient. Although the lack of a delayed neutron emission model for time-dependent simulations did not hinder the demonstration of the time-dependent coupled calculation routines, realistic transient simulations cannot be executed without such models. For this reason, new delayed neutron emission models were implemented for time-dependent simulations in Publication **IV** and verified against analytic solutions in infinite medium systems. The implementation utilized a delayed neutron group structure in the tracking of the delayed neutron precursors and in the various sampling routines used for, e.g. precursor production, delayed neutron emission time and energy. This group structure is generated automatically by the implementation based on the cross section libraries used in the simulation and the fissionable nuclides present in the modeled system. Some evaluated nuclear data libraries, such as ENDF/B libraries utilize a six group structure for the delayed neutrons, where the six groups are unique for each fissionable nuclide. JEFF libraries, however use eight groups that are common to all fissionable nuclides. The implementation uses a group structure, where each unique delayed neutron group is handled separately, meaning that simulations using JEFF based cross sections have eight delayed neutron groups and simulations using ENDF/B based cross sections have 6×(number of fissionable nuclides) groups. In Publication **IV** the models were not yet used for coupled calculations, but they have since been applied to both internally and externally coupled time-dependent simulations, although no results have yet been published.

7.5 Finalized unified coupling methodology

As the coupling approach between Serpent and coupled solvers, both internal and external, evolved during the thesis work, the last two publications focused on describing the final form of the coupling in steady state and burnup applications.

Publication **V** described the implementation of a unified coupling scheme used in both internally (Fig. 19) and externally (Fig. 20) coupled calculations for steady state applications. The stochastic approximation-based solution scheme (Algorithm 5)

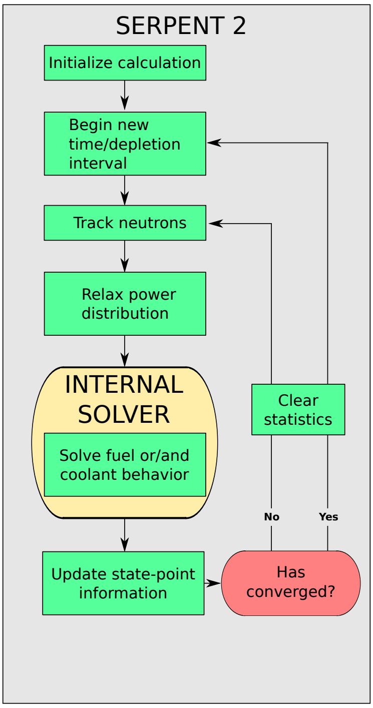


Figure 19. Solution flow in internally coupled calculations using the finalized coupled calculation routines.

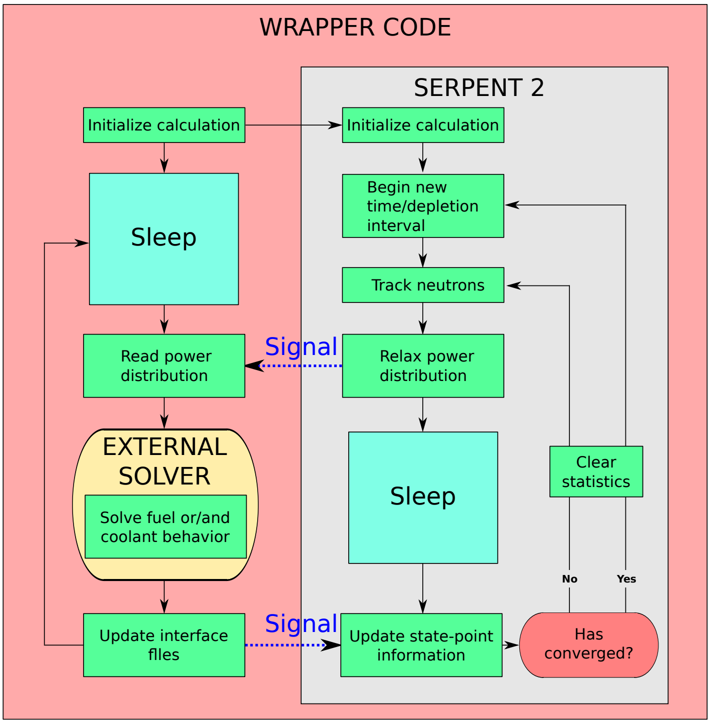


Figure 20. Solution flow in externally coupled calculations using the finalized coupled calculation routines.

was adopted in full for the solution relaxation. The finalized steady state capabilities were demonstrated in a coupled solution of neutronics and fuel behavior for a 3D BWR assembly geometry using both an internally and an externally coupled approach. Finally, Publication **VI** combined the steady state solution scheme used in the previous publication with the Stochastic Implicit Euler burnup scheme with thermal feedback [43] for coupled burnup calculations. The coupled burnup capabilities were applied to a series of 2D assembly burnup calculations using realistic fuel temperatures provided by the externally coupled ENIGMA fuel performance code.

At the end of the thesis project, the same coupling scheme (Figs. 19 and 20) can be used in steady state, burnup and transient calculations with internally or externally coupled solvers. Although the applications in this thesis were focused on fuel temperature coupling using the fuel behavior interface, the same coupled calculation approach can be used to couple any external solver with Serpent using one of the various multi-physics interface formats in Serpent. The coupled burnup calculations in Publication **VI** were conducted using the Stochastic Implicit Euler burnup scheme, but it is also possible to use the standard explicit Euler and predictor-corrector-based burnup schemes in Serpent with a stochastic approximation-based iteration for each neutron transport solution.

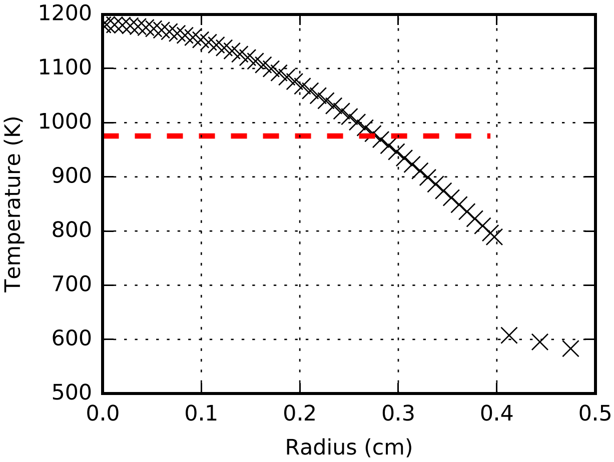


Figure 21. Realistic radial temperature distribution in fuel and in cladding at discrete nodes (x) and a radially constant effective fuel temperature (red dashed line) typically used in lattice calculations.

8. Application of the coupled calculation capabilities to effective fuel temperature problems

The multi-physics capabilities developed during this work were applied in Publications **I**, **II**, **V** and **VI**, partly as a demonstration, to the problem of effective fuel temperatures typically used in neutronics calculations. In order to understand this problem, consider the typical radial fuel temperature from the fuel rod centerline to the cladding outer surface shown in Fig. 21 with black x. Due to limitations in computational resources it is very common to run neutronics calculations using a single effective temperature (red dashed line) instead of a continuous temperature profile or a step profile [44]. As has been described previously in this thesis, the fuel temperature distribution has a strong effect on the neutronics of the system. The problem of the effective temperature is, then, how to choose the dashed line in Fig. 21 so that the neutron transport solution is preserved as well as possible.

The research into the optimal choice for this effective temperature has mostly focused on preserving the rate of resonance absorption in the system [45, 46, 47], i.e. choosing the effective temperature so that the number of neutrons lost to radiative capture by ^{238}U is conserved. Although the choice of an optimal effective temperature is generally a problem in multiple-objective optimization, the conservation of the radiative capture rate by ^{238}U is a good choice for an important objective, as it is the main basis of the fuel temperature reactivity feedback in low-enriched uranium. Moreover, conserving the radiative capture rate of ^{238}U should also automatically conserve the production rate for the fissile ^{239}Pu.

Several methods for calculating an estimate for this effective temperature have been used: A volume averaged fuel temperature is a simple and common approach to calculating an effective temperature, as was noted in [44]. Rowlands proposed already in [45] that the effective temperature should be calculated as the average temperature over all possible neutron track-lengths through the pellet, whereas Goltsev *et al.* suggested in [47] to use volume-averaging with an additional $1/\sqrt{T(r)}$ weighting function.

As the coupled calculation routines implemented in this work make it possible to use the accurate continuous fuel temperature profile even in large systems containing hundreds of fuel rods, it is possible to obtain accurate reference results for the effective temperature models (such comparisons were made in Publications **I**, **II** and **V**). Furthermore, the accurate flux solution obtained from the reference simulation can be used to accurately calculate various new effective temperatures. In Publication **V** this was demonstrated by calculating volume averaged effective temperatures using either the local collision rate or the local radiative capture rate as a weighting function.

8.1 Evaluating the performance of effective fuel temperature models in steady state calculations

The internal fuel temperature solver implemented in Publication **I** made it possible to obtain the accurate coupled neutronics–fuel temperature solution using the realistic continuous fuel temperature fields. In Publication **I** such a solution was obtained for a 2D BWR assembly geometry. The multiplication factor of the detailed solution was then compared to that of a simulation using an effective temperature obtained by volume averaging the accurate temperature solution over the total fuel volume of the assembly. The volume averaged effective temperature reproduced the reactivity of the assembly very well, leading to a reactivity difference of only 35 ± 4 pcm. However, the reactivity of the system is an integral parameter and does not give any indication of possible differences in local parameters.

Publication **V** applied a more robust evaluation of the effect of the effective radial fuel temperature by comparing three different pre-existing methods (Volume averaging, simplified Rowland's method and Goltsev's volume averaging with $1/\sqrt{T}$ weighting) and two new effective temperature models, that collapsed the realistic temperature distribution using volume averaging with local reaction rate weighting. The comparison was made against a reference solution using a realistic temperature profile at several axial levels of a 3D BWR assembly geometry. This time the comparison also included the differences in local reaction rates.

The main findings of publication **V**, with regard to effective fuel temperatures, can be summarized as follows: The radially constant effective fuel temperatures can reproduce the global radiative capture rate by ^{238}U rather well: The poorest performing effective temperature model led to a $-0.20\,\%$ relative error in the global ^{238}U radiative capture rate. The global radiative capture rate was best reproduced by one of the novel effective temperature models, namely the ^{238}U radiative capture weighted effective temperature. However, all of the radially constant effective temperatures led to significant errors in the local reaction rates. With the use of effective temperatures, the radiative capture was overestimated in regions close to the pellet surface and underestimated in the pellet inner regions. This is explained by the fact that the effective temperatures are significantly higher than the correct temperature at the pellet surface and significantly lower at the pellet center (see Fig. 21).

8.2 On the use of effective fuel temperatures in burnup calculations

The fact that large local differences in the capture reaction rates could be seen with the use of effective radial fuel temperatures prompted the question of possible effects on burnup calculations as differences in capture rates also indicate differences in the production of ^{239}Pu and other actinides. This was studied initially in publication **II** and later with the finalized coupled calculation routines in publication **VI**. Publication **VI** looked at the burnup effects in the context of generation of group constants for a core simulator for the simulation of the initial cycle of the EPR reactor [48], as briefly described in the following paragraphs.

The group constant generation for core simulators for a fuel cycle of a certain reactor consists of

1. a burnup calculation part to obtain the nuclide compositions for different fuel assemblies at different points of the fuel cycle and
2. a branch calculation part to generate the group constants using the nuclide compositions obtained from the burnup calculation.

It is quite normal to conduct the burnup calculation in the first part using a single effective temperature for all the fuel rods in the assembly. This not only ignores the radial variation in each fuel rod, but also does not consider the variation between the different fuel rods, which can be important if some of the fuel rods contain burnable absorber, which reduces their temperature by hundreds of Kelvin in the beginning of the fuel cycle.

The effects of the effective fuel temperature simplification on the local and assembly-wide nuclide inventories in the burnup calculation, as well as on the generated group constants and final core simulator results, was studied by generating one set of group constants using the traditional effective temperature approach for the history calculation and a second set of group constants by using realistic fuel rod and burnup-dependent radial temperature distributions in the burnup calculation. The realistic temperature distributions were obtained by coupling Serpent externally with the fuel performance code ENIGMA using the coupled calculation routines developed in this work.

The main findings of publication **VI** regarding the effective fuel temperatures were as follows: The use of the effective temperature resulted in a significant overestimation of the actinide production, fission power and thus fission product concentrations in the areas close to the fuel rod surface[10]. On the other hand, the same variables were underestimated for the fuel pellet inner regions, resulting on average in a conservation of the assembly-wide fission product concentrations. The results indicated that, with an optimal choice of the assembly-wide effective temperature, it may be possible to conserve reasonably well *either* the assembly-wide actinide concentrations *or* the gadolinium burnout from the burnable absorber (BA) rods. This suggests that it would be beneficial to use a separate effective temperature for the BA rods in the burnup calculation. The assembly-wise homogenized group

[10] Similar trends were also observed for the radial power distribution in Publication **II**.

constants were reproduced rather well by the effective temperature model and the effects on the simulation of the EPR initial cycle were minor. The study did indicate that the fuel temperature history of the fuel assemblies should be treated as an important history parameter in the parametrization of the cross sections for core simulators.

9. Future prospects

The last section of this manuscript will be devoted to describing some interesting prospects currently on the horizon for Monte Carlo multi-physics. The section is divided into two subsections, the first focuses on some possible upcoming methods and the second lists some possible future applications for the Monte Carlo multi-physics calculations.

9.1 Methodology

Although the coupled calculation capabilities implemented in this thesis allow the user to execute coupled steady state, burnup and transient calculations in a standardized manner, many possible avenues exist for the future development of the coupled calculation capabilities:

9.1.1 New algorithms for coupled burnup calculations

The coupled burnup scheme based on the Stochastic Implicit Euler method is known to have poor accuracy in applications that have rapid shifts in the neutron spectrum due to the reliance of the method on the use of end-of-step reaction rates [49]. Several improved techniques have already been proposed [50, 49, 51] and the search for efficient burnup algorithms for Monte Carlo applications is still ongoing. There is certainly a need to replace the SIE-based algorithm with a more efficient one, although it is not yet clear which algorithm will be the final choice for Monte Carlo multi-physics calculations.

9.1.2 Cheaper calculation of the Jacobian for Newton's method iteration

The history-based sensitivity and perturbation calculation methodology developed by Aufiero *et al.* [52] can be used to calculate the sensitivity of various results of the Monte Carlo simulation, such as the power distribution, to various input parameters such as nuclear data, material densities and material temperatures using a single Monte Carlo neutron transport simulation. For the Monte Carlo multi-physics problem, this means that the sensitivities required in the application of Newton's method could be obtained without the need to run hundreds or thousands of Monte Carlo neutron transport calculations. The methodology has already been applied to the calculation of the sensitivities in $k_{\text{eff.}}$ to the coolant density distribution in LFRs and SFRs [53]. Calculating the sensitivity of the power distribution tallied by the Monte Carlo code to the distributions provided by the coupled codes would pave the way for the possibility to couple Monte Carlo codes in a JFNK framework in an efficient manner.

9.1.3 Cheaper time-dependent calculations with hybrid methods

Time-dependent Monte Carlo neutron transport simulations are computationally expensive. Although the computational efficiency of the simulations may be improved using various methods for variance reduction [37], hybrid methods such as the Improved Quasi-Static (IQS) method [54] provide an attractive alternative due to the reduced computational effort. In the IQS method, the neutron flux solution is separated in space and time, at relatively short time intervals, into a time-independent flux shape and a space-independent flux-amplitude. By solving the time-independent flux shape after certain time intervals using detailed methods such as Monte Carlo neutronics, and leaving the time dependence to be solved by simplified methods such as the point-kinetics approximation, the computational effort is significantly reduced. The full Monte Carlo time-dependent capabilities can be used to verify the accuracy of the hybrid methods. The IQS method has already been implemented in the multi-group Monte Carlo code KENO [55] and in the three dimensional discrete ordinates code TORT [56].

9.1.4 Accurate calculation of the heat deposition

The simulations conducted in this work approximated the heat generation distribution in the simulated systems by assuming a nuclide dependent value for locally deposited heat due to each fission and no heat deposition to other materials due to direct neutron or photon heating. This was an acceptable approximation for the development and testing of the coupled calculation methodology, but it is a large simplification especially considering thermal-hydraulic calculations in which the heat generation in moderator, neutron absorbers and structural materials serves as a significant heat source.

The recent implementation of photoatomic interactions into Serpent [57] makes it possible to execute a photon transport simulations based on the results of a neutron transport calculation [58]. Using these capabilities, Serpent can calculate the direct heat deposition by both neutrons and photons as was demonstrated in [59]. By combining a separate model for the calculation of decay heat generated in the fuel, a more accurate representation for the power distribution could be calculated. Such an implementation will be of great use in future multi-physics calculations.

9.1.5 Delayed neutron group structure used in time dependent simulations

The delayed neutron emission routines implemented in Publication **IV** utilized a delayed neutron group structure in the handling of the delayed neutron precursors and in the sampling of the delayed neutrons. It would be possible, however, to do away with the group structure and instead tally each precursor nuclide separately. The production data for each nuclide can be obtained from the neutron induced fission yield data included in the evaluated nuclear data libraries and the data for the beta-decay of the precursors is present in the radioactive decay data in the libraries.

Tracking each delayed neutron precursor nuclide separately would most likely be costly in terms of memory demand, however it would make it possible to estimate the effects of the group structure approximation on time-dependent neutron transport simulations. Another potentially interesting research direction would be the use of alternate delayed neutron group structures generated using the detailed data for each precursor nuclide separately and specifically for each simulation.

9.2 Applications

The obvious application for the Monte Carlo multi-physics capabilities is the production of accurate high-fidelity solutions to various coupled problems. Although the significant computational effort required for such calculations precludes the use of high-fidelity methods in day-to-day reactor analyses, the high-fidelity methods can be used to produce reference results for the fast reduced-order models used in the daily work.

Due to the versatility of the Monte Carlo method and the wide variety of input and output formats, it should be possible to use the same thermal-hydraulic solvers for the Monte Carlo solution and the reduced order solver. An example of such an application is given in [60], in which the neutronics solver in the code system DYN3D/SUBCHANFLOW, namely DYN3D, is benchmarked against the Seprent/SUBCHANFLOW code system. If the same Monte Carlo code is also used to generate the homogenized group constants for the reduced order solver, the differences related to cross section data libraries as well as to thermal hydraulic models can be eliminated.

Thus far, such comparisons have been made only for steady state simulations [60, 61], but time-dependent reference calculations for deterministic transient solvers are also possible, although the computational effort may limit the comparisons to relatively small and simple systems.

9.2.1 Novel options for group constant parametrization

The main development direction of Serpent has always been application of the code to the generation of spatially homogenized group constants for core simulators [62, 61, 63]. The traditional approach to modeling the fuel temperature feedback in the core simulator is to generate the group constants using several different flat fuel temperature profiles (for example at 600, 900 and 1200 K) and generating a functional expansion of the group constants with respect to the fuel temperature. During the execution of the core simulator, an assembly averaged effective fuel temperature is solved based on the assembly power and local thermal-hydraulic conditions. This effective temperature is then used to evaluate the group constants based on the functional expansion.

The flat homogeneous fuel temperature profiles used in the group constant generation do not correspond in a very meaningful way to the realistic fuel temperature profiles that would result from a certain node power/coolant temperature combination. Since the core simulator is usually only interested in the interaction properties for neutrons in the assembly at power P and coolant temperature T, the group constants could be directly generated for a set of node powers and coolant temperatures. With the new multi-physics capabilities, the group constant generation can use the realistic fuel temperature distribution corresponding to those conditions solved either by the Monte Carlo code or an external fuel behavior solver.

9.2.2 Detailed simulation of xenon transients

An interesting application for the developed coupled calculation routines would be the xenon transients that occur in PWR cores. Any perturbations to the power distribution will initiate a change in the concentrations in the concentrations of the short lived neutron absorber ^{135}Xe. The changes in the xenon concentrations will have an additional effect on the power distribution in the core with couplings to the fuel behavior and the thermal hydraulics. The xenon transients couple the time dependent neutronics with the time dependent fuel behavior, core level thermal hydraulics and fuel depletion (production and burning of ^{135}Xe).

The capability to model such complex transients with several interconnected couplings would serve as a formidable demonstrations of the coupled calculation capabilities of Serpent 2.

References

[1] IPCC, 2014: Climate change 2014: Mitigation of climate change. Contribution of working group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

[2] OECD-NEA, 2011: ”Technical and economic aspects of load following with nuclear power plants”, online report available at http://www.oecd-nea.org/ndd/reports/2011/load-following-npp.pdf.

[3] C. L. Waata. *Coupled Neutronics/Thermal-hydraulics Analysis of a High-Performance Light-Water Reactor Fuel Assembly*. PhD thesis, University of Stuttgart, 2006.

[4] V. Seker, J.W. Thomas, and T.J. Downar. Reactor simulation with coupled Monte Carlo and computational fluid dynamics. In *Proc. Joint International Topical Meeting on Mathematics and Computations and Supercomputing in Nuclear Applications, M and C + SNA 2007*, Monterey, California., 15–19 Apr. 2007.

[5] N. Capellan, J. Wilson, O. Méplan, J. Brizi, A. Bidaud, A. Nuttin, and P. Guillemin. 3D coupling of Monte Carlo neutronics and thermal-hydraulics calculations as a simulation tool for innovative reactor concepts. In *Proc. Global 2009*, Paris, France, Sept. 6–11 2009.

[6] 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 *Proc. M&C 2011*, Rio de Janeiro, Brazil, May 8–12 2011.

[7] D. Kotlyar, Y. Shaposhnik, E. Fridman, and E. Shwageraus. Coupled neutronic thermo-hydraulic analysis of full PWR core with Monte-Carlo based BGCore system. *Nuclear Engineering and Design*, **241**, 3777–3786, 2011.

[8] Jaakko Leppänen, Maria Pusa, Tuomas Viitanen, Ville Valtavirta, and Toni Kaltiaisenaho. The Serpent Monte Carlo code: Status, development and applications in 2013. *Annals of Nuclear Energy*, **82**, 142 – 150, 2015.

[9] J. Lamarsh. *Introduction to Nuclear Engineering, Massachusetts (1983).* Addison-Wesley Publishing Company, Massachussets, 1983.

[10] Weston M. Stacey. *Nuclear Reactor Physics*. Wiley-VCH, 6 2007.

[11] Bernd Pfeiffer, Karl-Ludwig Kratz, and Peter Möller. Status of delayed-neutron precursor data: half-lives and neutron emission probabilities. *Progress in Nuclear Energy*, **41**, 39 – 69, 2002.

[12] R. E. Kelley and E. D. Clayton. Fissible: A proposed new term in nuclear engineering. *Letter to the Editor, Nucl. Sci. Eng.*, **91**, 481, 1985.

[13] L.W.G. Morgan and D. Kotlyar. Weighted-delta-tracking for Monte Carlo particle transport. *Annals of Nuclear Energy*, **85**, 1184 – 1188, 2015.

[14] E. Woodcock, T. Murphy, P. Hemmings, and S. Longworth. Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. Technical report, Argonne National Laboratory, 1965.

[15] W. A. Coleman. Mathematical verification of a certain Monte Carlo sampling technique and applications of the technique to radiation transport problems. *Nucl. Sci. Eng.*, **32**, 76–81, 1968.

[16] Massachussets Institute of Technology. Theory and methodology description of the OpenMC Monte Carlo code: https://mit-crpg.github.io/openmc/methods/physics.html, cited on 23 Nov. 2016.

[17] K. Ivanov et al. Benchmark for uncertainty analysis in modelling (UAM) for the design, operation and safety analysis of LWRs, volume I: Specification and support data for neutronics cases (Phase I). OECD/NEA, 2013.

[18] R. Pawlowski, R. Bartlett, N. Belcourt, R. Hooper, and R. Schmidt. A theory manual for multi-physics code coupling in LIME. Technical report, Sandia National Laboratories, 2011.

[19] D. A. Knoll and D. E. Keyes. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *J. Comput. Phys.*, **193**, 357, 2004.

[20] T.M. Evans, T.J. Urbatsch, H. Lichtenstein, and J.E. Morel. A residual Monte Carlo method for discrete thermal radiative diffusion. *Journal of Computational Physics*, **189**, 539 – 556, 2003.

[21] J. A. Willert, C. T. Kelley, D. A. Knoll, and H. Park. A hybrid approach to the neutron transport k-eigenvalue problem using NDA-based algorithms. In *Proc. M&C 2013*, Sun Valley, ID, 5–9 May 2013.

[22] Jeffrey Willert and H. Park. Residual Monte Carlo high-order solver for Moment-Based Accelerated Thermal Radiative Transfer equations. *Journal of Computational Physics*, **276**, 405 – 421, 2014.

[23] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statist.*, **22**, 400, 1951.

[24] J. Dufek and W. Gudowski. Stochastic approximation for Monte Carlo calculation of steady-state conditions in thermal reactors. *Nuclear Science and Engineering*, **152**, 274 – 283, 2006.

[25] R. E. MacFarlane and D. W. Muir. *NJOY99.0 Code System for Producing Pointwise and Multigroup Neutron and Photon Cross Sections from ENDF/B Data*. PSR-480, Los Alamos National Laboratory, 2000.

[26] T. Trumbull. Treatment of nuclear data for transport problems containing detailed temperature distributions. *Nucl. Tech.*, **156**, 75, 2006.

[27] J. Donnelly. Interpolation of temperature-dependent nuclide data in MCNP. *Nucl. Sci. Eng.*, **168**, 180–184, 2011.

[28] S. Li, K. Wang, and G. Yu. Research on Fast-Doppler-Broadening of neutron cross sections. In *PHYSOR 2012*, Knoxville TN, Apr. 15-20, 2012.

[29] C. J. Dean, R. Perry, R. Neal, and A. Kyrieleis. Validation of run-time Doppler broadening in MONK with JEFF3.1. *J. Kor. Phys. Soc.*, **59**, 1162, 2011.

[30] Paul K. Romano and Timothy H. Trumbull. Comparison of algorithms for doppler broadening pointwise tabulated cross sections. *Annals of Nuclear Energy*, **75**, 358, 2015.

[31] W. R. M. G. Yesilyurt and F. B. Brown. On-the-fly Doppler broadening for Monte Carlo codes. *Nucl. Sci. Eng.*, **171**, 239 – 257, 2012.

[32] Benoit Forget, Sheng Xu, and Kord Smith. Direct Doppler broadening in Monte Carlo simulations using the multipole representation. *Ann. Nucl. Energy*, **64**, 78 – 85, 2014.

[33] T. Viitanen. *Development of a stochastic temperature treatment technique for Monte Carlo neutron tracking*. PhD thesis, Aalto University, 2015.

[34] T. Viitanen and J. Leppänen. Explicit treatment of thermal motion in continuous-energy Monte Carlo tracking routines. *Nucl. Sci. Eng.*, **171**, 165 – 173, 2012.

[35] J. Leppänen. Modeling of non-uniform density distributions in the Serpent 2 Monte Carlo code. *Nucl. Sci. Eng.*, **174**, 318, 2013.

[36] Matt Ellis, Benoit Forget, Kord Smith, and Derek Gaston. Preliminary coupling of the Monte Carlo code OpenMC and the Multiphysics Object-Oriented Simulation Environment (MOOSE) for analyzing Doppler feedback in Monte Carlo simulations. In *Proc. M&C + SNA + MC 2015*, Nashville, TN, 19–23 Apr. 2015.

[37] B. Sjenitzer. *The Dynamic Monte Carlo Method for Transient Analysis of Nuclear Reactors*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2013.

[38] J. Leppänen. Development of a dynamic simulation mode in Serpent 2 Monte Carlo code. In *Proc. M&C 2013*, Sun Valley, ID, 5–9 May 2013.

[39] J. Leppänen, T. Viitanen, and V. Valtavirta. Multi-physics coupling scheme in the Serpent 2 Monte Carlo code. *Trans. Am. Nucl. Soc.*, **107**, 1165, 2012.

[40] T. Ikonen, V. Tulkki, E. Syrjälahti, V. Valtavirta, and J. Leppänen. FINIX–fuel behavior model and interface for multiphysics applications. In *Proc. 2013 Fuel Performance Meeting TopFuel*, Charlotte, USA, 2013.

[41] Timo Ikonen, Henri Loukusa, Elina Syrjälahti, Ville Valtavirta, Jaakko Leppänen, and Ville Tulkki. Module for thermomechanical modeling of LWR fuel in multiphysics simulations. *Annals of Nuclear Energy*, **84**, 111 – 121, 2015.

[42] W. J. Kilgour, J. A. Turnbull, R. J. White, A. J. Bull, P. A. Jackson, and I. D. Palmer. Capabilities and validation of the ENIGMA fuel performance code. In *Proc. ANS/ENS Int. Topical Meeting on LWR Fuel Performance*, Avignon, France, 21–24 April 1991.

[43] J. Dufek and H. Anglart. Derivation of a stable coupling scheme for Monte Carlo burnup calculations with the thermal-hydraulic feedback. *Ann. Nucl. Energy*, **62**, 260, 2013.

[44] G. Grandi, K. Smith, Z. Xu, and J. Rhodes. Effect of CASMO-5 cross-section data and Doppler temperature definitions on LWR reactivity initiated accidents. In *PHYSOR 2010*, Pittsburgh, PA, May 9-14, 2010.

[45] G. Rowlands. Resonance absorption and non-uniform temperature distributions. *Journal of Nuclear Energy. Parts A/B. Reactor Science and Technology*, **16**(4), 235 – 236, 1962.

[46] W. J. M. De Kruijf and A. J. Janssen. The effective fuel temperature to be used for calculating resonance absorption in a 238UO2 lump with a nonuniform temperature profile. *Nuclear Science and Engineering*, **123**, 121 – 135, 1996.

[47] A.O. Goltsev, V.D. Davidenko, V.F. Tsibulsky, and A.A. Lekomtsev. Computational problems in the calculation of temperature effects for heterogeneous nuclear reactor unit cells. *Annals of Nuclear Energy*, **27**, 175 – 183, 2000.

[48] G. Sengler, F. Forêt, G. Schlosser, R. Lisdat, and S. Stelletta. EPR core design. *Nuclear Engineering and Design*, **187**, 79 – 119, 1999.

[49] D. Kotlyar and E. Shwageraus. Stochastic semi-implicit substep method for coupled depletion Monte-Carlo codes. *Annals of Nuclear Energy*, **92**, 52 – 60, 2016.

[50] D. Kotlyar and E. Shwageraus. Numerically stable Monte Carlo-burnup-thermal hydraulic coupling schemes. *Annals of Nuclear Energy*, **63**, 371 – 381, 2014.

[51] D. Kotlyar and E. Shwageraus. Sub-step methodology for coupled Monte Carlo depletion and thermal hydraulic codes. *Annals of Nuclear Energy*, **96**, 61 – 75, 2016.

[52] Manuele Aufiero, Adrien Bidaud, Mathieu Hursin, Jaakko Leppänen, Giuseppe Palmiotti, Sandro Pelloni, and Pablo Rubiolo. A collision history-based approach to sensitivity/perturbation calculations in the continuous energy Monte Carlo code SERPENT. *Annals of Nuclear Energy*, **85**, 245 – 258, 2015.

[53] Manuele Aufiero, Michael Vincent Martin, Massimiliano Fratoni, and Stefano Lorenzi. Analysis of the coolant density reactivity coefficient in LFRs and SFRs via Monte Carlo perturbation/sensitivity. In *Proc. PHYSOR 2016*, Sun Valley, ID, May USA, May 1–5 2016.

[54] Karl Ott. Quasistatic treatment of spatial phenomena in reactor dynamics. *Nucl. Sci. Eng.*, **26**, 563–565, 1966.

[55] K.L. Goluoglu, R. V. Demeglio, R. E. Pevey, I. Suslov, H. L. Dodds, C.L. Bentley, and S. Goluoglu. Application of TDKENO for analysis of criticality excursion experiments. *Transactions of the American Nuclear Society*, **78**, 145–146, 1998.

[56] S. Goluoglu and H. L. Dodds. A time-dependent, three-dimensional neutron transport methodology. *Nucl. Sci. Eng.*, **139**, 248–261, 2001.

[57] T. Kaltiaisenaho. Implementing a photon physics model in Serpent 2. Master's thesis, Aalto University, School of Science, 2016.

[58] J. Leppänen, T. Kaltiaisenaho, V. Valtavirta, and M. Metsälä. Development of a coupled neutron / photon transport mode in the Serpent 2 monte carlo code. In *M&C 2017*, Jeju, Korea, April 16-20 2017.

[59] M. Metsälä. Accurate heat deposition calculations in nuclear reactors using Monte Carlo neutron and photon transport. Special Assignment, Aalto University, School of Science, 2016.

[60] Miriam Knebel, Luigi Mercatali, Victor Sanchez, Robert Stieglitz, and Rafael Macian-Juan. Validation of the Serpent 2-DYNSUB code sequence using the Special Power Excursion Reactor Test III (SPERT III). *Annals of Nuclear Energy*, **91**, 79 – 91, 2016.

[61] Jaakko Leppänen and Riku Mattila. Validation of the Serpent–ARES code sequence using the MIT BEAVRS benchmark – HFP conditions and fuel cycle 1 simulations. *Annals of Nuclear Energy*, **96**, 324 – 331, 2016.

[62] Jaakko Leppänen, Riku Mattila, and Maria Pusa. Validation of the Serpent–ARES code sequence using the MIT BEAVRS benchmark – initial core at HZP conditions. *Annals of Nuclear Energy*, **69**, 212 – 225, 2014.

[63] Jaakko Leppänen, Maria Pusa, and Emil Fridman. Overview of methodology for spatial homogenization in the Serpent 2 Monte Carlo code. *Annals of Nuclear Energy*, **96**, 126 – 136, 2016.

[64] V. Valtavirta. Designing and implementing a temperature solver routine for Serpent. Master's thesis, Aalto University, School of Science, 2012.

[65] V. Valtavirta. Internal neutronics temperature coupling in Serpent 2 – reactivity differences resulting from choice of material property correlations. In *Proc. M&C 2013*, Sun Valley, ID, May 5–9 2013.

Errata

Publication **II**:
-The third sentence of section IV.A. ”The user input consists of a list of three variables, the cold condition radial coordinate of the point, the hot coordinate radial coordinate of the point and the temperature at the point” should read ”The user input consists of a list of three variables, the cold condition radial coordinate of the point, the hot *condition* radial coordinate of the point and the temperature at the point”.

Publication **III**:
-The first sentence of section 2.1 ”The dynamic simulation mode was implemented in an earlier version of Serpent 2, as an extension of the conventional time-dependent criticality source mode.” should read ”The dynamic simulation mode was implemented in an earlier version of Serpent 2, as an extension of the conventional time-dependent *external* source mode.”

-Two references show up as question marks in section 2.1. The intended references were [64] and [65].

-The first sentence of the third paragraph of section 2.2 starting ”The TMS-method was introduced for the first time in english-speaking journals...” should start ”The TMS-method was introduced for the first time in English-speaking journals...”

-The reader should note that the dashes in place of the authors’ names in the list of references indicate that the authors of that paper were the same as in the previous reference.

Publication **IV**:
-The reader should note that the dashes in place of the authors’ names in the list of references indicate that the authors of that paper were the same as in the previous reference.

Publication **V**:
-The denominator of Eq. 11 has an additonal $\phi(r)$ term. Moreover, the variable r is missing from both the numerator and denominator of Eqs. 11 and 12. The equations should thus read: Equation 11

$$T_{\text{eff.}} = \frac{\int_r T(r) r \mathrm{d}r}{\int_r r \mathrm{d}r}$$

and Equation 12:

$$T_{\text{eff.}} = \frac{\int_r w(r) T(r) r \mathrm{d}r}{\int_r w(r) r \mathrm{d}r}.$$

-In the first sentence of the first text paragraph on page 60, the symbol ^{283}U should read ^{238}U.

Publication I

V. Valtavirta, T. Viitanen and J. Leppänen. Internal neutronics–temperature coupling in Serpent 2. *Nuclear Science and Engineering*, 177, pp. 193-202, https://dx.doi.org/10.13182/NSE13-3, 2014.

Internal Neutronics-Temperature Coupling in Serpent 2

Ville Valtavirta,* Tuomas Viitanen, and Jaakko Leppänen

VTT Technical Research Centre of Finland
P.O. Box 1000, FI-02044 VTT, Finland

Received January 10, 2013
Accepted October 4, 2013
http://dx.doi.org/10.13182/NSE13-3

Abstract—*This paper describes the built-in calculation routines in the reactor physics code Serpent 2 that provide a novel method for solving the coupled problem of the power distribution, temperature distribution, and material property distributions in nuclear fuel elements. All of the coupled distributions are solved during a single simulation with no coupling to external codes. The temperature feedback system consists of three separate built-in parts: an explicit treatment of the thermal motion of target nuclides during the transport calculation, an internal analytic radial temperature profile solver, and internal material property correlations.*

The internal structure and couplings of the calculation routines are described in detail, after which the results of an assembly-level problem are presented to demonstrate the capabilities and functionality of the system.

I. INTRODUCTION

The Serpent Monte Carlo code[a] is currently used in more than 90 universities and research organizations for reactor physics applications ranging from homogenized group constant generation to burnup calculations and the modeling of small research reactor cores. In addition to the publicly distributed version available at the Organisation for Economic Co-operation and Development/Nuclear Energy Agency Data Bank and the Radiation Safety Informational Computation Center, an entirely new version of the code, Serpent 2, is currently under development at the VTT Technical Research Centre of Finland. Considerable effort in Serpent development is currently devoted to multiphysics, i.e., the coupling of continuous-energy Monte Carlo neutron transport simulation to state-of-the art methods in thermal hydraulics and fuel research.

The multiphysics calculation scheme in Serpent 2 is designed to handle coupling to computational fluid dynamics, system-scale thermal-hydraulics codes, and fuel performance codes at two levels.[1] A universal multiphysics interface, based on the capability to model continuous nonuniform temperature and density distributions,[2,3] is used to handle data transfer between Serpent and any externally coupled solver code. In addition to the external coupling, a set of built-in calculation routines can be used to solve the temperature distributions inside the fuel as the neutronics simulation is run. This internal temperature feedback module is the topic of this paper. The work is still under way, and the methodology is intended to be used for both complementing the external coupling and acting as a stand-alone temperature solver.

Neutronics calculations typically do not explicitly consider the coupling of the neutronics solution to the fuel temperature and behavior and their effect on the original solution. The coupled solution can, however, be obtained by alternately solving the neutronics and the fuel temperature and behavior with dedicated codes until convergence is found.[4–6] Typically, the simultaneous modeling of neutronics, fuel temperature, and temperature-related material properties requires at least two separate code systems with a distinct wrapper code that handles the coupling of the codes and the expertise to use these codes. It is thus no wonder if a reactor physicist considers the coupled solution to be outside the scope of his or her study and chooses to disregard the temperature effects.

*E-mail: Ville.Valtavirta@vtt.fi

[a] For a complete description of the code and the latest news, see the project Web site: http://montecarlo.vtt.fi.

Serpent 2 offers no instant cure for the segregation of the reactor physics and fuel-modeling branches of nuclear science and engineering. It will however permit users to better consider temperature-related effects on their reactor physics calculations. The temperature feedback module can be used to address the various effects that the fuel temperature has on neutronics, such as the negative effect on reactivity. Serpent 2 is exceptionally well suited for such studies based on its capability to model continuous nonuniform temperature distributions without the memory limitations related to storing cross-section data at multiple temperatures. The module solves the temperature distribution inside the fuel pins in the problem geometry as well as the associated thermal conductivities, thermal expansions, and pellet relocation for the different material zones in the pins. The solution is updated after each neutron batch.

Section II gives the reader an overview of the coupled problem between the power distribution, temperature distribution, and material property distributions in nuclear fuel elements. It also presents the solution flow between the different internal parts of Serpent 2. The next three sections give a more detailed description of the three main parts of the temperature feedback system, namely, the on-the-fly temperature treatment in Sec. III, the internal temperature solver in Sec. IV, and the internal material property correlations in Sec. V. The capabilities of the module are presented through an assembly-level calculation in Sec. VI, and finally a summary and some discussion on the implications of this work are given in Sec. VII.

II. OVERVIEW

The power distribution in a nuclear fuel pin is intrinsically coupled to the temperature distribution. As the temperature distribution depends on the temperature-dependent thermal conductivities and dimensions of the pin, a method is needed for simultaneously solving the power, temperature, and thermal conductivity distributions as well as the dimensions of the pin. Traditionally, this solution has been obtained by solving the power distribution with a neutronics code, passing the solution to a fuel performance or a thermal hydraulics code, and updating the neutronics input according to the results obtained from these codes. The different types of codes are executed in turn until a convergent solution is found.[4–6] This traditional way of obtaining the coupled solution can waste much time in obtaining convergent neutronics solutions for configurations far from the final solution.

The novel method of obtaining the coupled solution, implemented in Serpent 2, is based on an internal temperature solver and material property correlations for the neutronics code. This permits solving the material properties and the temperature distribution after each neutron batch, rather than after each neutronics solution. When both the fission source and the temperature distribution are allowed to converge at the same time, during the inactive cycles, a significant reduction of simulation time is obtained.

II.A. Solution Flow

Figure 1 shows the solution flow between the neutron transport, material property, and temperature solver parts of Serpent 2. The power distribution is passed to the material property module, which uses the power distribution along with the temperature distribution from the previous iteration to update the thermal conductivities and dimensions of different material zones. The power distribution is used for the calculation of the initial pellet relocations while the temperature distribution is needed for the thermal conductivities and strains. On the first iteration, the user-given cladding outer temperature is used throughout the pin. The updated material properties are then used together with the power distribution to obtain a new analytical solution for the temperature distribution. Finally, this temperature distribution is saved to be used in the simulation of the next batch of neutrons.

The routine works in the two-dimensional fuel pin geometry type of Serpent, and it will be later extended to three dimensions and other geometry types. The power density and the material properties are assumed to be constant in each material zone. Therefore, the division of the pellet into multiple concentric rings increases the accuracy of the solution.

III. TARGET MOTION SAMPLING TEMPERATURE TREATMENT TECHNIQUE

The simultaneous solution of neutronics and the temperature distributions requires frequent updating of the

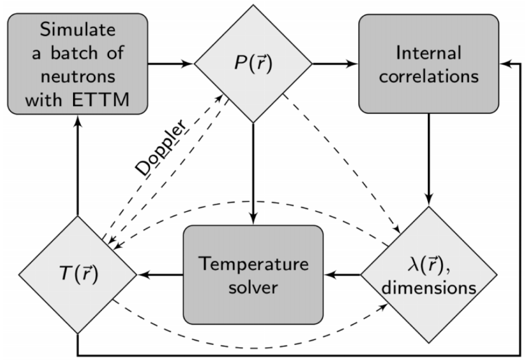


Fig. 1. Internal coupling between neutron transport, internal correlations, and temperature solver parts of Serpent. Solid lines indicate data flow, and dashed lines indicate feedbacks.

material temperatures. Hence, also the cross-section temperatures need to be updated after each temperature iteration—in the current implementation after each neutron batch. Since calculating new effective cross sections for each nuclide and each temperature between the neutron batches would be a disaster in terms of performance, on-the-fly temperature treatment techniques need to be introduced for the calculation to become feasible.

The target motion sampling (TMS) temperature treatment technique is a promising method for taking the effect of thermal motion on reaction probabilities into account on the fly during Monte Carlo neutron tracking. It is basically a rejection-based tracking technique in which the target temperatures are adjusted stochastically at each collision site. The method is being developed as a part of Serpent 2 and is adopted in the current work.

In the TMS technique, the neutron path lengths are sampled based on a majorant cross section, which takes into account the variation of cross sections within the range of thermal motion. More precisely, the nuclidewise majorant cross sections for neutron energy E are greater than or equal to the maximum total cross section within a thermal motion–dependent energy range around E, and the sum over the nuclidewise majorants is, in fact, used in the sampling of path lengths. At each collision point, the target nuclide is first sampled based on the ratios of nuclidewise majorants, and then, the target velocity is sampled from a Maxwellian-based distribution. The collision point is accepted or rejected according to the ratio of the total cross section of the target nuclide at the target-at-rest energy to the nuclidewise majorant at E. In the case of a rejected collision, a new path length is sampled starting from the newly rejected collision point candidate. If the collision point is accepted, a reaction is sampled using the reaction cross sections at the target-at-rest energy, and the neutron tracking proceeds according to the sampled reaction.

The total cross section and all other nonmajorant cross sections are at the so-called basis temperature of the cross sections, which can be chosen arbitrarily as long as it remains smaller than or equal to the minimum temperature of the materials to which the TMS method is applied. It should be noted that all the cross sections remain unchanged during the transport simulation regardless of constant updating of the material temperatures by the thermal feedback system.

The TMS method is, as a novel feature, capable of modeling continuous temperature distributions such that the tracking routine sees temperatures only through an arbitrary, spatially dependent variable $T(\vec{r})$. This property makes TMS particularly convenient to use in the current study: After resolving the functional form of the temperature distribution, it can be used in the tracking routine as is, without the need to approximate the continuous distribution with a step function, which would be the case with other current neutron-tracking methods.

The TMS method was first introduced in Ref. 2 under the name "explicit treatment of target motion." In this very first implementation, the basis temperature was 0 K, and hence, the target velocities sampled during the transport calculation corresponded to the actual, or "explicit," motion of target nuclei. However, it was noticed in Refs. 7 and 8 that the TMS method with 0 K basis is relatively slow compared to transport with prebroadened cross sections and that the efficiency of the method could be significantly increased by elevating the basis temperature above 0 K. The theory behind the elevation of the basis temperature is discussed further in Ref. 9.

IV. TEMPERATURE SOLVER

Solving the temperature distribution inside the fuel pin requires solving the heat equation

$$\frac{\partial T(\vec{r},t)}{\partial t} = \nabla \cdot [\alpha(\vec{r},t) \nabla T(\vec{r},t)] + \frac{1}{c_p(\vec{r},t)\rho(\vec{r},t)} q'''(\vec{r},t) \ , \quad (1)$$

where

$T(\vec{r},t)$ = temperature distribution

$q'''(\vec{r},t)$ = heat generation density

$\alpha(\vec{r},t)$ = thermal diffusivity of medium

$c_p(\vec{r},t)$ = specific heat capacity of medium

$\rho(\vec{r},t)$ = density of medium.

This routine solves the heat equation in each material zone by assuming axial and polar symmetry as well as independence of all variables from time. Furthermore, the heat generation density and the thermal conductivity of the medium are assumed to be constant inside each material zone.

With these assumptions, the heat equation for ring i reduces to

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial}{\partial r} T^i(r) \right) + \frac{(q''')^i}{\lambda^i} = 0 \ , \quad (2)$$

where λ^i is the thermal conductivity of ring i. This equation has a unique analytical solution if continuity of the temperature distribution is required and the cladding outer temperature is used as a boundary condition. The general solution for Eq. (2) is

$$T^i(r) = C_0^i r^2 + C_1^i \ln r + C_2^i \ , \quad (3)$$

where C_0^i, C_1^i, and C_2^i are constants depending on the local thermal conductivity and heat generation density as well as the temperature on the outer material boundary and the heat flux at the inner boundary.

At the outermost zone boundary, the temperature is specified by the user and can be expressed as

$$T(R^i_{out}) = C^i_0 (R^i_{out})^2 + C^i_1 \ln R^i_{out} + C^i_2 = T^i_{out} \ , \qquad (4)$$

which is a user-given constant. Furthermore, we can express the inner temperature in a similar manner:

$$T(R^i_{in}) = C^i_0 (R^i_{in})^2 + C^i_1 \ln R^i_{in} + C_2 \ , \qquad (5)$$

and finally we note that in a steady state, the heat flux through the inner surface of the material zone must equal the total power generated in the material zones inside the inner surface, q^i_{in}, divided by the inner surface area:

$$-\lambda^i \nabla_{r=R^i_{in}} T = \frac{q^i_{in}}{2\pi R^i_{in} h^i} = \frac{P^i_{in}}{2\pi R^i_{in}} \ , \qquad (6)$$

where P^i_{in} denotes the total linear power generated inside the inner surface $r = R^i_{in}$. Equation (2) along with Eqs. (4), (5), and (6) are sufficient constraints to solve for the four unknowns, yielding

$$T^i_{in} = T^i_{out} + \frac{P^i}{4\pi\lambda^i} + \left(\frac{P^i_{in}}{2\pi\lambda^i} - \frac{P^i}{2\pi\lambda^i} \frac{(R^i_{in})^2}{[(R^i_{out})^2 - (R^i_{in})^2]} \right) \ln(R^i_{out}/R^i_{in}) \ ,$$

$$C^i_0 = -G^i \ ,$$

$$C^i_1 = \frac{G^i[(R^i_{out})^2 - (R^i_{in})^2] + (T^i_{out} - T^i_{in})}{\ln(R^i_{out}/R^i_{in})} \ ,$$

and

$$C^i_2 = \frac{[T^i_{in} + G^i(R^i_{in})^2] \ln R^i_{out} - [T^i_{out} + G^i(R^i_{out})^2] \ln R^i_{in}}{\ln(R^i_{out}/R^i_{in})} \ ,$$

where

$$G^i = \frac{P^i}{4\pi[(R^i_{out})^2 - (R^i_{in})^2]\lambda^i} \qquad (7)$$

and P^i is the linear power of the material zone i. As the inner temperature of ring i is the outer temperature of ring $i-1$, the temperatures can be solved from outside in.

Since the logarithmic term diverges at the centerline, the central material zone has to be treated as a special case. Applying the zero-heat-flux constraint at the centerline $C^1_1 = 0$ yields new formulas for the unknowns:

$$T^1_{in} = T^i_{out} + \frac{P^i}{4\pi\lambda^i} \ ,$$

$$C^1_0 = -\frac{P^i}{4\pi (R^i_{out})^2 \lambda^i} \ ,$$

$$C^1_1 = 0 \ ,$$

and

$$C^1_2 = T^i_{out} + \frac{P^i}{4\pi\lambda^i} \quad .$$

The temperature feedback routine solves for the constants C_i at each material zone thus obtaining a complete analytical solution for the temperature distribution in the pin. The constants are then stored for an easy and fast calculation of the temperature at any given point of interaction, with minimal memory usage.

V. MATERIAL PROPERTIES

Since the temperature distribution inside a fuel pin depends strongly on the local thermal conductivity as well as the dimensions of the material zones, most notably the width of the gas gap, it is imperative that this side of the temperature feedback is also addressed. To calculate the updated material properties, internal correlations were implemented for the thermal conductivities and thermal expansions of different materials, as well as for the initial pellet relocation. To offer a good comparability of the results, the correlations implemented were those used in the FRAPCON/FRAPTRAN code families as described in Ref. 10. The reactivity changes associated with the selection of different correlations are addressed separately.[11]

V.A. Thermal Conductivities

In this paper, only the temperature dependencies of the thermal conductivities are considered. The thermal conductivities for each material zone λ_i are averaged with respect to temperature:

$$\lambda_i = \frac{1}{T^{in}_i - T^{out}_i} \int_{T^{out}_i}^{T^{in}_i} \lambda(\Theta)\, d\Theta \ , \qquad (8)$$

where T^{in}_i and T^{out}_i are the inner and outer boundary temperatures of the material zone, respectively, and $\lambda(\Theta)$ is the functional form of the temperature-dependent thermal conductivity, with the temperature Θ as an integral variable. This is probably not the optimal method of averaging, but it was chosen at this stage for the sake of simplicity.

For this study, the fuel was assumed to be nondoped, fresh UO_2 fuel at 95% of theoretical density. The thermal conductivity of the fuel thus follows the correlation

$$\lambda = \frac{1}{A+BT} + \frac{C}{T^2}\exp\left(-\frac{D}{T}\right) , \tag{9}$$

where the temperature is given in kelvin, the result is in W/(m·K), and the constants A through D are

$$A = 0.0452 \text{ m·K/W} ,$$

$$B = 2.46 \times 10^{-4} \text{ m·K/(W·K)} ,$$

$$C = 3.5 \times 10^{9} \text{ W·K/m} ,$$

and

$$D = 16\,361 \text{ K} .$$

The cladding material used in this study was Zircaloy, with a thermal conductivity of

$$\lambda = 7.51 + 2.09 \times 10^{-2}T + 1.45 \times 10^{-5}T^2 + 7.67 \times 10^{-9}T^3 , \tag{10}$$

where the temperature is given in kelvins and the result is in W/(m·K). The minor contributions to thermal conductivity by the surface oxide layers were disregarded for simplicity.

The gas in the diametral gap was assumed to be pure helium with a thermal conductivity of

$$\lambda = 2.531 \times 10^{-3}T^{0.7146} , \tag{11}$$

where again the temperature is given in kelvins and the result is in W/(m·K). The radiative heat transfer over the gas gap was disregarded for simplicity as it can be considered to be mostly insignificant in normal operating conditions. It could, however, be easily accounted for in the routine.

V.B. Thermomechanical Changes in Geometry

The correlation used for calculating the radial thermal expansion of the fuel part was

$$\epsilon = \frac{\Delta R}{R_0} = K_1T - K_2 + K_3 \exp\left(-\frac{E_D}{kT}\right) , \tag{12}$$

where the temperature is given in kelvins, k is the Boltzmann's constant, and the constants are

$$K_1 = 9.8 \times 10^{-6} \text{ K}^{-1} ,$$

$$K_2 = 2.61 \times 10^{-3} ,$$

$$K_3 = 3.16 \times 10^{-1} ,$$

and

$$E_D = 1.32 \times 10^{-19} \text{ J} .$$

For the Zircaloy cladding, the thermal expansion was modeled as linear:

$$\epsilon = \frac{\Delta R}{R_0} = \alpha(T - T_0) , \tag{13}$$

with the coefficient of thermal expansion $\alpha = 6.7210 \times 10^{-6}$. The reference temperature T_0 was set to 300 K.

As the temperature distribution is typically not constant in each material zone, the thermal strain was summed over the region by integrating:

$$\Delta R_{out} = \Delta R_{in} + \int_{R_{in}}^{R_{out}} \epsilon(T(r))\, dr , \tag{14}$$

where ΔR_{out} and ΔR_{in} are the total displacements of the material zone boundaries due to thermal expansion in the whole pellet, and the thermal strain ϵ [from Eq. (12) for fuel and Eq. (13) for cladding] is integrated numerically over the material zone, from R_{in} to R_{out}, with the functional form of the temperature distribution $T(r)$ presented in Eq. (3).

Finally, the cracking of the pellet during the initial rise to power, i.e., the pellet relocation,[12] is modeled according to the FRAP model as a change in the gas gap width G relative to the as-fabricated gas gap width G_0:

$$\Delta G/G_0 = \begin{cases} 0.30, & LHR < 20 \text{ kW/m}, \\ 0.28 + 0.0025(LHR - 20), & 20 \text{ kW/m} \le LHR < 40 \text{ kW/m}, \\ 0.32, & 40 \text{ kW/m} \le LHR, \end{cases}$$

where the linear heat rate LHR is given in kW/m.

To focus on the temperature effect of the feedback, the updated dimensions of the radial zones were not passed to the neutronics model in this study. They could, however, be updated to also take in account the effect of the geometry changes in the neutronics solution. In that case, the densities of the materials should also be updated to conserve their mass.

VI. DEMONSTRATION RESULTS

VI.A. Problem Definition

The functionality and the capabilities of the code system are illustrated by an assembly-level calculation. First, the internal power, temperature, and material property distributions of three pins of interest are analyzed in more detail; second, the integral results of the assembly calculation are compared with results obtained without the temperature feedback system.

The geometry and the material properties of the assembly are taken from the uranium dioxide boiling water reactor (BWR) case of the "Benchmark Problem Suite for Reactor Physics Study of LWR Next Generation Fuels."[13] To allow for a more realistic simulation, a gas gap was inserted into the fuel pins by reducing the pellet outer radius by 0.0050 cm. The fuel density was scaled up to preserve the total fuel mass in the problem.

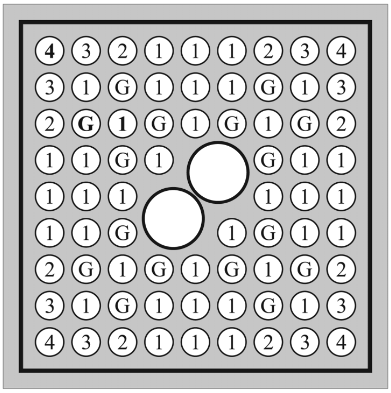


Fig. 2. Geometry of the BWR assembly used in the test case. The labels of the pins refer to the fuel types presented in Table II, with the bold labels indicating the three pins analyzed in greater detail.

TABLE I

Geometric Dimensions of the Simulated Assembly*

	Fuel Pins	Coolant Channels
Cladding outer diameter	1.120	2.490
Cladding inner diameter	0.980	2.350
Pellet outer diameter	0.970	n/a
Pin pitch	1.440	
Assembly pitch	15.240	
Channel box inner width	13.400	
Channel box outer width	13.900	

*In units of centimeters.

The assembly geometry is shown in Fig. 2, and its dimensions are given in Table I. The labels in the pins refer to the different fuel types presented in Table II, with the bold labels in Fig. 2 indicating the three pins to be discussed in more detail. The structural material in the materials table covers the pin cladding, the water rod cladding, and the channel box.

The neutron flux was normalized to correspond to a power of 25 W/g U $\approx$ 12.2 kW/cm. The simulation consisted of 4000 neutron batches, with one million neutrons in each batch. The number of inactive iteration cycles was set to 200. Such a large number of histories for a two-dimensional assembly calculation was used to obtain good statistical accuracy for the multiplication factor. Reflective boundary conditions were used. The fuel part of the pins was divided radially into ten zones of equal volume, and the cladding outer temperature was set to be 600 K for all of the pins.

VI.B. Verification of the Temperature Solution

To ascertain that the implemented temperature solver yields realistic results in this case, a comparison calculation was done against FRAPCON-3.4, with the aforementioned problem geometry. It should be noted that the temperature solver implemented for this study will be replaced with a more realistic fuel behavior module, which will be validated more thoroughly against experimental data. The temperature solution was calculated for rod type 1. The length of the rod was set to 3.0 m in the FRAPCON calculation, the fill gas pressure was set to 3 bars, and the plenum length was set to 5 cm. The cladding outer temperature was set to 600 K in both cases.

The Serpent calculation was done with 25 inactive cycles to converge the source distribution as well as the thermomechanical solution. The temperature for the comparison was then calculated from a power distribution tallied with a single batch of 30 million neutrons.

Figure 3 shows the relative differences in temperatures between the two solutions at linear powers between 10 and 300 W/cm. The results show that Serpent underestimates the gap conductance leading to slightly higher fuel temperatures both at the surface of the fuel and at the centerline. The relative differences stay under 2%, which can be viewed as a good result for such a simplistic model. It should be noted, however, that this does not verify the accuracy of Serpent's temperature solution in different rod geometries.

VI.C. The Coupled Solution

The coupled problem considered in this paper is that of the power distribution, the temperature distribution, and the material properties and geometry in nuclear fuel elements. To demonstrate the obtained solution, the distributions obtained for these coupled variables are now presented for the three pins indicated with bold labels in Fig. 2.

TABLE II

Material Specifications for the Materials in the Simulated Assembly*

Fuel Materials					
	Fuel 1	Fuel 2	Fuel 3	Fuel 4	Gd Fuel
Density (g/cm^3)	10.309	10.309	10.309	10.309	10.003
^{235}U	0.02125	0.01687	0.01350	0.01012	0.01563
^{238}U	0.31208	0.31647	0.31984	0.32321	0.29324
^{154}Gd					0.00063
^{155}Gd					0.00432
^{156}Gd					0.00601
^{157}Gd					0.00460
^{158}Gd					0.00730
^{160}Gd					0.00648
^{16}O	0.66667	0.66667	0.66667	0.66667	0.66178
Enrichment	6.3 wt%	5.0 wt%	4.0 wt%	3.0 wt%	5.0 wt%
Gd_2O_3 content					6.0 wt%

Nonfuel Materials			
	Coolant	Structural Material	Gas Gap
Density (g/cm^3)	0.457	6.53	0.00027
^{1}H	0.66667		
^{16}O	0.33333		
^{90}Zr		0.5150	
^{91}Zr		0.1120	
^{92}Zr		0.1710	
^{94}Zr		0.1740	
^{4}He			1.000

*In atomic fractions.

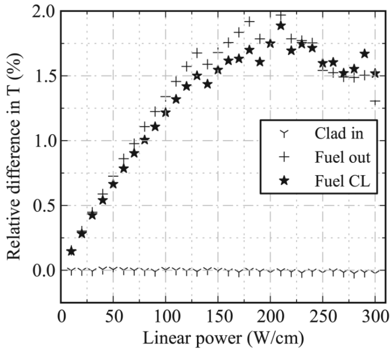
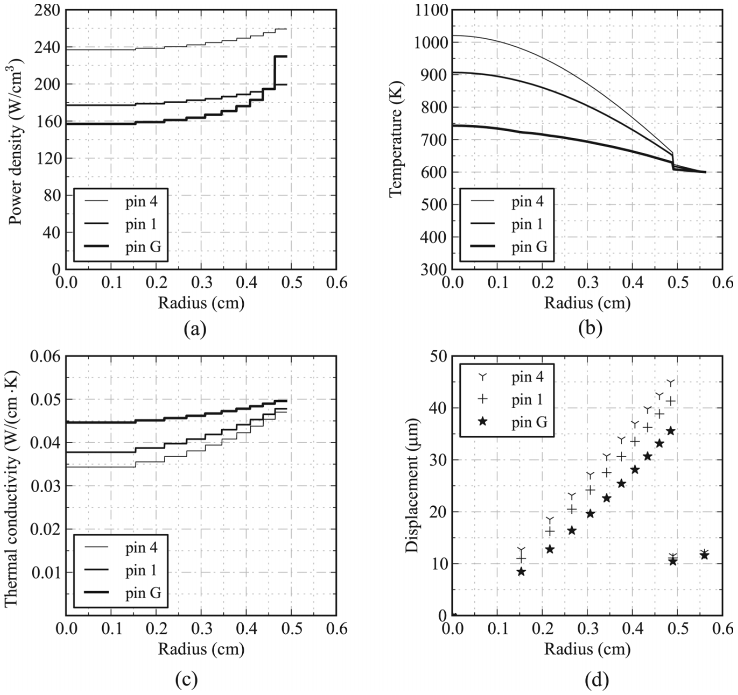


Fig. 3. Relative differences in the temperature solution (Serpent − FRAPCON)/FRAPCON at the cladding inner surface, the fuel outer surface, and the fuel centerline.

The total linear powers of the pins were 185.6 W/cm for pin **4**, 140.1 W/cm for pin **1**, and 132.2 W/cm for pin **G**. The average power densities of the different material zones are shown in Fig. 4a. The self-shielding of the gadolinium-doped pin **G** results in peaking of the power in the surface parts of the pin. The power densities of the two undoped pins have less variation, although the outer parts of the fuel do exhibit a higher power density also in these pins.

The temperature distributions of the pins are shown in Fig. 4b. An interesting result is that although the linear power of pin **1** is only slightly higher than that of pin **G**, the difference in centerline temperatures is significant. This is a direct consequence of the different shapes of the power distributions in the two pins. A considerably larger amount of the total pin power is generated in the inner parts of the pellet for pin **1** than for pin **G**.

The fuel thermal conductivities obtained from the coupled solution are shown in Fig. 4c. The degradation of



(a)

(b)

(c)

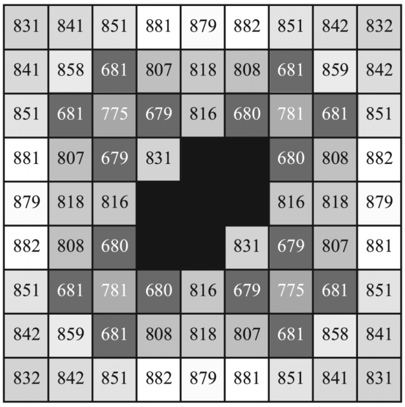
(d)

Fig. 4. Coupled solution of (a) power distributions, (b) temperature distributions, (c) thermal conductivities, and (d) radial displacements in three pins of the assembly.

the fuel thermal conductivity with respect to increasing temperature is clearly seen. The thermal conductivities of helium and cladding in the different pins are not shown here as they did not exhibit large variations due to the temperatures of the outer material zones being close to each other in the different pins.

The average displacements of the material zone boundaries are shown in Fig. 4d. The dependence between the boundary radius and the boundary displacement in the fuel part is slightly sublinear since the outer zones are in a lower temperature and thus do not expand as much as the inner zones.

A similar coupled solution of the power distribution, temperature distribution, thermal conductivities, and geometry is obtained simultaneously for each pin of the assembly. Based on the output of Serpent 2, this type of closer examination can be carried out for whichever pins are included in the temperature feedback consideration. Moreover, the volume-averaged temperatures of the different pins in the assembly can be easily calculated and are shown in Fig. 5 for the fuel parts of the pins.



831	841	851	881	879	882	851	842	832
841	858	681	807	818	808	681	859	842
851	681	775	679	816	680	781	681	851
881	807	679	831			680	808	882
879	818	816				816	818	879
882	808	680			831	679	807	881
851	681	781	680	816	679	775	681	851
842	859	681	808	818	807	681	858	841
832	842	851	882	879	881	851	841	831

Fig. 5. Volume-averaged temperatures of the fuel parts of each pin in the assembly.

VI.D. Comparison with an Effective Temperature Case

To assess the effect of the realistic temperature distributions on the neutronics solution, the assembly problem was recalculated with all of the fuel at a single homogeneous temperature. The homogenized temperature was determined by taking a volume average over the fuel

TABLE III

Results of Quantitative BWR Analysis

	Reference	Homogenized
k_{eff}^{ana}	1.03180 ± 0.00002	1.03143 ± 0.00002
Reactivity (pcm)	3082 ± 2	3047 ± 2
Integral flux	$(5.2474 \pm 0.0001) \times 10^{16}$	$(5.2490 \pm 0.0001) \times 10^{16}$

temperature distribution in the reference case.[b] This yielded a temperature of 804 K. The simulation was normalized to the same power as the coupled case, and a TMS Doppler treatment was used.

Both of the simulations were run with 12 OpenMP threads on a single computer cluster node with 12 Intel Xeon X5690 CPUs at a 3.47-GHz clock frequency. The number of simulated neutron histories was high to produce small statistical uncertainties. The running times of the simulations were 33.8 h for the reference case and 43.8 h for the feedback simulation. The same simulation with homogeneous fuel temperature, preprocessed cross sections, and no target motion sampling ran in 16.5 h. The cross sections used 760 Mbyte of memory in both simulations, and the total memory size was 2.6 Gbytes in the reference simulation and 2.9 Gbytes in the feedback case.

The results from the comparison are listed in Table III. It can be seen that the volume-averaged temperature reproduces the reactivity of the system with a small difference of 35 ± 4 pcm. This might not be the case generally, and the effects of realistic temperature distributions on depletion calculations and different reactor types still have to be assessed. Regardless, this does seem to suggest that a short temperature feedback simulation could, in some cases, be used to calculate an effective temperature for a fuel pin or an assembly to be used in a calculation without the temperature feedback. The current routine also makes it possible to calculate the effective temperature by weighting the temperatures of the system with a variable of interest such as the flux, collision density, or absorption density.

VII. SUMMARY AND DISCUSSION

A novel method for calculating the coupled solution between the power distribution, temperature distribution, thermal conductivities, and pin geometry has been implemented in Serpent 2. The system is based on a temperature solver and material property correlations incorporated directly into the neutronics code. Together with the on-the-fly temperature treatment, this system offers a quick way of obtaining the coupled solution.

The internal temperature solver is based on the analytical solution of the time-independent radial heat equation. The functional form solution has the benefit of allowing the use of a continuous temperature profile by storing only three constant values for each material zone as the recalculation of the temperature at any radius is quick and straightforward. The internal material property correlations used in this work were obtained from the FRAPCON/FRAPTRAN code family. Further correlations may be added to include other effects that affect the heat transfer in the pins, such as cladding creep and pellet swelling.

The capabilities of the temperature feedback module were demonstrated with a simulation of a BWR assembly containing 74 fuel pins monitored by the module. The coupled solutions of the different distributions were presented and discussed for three of the pins. The temperature distribution in the assembly was volume-averaged over the fuel parts of the pins, and a second simulation was carried out with all of the fuel in the assembly at this volume-averaged temperature. The difference in the reactivity of the system was surprisingly small, only 35 ± 4 pcm.

Calculating pin or assembly-level effective temperatures might prove to be an important application of the temperature feedback system. This line of research involves comparing different methods for deriving a simple representative temperature from a continuous nonuniform temperature distribution. This kind of comparison would be easy to achieve with the current routine as it can average the temperatures over a chosen domain weighted with various parameters such as the local flux or absorption density.

The temperature solver will be replaced by a more conventional, nonanalytic, fuel behavior solver. Further development of the temperature feedback system will concentrate on transient heat conduction as well as considering the axial temperature distribution in fuel elements. Further topics to be addressed are the radial and axial binnings of the fission power as a higher number of bins on the one hand increases the spatial detail but on the other hand impairs the statistics.

[b] This can be thought of as an extremely good guess by the user. Using the Rowlands formula[14] might yield an even better guess.

ACKNOWLEDGMENTS

This work has been funded by the NEPAL and KÄÄRME projects under the Finnish National Research Programme on Nuclear Power Plant Safety, SAFIR-2014. The writers would also like to thank P. A. Aarnio for fruitful discussions on fuel temperature–related subjects.

REFERENCES

1. J. LEPPÄNEN, T. VIITANEN, and V. VALTAVIRTA, "Multiphysics Coupling Scheme in the Serpent 2 Monte Carlo Code," *Trans. Am. Nucl. Soc.*, **107**, 1165 (2012).

2. T. VIITANEN and J. LEPPÄNEN, "Explicit Treatment of Thermal Motion in Continuous-Energy Monte Carlo Tracking Routines," *Nucl. Sci. Eng.*, **171**, 165 (2012); http://dx.doi.org/10.13182/NSE11-36.

3. J. LEPPÄNEN, "Modeling of Nonuniform Density Distributions in the Serpent 2 Monte Carlo Code," *Nucl. Sci. Eng.*, **174**, 318 (2013); http://dx.doi.org/10.13182/NSE12-54.

4. J. E. HOOGENBOOM et al., "A Flexible Coupling Scheme for Monte Carlo and Thermal-Hydraulics Codes," *Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuclear Science and Engineering (MC 2011)*, Rio de Janeiro, Brazil, May 8–12, 2011.

5. D. KOTLYAR et al., "Coupled Neutronic Thermo-Hydraulic Analysis of Full PWR Core with Monte-Carlo Based BGCore System," *Nucl. Eng. Des.*, **241**, 3777 (2011); http://dx.doi.org/10.1016/j.nucengdes.2011.07.028.

6. M. VAZQUEZ et al., "Coupled Neutronics Thermal-Hydraulics Analysis Using Monte Carlo and Sub-Channel Codes," *Nucl. Eng. Des.*, **250**, 403 (2012); http://dx.doi.org/10.1016/j.nucengdes.2012.06.007.

7. T. VIITANEN and J. LEPPÄNEN, "Explicit Temperature Treatment in Monte Carlo Neutron Tracking Routines—First Results," *Proc. PHYSOR 2012*, Knoxville, Tennessee, April 15–20 2012, American Nuclear Society (2012).

8. T. VIITANEN and J. LEPPÄNEN, "Optimizing the Implementation of the Explicit Treatment of Thermal Motion—How Fast Can It Get?," *Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013)*, Sun Valley, Idaho, May 5–9, 2013, American Nuclear Society (2013).

9. T. VIITANEN and J. LEPPÄNEN, "Target Motion Sampling Temperature Treatment Technique with Elevated Basis Cross Section Temperatures," *Nucl. Sci. Eng.*, **177**, 77 (2014); http://dx.doi.org/10.13182/NSE13-37.

10. W. G. LUSCHER and K. J. GEELHOOD, "Material Property Correlations: Comparisons Between FRAPCON-3.4, FRAPTRAN 1.4, and MATPRO," NUREG/CR-7024, PNNL-19417, Pacific Northwest National Laboratory (Mar. 2011).

11. V. VALTAVIRTA, "Internal Neutronics-Temperature Coupling in Serpent 2—Reactivity Differences Resulting from Choice of Material Property Correlations," *Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013)*, Sun Valley, Idaho, May 5–9, 2013, American Nuclear Society (2013).

12. M. OGUMA, "Cracking and Relocation Behavior of Nuclear Fuel Pellets During Rise to Power," *Nucl. Eng. Des.*, **76**, 35 (1983); http://dx.doi.org/10.1016/0029-5493(83)90045-6.

13. A. YAMAMOTO et al., "Benchmark Problem Suite for Reactor Physics Study of LWR Next Generation Fuels," *J. Nucl. Sci. Technol.*, **39**, 900 (2002); http://dx.doi.org/10.1080/18811248.2002.9715275.

14. G. ROWLANDS, "Resonance Absorption and Non-Uniform Temperature Distributions," *J. Nucl. Energy A/B*, **16**, 235 (1962); http://dx.doi.org/10.1016/0368-3230(62)90294-X.

Publication II

V. Valtavirta, V. Tulkki, J. Leppänen and T. Viitanen. The universal fuel performance code interface in Serpent 2. In *Proceedings of TopFuel 2013*, Charlotte, NC, Sept. 15–19, 2013.

THE UNIVERSAL FUEL PERFORMANCE CODE INTERFACE IN SERPENT 2

V. Valtavirta, V. Tulkki, J. Leppänen, T. Viitanen

VTT Technical Research Centre of Finland, P.O. Box 1000, FI-02044 VTT, Finland,ville.valtavirta@vtt.fi

The increasing interest in high fidelity modeling of power and test reactors drives the development of new tools for multi-physics modeling as well as the coupling capabilities of the current, well established, codes. The simultaneous modeling of fuel performance and core neutronics enables solving the coupled problem between fission power, fuel temperature and mechanical behavior of fuel elements at different parts of the system.

To make such considerations easier, an universal fuel performance code interface has been implemented in the Monte Carlo reactor physics code Serpent 2. The interface allows to take in account the temperature and strain distributions, calculated by fuel performance codes, in the neutron transport simulations. Information is transmitted to Serpent 2 via a separate file containing 2D axisymmetric temperature and strain distributions in the fuel elements. The output in the coupled calculation mode includes 2D axisymmetric power distributions and axial fast fluence distributions in different lattice positions.

Results from an initial test problem involving the depletion of a short 3D VVER assembly are presented and compared to a similar calculation without the thermal and mechanical feedback.

I. INTRODUCTION

Fuel performance modeling and neutron transport simulations have typically focused on their own, apparently separate, aspects of reactor modeling. Fuel performance codes are built to provide information on the structural integrity of the fuel elements. These simulations involve calculating the temperature distributions in the elements as well as their structural deformations. Neutron transport codes, for their part, focus on criticality safety and solving the reaction rates at different parts of the system.

While these problems may appear disjoint, they are everything but. The distribution of the fission power in and between the fuel elements, an important prerequisite for any fuel performance simulation, can be accurately obtained from neutron transport simulations. Likewise, the temperature distributions inside the fuel elements as well as their dimensions in power producing conditions have to be taken in account in the neutronics simulation and can be obtained from fuel performance studies.

As the computational tools and resources have developed, the simultaneous or iterative calculation of more than one aspect of the coupled problem, i.e. multi-physics modeling, has become ever more feasible. With the safety analyses moving slowly from conservative modeling to the 'best estimate plus uncertainty' approach, there is a need for more detailed modeling of nuclear systems.

As the scientific community moves towards high fidelity modeling of power reactors, addressing the couplings between the different physics involved is one of the big challenges[1]. The recently published Benchmark for Evaluation and Validation of Reactor Simulations (BEAVRS)[2] is an exceptional tool for the verification and validation of multiphysics simulations as it provides extremely detailed core specifications and operating data from the first two cycles of an actual commercial reactor.

Simultaneous fuel performance and neutronics analyses have already been performed by coupling discrete ordinate[3] or hybrid MOC/diffusion[4] neutronics solvers with fuel performance codes. Monte Carlo neutron transport codes offer some interesting abilities such as their explicit resonance treatment as well as their detailed and flexible geometry modeling.

This paper describes, for the first time, the fuel performance code interface in Serpent 2 and presents initial results from a coupled calculation utilizing the interface.

II. SERPENT

Serpent is a continuous energy Monte Carlo reactor physics code developed originally at Technical Research Centre of Finland, and currently used at 90 universities and research organizations around the world in applications ranging from homogenized group constant generation to depletion calculations and the modeling of small reactor cores. Serpent has been available in public distribution since 2009. The development of a new, almost completely rewritten, version of the code Serpent 2 has been going on since 2010. Some motivators for the rewriting of the source code included a better parallelizability and the extension of the depletion capabilities of the code from 2D-lattice problems to three dimensional full core calculations. Serpent 2 is currently in beta testing phase.

A special focus of Serpent 2 is in the multiphysics capabilities of the code and coupled calculations with fuel performance, CFD and system-scale thermal hydraulic codes. A solid base for multiphysics modeling is given by the capability to model continuous temperature and density distributions. The temperature effect on the cross section data is modeled through the Target Motion Sampling (TMS) method developed by T. Viitanen[5,6] and the varying densities are handled through rejection sampling[7].

Fuel behavior data can be calculated externally and passed to Serpent 2 via the fuel performance code interface, or it can be calculated in-code by the fuel model FINIX[8] implemented to Serpent 2 on source code level.

III. MOTIVATION

The ability to model realistic fuel temperature and density distributions as well as the thermomechanical changes in the problem geometry is important if one wants to accurately model the coupled problem between the power distribution, temperature distribution and the changes in geometry. Accurate representations are needed for the radial and axial power distributions in a fuel rod in order to obtain a realistic temperature distribution inside the fuel element. The realistic temperature distribution itself is in turn needed to obtain the accurate radial and axial power distributions. This coupled problem also involving the changes in fuel element dimensions, leading to changes in the moderator volume in the flow channel, requires a coupled solution of the fission power, temperature distribution and geometry changes.

Serpent has been utilized in coupled fuel performance – neutronics simulations in the past via running a separate Serpent calculation for each iteration[9]. While such calculations can be done by re-writing the inputs between the iterations, the novel fuel performance interface is meant

1) to offer new capabilities, not available via the normal input such as linear interpolation of temperatures, separation of temperature zones and material zones etc.

2) to allow easier access to the variables of interest by simplifying the required input and output formats to better suit the needs of external coupling scripts.

3) to save calculation time by not having to re-initialize the problem at each iteration.

IV. INTERFACE FOR FUEL PERFORMANCE CODES IN SERPENT 2

This section will describe the additional input and output of Serpent 2 in calculations using the fuel performance interface.

IV.A. Input

The input format of the interface was designed to be as general and as simple as possible. The data passing through the interface uses a 2D axisymmetric (1.5D) coordinate system. The user input consists of a list of three variables, the cold condition radial coordinate of the point, the hot coordinate radial coordinate of the point and the temperature at the point:

```
<r1 cold>        <r1 hot>        <T@r1>
<r2 cold>        <r2 hot>        <T@r2>
…
```

Different fuel rods and different axial zones are separated by header lines defining the lattice position and axial interval in question. The radial nodalization and the axial zones are completely independent from the material zone boundaries, which means for example that dividing the fuel into axial, radial or angular depletion zones does not complicate the interface input.

One should notice that the masses and densities of the materials are only given in cold conditions as a part of the normal Serpent input, and the changes in density are calculated from changes in fuel dimensions in a mass conserving manner.

IV.B. Output

In addition to the default output, the user can define a number of radial and axial zones for each lattice position, where the fission power is calculated. The fast flux is also calculated at the user defined axial zones. The output is written to a plain text file in a format that is meant to be easy to read by external coupling scripts or programs.

V. FUEL BEHAVIOR CONSIDERATION IN NEUTRON TRACKING

V.A. Geometry

In order to accurately model the coupled problem, the changes in the problem geometry due to thermomechanical phenomena need to be addressed in the neutronics model. They affect the reaction rates for example through the changes in the moderator volume resulting from the changes in the cladding outer radius of the fuel elements.

The Serpent geometry model is based on material cells, which are defined by elementary surfaces such as cylinders and planes in the case of three dimensional fuel rods. The updated geometry has to be taken in account in two subroutines. First, scoring the neutron interactions in the correct material cells requires translating the interaction coordinates to the material cell the coordinates fall into. Second, the surface tracking routine requires the distance to the closest cell boundary for a neutron in certain coordinates traveling in a certain direction. Both the cell search by coordinates and the calculation of the distance to the first boundary crossing were slightly modified to take in account the updated geometry.

A simple hot-to-cold interaction coordinate transformation makes it possible to use the standard cell search routine. Secondly, in order to calculate the distance to the closest cell boundary, one has to first determine the current material cell and perform a cold-to-hot coordinate transformation for the cylindrical inner and outer surfaces of the material zone. After that, the default search for the nearest boundary can be utilized.

At some point during normal fuel life the gas gap between the pellet and the cladding closes i.e. the pellet outer surface and the cladding inner surface share the same radial coordinate. This provides no difficulty for the geometry consideration implemented in Serpent 2.

V.B. Temperature

The temperature of the fuel material has an important effect on the reactivity of the system, as the cross sections are dependent on the temperature of the material. Broadening of the resonance peaks in the cross sections with increasing temperature increases the losses of neutrons to resonance absorption (U238 being the dominant affecting nuclide), which results in a reduced reactivity of the system and a reduced fission rate at high temperature regions.

In depletion calculations, the failure to utilize accurate temperature distributions can lead to wrong estimates on fission and absorption reaction-rates in different depletion zones, which can lead to significant differences in the nuclide inventory over the depletion history.

The Target Motion Sampling in Serpent 2 scores the material temperature only at the points of interaction during the tracking routine, which allows the calculation of the local temperature on-the-fly based on the precise coordinates of the interaction. Serpent currently offers two options for the calculation of the local temperature from the points given in the interface input. First one is the piecewise homogeneous, where <T r_i> gives the constant temperature from r_{i-1} to r_i. The second option is the linear interpolation of temperature from the adjacent nodes.

After the temperature has been calculated at the interaction point, it is used in the Target Motion Sampling to correctly accommodate for the temperature dependence of the cross sections.

V.C. Density

The atomic densities of materials are used in the calculation of the material-wise cross sections, which in their part are used in Monte Carlo neutron transport for the sampling of the neutron path lengths between interaction points.

The absolute changes in the material densities are not explicitly needed or calculated by Serpent, instead the ratio between the hot and cold condition densities is calculated and used. The density ratio between nodes *i* and *i+1* can be calculated from the radial coordinates given in the interface input:

$$\frac{\rho_{hot}}{\rho_{cold}} = \frac{(r_{i+1}^{cold})^2 - (r_i^{cold})^2}{(r_{i+1}^{hot})^2 - (r_i^{hot})^2}, \qquad (1)$$

where the subscripts refer to the node indices and the superscripts refer to the hot and cold conditions. The density ratio is used in the rejection sampling of the neutron path lengths.

V.D. Coupling

To eliminate the need to execute a new Serpent calculation each time the interface file is updated, POSIX signaling capabilities have been introduced to Serpent 2. This enables the direct communication between Serpent and the coupling code.

If a parent process id is specified in Serpent input, Serpent will wait for updated interface files before each transport calculation. Serpent will send a SIGUSR1 signal to the parent process and go to a waiting mode to allow time for executing the external fuel performance code. Serpent will wait until SIGUSR1 is received back from the parent process. After receiving the continue signal, Serpent will proceed to read the updated interface data and carry out the transport cycle. This eliminates the need to rewrite the input for each transport calculation, which comes especially in handy with depletion calculations containing several steps. The fission source is completely re-calculated after each update. While re-calculating the fission source each time reduces step-to-step correlations yielding better statistics, it increases the computation time. The fission source from the last step should provide at least a good initial guess for the fission source, an option, which will be further investigated.

V.E. Capabilities and limitations

As the fuel performance interface itself is very simple and because the temperature and density regions as well as the power and flux tallies are handled separately from the underlying geometry, the use of the interface is not limited to certain fuel, cladding or coolant materials. The only built-in limitation at the moment is the 2D axisymmetric geometry used in the input and output. The azimuthal dependency of variables will be straightforward to implement, while less structured meshes will bring difficulties to the surface tracking as the distance to the next cell boundary will not be as easy to calculate.

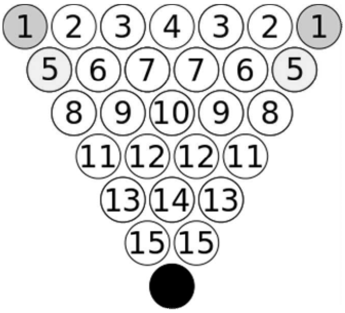


Fig. 1 The numbering of the unique lattice positions, referred in the text. One sixth of the assembly is shown. Black position is the central instrumentation tube.

TABLE I. Geometry specifications

Active length (cm)	60.0
Shroud tube inner width (cm) (flat to flat)	14.2
Shroud tube thickness (cm)	0.15
Assembly pitch (cm)	14.7
Central hole radius (cm)	0.060
Pellet radius (cm)	0.380
Clad inner radius (cm)	0.3865
Clad outer radius (cm)	0.4535
Pin pitch (cm)	1.23

Changes in the axial dimensions of the rod are not yet modeled, which means that axial elongation of the fuel cannot be taken in account. The interface does not limit the number of simulated fuel rods or the detail of power, temperature or depletion zone discretization.

VI. TEST CASE DESCRIPTION

To give an example of the capabilities and possible end uses of the interface a simple depletion calculation of a three dimensional short VVER-440 assembly is presented. The fuel performance code chosen for this example was the steady-state fuel performance code ENIGMA, developed by Nuclear Electric and BNFL in the United Kingdom[10]. The version of the code used in this study is based on ENIGMA v.5.9b, but has been modified at VTT over the years, mostly for model extension and addition purposes concerning for example cladding material properties, fission gas release and gadolinia-doped fuel rods.

VI.B. General properties

The test case was a short VVER-440 assembly with an active length of 60 cm and six gadolinium doped fuel rods. The geometry is described in Table 1. Figure 1 shows the numbering of the unique rods in one sixth of the hexagonal assembly (twelve fold symmetry was applied). Pin 5 is the burnable absorber rod containing 3.35 wt-% of Gd2O3 with an enrichment of 4.37 wt-%. The fuel in rod 1 is enriched to 4.2 wt-%, slightly lower than the 4.4 wt-% in the other pins. The coolant temperature was varied linearly from the temperature at the bottom of the assembly (537.15 K) to the temperature at the top of assembly (572.15 K). The power of the assembly was set to 1.198 MW, corresponding to an approximate linear power of 16 kW/m averaged over all rods. The depletion calculation was extended to 1115 days corresponding to an end of simulation burnup of 43.1 GWd/tU.

VI.B.1. Serpent input

For the Serpent model, reflective boundary conditions were utilized in the radial direction and black boundary conditions in the axial direction. For simplicity, only the active length of the assembly was modeled in the Serpent model, however the inactive parts of the assembly as well as other important details can easily be modeled with Serpent. 10 cm high water reflectors were added both to the top and the bottom of the active part. The coolant temperature was kept at a constant 550 K in the Serpent model to allow for the use of thermal scattering libraries. The density of the coolant was, however, varied as the decrease in the moderator density at the upper parts of the assembly gives rise to decreased moderation, an important feature strongly affecting the axial power profile.

The density was given to Serpent as an axial mesh with the mesh point spacing of 1 cm, with linear interpolation between the points. The density of the coolant was calculated from the coolant temperature and pressure (12.3 MPa) according to the revised IAPWS formulation[11].

The fuel parts of the rods were divided into depletion zones both in the radial and axial direction. The radial division consisted of ten zones of equal area in the gadolinia doped reds and of five zones of equal area in the undoped rods. The axial division had 15 zones of equal height. The fission power was calculated at 15 axial zones with 10 equal area radial rings.

VI.B.2. Enigma input

The ENIGMA model consisted of 15 equal height axial elements. The fuel part of the pellet was divided into 80 equally thick segments. The coolant pressure was 12.3 MPa. The default VVER correlations were used in ENIGMA. The restart capability of ENIGMA was used to eliminate the need to run the simulation from start at each step and the power depression profile was provided via a separate input file.

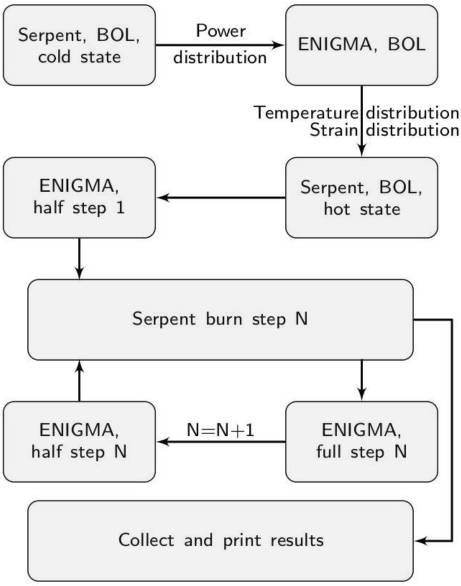


Fig. 2 The iteration scheme applied in the test calculation.

VI.C. Coupling

For the coupling a small program was written to handle the writing of the input to ENIGMA as well as the updating of the Serpent interface file. The cold and hot coordinates and the temperatures were passed to Serpent at 83 radial areas: At the central hole, at the 80 radial segments calculated by ENIGMA, at the gas gap and at the cladding. The piecewise homogeneous temperature profile was used in the Serpent calculation.

The fission power calculation in Serpent allowed the passing of the axial and radial power distributions to ENIGMA as well as the calculation of the total heat generation rates of the individual fuel elements.

The iteration scheme applied in this calculation is illustrated in Figure 2. At the beginning of each depletion step, the fuel rod state was solved with ENIGMA at the middle of the step (MOS) using beginning of step (BOS) power distributions. The ENIGMA calculated MOS distributions were used in the Serpent calculation of the full depletion step. After the power distribution was obtained from Serpent for the end of step (EOS), the fuel state was solved with ENIGMA for the EOS.

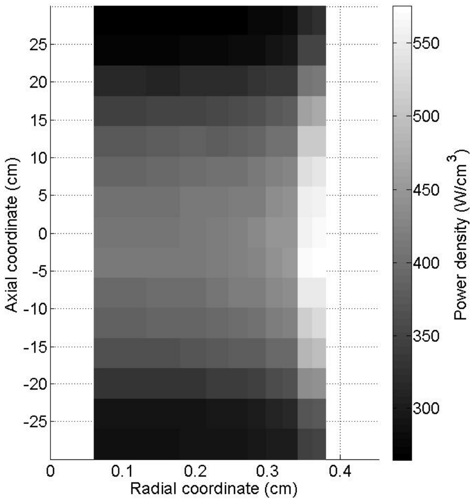


Fig. 3 The power distribution at lattice position 1 at the burnup of 21.4 GWd/tU.

This iteration scheme is by no means optimal and does not even involve iterating for convergence at any step. A more detailed study on the optimal iteration scheme in depletion calculations will be conducted later. When combined with the predictor-corrector depletion methods and given the chance to exchange information mid-step, this is in no way a simple problem.

VI.D. Selected results of the test case

Figure 3 shows the calculated power distribution at lattice position 1 at the burnup of 21.4 GWd/tU. The 15 axial segments are clearly visible as are the 10 equal volume radial segments. The central hole, the gas gap and the cladding are not visible in this figure. The power density is slightly axially asymmetric and peaks below the axial midpoint. This is due to the decreased moderator density, and the resulting decrease in moderation and thermal flux, at higher axial segments. Typically in PWRs the decrease in moderation at lower coolant densities is countered by decreased neutron absorption by soluble absorbers. Figure 3 shows the increased power density near the surface of the fuel resulting from the self-shielding effect combined to the transmutation of U238 into the fissile isotope Pu239.

The temperature distribution calculated by ENIGMA based on the power distribution in Figure 3 is shown in Figure 4 and the ratio between the cold state and hot state densities (Density Factor) is shown in Figure 5.

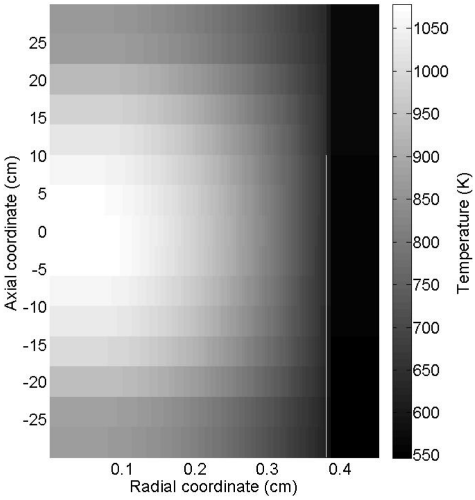


Fig. 4 The temperature distribution solved from the power distribution in Fig. 3 by ENIGMA.

VI.E.Comparison case

A separate simulation was executed without thermomechanical feedback. The geometry was expanded to the hot conditions for the beginning of life, obtained from the feedback simulation. The fuel, cladding and fill gas temperature was kept at a constant 800 K, which was close to the fuel temperature averaged over the whole assembly at the beginning of the fuel life. One might easily come up with more clever methods of choosing the effective temperature, based on the coupled calculation results. A tempting candidate would be the temperature averaged over each interaction point in the neutronics calculation.

The absolute and relative differences are calculated in reference to the feedback calculation i.e. the absolute difference in the variable X is calculated from

$$\Delta X = X_{COMPARISON} - X_{FEEDBACK}, \qquad (2)$$

and the relative difference from

$$\Delta X_{rel} = \frac{X_{COMPARISON} - X_{FEEDBACK}}{X_{FEEDBACK}}. \qquad (3)$$

VI.C.1. Multiplication factor

Differences in the multiplication factor of the system are interesting because of criticality safety and reactivity control considerations.

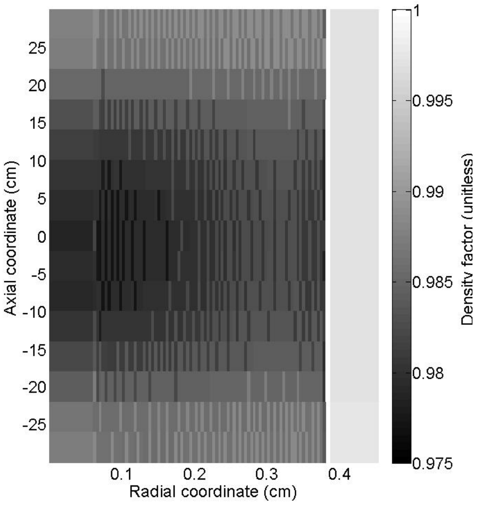


Fig. 5 The Density factor at lattice position 1 at the burnup of 21.4 GWd/tU.

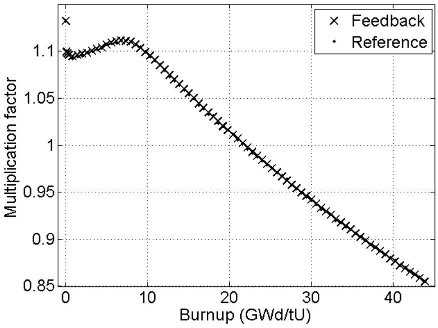


Fig. 6 The evolution of the multiplication factor of the system during the depletion calculation.

A higher fuel temperature lowers the resonance escape probability decreasing the multiplication factor of the system. Figure 6 shows the evolution of the multiplication factor of the system during the depletion calculation and Figure 7 shows the difference in the reactivity of the system between the coupled case and the reference

The effective temperature of 800 K yields surprisingly good results showing no clear deviations, although the statistical uncertainties are large.

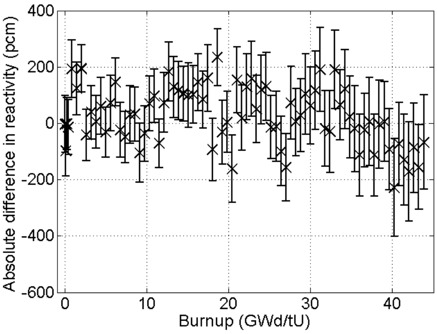


Fig. 7 The absolute difference in the reactivity of the assembly at different burnups (1σ error bars)

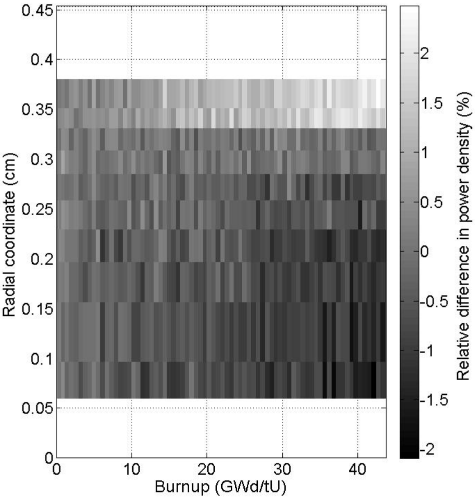


Fig. 8 The relative difference in the radial power profile at lattice position 1 at different burnups.

VI.C.2 Pin power distribution

As the problem of constructing a pin-power distribution from less detailed data such as the assembly linear power is quite common and many times required for a more detailed modeling of the individual pins, it will be interesting to compare the power produced at different lattice positions with and without the thermal and mechanical feedback effects.

The relative differences in pin powers between the two simulations stay inside ±0.6 % without showing any clear trend. The deposition of fission power within the fuel elements does differ, between the simulations.

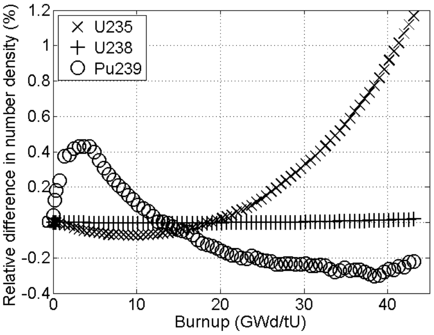


Fig. 9 Relative differences in atomic densities of U235, U238 and Pu239.

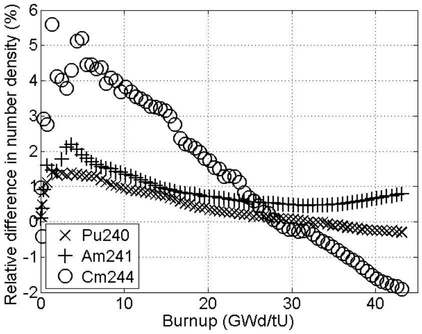


Fig. 10 The relative difference in atomic densities of three important actinides.

Figure 8 shows the relative difference in the axially averaged, radial power profile in lattice position 1 over the fuel life. It is easily seen that the radial power peaking is greater in the comparison calculation. This results from the higher pellet surface temperature in the comparison simulation, which increases resonance absorption in U238, leading to increased formation of the fissile isotope Pu239 near the surface of the fuel.

VI.C.3 Nuclide concentrations

Nuclide inventories are an important result of depletion calculations. They are essential source data in criticality safety, radiation shielding, and accident modeling. This section will compare the effects of the thermal and mechanical feedback on the concentrations of some important nuclides. One should notice, that the Monte Carlo method does not produce estimates for the statistical uncertainties in the nuclide inventories.

Figure 9 shows the relative difference in the nuclide concentrations of U235, U238 and Pu239 between the simulations. Both U235 and Pu239 contribute to the power production, with the importance of Pu239 increasing with higher burnup. Fig. 9 shows that the division of fission power between these nuclides seems to depend on the modeling of the temperature distribution in the neutronics simulation: In this case the flat temperature profile leads to underutilization of the initial fissile isotope U235, compared to the feedback simulation. The relative differences in the nuclide concentrations are shown for some important minor actinides in Figure 10. The differences in these nuclide concentrations reach their highest level at low burnups and decrease after that to EOL differences between +1 and -2 %.

VI.C.4 Computational time

The feedback simulation took 511 hours on 12 Intel Xeon X5690 processors. A reference simulation with similar statistical accuracy would take 216 hours on the same hardware if on-the-fly Doppler treatment is used.

VII. CONCLUSIONS

A fuel performance code interface with a standard output and input format has been implemented in the Monte Carl neutron transport code Serpent 2. The main design choices include the independence of the temperature and strain meshing from the Serpent 2 geometry, providing generality to the interface. The temperature and density considerations in the neutron tracking are handled via the Target Motion Sampling and rejection sampling of neutron path lengths, methods that allow the use of continuous temperature and density distributions.

The functionality of the coupling was demonstrated through a Serpent 2/ENIGMA depletion calculation of a short VVER assembly. While the obtained power, temperature and strain distributions seem plausible, further validation and verification of the coupled calculation performance is needed. Comparison to a neutronics simulation without thermal or mechanical feedback, showed that the reactivity and pin power distribution are reproduced surprisingly well without the feedback. The radial and axial power distributions are, however, affected by the temperature distribution used in the simulation.

The universal fuel performance code interface opens up 0061n easy way to study the challenging field of multi-physics from the fuel behavior – neutronics point of view. The coupled modeling requires special considerations such as assessing the convergence of the solution, the acceleration of convergence, the iteration scheme between the fuel performance and neutronics code in depletion calculations, the stability of the solution and the inclusion of a coupling to the thermal hydraulics. These topics will provide many interesting challenges in the years to come.

ACKNOWLEDGMENTS

This work has been funded by the KÄÄRME and PALAMA projects under the Finnish National Research Programme on Nuclear Power Plant Safety, SAFIR-2014.

REFERENCES

1. K. SMITH and B. FORGET, “Challenges in the Development of High-Fidelity LWR Core Neutronics Tools,” *In Proc. M&C 2013,* Sun Valley, ID, 5-9 May, 2013.
2. N. HORELIK *et al.* ”Benchmark for Evaluation and Validation of Reactor Simulations (BEAVRS),“ *In Proc. M&C 2013,* Sun Valley, ID, 5-9 May, 2013.
3. K. CLARNO et al., “Integrated Radiation Transport and Nuclear Fuel Performance for Assembly - Level Simulations,” *In Proc. PHYSOR-2012*. Knoxville, TN, 15-20 April, 2012.
4. F. N. GLEICHER, B. SPENCER, S. NOVASCONE, R. WILLIAMSON, R. C. MARTINEAU, “ Coupling the Core Analysis program DeCART to the Fuel Performance Application BISON,” *In Proc. M&C 2013,* Sun Valley, ID, 5-9 May, 2013.
5. T. VIITANEN and J. LEPPÄNEN, “Explicit treatment of thermal motion in continuous-energy Monte Carlo tracking routines,” *Nucl. Sci. Eng.*, **171**, 165 (2012)
6. T. VIITANEN and J. LEPPÄNEN, “Explicit Temperature Treatment in Monte Carlo Neutron Tracking Routines -- First Results,” *In Proc. PHYSOR-2012*. Knoxville, TN, 15-20 April, 2012.
7. J. LEPPÄNEN, “Modeling of Non-uniform Density Distributions in the Serpent 2 Monte Carlo Code,” accepted for publication in *Nucl. Sci. Eng., (2013)*
8. T. IKONEN *et al.* ” FINIX – fuel behavior model and interface for multiphysics applications,” *In Proc. 2013 LWR Fuel Performance Meeting / Top Fuel,* Charlotte, NC, 15-19 September, 2013 (submitted).
9. T. VIITANEN and V. TULKKI, “Combining Reactor Physics and Fuel Performance Calculations”, *In Proc. 2012 LWR Fuel Performance Meeting / Top Fuel,* Manchester, UK, 2-6 September, 2012.
10. W. J. KILGOUR et al., “Capabilities and Validation of the ENIGMA Fuel Performance Code”, *In Proc. ANS/ENS Int. Topical Meeting on LWR Fuel Performance*, Avignon, France, 21-24 April 1991.
11. IAPWS, “Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam”, Lucerne, Switzerland (2007)

Publication III

V. Valtavirta, T. Ikonen, T. Viitanen and J. Leppänen. Simulating fast transients with fuel behavior feedback using the Serpent 2 Monte Carlo code. In *Proceedings of PHYSOR 2014*, Kyoto, Japan, Sept. 28–Oct. 3, 2014.

SIMULATING FAST TRANSIENTS WITH FUEL BEHAVIOR FEEDBACK USING THE SERPENT 2 MONTE CARLO CODE

V. Valtavirta, T. Ikonen, T. Viitanen, J. Leppänen
VTT Research Centre of Finland,
P.O. Box 1000, FI-02044 VTT, Finland
ville.valtavirta@vtt.fi

ABSTRACT

Simulating transients with reactivity feedback effects using Monte Carlo neutron transport codes can be used for validating deterministic transient codes or estimating for example the total deposited energy in a fuel rod following a known reactivity insertion in the system. Recent increases in computational power as well as developments in calculation methodology makes it possible to obtain a coupled solution for several aspects of the multi-physics problem in a single calculation. This paper describes the different methods implemented in Serpent 2 Monte Carlo code that enable it to model fast transients with fuel behavior feedback. The capability is demonstrated in a prompt critical pin-cell case, where the transient is shut down by the negative reactivity from rising fuel temperature.

Key Words: **Monte Carlo, multi-physics, fast transient, coupling, neutronics, fuel behavior**

1. INTRODUCTION

Monte Carlo neutronics simulations are traditionally used mainly in criticality and shielding calculations as well as in the validation of deterministic transport codes. The Serpent[1] Monte Carlo code developed at VTT Technical Research Centre[1] has been specifically optimized for 2D lattice calculations such as group constant generation for higher level codes. The completely rewritten version, Serpent 2, extends the capabilities of Serpent 1 in 3D-modeling, full core depletion problems and multi-physics applications while offering good parallelization to take advantage of the present computer clusters and supercomputers.

An interesting problem in multi-physics is the modeling of fast (feedback from fuel temperature) and slow (feedback from fuel temperature and coolant conditions) transients with corresponding feedback effects. This paper focuses on the coupled modeling of neutronics and fuel behavior in fast transients where the time scales are so short that the coolant conditions can be modeled as static and delayed neutron emission can be omitted. This paper outlines the functionalities required from a Monte Carlo neutron transport code in order to simulate coupled fast transients and provides a demonstration calculation with the current capabilities of Serpent 2.

[1] For the latest news about Serpent, visit the project homepage at http://montecarlo.vtt.fi

Section 2 discusses the different capabilities required for the modeling of coupled transients and Section 3 describes the iterative solution scheme applied in this study. The specifics on the demonstration case are given in Section 4 and its results are discussed in Section 5. Finally, the work is summarized in Section 6.

2. REQUIREMENTS FOR COUPLED TRANSIENT SIMULATIONS

A fundamental requirement for transient simulations with Monte Carlo neutronics is the capability for dynamic, i.e. time-dependent, simulation of the neutron interactions. The dynamic simulation mode in Serpent 2 is described in subsection 2.1.

The most important feedback effect in a fast transient is the fuel temperature reactivity feedback. This feedback is the result of changes in the microscopic cross sections of different nuclides with increasing temperature. These temperature effects on the cross sections must be modeled accurately in order to describe the behavior of the system. Serpent 2 can handle this on-the-fly via the Target Motion Sampling (TMS) temperature treatment technique described in subsection 2.2.

The fuel behavior has to be solved either internally or by an external program and the neutron tracking routine has to be able to easily access the resulting temperature and strain distributions. In Serpent 2 the transfer of material temperatures and densities from solvers to neutron tracking is managed by a multi-physics interface described in subsection 2.3. The internal fuel behavior module FINIX is used in this study. Its main features are addressed in subsection 2.4.

2.1. Dynamic Simulation Mode

The dynamic simulation mode was implemented in an earlier version of Serpent 2, as an extension of the conventional time-dependent criticality source mode. Instead of allowing the neutron population to grow ($k > 1$) or decay ($k < 1$) exponentially, the simulation is divided into discrete time intervals, with the population size re-adjusted at the beginning of each interval, while preserving the total weight. The method is described in detail in Ref. [2]. This sequential population control essentially allows the modeling of prompt super-critical excursions, in which the initial population is eventually multiplied by a factor of thousand or more, as well as sub-critical states without running out of simulated neutrons. The methodology was validated for short time intervals in comparison to MCNP5, and coupled to a preliminary steady-state temperature feedback solver [**? ?**] to demonstrate the practical feasibility of the coupling. In this paper, the steady-state solver is replaced by the FINIX module, with a more refined transient model for the thermal-mechanical behavior of the fuel rod.

2.2. Target Motion Sampling Temperature Treatment Technique

In Serpent 2 the modeling of temperature distributions relies on the Target Motion Sampling (TMS) temperature treatment technique. It is a stochastic technique for taking the effect of thermal motion on reaction rates and path lengths into account on-the-fly during Monte Carlo transport calculation. Thus, functionally TMS is an on-the-fly Doppler-broadening technique, but since no cross sections are actually Doppler-broadened in the method, the term "temperature treatment technique" is preferred.

The basic idea of the method is to sample thermal motion of target nuclei at each collision site and to deal with the collisions in target-at-rest frame using cross sections at a temperature lower than that of the collision site. The fact that in this tracking scheme the total cross section becomes a distributed quantity is dealt with using rejection sampling methods based on a temperature majorant cross section. A similar rejection sampling technique is utilized for variable material densities in Serpent 2. This kind of a scheme also provides for modeling of arbitrary temperature and density distributions that can be continuous in both space and time, independent of the material boundaries in the problem geometry. This property makes TMS and Serpent 2 particularly convenient to use together with external or internal temperature solvers, for instance FINIX.

The TMS-method was introduced for the first time in english-speaking journals[2] in 2012 [4] and ever since the main emphasis of the development has been on the optimization of the method. Preliminary results of different optimization techniques were presented at M&C 2013 conference [5] and the optimization ideas were more thoroughly discussed in References [6, 7], the first of which covers the usage of the method with elevated basis cross section temperatures and the second of which discusses the truncation of the thermal motion in the temperature majorant generation.

To be precise, the modeling of temperatures in Serpent is three-fold: The cross sections are first processed in 300 K temperature intervals to an ACE-formatted cross section library using the NJOY cross section processing code [8]. Then, in the problem initialization phase the cross sections are further Doppler-broadened to the minimum temperature of each nuclide using the Doppler-preprocessor of Serpent [9]. The TMS method is only responsible of the necessary on-the-fly temperature corrections between the nuclide minimum and maximum temperatures in the system.

2.3. Multi-Physics Interface for Fuel Behavior

Serpent 2 includes dedicated multi-physics interfaces for fuel behavior and for thermal hydraulic/CFD-calculations. The purpose of these interfaces is to read externally calculated temperature and density distributions of materials, store them in a suitable format and provide the tracking routine with the local temperature and material density at interaction points[10, 11].

The fuel behavior interface handles radial temperature and strain distributions in nested geometries such as fuel pins. Separate temperature and strain distributions can be stored for different pins,

[2] At the SNA&MC conference 2013 it was learned that a similar method has been utilized in, for example, Russian Monte Carlo code PRIZMA for many years [3].

axial elevations and angular segments. Furthermore, the temperature and strain distributions can be stored separately for separate time steps yielding the possibility to use temperature distributions that vary both in space and in time.

2.3.1. Temperatures

The temperature is required by the TMS method only at the interaction points, which allows the calculation of the local temperature on-the-fly based on the precise coordinates of the interaction. In this study, linear interpolation is used with respect to the radial coordinate as well as time, allowing Serpent to use temperature distributions that are continuous in both space and time. Higher order interpolation methods or functional dependencies can be implemented in future if they are deemed necessary.

2.3.2. Geometry

In order to accurately model the coupled problem, the changes in the problem geometry due to thermomechanical phenomena need to be addressed in the neutronics model. They affect the reaction rates for example through the changes in the moderator volume resulting from the changes in the cladding outer radius of the fuel elements.

The Serpent geometry model is based on material cells, which are defined by elementary surfaces such as cylinders and planes in the case of three dimensional fuel rods. The updated geometry has to be taken in account in two subroutines. First, sampling a specific interaction type at interaction points requires the local material cross sections. This means that the interaction coordinates have to be translated to the material cell the coordinates fall into. Second, the surface tracking routine requires the distance to the closest cell boundary for a neutron in certain coordinates traveling in a certain direction. Both the cell search by coordinates and the calculation of the distance to the first boundary crossing were slightly modified to take in account the updated geometry.

A simple hot-to-cold interaction coordinate transformation makes it possible to use the standard cell search routine. Secondly, in order to calculate the distance to the closest cell boundary, one has to first determine the current material cell and perform a cold-to-hot coordinate transformation for the cylindrical inner and outer surfaces of the material zone. After that, the default search for the nearest boundary can be utilized.

All coordinates that are stored in one timestep and accessed in another (such as source points) are transformed into the cold (or initial) coordinate system before storage.

2.3.3. Material densities

The atomic densities of materials are used in the calculation of the material-wise cross sections, which in their part are used in Monte Carlo neutron transport for the sampling of the neutron path lengths between interaction points.

The absolute changes in the material densities are not explicitly needed or calculated by Serpent, instead the ratio between the hot and cold condition densities is calculated and used. The density ratio between nodes i and i+1 can be calculated from the radial coordinates given in the interface input:

$$DF(r,t) = \frac{\rho^{\mathrm{hot}}(r,t)}{\rho^{\mathrm{cold}}(r,t)} = \frac{(r_{i+1}^{\mathrm{cold}})^2 - (r_i^{\mathrm{cold}})^2}{(r_{i+1}^{\mathrm{hot}})^2 - (r_i^{\mathrm{hot}})^2}, \tag{1}$$

where ρ is the material density, the subscripts refer to the node indices and the superscripts refer to the hot and cold conditions. The density ratio is used in the rejection sampling of the neutron path lengths.

2.4. FINIX

In Serpent 2, the thermal and mechanical behavior of the fuel rod is simulated with a dedicated fuel performance module, FINIX [12–14]. FINIX is a general purpose fuel module built especially for multi-physics simulations such as coupled reactor physics – fuel performance, or thermal hydraulics – fuel performance simulations. It simulates the time-dependent thermal and mechanical behavior of the rod, taking into account the dependence of material properties such as thermal conductivity, thermal expansion, Young's modulus, etc., on various parameters, including temperature and burnup. The thermal and mechanical solutions are coupled via the gas pressure and conductance of the pellet-cladding gap. The numerical models have been chosen to be computationally inexpensive, but on par with other currently employed one dimensional fuel performance codes.

Current capabilities of FINIX concentrate on the simulation of transient behavior. The time-dependent temperature distribution is computed taking into account radial heat transfer in the pellet, gap, cladding, and into the coolant. Thermal expansion of the pellet and the cladding, and the elastic response of the cladding is calculated, as well as their influence on the gap conductivity. Plastic deformation and oxidation of the cladding and release of fission gases into the gap are yet to be included in FINIX. Consequently, effects of accumulating burnup during steady state irradiation cannot be simulated, but a transient for a non-fresh fuel can be modeled if the initial state is known. Currently initialization for non-zero burnup can be done, e.g., from a FRAPCON simulation. Additional initialization methods are under development.

FINIX has been validated against the FRAPTRAN fuel performance code as well as experimental Halden reactor data [14, 15]. The comparison shows very good agreement, especially for the temperature distribution, which is the most relevant output for coupled neutronics simulations.

FINIX has been coupled to Serpent 2 on source code level. The communication between Serpent and FINIX is via straight memory access, which requires no writing and reading of files. The coupled solution is further described in the next section.

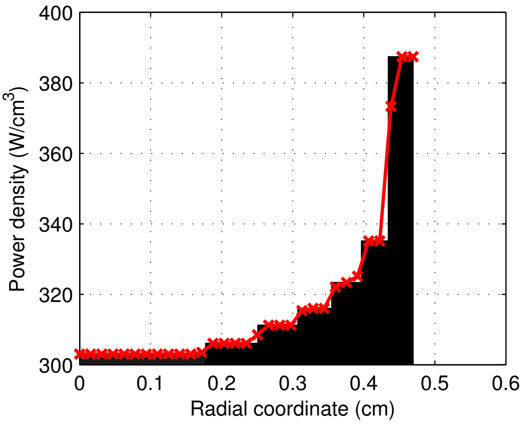


Figure 1. Radial power distributions in Serpent (black) and FINIX (red).

3. COUPLING OF SERPENT AND FINIX

3.1. Solution procedure

The solution scheme utilized in this study involves sequential and iterative solving of the fission power distribution by Serpent and the temperature and strain distributions by FINIX. At the beginning of the dynamic simulation a steady state solution of the fuel behavior is calculated to be used on the first timestep.

At the first iteration of a time step the temperature and strain distributions are taken from the previous step and at the later iterations they are interpolated between the beginning of step (BOS) and end of step (EOS) distributions. First the neutronics solution is obtained for the new time interval, after which the temperature distribution at the end of the interval is calculated by FINIX. A convergence criterion is applied after this and if the convergence of the coupled solution is not deemed sufficient, the neutronics solution for the next iteration is obtained.

3.2. Grid matching

As lined out in the previous sections, the temperature and strain distributions calculated by FINIX can be used in Serpent 2 as is, without further mapping operations. The power distribution calculated by Serpent 2, as with any Monte Carlo Code, is the integral power in a macroscopic volume (tally region). As FINIX receives the power densities in radial points and assumes a linear variation between the node points, some thought needs to be set on the mapping between the volumetric tally region and the pointwise radial node points.

Figure 1 shows an example of the grid matching between Serpent 2 (7 power tallies, 81 temperature points) and FINIX (81 radial nodes) with regards to power (left) and temperature (right), the material zone boundaries are also plotted in the figure (blue dashed lines, the wider lines indicate pellet surface and cladding inner and outer surfaces). The Serpent power tallies are shown in black in the left axes. If Serpent 2 is coupled to an external solver, the values of the power tallies are printed out and the user can then process them for the fuel performance code. Coupling with FINIX, the integral power tallies have to be mapped to the radial nodes of FINIX (red markers), between which the power density varies linearly (red line). In the mapping, the total power in the pin is conserved. The linear variation of FINIX is equal to the piecewise constant tallies of Serpent everywhere but at the tally boundaries. For the FINIX nodes closest to the tally boundaries, the power density has to be calculated separately so that the total pin power is conserved.

3.3. Convergence criterion

When the EOS temperature is solved for a timestep $[t_j, t_{j+1})$ the result is a pointwise distribution $T(r_i, t_{j+1})$. For the first iteration of this timestep, the end of step temperature distribution is compared to the beginning of step temperature distribution. On further iterations, the new EOS solution is compared to the previous EOS solution. The convergence criterion used in this study was the absolute pointwise difference in the temperature distributions. If the difference was below 3 K further iterations were omitted. The value of 3 K was arbitrarily chosen and can be fine tuned or user defined. Relative differences can also be used as a criterion.

The advantage of applying the convergence criterion on the temperature distribution rather than the power distribution is the fact that small differences in power result in negligible differences in temperature on short timesteps. For example, the difference in deposited energy even between the linear powers 300 W/cm and 3000 W/cm during a time step of 100 μs is only 0.27 J/cm. This is far too small to lead to a noticeable difference in EOS temperature distributions. Consequently, if the temperature distribution does not change, the interdependence between the fission power and temperature distribution is already resolved and does not have to be iterated further.

If a certain timestep is calculated more than once, the relaxation is applied to the power distribution. The relaxed power distribution after i iterations is calculated from:

$$\overline{P}^i(r,t) = \begin{cases} P^1(r,t) & i = 1 \\ \frac{1}{i-1}\sum_{n=2}^{i} P^n(r,t) & i > 1 \end{cases}$$

Where $P^n(r,t)$ is the solution for the local power density on iteration n. In essence, if multiple iterations are required, the first solution is discarded and a simple average over the subsequent solutions is used.

4. DESCRIPTION OF TEST CASE

An extremely simple problem case was chosen to illustrate the coupled neutronics/fuel behavior solution. The geometry was a square 2D pincell similar to a TMI-1 fuel rod with 0.4695 cm pellet radius, 0.4791 cm clad inner and 0.5464 cm outer radius with a lattice pitch of 1.4427 cm. The fuel was undoped UO_2 with an initial enrichment of 4.03 wt%. The fuel was depleted until a burnup of 8.84 MWd/kgU with 10 radial depletion zones to obtain a more realistic radial nuclide distribution. The initial conditions of the transient were hot full power (233 W/cm), where the system was kept at critical state with soluble boron absorber in the coolant. To onset the transient the coolant boron concentration was reduced from 970 ppm to 860 ppm yielding an excess reactivity of 1865 pcm.

For the fuel model, characteristics of a standard TMI-1 fuel rod as defined in Ref. [16] were used, apart from the total rod length (needed by FINIX to calculate the total fill gas volume) which was set to 200 cm. Because of the 2D neutronics model, the power profile of the rod was axially invariant. The boundary condition for the temperature solution was a coolant bulk temperature of 500 K.

The simulation was run with the dynamic simulation mode of Serpent 2 for 56 milliseconds with 2000 time intervals. The fission power was tallied in 10 radial zones of equal volume. The fuel model of FINIX contained 101 radial nodes in the fuel pellet as well as 51 nodes in the cladding. The modeled neutron population consisted of 200 batches of 2000 neutrons each. The calculation time was approximately 12 days on 12 Intel Xeon X5690 cores, with one thread each.

The results obtained from these calculations should mostly be viewed as a demonstration of a coupled solution of neutronics and fuel behavior as the fuel model used here was not initialized with respect to burnup and thus lacks fuel swelling due to fission products and degradation of fuel thermal conductivity.

5. RESULTS

5.1. Time evolution of the system

Figure 2 shows the time evolution of the linear power (a) and the pellet centerline and surface temperatures (b). As expected the system linear power starts to increase exponentially from the start. It reaches a peak value of approximately 1.36 MW/cm (almost 6000 times the initial linear power) at t=6.440 ms. At this time, the reactivity of the system is zero and the power does not grow anymore, but the temperature of the fuel is still rising rapidly (see black dashed line in Fig. 2). The temperature overshoots the point of zero reactivity resulting in the exponentially decreasing system power.

Calculating the expected response of the system to the reactivity insertion from point kinetics parameters obtained from a criticality source simulation with the onset conditions for the transient (860 ppm boron) yields an exponential curve plotted in Fig. 2(a) in red. The point kinetics approximation used here does not include any feedback effects. The slope corresponds well to the initial

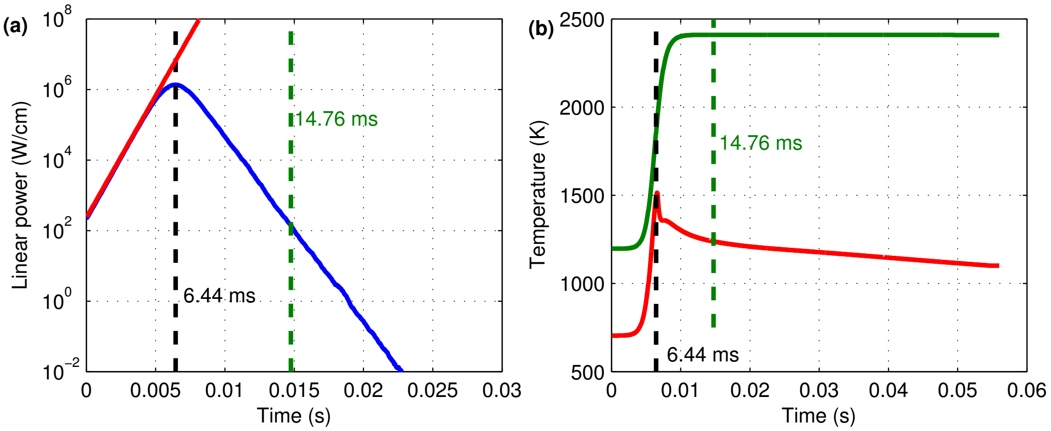


Figure 2. a: The linear power of the system as a function of time (blue) with point kinetics prediction (red). b: The pellet centerline and surface temperatures as a function of time.

increase in power in the dynamic simulation, but the negative reactivity feedback from increasing fuel temperature means that the dynamic simulation eventually diverges from the no-feedback point kinetics prediction as expected.

The deposited energy per unit length can be obtained by integrating the tallied power with respect to time, yielding 3.75 kJ/cm. Calculating from this energy and using approximate values for UO_2 specific heat and density (0.32 J/g/K and 10.3 g/cm^3 respectively), the mean temperature of the pellet should rise by an approximate 1640 K. This seems to be in line with the initial and final radial temperature distributions shown in Fig. 3(a).

5.2. Radial temperatures and strains

The radial temperature distributions at times 0 ms, 6.440 ms, 14.756 ms and 56 ms are plotted in Fig. 3(a). The markers correspond to the nodal solution points by FINIX, between which the solution was interpolated in Serpent yielding the continuous temperature distributions. The initial temperature distribution (red in Fig. 3(a)) has the typical parabolic shape in the fuel part reaching 1200 K at the centerline and being approximately 700 K at the pellet surface. The later temperature distributions (black at peak power, green at 14.756 ms) exhibit a peak near the pellet surface. This is due to the high amount of fissile 239Pu in these parts of the pellet. The maximum centerline temperature of 2410 K is reached at t=14.756 ms, when the linear power of the system has already dropped down to approximately 140 W/cm. After this time, the loss of local heat due to heat transfer exceeds the amount of local heat generation even in the fuel centerline. The surface temperature has already peaked at innermost regions. The blue curve indicates the temperature distribution in the end of the simulation at 56 ms, when the temperature of the surface parts of the pellets has already begun to decrease from the maximum of 1515 K at 6.580 ms.

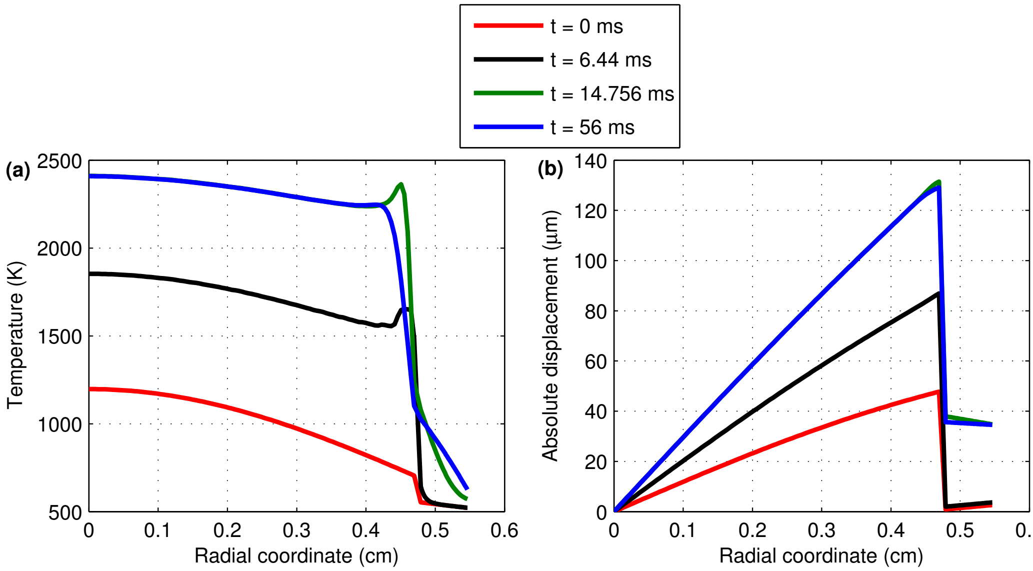


Figure 3. a: The radial temperature distribution at different times. b: The displacement of radial nodes at different times.

The absolute displacement of radial nodes due to thermal strain is shown in Fig. 3(b). The gas gap closes fully at 6.720 ms, after the system power has already begun to fall, but while the fuel temperature is still rising.

6. SUMMARY

The basic parts for calculating fast transients with fuel behavior have been implemented in Serpent 2 and are connected in a way that allows the modeling of coupled fast transients. This was demonstrated via a simple simulation of a pin-cell geometry with a large reactivity insertion, which showed the shutting down of the transient by fuel temperature feedback. More complex system, lattices, parts of core and/or 3D geometries are possible to model but computational cost may be prohibitive at the moment as the coupled simulation mode has not been optimized in any way. The responses of larger systems are also much more complex due to the limited travel time of neutrons.

The modeling of slow transients in the future requires a temperature treatment method for bound scatterers in the coolant as well as the inclusion of delayed neutrons to the dynamic simulation mode. Further topics of research and development pertaining to transient modeling is the capability for coupled neutron/gamma transport to account for the quick deposition of fission energy in the coolant due to the fission gammas as well as the optimization of the calculation routines to decrease the calculation time needed for such studies.

ACKNOWLEDGMENTS

This work has been funded by the NUMPS project of the Finnish Academy as well as the KÄÄRME and PALAMA projects under the Finnish National Research Programme on Nuclear Power Plant Safety, SAFIR-2014.

REFERENCES

[1] J. Leppänen. *Development of a New Monte Carlo Reactor Physics Code*. Ph.D. thesis, Helsinki University of Technology (2007).

[2] —. “Development of a dynamic simulation mode in serpent 2 Monte Carlo code.” In: *M&C 2013*. Sun Valley, ID (2013).

[3] V. Ogibin and A. Orlov. “Majorized cross-section method for tracking neutrons in moving media.” *Voprosy Atomnoj Nauki i Techniki, Metodoki i Programmy*, **2**: pp. 6–9 (1984 (In Russian)).

[4] T. Viitanen and J. Leppänen. “Explicit treatment of thermal motion in continuous-energy Monte Carlo tracking routines.” *Nucl. Sci. Eng.*, **171**: pp. 165 – 173 (2012).

[5] T. Viitanen and J. Leppänen. “Optimizing the implementation of the explicit treatment of thermal motion – how fast can it get?” In: *M&C 2013*. Sun Valley, ID (2013).

[6] —. “Target motion sampling temperature treatment technique with elevated basis cross section temperatures.” *Nucl. Sci. Eng.* (accepted for publication).

[7] —. “Temperature majorant cross sections in Monte Carlo neutron tracking.” *Nucl. Sci. Eng.* (in preparation).

[8] R. E. MacFarlane and D. W. Muir. *NJOY99.0 Code System for Producing Pointwise and Multigroup Neutron and Photon Cross Sections from ENDF/B Data*. PSR-480, Los Alamos National Laboratory (2000).

[9] J. Leppänen and T. Viitanen. “New data processing features in the serpent Monte Carlo code.” *J. Korean Phys. Soc.*, **59**: pp. 1365–1368 (2011).

[10] J. Leppänen, T. Viitanen, and V. Valtavirta. “Multi-physics coupling scheme in the serpent 2 Monte Carlo code.” *Trans. Am. Nucl. Soc.*, **107**: p. 1165 (2012).

[11] J. Leppänen *et al.* “Unstructured mesh based multi-physics interface for CFD code coupling in the serpent 2 Monte Carlo code.” In: *Physor 2014*. Kyoto, Japan (2014).

[12] T. Ikonen. *FINIX - fuel behavior model and interface for multiphysics applications - code documentation for version 0.13.9. Technical report*, VTT Technical Research Centre of Finland (2013).

[13] T. Ikonen *et al.* “FINIX – fuel behavior model and interface for multiphysics applications.” In: *2013 Fuel Performance Meeting TopFuel*. Charlotte, USA (2013).

[14] —. “Module for thermomechanical modeling of LWR fuel in multiphysics simulations.” *Annals of Nuclear Energy* ((In preparation)).

[15] H. Loukusa. *Validation of the FINIX fuel behavior code version 0.13.9. Technical report*, VTT Technical Research Centre of Finland (2013).

[16] T. Blyth *et al.* “Benchmark for uncertainty analysis in modelling (UAM) for design, operation and safety analysis of LWRs, volume II: Specification and support data for the core cases (phase II).” OECD/NEA (2013).

Publication IV

V. Valtavirta, M. Hessan and J. Leppänen. Delayed neutron emission model for time dependent simulations with the Serpent 2 Monte Carlo code – first results. In *Proceedings of PHYSOR 2016*, Sun Valley, ID, May 1–5, 2016.

DELAYED NEUTRON EMISSION MODEL FOR TIME DEPENDENT SIMULATIONS WITH THE SERPENT 2 MONTE CARLO CODE – FIRST RESULTS

V. Valtavirta†, M. Hessan‡ & J. Leppänen†

† VTT Technical Research Centre of Finland
Ville.Valtavirta@vtt.fi
Jaakko.Leppanen@vtt.fi

‡ RWTH Aachen
Hessan@lrst.rwth-aachen.de

ABSTRACT

Extending the applicability of the Monte Carlo code Serpent 2 from steady state coupled multi-physics calculations to time dependent reactor dynamics problems requires a new model for delayed neutron emission in the dynamic simulation mode of the code. The implementation of new routines in Serpent responsible for the generation of the initial precursor source for dynamic simulations as well as producing, keeping track of and decaying precursors during the time dependent simulation and emitting delayed neutrons are described. First verification of the functionality of the new routines is obtained by simulating the response of an infinite homogeneous reactor with LWR-like composition to various reactivity insertions and comparing the results to the point kinetics prediction. The comparison shows good agreement between the calculated results and the analytical solution. Further verification and validation efforts are discussed alongside with potential challenges that might be encountered in the simulation of more realistic problems.

Key Words: **Monte Carlo, dynamic simulation, delayed neutron, precursor**

1. INTRODUCTION

Serpent[1] is a Monte Carlo neutronics code developed at VTT Technical Research Centre of Finland[1]. The code has approximately 500 registered users in 48 organizations in 37 countries. The original version of Serpent was developed for spatial homogenization, whereas the new, completely rewritten, version Serpent 2 has a very much broader base of application. One of the major focus areas of Serpent 2 is coupled multi-physics calculations and much of the recent work has been devoted to developing new methods for applying Monte Carlo neutronics to multi-physics problems[2–8]. Steady state

[1] For the latest news, see the project website: http://montecarlo.vtt.fi

multi-physics simulations with high-fidelity thermal hydraulic feedback can now be executed with Serpent 2 on full core level[9]. Monte Carlo multi-physics simulations can be used to provide a reference solution for lower order steady state neutronics solvers.

Since Monte Carlo neutronics solves the neutron transport problem without major approximations, it can be used to obtain reference solutions for other solution methods. The next point of interest would be to provide reference solutions for reactor dynamics codes. This requires the capability for time dependent neutronics from the Monte Carlo code as well as a way to incorporate thermal hydraulic feedback into the simulation. The time dependent simulation mode in Serpent[10] as well as methodology for time dependent multi-physics calculations with Serpent 2[11] are already in place. However, certain limitations concerning the application of the delayed neutron emission routines in Serpent to time dependent problems have, thus far, limited the capabilities to scenarios where the effect of delayed neutron emission is negligible such as the super-prompt critical cases simulated in [11–13]. This paper details the work done in implementing a new set of delayed neutron emission routines for Serpent 2 and compares the results from the new routines to a point-kinetics prediction in an LWR-like infinite homogeneous reactor.

For an extensive overview on the problems and solutions for the dynamic Monte Carlo simulations one can consult the Ph.D. thesis of B. L. Sjenitzer[14]. Previous implementations for delayed neutron treatment in Monte Carlo codes, see [14–16] for TRIPOLI, [16] for MCNP5 and [17] for OpenMC.

2. DYNAMIC SIMULATION MODE

The time dependent simulation mode in Serpent was implemented in 2012[10] based on the pre-existing external source simulation mode. Starting from an initial neutron source, the neutrons are tracked in time dependent fashion until a cut off time given by the user. The simulation time can be divided into multiple time-intervals with population control applied on each time interval boundary to keep the number of neutrons to simulate (and thus the computational cost) constant. This makes it possible to simulate systems where the neutron population decreases or increases considerably with good statistical accuracy.

Dividing the total simulation time into intervals also makes it possible to exchange information with a coupled code at the time interval boundaries in multi-physics calculations.

3. DELAYED NEUTRON EMISSION

The time-independent criticality source simulation mode in Serpent is generation based, i.e., the fission neutrons produced by a neutron batch will serve as the source for the next neutron batch. From the generation point of view, there is no difference between a prompt and delayed neutron born in a fission.

Both will be increase the generation number by 1 even though their emission may happen even minutes apart.

In time dependent simulations the difference between prompt and delayed neutrons is evident. Prompt neutrons are generated practically instantaneously at the time of the fission event, whereas delayed neutrons are emitted far in the future (compared to the prompt neutrons), typically on some upcoming time interval. These delayed neutrons can be either stored into the memory to wait until the simulation reaches the time-interval of their emission, or they can be tracked immediately.

Tracking the delayed neutrons immediately in a generation based manner saves memory, but it brings about other problems: In coupled calculations, the material temperatures and densities on the emission time interval might significantly differ from those in the precursor generation time interval. Even in uncoupled calculations, the neutron population may have increased or decreased significantly between the generation and emission times meaning that the statistical weight of the delayed neutron will be too low or too high to generate good statistics.

Storing the delayed neutrons in the memory causes other problems. The precursor population of a reactor (and thus the number of delayed neutrons to store) is typically many orders of magnitude larger than the neutron population at any given instant. This means that a considerable amount of memory has to be allocated into storing neutrons that will be emitted some time far away in the future.

A solution for this problem is to create delayed neutron precursors instead of delayed neutrons at fission events. Tracking precursors means that we can force them to emit delayed neutrons on every time interval[14] so memory is not wasted in storing neutrons that will only affect the simulation in the far future. This requires some additional routines for the generation of the initial precursor source for dynamic simulations as well as producing precursors and keeping track of them as they decay during the time dependent simulation. The implementation of these routines in Serpent is described in the following sections.

4. NEW IMPLEMENTATION

4.1. Source Generation

In order to run time dependent simulations with delayed neutrons, the first step is to set up the initial conditions. A representative neutron source at the initial time $t = 0$ has to be created for a steady state system. The neutron source actually has two parts:

1. The live neutrons moving in the geometry at distinct locations $\vec{r}$ with distinct velocities $\vec{v}$ at the time instant $t = 0$.

2. The delayed neutron precursors at distinct locations $\vec{r}$ in the geometry.

The number of delayed neutron precursors is considerably higher than the number of live neutrons. In the example case presented in this paper, the instantaneous precursor population exceeded the instantaneous live neutron population by the factor of 4.2×10^4 in the critical state. Both of these sources as well as their normalization can be generated for a critical system with a single criticality source simulation:

4.1.1. Live Neutron Source

For the live neutron source we want to store neutrons at random times during their lifetime. We can do this by saving neutrons during a criticality source simulation at tentative interaction sites. The interactions of neutrons are not distributed uniformly in time, but depend on the neutron energy and the local interaction cross section: The mean path length between two interactions (mean free path) for a neutron with energy E travelling over a path, where the interaction probability is constant over the path length, is

$$\lambda_{\text{mean}}(E) = \frac{1}{\Sigma_{\text{tot}}(E)}, \tag{1}$$

where Σ_{tot} is the total macroscopic cross section over the path. The time it takes for the neutron to travel this path length gives the mean time between two interactions

$$t_{\text{mean}}(E) = \frac{\lambda_{\text{mean}}(E)}{v(E)} = \frac{1}{\Sigma_{\text{tot}}(E)v(E)}, \tag{2}$$

where $v(E)$ is the velocity of the neutron. This means that the mean interaction frequency depends on the energy of neutrons as

$$f_{\text{mean}}(E) = \frac{1}{t_{\text{mean}}(E)} = \Sigma_{\text{tot}}(E)v(E). \tag{3}$$

If we store neutrons at sampled interactions and want our stored neutron source to represent the physical neutron source at random time in the system, we should store each neutron with a probability proportional to $P \propto 1/(\Sigma_{\text{tot}}(E)v(E))$.

We will thus save neutrons in tentative collision sites with the probability of

$$P = A\frac{w}{v\Sigma_{\text{path}}}, \tag{4}$$

where w is the weight of the incoming neutron, v is its velocity, Σ_{path} is the macroscopic cross section used in sampling the path-length to the tentative collision site and A is a normalization factor depending on the minimum neutron speed and the minimum cross section to keep the probability P below unity for all cases. Here Σ_{path} can be either the local material total cross section if surface tracking is used or a majorant cross section if delta tracking is used.

For each live neutron, the location, direction cosines, the neutron energy and the weight of the neutron are saved.

The total live neutron population is also tallied for normalization using an implicit estimator for

$$N_{\text{live}}^{\text{phys.}} = \int_V \int_E \frac{1}{v}\phi(\vec{r}, E)\mathrm{d}E\mathrm{d}r^3, \tag{5}$$

where the integral is of the one over velocity multiple of scalar flux over the whole geometry and energy range.

4.1.2. Delayed Neutron Precursor Source

The delayed neutron precursor production rate is tallied on a regular spatial mesh using an implicit estimator for

$$P_{\text{prec},g,i,j,k} = \int_{V_{i,j,k}} \int_E \beta_g(\vec{r}, E)\nu(\vec{r}, E)\Sigma_f(\vec{r}, E)\phi(\vec{r}, E)\mathrm{d}E\mathrm{d}r^3, \tag{6}$$

where $P_{\text{prec},g,i,j,k}$ is the precursor production rate for group g at the mesh index i, j, k and $V_{i,j,k}$ is the volume corresponding to that mesh index. $\beta_g(\vec{r}, E)$ is the fraction of fission neutrons being delayed in group g, ν is the number of fission neutrons produced per fission, Σ_f is the macroscopic fission cross section and ϕ is the scalar flux. In a critical system these production rates can be converted to stable populations by dividing them with the decay constant for the corresponding group λ_g

$$N_{\text{prec},g,i,j,k}^{\text{phys}} = \frac{1}{\lambda_g}P_{\text{prec},g,i,j,k}. \tag{7}$$

These stable populations are then stored for all g, i, j and k.

4.2. DYNAMIC SIMULATION

The time dependent simulations in Serpent are based on dividing the total simulation time into a number of intervals and applying population control at each interval boundary. At the initial time interval boundary ($t_0 = 0$), the initial neutron source has to be sampled. This consists of two parts

- Live neutrons at $t_0 = 0$.
- Delayed neutrons emitted from the precursors existing at t_0 during the time interval (between t_0 and the end of the interval (EOI) t_1).

One should note that in addition to these neutron sources there will be additional delayed neutrons emitted by precursors that are created and decay during the time interval. Their emission is handled at the production of the precursors (see Section 4.2.3).

We can choose to sample N^s_{live} live neutrons and N^s_{emit} delayed neutrons for the interval. To avoid statistical problems, the weights of the live and emitted neutrons should be close to each other

$$\frac{N^s_{\mathrm{live}}}{N^s_{\mathrm{emit}}} \approx \frac{N^{\mathrm{phys.}}_{\mathrm{live}}}{N^{\mathrm{phys}}_{\mathrm{emit}}}, \tag{8}$$

where $N^{\mathrm{phys.}}_{\mathrm{live}}$ is the physical number of live neutrons at the beginning of the interval (tallied in criticality source simulation, Eq. 5) and $N^{\mathrm{phys}}_{\mathrm{emit}}$ is the physical number of neutrons that precursors present at $t = 0$ will emit during the time-interval (can be calculated from known precursor populations and decay constants). This means that for a short time interval most of the sampled source should consist of live neutrons, whereas for a long time-interval most of the sampled source should consist of delayed neutrons emitted from the precursors.

4.2.1. Live Neutron Source at Initial Time

The live neutrons can be sampled from the neutron source stored during the criticality source simulation. If the weight of each sampled neutron w_i represents the number of physical neutrons this neutron represents, the weights have to be scaled with a factor

$$a = \frac{N_{\text{live}}^{\text{phys.}}}{\sum_i^{N_{\text{live}}^s} w_i}, \tag{9}$$

i.e., the sum of the weights of the sampled live neutrons must be equal to the physical number of live neutrons in the system tallied in the criticality source simulation (Eq. 5). The scaling factor can be further modified if the initial power level of the system is to be adjusted.

4.2.2. Delayed Neutron Source at Initial Time

The delayed neutrons can be sampled for the upcoming time interval from the precursor populations at the beginning of the interval (BOI). In the simplest case we want to emit N_{emit}^s delayed neutrons with equal weight w_{emit}. To determine the weight of each emitted neutron w_{emit} we can first calculate the physical number of delayed neutrons emitted over the time interval from the initial precursor source:

$$N_{\text{emit}}^{\text{phys}} = \sum_{g,i,j,k} \left(1 - \exp\left[-\lambda_g(t_1 - t_0)\right]\right) N_{\text{prec},g,i,j,k}^{\text{phys}} \tag{10}$$

We'll then simply divide this total weight over all of our emitted neutrons resulting in

$$w_{\text{emit}} = \frac{N_{\text{emit}}^{\text{phys}}}{N_{\text{emit}}^s}. \tag{11}$$

The spatial sampling of the delayed neutrons can be done by, e.g., sampling first the bin (g, i, j, k) to emit from with the probability to emit from a certain bin is equal to the number of physical delayed neutrons emitted from that bin divided by the total number of physical delayed neutrons emitted. After the bin has been sampled, emission coordinates can be randomly sampled from the spatial domain of the bin with resampling used if the sampled point does not hit a fissile (or precursor containing) material. Another sampling method is to sample points randomly from the whole geometry and adjust the weights of the emitted neutrons based on the local emitted population.

The emission times are sampled using the decay-law based relation

$$t_{\text{emit}} = t_{\text{BOI}} - \frac{1}{\lambda} \log\left[1 - \xi(1 - \exp\left[-\lambda(t_1 - t_0)\right])\right], \tag{12}$$

where ξ is a sample from the uniform random distribution over the interval [0, 1). The energies of the emitted neutrons are sampled based on the nuclear data of the delayed neutron group and the emission direction is sampled isotropically.

4.2.3. Precursor Production During Time Interval

Delayed neutron emission from fission events is currently handled in an implicit manner, but an analogous treatment is also possible. Using the implicit model, we can tally the produced delayed neutron precursor population in each interaction. The produced precursor weight for group g from an interaction by a neutron with an incoming weight of w_{in} is calculated from

$$w_g = \beta_g \nu \frac{\Sigma_f}{\Sigma_{\mathrm{tot.}}} w_{\mathrm{in}}, \tag{13}$$

where w_{in} is the weight of the incoming neutron. Part of the produced weight in each interaction will decay over the remainder of the current time interval and the remaining weight will survive until the end of the interval. These parts can be calculated by the decay law in a straightforward manner:

$$w_g^{\mathrm{emit}} = w_g(1 - \exp\left[-\lambda_g(t_{\mathrm{EOI}} - t_{\mathrm{int.}})\right]) \tag{14}$$

and

$$w_g^{\mathrm{surv.}} = w_g \exp\left[-\lambda_g(t_{\mathrm{EOI}} - t_{\mathrm{int.}})\right], \tag{15}$$

where $t_{\mathrm{int.}}$ is the interaction time and t_{EOI} is the time at the end of the current time interval.

The surviving weight will be added to the corresponding precursor population tally at the end of the time interval. The proportion of the weight that decays corresponds to the weight of delayed neutrons to be emitted before the end of the time interval. Russian roulette will be played for w_{emit} to either increase the weight to a suitable level (such as w_{in}) followed by the creation/emission of the delayed neutron or not emitting anything at all. If the delayed neutron survives the Russian roulette, its emission time is sampled using Eq. 12, where t_0 is replaced by the interaction time and t_1 is replaced by t_{EOI}. One should note that the emission coordinates of these delayed neutrons are not sampled from the precursor mesh but the accurate interaction coordinates are used instead.

Table I. Nuclide composition for the infinite reactor case.

Nuclide	mass fraction
H-1	3.626×10^{-2}
O-16	2.877×10^{-1}
U-235	2.003×10^{-2}
U-238	6.559×10^{-1}
B-10	6.587×10^{-5}

4.2.4. Live Neutron Source at Further Time Intervals

On the subsequent time intervals, the live neutron source consists of all of the neutrons that survived until the end of the previous time interval. Population control can be applied to the live neutron source controlling the number and weight distribution of the neutrons to be simulated on the next time interval. In this work we use stochastic splitting and Russian roulette to resize the live population to approximately

$$N^s_{\text{live}} = \text{round}\left[\frac{N^{\text{phys.}}_{\text{live}}}{N^{\text{phys}}_{\text{live}} + N^{\text{phys}}_{\text{emit}}}\right], \tag{16}$$

live neutrons.

4.3. Delayed Neutron Source at Further Time Intervals

The delayed neutron emission on further time intervals is handled similarly to the initial time-interval. The only difference is that the precursor populations corresponding to the new BOI are used for the sampling rather than the initial populations.

5. COMPARISON TO POINT-KINETICS

The first physical test case for the model is an infinite homogeneous reactor. The solution to the infinite homogeneous reactor problem is known and given by point-kinetics, which makes the problem a good test case for the new implementation.

The composition of the system is based on the mixture of water and 3 wt-% enriched uranium and

Table II. Delayed neutron group structure in the test case.

Group	decay constant (1/s)	β (pcm)
1	1.25×10^{-2}	21.1
2	2.83×10^{-2}	102.1
3	4.25×10^{-2}	60.0
4	1.33×10^{-1}	131.5
5	2.92×10^{-1}	222.9
6	6.66×10^{-1}	68.2
7	1.63	58.7
8	3.55	19.7

the reactivity of the system is adjusted by varying the amount of Boron-10 in the composition (see Table I). A criticality source simulation was executed to generate the initial live neutron and precursor sources for the dynamic simulations. The cross section libraries used in the simulation were based on JEFF-3.1.1. For the dynamic simulations, the amount of Boron in the composition was slightly altered resulting in a reactivity insertion or extraction. Four cases with non-zero reactivity insertions were simulated in addition to the case with the critical Boron concentration. Neutrons were simulated for 3 ms with the total simulation time divided into 100 intervals for population control. The neutron population during the simulation was tallied in 100 time bins of equal length. The simulation times for the transient calculations were from 90 to 100 wall-clock hours using 20 OpenMP threads on a 2 × 10 core Intel Xeon E5-2690 v2 node.

The generation time for the system was $\Lambda_{\text{gen}} = 2.46 \pm 0.0001$ µs and the effective delayed neutron fraction was $\beta_{\text{eff.}} = 684 \pm 0.06$ pcm.

The point-kinetics equations for the neutron and precursor populations are (derivation detailed in basic reactor physics textbooks, e.g., [18])

$$\frac{\partial n(t)}{\partial t} = \frac{\rho(t) - \beta}{\Lambda} n(t) + \sum_{g=1}^{n_g} \lambda_g C_g(t) \tag{17}$$

$$\frac{\partial C_g(t)}{\partial t} = \frac{\beta_g}{\Lambda} n(t) - \lambda_g C_g(t), \tag{18}$$

where $n(t)$ and $\rho(t)$ are the neutron concentration/population and system reactivity at time t and Λ is the mean generation time of the system. Starting from known initial neutron and precursor populations and a known reactivity insertion the point-kinetics equations are integrated numerically using the forward

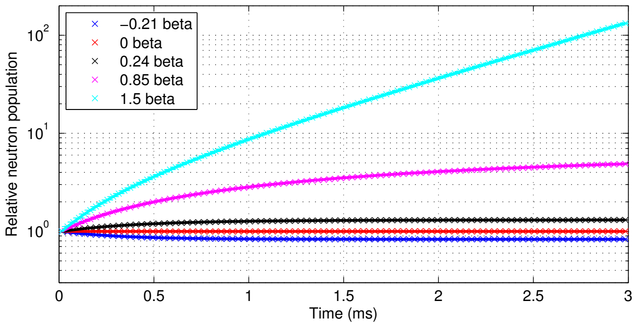


Figure 1. Relative population sizes tallied in test simulations (x, with one sigma error bars) with different reactivity insertions show good agreement with the point-kinetics predictions (solid line)

euler scheme with a very small step size to ensure stability and a small truncation error. Finally, since the Monte Carlo simulation results are integrated over the time bin lengths, the neutron populations were integrated into the same bin structure used in the Monte Carlo simulation.

The bin values relative to the first bin value from the Monte Carlo simulations and the theoretical predictions are plotted in Figure 1. The simulation results are indicated with x-markers and one sigma error bars, while the theoretical predictions are plotted as solid lines. The vertical axis is logarithmic, which means that exponential change in neutron population, as would be the case without delayed neutron emission, could be seen as straight lines. Judging by the curved lines, the effect of delayed neutrons on the transient is certainly visible even in the super-prompt critical case. The comparison of the neutron populations calculated by Serpent to the point-kinetics predictions show a good agreement.

6. FUTURE WORK & DISCUSSION

6.1. Spatial Fidelity and Memory Consumption

Using a mesh to tally the precursor populations means that the precursor concentrations tallied in the mesh are homogenized over the mesh cells. In the current implementation the homogenization is done only for the initial populations and the tallied part of the weight produced in interactions (Eq. 15). The emitted part of the weight (Eq. 14) is emitted from the interaction coordinates. The alternative for the mesh based tallying is to generate point-wise precursors in interactions[14].

The memory needed to tally three dimensional precursor populations on a regular mesh goes up as the spatial resolution of the mesh increases. One important question that remains is the required level of spatial fidelity for tallying the precursor populations. If the populations of each group have to be tallied at sub-centimeter scale, the memory consumption for large scale problems may prove to be prohibitive.

To match the resolution of a $n_x \times n_y \times n_z$ cartesian mesh one has to store a comparable number of point-wise precursors. The advantage that the point-wise precursors have over the mesh based approach is that in linear neutron transport problems, all of the point-wise precursor do not have to be stored in memory at the same time. Instead, the precursors can be divided into batches and each batch can be tracked through the simulation time intervals sequentially. In coupled multi-physics simulations this is not generally the case as the neutron batches are no longer independent, but coupled through the coupled thermal hydraulic fields. The advantage that the mesh based approach has, however, is that implicit estimators can be used to tally precursor production to the mesh in each interaction yielding a large amount of statistics for the tally, whereas creating a new point-wise precursor in each interaction is not feasible from memory consumption point of view.

6.2. Calculation Times and Variance Reduction

The test simulations in the infinite reactor geometry were 3 milliseconds of simulated time and took 90 wall clock hours on one node of our simulation cluster. To provide computational benchmarks for reactor dynamics codes on relevant test cases, the simulations have to be extended to the second or minute scale. The tally resolution in those cases does not need to be on microsecond scale as here, but the required computational effort is still very large.

Due to the random walk nature of Monte Carlo simulations, one of the problems special to time dependent simulations is the fact that the further the simulation gets in time from the fixed initial source, the larger the variance in the results is. Methods for variance reduction are very much needed to make the computational efforts in time dependent simulations bearable. One method developed specifically for time dependent simulations is the branchless collision method suggested by Sjenitzer[14]. There are also already implicit reaction modes for nxn, capture and fission reactions present in Serpent. Utilizing these methods decreases the branching of fission chains, but requires additional intelligent weight control for the neutron population to ensure that the statistical weight distribution of the neutrons stays bounded.

6.3. Delayed Neutron Group Structures

Delayed neutron treatment is typically based on a precursor group approximation, where precursors with similar decay constants are lumped together to generate a small number of precursor (and delayed neutron) groups. Currently, the JEFF cross sections have eight precursor groups that are the same for all fissionable nuclides, whereas ENDF definitions, for example, have six precursor groups that are specific to the nuclides. This means that using ENDF cross sections the total number of precursor groups will go up with increasing number of fissionable nuclides in the system. Currently, Serpent tracks groups with distinct decay constants separately.

If the number of distinct delayed neutron groups in a simulation gets very large it might be beneficial to remember that the group structures are an approximation in the first place and track individual precursor nuclides instead. The neutron induced fission yields for precursor nuclides can be found in the nuclear data libraries alongside with their decay constants and neutron emission spectra. This data should be enough to treat the delayed neutron emission directly through individual precursor nuclides.

7. V&V

The verification and validation of the new methodology is very important. The effort is divided into multiple stages moving from simple cases with analytic solutions, such as presented in this paper, to more realistic systems with comparisons to experimental data. In the V&V calculations, it should be noted that the simulation of critical systems should always result in a stable neutron population over time. For perturbed configurations experimental data will be needed. The ultimate application for the dynamic simulation mode will be time dependent coupled multi-physics calculations with thermal hydraulic and/or fuel behavior feedback, however, before the coupled simulations the methodology has to be validated in time-dependent configurations without any feedback effects. Negative reactivity insertion in control rod drop experiments done in test reactors at close to zero power level can serve as a good benchmark case. The V&V effort has been divided into multiple stages with increasing complexity:

1. Infinite homogeneous reactor.
   - Critical system.
   - Comparison to point-kinetics prediction.
2. Realistic geometries with leakage: critical system.
   - Horizontally reflected 3D assembly geometry.
   - TRIGA reactor core.
   - SPERT-III reactor core.
3. Realistic geometries with leakage: transient scenario without feedback
   - TRIGA reactor control rod drop.
   - SPERT-III reactor, beginning of rod ejection transients for the low initial power cases.
4. Realistic geometries with leakage: transient scenario with feedback
   - TRIGA reactor power pulse (fuel behavior feedback).
   - SPERT-III reactor rod ejection transients (thermal hydraulics feedback).

8. CONCLUSION

The extension of the applicability of the Serpent Monte Carlo code to reactor dynamics problems has been started by implementing routines for delayed neutron precursor tracking and delayed neutron emission in dynamic simulations. The initial implementation is based on tracking the populations of delayed neutron precursor groups on a regular mesh. The initial neutron and precursor source for the dynamic simulation are generated with a steady-state criticality source simulation.

The methodology was tested in an infinite homogeneous reactor geometry with LWR-like composition. Comparing the results to the theoretical prediction given by solving the point-kinetics equations show good agreement and suggests that the source generation by the criticality source simulation as well as the delayed neutron emission in the dynamic simulation mode are working as intended. The spatial distribution of neutrons did not play a role in the presented case meaning that the capability of the mesh based precursor tracking to sufficiently represent the precursor populations in a realistic geometry has still to be evaluated. An alternative approach to the mesh-based method is to track point-wise precursors instead.

The computational cost of dynamic simulations is very high and applying non-analogue reaction modes or the branchless-collision method could be useful in reducing the cost. Reaction modes that change the statistical weight of the neutrons require the implementation of some additional weight control routines to keep the statistical weight distribution of the neutrons bounded as a large weight-disparity will mean that some neutrons have negligible effects on the results whereas the behavior of a few high weight neutrons may dominate the results.

Verification and validation of the new methodology has been divided into multiple stages, moving from simple systems with theoretically known solutions to more realistic geometries with either theoretically known solutions, as is the case in critical systems, or experimental data. As the objective of this work is to provide a way for Serpent to tackle reactor dynamics problems with relevant feedback from moving control elements or material temperature and density fields, the two of the validation cases will be the prompt power pulse in a TRIGA-reactor caused by the pulse-rod ejection and relevant experiments from the experimental series of SPERT-III reactor rod ejection accidents.

REFERENCES

[1] J. Leppänen. *Development of a New Monte Carlo Reactor Physics Code*. Ph.D. thesis, Helsinki University of Technology (2007).

[2] J. Leppänen, T. Viitanen, and V. Valtavirta. "Multi-physics coupling scheme in the serpent 2 Monte Carlo code." *Trans. Am. Nucl. Soc.*, **107**: p. 1165 (2012).

[3] V. Valtavirta, T. Viitanen, and J. Leppänen. "Internal neutronics temperature coupling in serpent 2." *Nucl. Sci. Eng.*, **177**: pp. 193 – 202 (2014).

[4] J. Leppänen *et al.* "The numerical multi-physics project (numps) at vtt technical research centre of finland." *Ann. Nucl. Energy* (Accepted for publication).

[5] T. Viitanen and J. Leppänen. "Explicit treatment of thermal motion in continuous-energy Monte Carlo tracking routines." *Nucl. Sci. Eng.*, **171**: pp. 165 – 173 (2012).

[6] T. Viitanen and J. Leppänen. "Target motion sampling temperature treatment technique with elevated basis cross section temperatures." *Nucl. Sci. Eng.*, **177**: pp. 77–89 (2014).

[7] —. "Temperature majorant cross sections in Monte Carlo neutron tracking." *Nucl. Sci. Eng.* (Accepted for publication).

[8] J. Leppänen *et al.* "Unstructured mesh based multi-physics interface for CFD code coupling in the serpent 2 Monte Carlo code." In: *Physor 2014*. Kyoto, Japan (2014).

[9] M. Daeubler *et al.* "High-fidelity coupled Monte Carlo neutron transport and thermal-hydraulic simulations using Serpent 2/SUBCHANFLOW." *Ann. Nucl. Energy*, **83**: pp. 352–375 (2015).

[10] J. Leppänen. "Development of a dynamic simulation mode in serpent 2 Monte Carlo code." In: *M&C 2013*. Sun Valley, ID (2013).

[11] V. Valtavirta *et al.* "Simulating fast transients with fuel behvior feedback using the Serpent 2 Monte Carlo code." In: *Physor 2014*. Kyoto, Japan (2014).

[12] T. Ikonen *et al.* "Module for thermomechanical modeling of LWR fuel in multiphysics simulations." *Ann. Nucl. Energy* (Accepted for publication).

[13] M. Aufiero *et al.* "Serpent-OpenFOAM coupling in transient mode: Simulation of a godiva prompt critical burst." In: *M&C + SNA + MC 2015*. Nashville, USA (2015 (submitted)).

[14] B. Sjenitzer. *The Dynamic Monte Carlo Method for Transient Analysis of Nuclear Reactors*. Ph.D. thesis, Delft University of Technology, Delft, The Netherlands (2013).

[15] B. L. Sjenitzer and J. E. Hoogenboom. "Dynamic Monte Carlo method for nuclear reactor kinetics calculations." *Nucl. Sci. Eng.*, **175**: pp. 94–107 (2013).

[16] —. "Extensions of the MCNP5 and TRIPOLI4 Monte Carlo codes for transient reactor analysis." In: *SNA+MC 2013*. Paris, France (2013).

[17] M. Mahjob and J. Koclas. "OpenMC - TD, a new module for Monte Carlo time dependent simulations used to simulate a CANDU-6 cell LOCA accident." In: *7th International Conference on Modelling and Simulation in Nuclear Science and Engineering*.

[18] W. M. Stacey. *Nuclear Reactor Physics*. Wiley-VCH (2007).

Publication V

V. Valtavirta, J. Leppänen and T. Viitanen. Coupled neutronics–fuel behavior calculations in steady state using the Serpent 2 Monte Carlo code. *Annals of Nuclear Energy*, 100, Part 2, pp. 50–64, https://dx.doi.org/10.1016/j.anucene.2016.10.015, 2017.

Annals of Nuclear Energy 100 (2017) 50–64

Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

Coupled neutronics–fuel behavior calculations in steady state using the Serpent 2 Monte Carlo code

Ville Valtavirta [*], Jaakko Leppänen, Tuomas Viitanen

VTT Technical Research Centre of Finland, P.O. Box 1000, FI-02044 VTT, Finland

ARTICLE INFO

Article history:
Received 19 July 2016
Received in revised form 17 October 2016
Accepted 19 October 2016

Keywords:
Monte Carlo
Multi-physics
Temperature feedback
Fuel behavior
Effective fuel temperature
Coupled calculation

ABSTRACT

This paper describes the recent developments in the multi-physics capabilities of the Serpent Monte Carlo code concerning coupled multi-physics calculations with fuel behavior feedback. We cover the structure and operation of the fuel behavior interface in Serpent 2 as well as the coupled calculation routines implemented for steady state multi-physics calculations with any internally or externally coupled solver. The intended solution flow and code-to-code communication in internally and externally coupled multi-physics simulations is described alongside with the stochastic approximation based solution relaxation methods implemented in Serpent.

The two-level multi-physics coupling scheme in Serpent 2 is demonstrated by obtaining a coupled solution for the neutronics–fuel behavior problem using first the internally coupled FINIX fuel behavior module and then the externally coupled ENIGMA fuel performance code in a 3D assembly geometry. Parameters such as maximum pellet centerline temperatures can be evaluated from the coupled solution. The temperature fields obtained from the coupled solution are also used to estimate the effect of the detailed radial representation of the fuel temperature distribution compared to various radially averaged effective fuel temperature representations. The convergence of the coupled solution is investigated alongside the possibility to speed up the convergence by using the Uniform Fission Sites method.

1. Introduction

The two-way coupling between the neutronics and the fuel temperature of a nuclear reactor is an important safety feature of current fission reactors. In reactors using low-enriched uranium (LEU), this feedback effect is largely based on the temperature dependent widening (or Doppler-broadening) of the resonance peaks in the radiative capture cross section of ^{238}U. This increase in the energy-width of the resonance peaks leads to an increased loss of neutrons to resonance absorption and a decrease in the reactivity of the system with increasing fuel temperature. Solving this coupled problem is essential in reactor modeling. While the radial fuel temperature profile is continuous in reality, the ability to model continuous temperature distributions in neutronics calculations is very recent (Viitanen and Leppänen, 2012). Traditionally, the continuous temperature profile, which spans hundreds of Kelvins between the fuel centerline and pellet surface, has been collapsed into a single effective temperature to be used throughout the whole pellet in the radial direction (Rowlands, 1962; de Kruijf and Janssen, 1996; Goltsev et al., 2000).

This paper describes the finalized fuel behavior interface in the Serpent 2 Monte Carlo code (Leppänen et al., 2015), which can be used to bring in realistic radially and axially varying fuel temperature and strain distributions for neutronic analyses. A new solution iteration scheme to be used when conducting internally or externally coupled calculations with Serpent 2 is also implemented and described. The convergence in this iteration scheme is enhanced by using a stochastic approximation based solution relaxation scheme as described by Dufek and Gudowski (2006) and Dufek and Hoogenboom (2014).

The interface and the iteration scheme are tested with two coupling types, using the externally coupled fuel performance code ENIGMA and the internally coupled fuel behavior module FINIX in a 3D assembly geometry based on the Peach Bottom 2 BWR model. The coupled solution provides relevant safety information concerning the maximum power densities and fuel temperatures in the assembly. The temperature distributions obtained from the coupled calculation are also used to estimate the differences resulting from the use of a radially averaged effective fuel temperature distribution compared to the more realistic radially varying fuel temperature distribution.

The calculations in this paper focus on steady-state conditions. Coupled depletion or transient calculations utilizing the new iter-

[*] Corresponding author.
E-mail address: ville.valtavirta@vtt.fi (V. Valtavirta).

http://dx.doi.org/10.1016/j.anucene.2016.10.015
0306-4549/© 2016 Elsevier Ltd. All rights reserved.

ation scheme, while interesting, are not in the scope of this paper. The main focus of the paper will be the description of the structure, capabilities and limitations of the fuel behavior interface as well as the coupled calculation iteration scheme and the associated power relaxation scheme. While a comparison of the realistic temperature distribution to various effective temperature models is made in Section 10.4, the problem of calculating the radial effective fuel temperature is not addressed in depth. The implementations presented in this work should also not be seen as pertaining to a certain coupled code system such as Serpent/FINIX or Serpent/ENIGMA. The two coupled codes presented here are only used to exemplify the use of the internal and the external coupling and the code systems should be separately validated, if actual safety analyses are to be conducted with them.

In this study we focus solely on the coupling between fuel behavior and neutronics to continue our previous studies on the subject (Viitanen and Tulkki, 2012; Valtavirta et al., 2013; Valtavirta et al., 2014a,b; Valtavirta, 2015). While the coupling between neutronics and fuel behavior can be well used to demonstrate the coupled multi-physics capabilities of Serpent 2 they cannot be considered to provide the actual multi-physics solution for the studied 3D assembly as the thermal–hydraulics of the system is ignored. In general, the coolant behavior should also be solved.

2. The Serpent Monte Carlo code

The Serpent[1] Monte Carlo code (Leppänen et al., 2015) is currently used in more than 150 universities and research organizations for reactor physics applications ranging from homogenized group constant generation to burnup calculations and the modeling of small research reactor cores. Serpent has been developed at VTT Technical Research Centre of Finland since 2004 and the current development version, Serpent 2, has notably diversified the applications of the code. A considerable effort in the current development is devoted to multi-physics, i.e., the coupling of continuous-energy Monte Carlo neutron transport simulation to state-of-the art methods in fields such as thermal–hydraulics and fuel research.

Multi-physics calculations with Serpent have initially focused on coupled neutronics and fuel behavior calculations: The first simple coupled calculation of fuel performance and neutronics using Serpent as the neutronics tool was conducted with Serpent 1 and the ENIGMA fuel performance code (Viitanen and Tulkki, 2012). In this initial study, a 3D system was approximated by using 2D neutronics in neutronically separated axial layers. An internally coupled simple fuel temperature solver was implemented later in Valtavirta et al. (2014b), which proved the concept for using internal two-way coupled temperature feedback with the Target Motion Sampling (TMS) on-the-fly temperature treatment technique, which itself is described in Viitanen (2015). Later studies have focused on subjects such as an externally coupled solver in depletion calculations (Valtavirta et al., 2013), as well as the internally coupled FINIX fuel behavior module in steady-state Ikonen et al., 2013 and prompt-super critical transient conditions (Valtavirta et al., 2014a). In recent years the multi-physics work with Serpent has expanded from fuel behavior coupling to coupling with thermal–hydraulics (Daeubler et al., 2015; Knebel et al., 2016; Tuominen et al., 2016) and solid mechanics (Aufiero et al., 2015).

2.1. Current multi-physics approach in Serpent 2

One of the main aspects of the current work on Serpent 2 is the expansion of the multi-physics capabilities of the code. The capability to easily model systems with varying material temperature and density distributions opens up the way for studying the important feedback effects in operating nuclear reactors using the accurate Monte Carlo neutronics solution coupled with high-fidelity solvers for the coupled temperature and density fields. Monte Carlo codes are also an excellent tool for calculating the reference neutronics solution to the three dimensional heterogeneous full-core problem. Without the capability to model a system with hundreds of material temperatures and densities the reference solution can only be obtained for the zero-power reactor. With the multi-physics capabilities in Serpent, the reference solution can also be obtained for the hot full-power system as long as the material temperature and density fields are provided by a separate solver (Leppänen and Mattila, 2016; Daeubler et al., 2015; Knebel et al., 2016). The multi-physics approach in Serpent is shortly described in the following.

To incorporate the temperature and density changes into the neutron tracking and interaction physics, Serpent 2 relies on two features:

1. The rejection sampling of neutron path lengths.
2. The capability to handle the temperature dependence of microscopic cross-sections on-the-fly by the Target Motion Sampling temperature treatment technique (Viitanen and Leppänen, 2012, 2014, 2015).

These two aspects of the code enable Serpent 2 to model materials with arbitrarily refined temperature and density distributions with good efficiency (Leppänen, 2013; Leppänen et al., 2014). The solution transfer between Serpent and different solvers is based on a universal multi-physics interface designed to separate the state-point (temperature and density) information from the actual geometry model (Leppänen et al., 2012). The routines allow the tracking routine to efficiently handle complex, or even continuous, temperature and density distributions while using delta-tracking, surface-tracking or a combination of the two. From the user's point of view the separation of state-point and geometry information makes it possible to bring in the temperature and density solutions without any modifications to the main input.

The solution coupling scheme in Serpent 2 has two levels as shown in Fig. 1. The internal lightweight solvers FINIX (Ikonen et al., 2015), the fuel behavior module used in this work, and COSY,

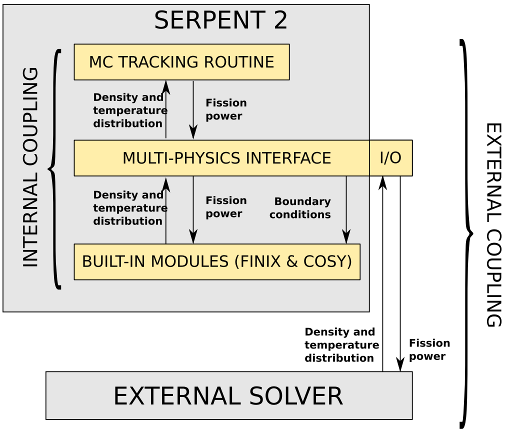


Fig. 1. The two-level coupling scheme in Serpent 2. Internal solvers give reasonably accurate solutions with small computational cost. High-fidelity solutions from state-of-the-art solvers can be coupled to Serpent 2 externally using the universal multi-physics interface.

[1] For a complete description of the code and the latest news, see the project Web site: http://montecarlo.vtt.fi.

a porous medium TH-solver (still under development but described shortly in Leppänen et al. (2015)), are meant to provide reasonably accurate solutions with a small computational cost, while the universal multi-physics interface can be used to pass solutions between Serpent and state-of-the-art high-fidelity external solvers. The internal coupling functions with direct memory access will be demonstrated in this paper with the internally coupled FINIX fuel behavior module. The external coupling via a universal multi-physics interface is based on sequential exchange of input and output files and is demonstrated by a coupling with the ENIGMA fuel performance code using the fuel behavior interface.

3. Fuel behavior interface in Serpent 2

While some results from simulations using the fuel behavior interface have been presented before (Valtavirta et al., 2013, 2014a; Ikonen et al., 2015), the previous studies have utilized various developmental versions of the fuel behavior interface and different naive iteration schemes between the neutronics and fuel behavior solvers. Moreover, the interface has not been described in detail in journals previously.

The fuel behavior interface handles radial temperature and strain distributions in fuel pin geometries and it has been designed specifically for solvers using the so-called 1.5-dimensional geometry model, where the fuel rods are divided into axial layers and the heat transfer and thermomechanical problems are solved separately for each axial layer. This is why separate radial temperature and strain distributions can be brought in for different pins and axial elevations. Furthermore, the temperature and strain distributions can be stored separately for different simulation times yielding the possibility to use temperature distributions that vary both in space and in time (Valtavirta et al., 2014a). The fuel behavior solution is brought in at radial nodes defined at the cold radial coordinates of the system (top part of Fig. 2). The fuel behavior solution itself is simply the temperature and hot state coordinate of each node (bottom part of Fig. 2).

The interface will thus contain three values for each radial node k

$$r_k^{\text{cold}}, r_k^{\text{hot}}, T_k \qquad (1)$$

namely the radial coordinate of the node in the cold state, the radial coordinate of the node in the hot state and the temperature at the node. This list allows calculating the material temperatures and densities anywhere between the innermost and outermost radial node. In the external coupling these lists are provided via an interface file separately for different fuel pins and their axial and angular zones. In the internal coupling these lists are populated by copying the values from the FINIX solution arrays using direct memory access. These three quantities are then used during the neutron transport to take in account the effects of temperature, geometry and density changes on the neutronics as described in the next sections.

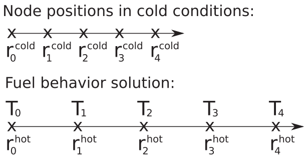


Fig. 2. The fuel behavior interface takes in a list of the cold radial coordinates of the solution nodes (x) accompanied with the fuel behavior solution consisting of the hot-state coordinates of the nodes and the temperature at each node. Separate lists can be given for different axial or angular segments.

3.1. Changes in material temperatures

Modeling the local material temperature correctly is important as the thermal motion of the target nuclei affects their microscopic cross sections. In low-enriched uranium-based fuel materials, the resonance peaks in the cross sections broaden with increasing temperature. This increases the losses of neutrons to resonance absorption (mainly by ^{238}U), which results in a reduced reactivity of the system and a reduced fission rate near high fuel temperature regions.

The TMS temperature treatment technique in Serpent only requires the material temperature at the interaction points. This allows the calculation of the local temperature on-the-fly based on the precise coordinates of the interaction as well as the temperatures and hot-state coordinates of the adjacent radial nodes. In this study, linear interpolation is used with respect to the radial coordinate, allowing Serpent to use temperature distributions that are continuous in the radial direction. No interpolation is currently done in the angular or axial direction.

For the TMS temperature treatment, the microscopic cross sections are initially broadened to a material-wise minimum temperature in the beginning of the simulation. A material-wise maximum temperature is used for calculating majorant cross sections used in the rejection sampling of path lengths in the TMS treatment. Rejection sampling is used to account for the fact that the material temperature, and thus the macroscopic cross section, is allowed to vary over the sampled path length. These minimum and maximum temperatures for each material can be specified in the input-file. The specified limits should cover the whole temperature range that is expected to be encountered during the iteration of the coupled solution. Later use of temperatures outside this material-wise pre-specified interval will lead to unphysical results, which is why the simulation will be terminated in such cases.

3.2. Changes in geometry

In order to accurately model the coupled problem, the changes in the problem geometry due to thermomechanical phenomena need to be addressed in the neutronics model. They have a slight impact on the reaction rates, e.g., through the changes in the moderator volume resulting from the changes in the cladding outer radius of the fuel rods.

The Serpent geometry model is based on material cells, which are defined by elementary surfaces such as cylinders and planes in the case of three dimensional fuel rods. The updated geometry has to be taken in account in two subroutines: First, sampling a specific interaction type at interaction points requires the local material cross sections. This means that the interaction coordinates have to be translated into the local material cell. Second, the surface tracking routine requires the distance to the closest cell boundary for a neutron with a certain position $\vec{r}$ and velocity $\vec{v}$. Both the cell search by coordinates and the calculation of the distance to the first boundary crossing have been slightly modified to take in account the updated geometry as explained in the following paragraph.

A simple hot-to-cold transformation for the interaction coordinates makes it possible to use the standard cell search routine in obtaining the geometry cell of the interaction. Secondly, in order to calculate the distance to the closest cell boundary, one has to first determine the current material cell and perform a cold-to-hot coordinate transformation for the cylindrical inner and outer surfaces of the material zone. After that, the default search for the nearest boundary can be utilized. The coordinate transformations are currently based on linear interpolation of the thermal expansion between the radial nodes.

3.3. Changes in material densities

The atomic densities of materials are used in the calculation of the material-wise cross sections, which in their part are used in Monte Carlo neutron transport for the sampling of the neutron path lengths between interaction points.

Serpent calculates the local macroscopic material cross sections based on the cross section at a base material density as well as the ratio of the local material density and the base material density. This ratio of densities is called the density factor. The density factor between nodes k and $k+1$ can be calculated from the radial coordinates given in the interface input:

$$DF(r) = \frac{\rho^{\text{hot}}(r)}{\rho^{\text{cold}}(r)} = \frac{(r_{k+1}^{\text{cold}})^2 - (r_k^{\text{cold}})^2}{(r_{k+1}^{\text{hot}})^2 - (r_k^{\text{hot}})^2}, \tag{2}$$

where ρ is the material density, the subscripts refer to the node indices and the superscripts refer to the hot and cold conditions. The density factor is equal to the ratio of the macroscopic cross sections in the hot and cold state and is used in the rejection sampling of the neutron path lengths. Rejection sampling is used by Serpent to allow the material density to vary over the sampled path lengths.

3.4. Interface output

The fuel behavior interface tallies the deposited fission power and fast flux using a user defined axial, angular and radial binning for each of the monitored fuel pins. In the internal coupling with FINIX, the fission power is transferred internally into the corresponding FINIX-arrays to be used in the next solution iteration by FINIX. In the external coupling, these variables are written into a plain text file, to be used by the external program.

4. Coupled calculation sequence

The neutron transport problem is a linear one as long as the material cross sections are independent of the neutron transport solution. The problem becomes non-linear, when the fission power distribution is coupled to the material temperatures and densities and subsequently to the cross sections. The traditional method for solving the non-linear problem, employed typically by core and fuel cycle simulator codes, is to use the so-called operator method, where the sub-problems are solved separately. Typically the different solvers are iterated until a converged solution is found. This solution process, consisting of the subsequent iteration of the neutronics and coupled field solvers, is also employed in this work.

The initialization of a Monte Carlo calculation can be quite time consuming with the cross section preparation and the initial source convergence taking a significant amount of time. Previously the coupled calculations have required multiple separate Serpent simulations with a separate initialization stage for each simulation. To increase the efficiency of coupled calculations, several new routines were written that allow Serpent to exchange information with other solvers and wait for an updated temperature and density solution, without terminating the Serpent simulation in between. Moreover, the fission source can be stored at the end of each iteration to be used as an initial source on the next iteration. This will reduce the number of inactive cycles required to reach a converged fission source on the later iterations, without the need to store the fission source to a file in between.

Using an internal module, such as FINIX, the fuel-behavior solution can be updated by only making the correct subroutine calls after simulating a specified number of neutron batches. By not terminating the simulation in-between, the initialization has to be done only once. Fig. 3 shows the solution flow using the internal coupling: After intializing the calculation (cross-section loading and pre-processing etc.) the first transport solution is obtained. After the neutronics solution, the power distribution can be relaxed before passing it to the internal solver. The updated solution is then calculated by the internal solver and the temperature and density distributions are updated for the neutron tracking routines. The convergence of the solution can be estimated based on the updated temperature distribution. If the solution has not converged to the required level, the next transport solution can be obtained. The fission source stored at the end of the previous transport solution can be used as the initial fission source for the next solution. The same iteration scheme can be used for solving fuel behavior, coolant behavior or both. The methodology can also be extended to depletion or transient calculations, but their description will be left to future publications.

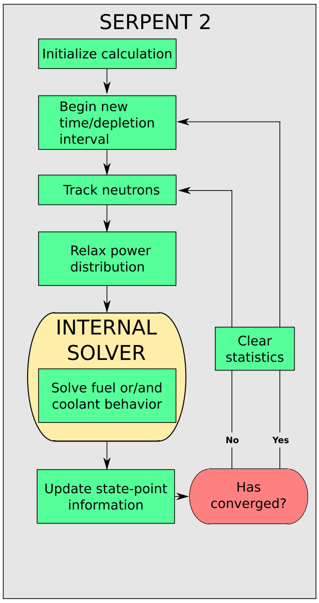


Fig. 3. The iterative coupled solution procedure used with internally coupled solvers.

With an externally coupled solver, the program flow has to be controlled by other means. Serpent 2 supports two methods for this: Reading and writing communications files and POSIX-signaling. In the file-based communication mode, Serpent and the external program will write their signals to separate files, that can be read by the other program to receive the signal. In the POSIX-signaling mode, the external coupling program provides Serpent input file with its own process ID number (pid) that Serpent will use for the communication, the external program can identify the process ID of Serpent based on the incoming signals sent by Serpent.

The signal numbers that are sent in each of the signaling modes will be the same: After simulating a specified number of neutron batches, Serpent sends the parent process a certain POSIX signal to indicate that Serpent is waiting for a new solution of the coupled fields. After the external program calculates the updated solution and sends a POSIX signal back, Serpent reads the updated multi-physics interface files and continues the simulation. Different signals are used to indicate the moving to the next time point (in transients or burn-up calculations) and the completion of the simulation. The schematic of the solution flow using external coupling is shown in Fig. 4. The solution flow is very similar to that of

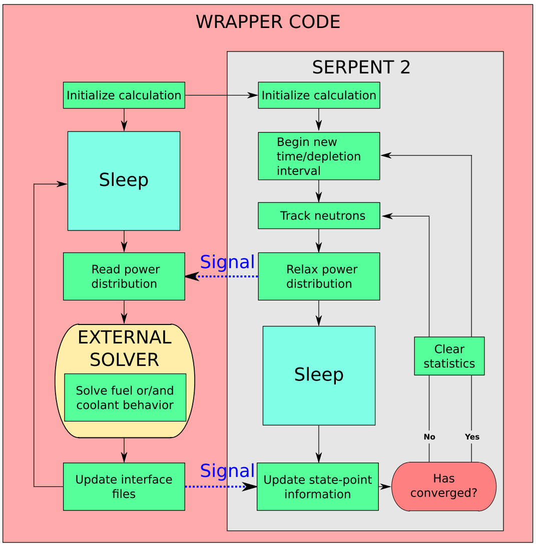


Fig. 4. The iterative coupled solution procedure used with externally coupled solvers.

the internal coupling, which means that many of the same interface routines can be used for both coupling types. This coupling mode is meant to be universal in the sense that it can be used with any type of external solver and any of the multi-physics interface formats provided by Serpent.

On the general level both the internal coupling and the external coupling accomplish the same thing: exchanging the relevant result fields between two solvers. Whichever approach can be used in code-to-code coupling but there are certainly some factors to consider when a choice about the coupling approach is made: The external coupling does not necessarily require any source code modifications to either solver, even more, the external coupling can many times be achieved without the source code of either solver, which may make it the only way to execute multi-physics calculations with tools that are distributed as pre-compiled binaries. The downside to this flexibility is then the constraint of the standard input/output format of both codes as well as the file based data transfer which is typically much slower than direct memory access. This is especially important if the amount of data that has to be transferred is large. The internal coupling gives, in general, more possibilities to a code-development oriented user. The execution order, data transfer and parallelization of the different solvers can be controlled at the source code level.

5. Solution relaxation

To enhance the convergence of the coupled solution, it is possible to apply solution relaxation methods to either the neutronics or the coupled solution, e.g., temperature or density solution. Serpent uses the stochastic approximation based relaxation for the relaxation of the fission power distribution in steady state coupled Monte-Carlo neutronics – thermal–hydraulic calculations described by Dufek and Gudowski (2006) and recapped in Dufek and Hoogenboom (2014). A separate derivation of applicability for a simulation with also fuel temperature variations was presented by Ivanov et al. (2014).

To restate the main results from Dufek and Gudowski (2006), the relaxed neutron flux at iteration n, $\phi^{(n)}$, can be calculated from the previous estimate for the flux $\phi^{(n-1)}$ and the new Monte Carlo estimate for the flux $\widehat{G}_{s_n}(\phi^{(n-1)})$ by

$$\phi^{(n)} = \phi^{(n-1)} - \alpha_n d\left(\phi^{(n-1)} - \widehat{G}_{s_n}(\phi^{(n-1)})\right), \tag{3}$$

where $d \in (0, 1]$ is a fixed underrelaxation factor, α_n is the step-size in the n:th step and s_n is the sample size used to calculate $\widehat{G}_{s_n}(\phi^{(n-1)})$. The initial guess for the neutron flux $\phi^{(0)}$ can be obtained, e.g., by a Monte Carlo solution with some initial conditions for the temperature and density fields. In a situation where $\phi^{(0)}$ is not well known, Dufek and Gudowski advice to set $d = 1$ yielding

$$\phi^{(n)} = (1 - \alpha_n)\phi^{(n-1)} - \alpha_n \widehat{G}_{s_n}(\phi^{(n-1)}). \tag{4}$$

The choice of the sample size and step size for iteration n can be done in several ways as described by Dufek and Gudowski. The most simple one is to use a fixed sample size s and use the corresponding step sizes

$$\alpha_n = \frac{1}{n}, \tag{5}$$

turning Eq. (4) into a simple average of the previous Monte Carlo solutions

$$\phi^{(n)} = \frac{1}{n}\sum_{i=1}^{n} \widehat{G}_s(\phi^{(i-1)}). \tag{6}$$

Dufek and Gudowski also noted that it is usually a good idea to start from a small sample size s_n and let it grow as the iteration number n grows. The first iterations may contain large systematic errors due to a non-converged coupled solution, this means that it is not practical to use a large sample size, and a long computing time, to obtain small statistical error in a case where there is still a large systematic error present. It is best if the statistical and systematic errors decrease in a similar manner. One solution for this, as stated by Dufek and Gudowski, would be to use a small and fixed s_n over all iteration steps yielding Eq. (6), as the statistical accuracy of the neutron flux estimate gets better as the number of iterations increases. The main drawback to this model according to Dufek and Gudowski was the time wasted in the initialization phase of the Monte Carlo code when running a large number of small population Monte Carlo simulations. It is now pointed out that using the coupled iteration scheme presented in Section 4 of this paper, the initialization phase of the Monte Carlo code is only run once, regardless of the number of iterations n that is to simulated. As such, there is no extra time wasted initializing the Monte Carlo code between the solutions and the small, fixed sample size was used in this paper.

However, it should also be stated that if the solution of the temperature and density fields is time consuming compared to the Monte Carlo solution, it may be beneficial to increase the sample size with increasing iteration number as this ultimately requires fewer calls to the coupled solver at the cost of updating the temperature solution less frequently. In those cases one of the methods for calculating an increasing sample size s_n and the corresponding step sizes α_n provided by Dufek and Gudowski (2006) will be useful. Increasing the sample size can be done easily by modifying the simulated neutron population during the simulation.

6. Monitoring the convergence of the coupled solution

To monitor the convergence of the solution during the simulation, it is not enough to follow the multiplication factor of the system (Salino and Hébert, 2016). Instead two methods applied to the thermal–hydraulic fields are used in Serpent. They both require the calculation of the local relative difference between subsequent solution fields. Denoting the number of the iteration by n, and the spatial node index (specifying pin, axial zone and radial node) by k the local relative difference in field Q will be

$$\epsilon_k^n = \frac{Q_k^n - Q_k^{n-1}}{Q_k^n}. \tag{7}$$

The two convergence criteria calculated by Serpent during to coupled calculations are the maximum absolute value of ϵ over the whole geometry

$$\epsilon_{\max}^n = \max_k |\epsilon_k^n| = \max_k \frac{|Q_k^n - Q_k^{n-1}|}{Q_k^n}. \tag{8}$$

and the ℓ^2-norm of the ϵ field:

$$\ell_n^2 = \sqrt{\sum_k (\epsilon_k^n)^2} = \sqrt{\sum_k \left(\frac{Q_k^n - Q_k^{n-1}}{Q_k^n}\right)^2}. \tag{9}$$

The first of the two criteria (Eq. (8)) measures the greatest local relative change in the solution field but does not consider the convergence of the field as a whole. The ℓ^2-norm, on the other hand, considers the relative changes in the whole solution field. Both of the convergence criteria are calculated for three fields: The unrelaxed (iteration based) power distribution, the relaxed power distribution and the temperature distribution obtained from the coupled solver.

7. Fuel behavior solvers

7.1. FINIX

The internally coupled solver FINIX is a novel light-weight fuel behavior module designed specifically for coupled calculations by Ikonen at VTT (Ikonen et al., 2013). In addition to the coupling with Serpent, FINIX has been coupled at source code level to the reactor dynamics codes TRAB3D and TRAB-1D (Ikonen et al., 2015) and HEXTRAN (Ikonen et al., 2015). FINIX solves the time dependent thermal and mechanical behavior of the fuel rod using the 1.5-dimensional geometry model typical for fuel performance codes. Each fuel rod is divided into several axisymmetric axial zones and the thermal and mechanical behavior are solved separately in each axial zone. The axial zones are only loosely coupled, based on the gas gap pressure, which is solved simultaneously for the whole rod.

FINIX solves the heat transfer in the fuel pellet, gas gap and cladding as well as heat transfer from cladding to coolant. The mechanical behavior of the pellet and cladding are also solved separately for each of the axial zones. While the current capabilities of FINIX are specifically directed to calculating the thermal response of fuel rods in fast transient scenarios, FINIX can also be used to calculate the steady state conditions for the fuel rod. Further information concerning FINIX can be found in Ikonen et al. (2013, 2015) and Ikonen (2013).

7.2. ENIGMA

The externally coupled solver used in this paper is the fuel performance code ENIGMA for steady-state and transient conditions, developed by Nuclear Electric and BNFL in the United Kingdom (Kilgour et al., 1991). The version of the code used in this study is based on ENIGMA v.5.9b, but has been modified at VTT over the years, mostly for model extension and addition for purposes concerning for example cladding material properties, fission gas release and gadolinia-doped fuel rods. The extension most relevant to this work makes it possible to read in the radial power distribution from a separate file in order to bring in the fission power profile from an external neutronics solver. The extension was initially written for modeling the radial power distribution in burnable absorber rod accurately during the absorber burn up, but the routines have also been utilized in previous multi-physics calculations with Serpent 1 and ENIGMA (Viitanen and Tulkki, 2012).

The geometry model of ENIGMA also follows the 1.5-dimensional model. The axial coupling between the different zones comes from the coolant temperature boundary condition, rod internal pressure as well as fission gas transport. A good overview of the history, structure and applications of ENIGMA as well as the recent development of the UK National Nuclear Laboratory version can be found in Rossiter (2011).

8. Effective fuel temperatures

Traditionally, neutronics calculations have been conducted with a radially constant fuel temperature distribution, which may be different for different fuel rods and/or axial elevations. The driving idea behind the choice of the effective temperature is the conservation of some parameter of interest, for example the multiplication factor or the resonance absorption rate of the system. A very simple way to calculate this effective temperature is to use a predetermined linear combination of the pellet centerline and surface temperatures

$$T_{\text{eff.}} = \alpha T_c + (1-\alpha) T_s, \tag{10}$$

where T_c and T_s are the pellet centerline and surface temperatures, respectively, but several other techniques exist, such as volume averaging the temperature in the radial direction

$$T_{\text{eff.}} = \frac{\int_r T(r)\mathrm{d}r}{\int_r \phi(r)\mathrm{d}r} \tag{11}$$

or calculating a weighted volume average

$$T_{\text{eff.}} = \frac{\int_r w(r)T(r)\mathrm{d}r}{\int_r w(r)\mathrm{d}r} \tag{12}$$

with a radial weighting distribution $w(r)$. A more advanced technique based on averaging the temperature distribution based on possible neutron path-lengths through the fuel pellet, called chord-averaged effective temperature, was proposed by Rowlands (1962). Many times a simplified form of the Rowlands' effective temperature formula is used:

$$T_{\text{eff.}} = T_s + \frac{4}{9}(T_c - T_s), \tag{13}$$

The idea of the chord-averaged temperatures was later extended by de Kruijf and Janssen (1996).

The capability to use the accurate temperature distribution in a neutronics simulation provides a way to calculate a reference solution that the effective temperature models can be compared to. It also enables other interesting means to choose the weighting function and calculate the volume integrals in Eq. (12): The Monte Carlo neutron transport can be used to evaluate integrals of the form

$$\int_V \int_E f(\vec{r},E)\phi(\vec{r},E)\mathrm{d}E\mathrm{d}V, \tag{14}$$

where $f(\vec{r},E)$ is a chosen response function. If we choose the response function to be

$$f(\vec{r},E) = T(\vec{r})g(\vec{r},E), \tag{15}$$

we can use the Monte Carlo method to calculate statistical estimates for integrals of the form

$$\int_V \int_E T(\vec{r})g(\vec{r},E)\phi(\vec{r},E)\mathrm{d}E\mathrm{d}V \tag{16}$$

which we can use to calculate the effective temperatures using different flux or reaction rate weightings. The flux weighted effective temperature, where the weighting function is the local one-group flux, is easily calculated by setting $g(\vec{r},E)=1$, using two different tallies and taking the fraction of the two:

$$T_{\text{flx.}} = \frac{\int_V \int_E T(\vec{r})\phi(\vec{r},E)\mathrm{d}E\mathrm{d}V}{\int_V \int_E \phi(\vec{r},E)\mathrm{d}E\mathrm{d}V}. \tag{17}$$

This equation is actually equal to setting

$$w(r) = \int_z \int_\theta \int_E \phi(\vec{r},E)\mathrm{d}E\mathrm{d}\theta\mathrm{d}z \tag{18}$$

in Eq. (12). Here z is the axial coordinate and θ is the azimuthal angle in cylindrical coordinates.

In a similar vein, we can set $g(\vec{r},E)=\Sigma_{\text{tot.}}(\vec{r},E)$ in Eq. (16) and calculate the local collision rate weighted temperature

$$T_{\text{col.}} = \frac{\int_V \int_E T(\vec{r})\Sigma_{\text{tot.}}(\vec{r},E)\phi(\vec{r},E)\mathrm{d}E\mathrm{d}V}{\int_V \int_E \Sigma_{\text{tot.}}(\vec{r},E)\phi(\vec{r},E)\mathrm{d}E\mathrm{d}V} \tag{19}$$

or even set the response function $g(\vec{r},E)$ to be the local macroscopic radiative capture cross section of ^{238}U: $\Sigma_{(n,\gamma)}^{^{238}\text{U}}(\vec{r},E)$ and calculate the local ^{238}U radiative capture rate weighted temperature

$$T_{\text{capt.}}^{^{238}\text{U}} = \frac{\int_V \int_E T(\vec{r})\Sigma_{(n,\gamma)}^{^{238}\text{U}}(\vec{r},E)\phi(\vec{r},E)\mathrm{d}E\mathrm{d}V}{\int_V \int_E \Sigma_{(n,\gamma)}^{^{238}\text{U}}(\vec{r},E)\phi(\vec{r},E)\mathrm{d}E\mathrm{d}V}. \tag{20}$$

Since these two effective temperatures directly weight the temperature with the physical reaction rates, they should be able to conserve those reaction rates to a high degree. However, as moving from realistic temperature distributions to effective temperatures will also perturb the local flux distribution, the reaction rates are not perfectly conserved.

We'll estimate the performance of several traditional radial effective temperature models as well as the new Monte Carlo effective temperatures (Eqs. (19) and (20)) in Section 10.4.

9. Test calculation

The calculation system was demonstrated on a realistic 3D BWR assembly geometry based on the Peach Bottom 2 model in the UAM benchmark (Ivanov et al., 2013). The pin configuration can be seen in Fig. 5 and Table 2. The system specifications are shown in Table 1. The modeled 3D geometry was finite in the axial direction with a black boundary condition. The boundary conditions in the horizontal plane were reflective. The height of the system was set to a realistic 365.76 cm. The flux level in the neutronics calculations was normalized to the total power produced in the assembly which was set to 3.9429 MW, resulting in an average linear power of 220 W/cm per rod. The coolant temperature was set to 557 K for the whole axial length. While the constant coolant temperature is non-physical for the chosen power level, the resulting axially symmetric model allows us to estimate the solution convergence based on the symmetry condition as described in Section 10.2.

The FINIX and ENIGMA inputs both used the relevant values from Table 1 for the rod and system parameters. The boundary condition chosen for the heat transfer problem was to fix cladding outer surface temperature to 557 K for each axial node in each rod in order to only compare the heat transfer models inside the fuel rod. Both fuel behavior solvers divided the fuel rod into 24 axial zones. The radial nodalization in the FINIX model used 51 nodes spaced evenly in radius between the pellet centerline and surface as well as 20 nodes spaced evenly in radius between the cladding inner and outer surface. The radial nodalization in the ENIGMA model consisted of 50 fuel rings of equal width and a single ring for the cladding. Some differences are expected to arise from different correlations or solution methodology used in the codes. As the purpose of this work is not to compare the codes, but to only demonstrate the use of both the internal and external couplings, these differences are only discussed shortly from the point of view of the neutronics solution in Section 10.3.

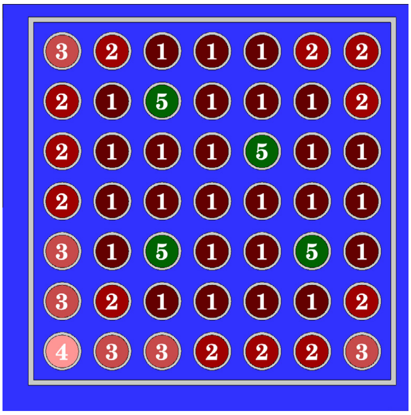


Fig. 5. The BWR assembly with different fuel types marked with numbers 1–5 (see Table. 2).

Table 1
System properties of the modeled BWR assembly.

Coolant properties (uniform)	
Coolant pressure	7.14 MPa
Coolant temperature	557 K
Coolant density	460.72 kg/m^3
Void fraction	40%
Fuel pin geometry	
Active height	365.76 cm
Cladding outer diameter	0.71501 cm
Cladding inner radius	0.62103 cm
Pellet outer radius	0.60579 cm
Fuel pellet properties	
Fuel enrichment	1.33–2.93 wt.%
Fuel density	95.1% of theoretical density
Fuel Gd_2O_3 content	0 or 3 wt.%
Fuel pellet height	1.067 cm
Fuel pellet chamfer	none
Fuel pellet dishing	none
Average grain size	20 µm
Cladding properties	
Cladding material	Zircaloy-2
Cladding surface roughness	0.50 µm
Cladding yield strength	350 MPa
Fill gas properties	
Fill gas composition	100% He
Fill gas pressure	0.69 MPa

Table 2
Composition of the different fuel materials in the BWR assembly.

Type	Uranium enrichment (wt.%)	Gd_2O_3 content (wt.%)
1	2.93	0.00
2	1.94	0.00
3	1.69	0.00
4	1.33	0.00
5	2.93	3.00

The fission power was tallied by Serpent individually for each unique[2] lattice position. The rod-wise power tallies were divided axially into 24 zones to match the axial nodalization of the coupled codes. Radially the fission power tally was divided into 10 rings of equal area covering the fuel pellet.

9.1. Solution transfer between Serpent and FINIX

The temperature and strain distributions calculated by FINIX can be used in Serpent 2 as is, without further mapping operations. Due to the source code level coupling, the solution transfer between the codes was simple. After each fuel behavior solution, the new temperature distribution as well as the deformed coordinates of each axial node were copied from the FINIX solution arrays to the Serpent multi-physics interface data structures using direct memory access. The power distribution calculated by Serpent 2 was also transferred to the FINIX data structures using direct memory access.

9.2. Solution transfer between Serpent and ENIGMA

A small wrapper program was written for the solution transfer between Serpent and ENIGMA as well as for controlling the program flow according to Fig. 4. The temperature solutions of ENIGMA were read from the human readable ENIGMA output file *.op8*. The temperatures written to the file are the pellet centerline and surface temperatures, temperatures at the cladding inner and outer surface and the mid-temperatures of each fuel ring. The new radial coordinates at the nodes can be obtained from the printed hot-inner radii of the fuel-rings as well as the hot pellet surface and cladding inner and outer surface radii. Because the temperatures are given at mid-radius of each fuel ring and the strains are given at the inner surfaces of the rings, some additional mapping is needed. The strains at the middle of each fuel ring are calculated as a linear average between the inner and outer radius of the annulus. After this, the temperatures and updated coordinates of the radial nodes can be given to Serpent at pellet centerline, at the middle of each fuel ring, at pellet outer surface and at cladding inner and outer surfaces.

The power is transferred from Serpent to ENIGMA separately for each rod by giving the total linear power of each axial zone, and a radial power depression distribution based on the volumetric radial power densities calculated by Serpent. With this input, ENIGMA calculates the local nodal power densities to be used in the fuel behavior calculation.

9.3. Run parameters

A reference solution was calculated with both code systems using 400 iterations with 20 million active neutron histories simulated in each iteration and stochastic approximation based relaxation applied to the power distribution. To minimize the effect of the initial guess for the temperature and strain solution, the initial guess was calculated using a similar coupled solution (400 iterations with 20 million active histories in each iteration). The initial guess in the reference calculation should be thus very close to the final result. The 20 million neutrons in each iteration were divided into 1000 source cycles with 20 000 neutrons in each cycle. The number of additional inactive neutron batches was set to 300 for the first iteration and 100 for the subsequent iteration, based on the evolution of the cycle-wise Shannon entropy of the fission source. The simulation used JEFF-3.1.1 based cross section libraries.

10. Results

10.1. Overview

This overview will use the results from the internally coupled calculation with FINIX. The results from the externally coupled calculations with ENIGMA were very similar and the slight differences are described in Section 10.3. The 400 internally coupled iterations took 57 h on 16 cores of an Intel Xeon E5-2698 with hyperthreading using 32 OpenMP threads (one for each logical core of the processor). The two convergence criteria (Eqs. (8) and (9)) for the temperature distribution calculated at the last iteration were $\epsilon^{400}_{\max} = 8.26 \times 10^{-5}$ and $\ell^2_{400} = 1.17 \times 10^{-5}$. The main result from the coupled simulation is the high-fidelity temperature distribution seen in Figs. 6 and 7. This temperature distribution can be used in calculating further neutronics results such as group constants, multiplication factor or feedback coefficients for the system.

The fuel centerline temperatures in each lattice position as well as the total rod power levels are shown in Figs. 8 and 9. We can see that the hottest pin is at lattice position 9 and the coldest pin is at lattice position 23, which is a burnable absorber rod. For our detailed sub-pin level analysis we'll focus on the hottest pin (lattice position 9) and the burnable absorber pin in the adjacent lattice position 14.

10.2. Estimating final convergence based on the symmetry condition

The final convergence criteria presented in the previous section only measure how much the temperature solution changed on the last iteration step. They do not directly indicate how far the solution is from the converged coupled solution. In general, the only way to accurately estimate the remaining error in the coupled Monte Carlo solution is to compare the solution to a known reference solution, which does not exist in the general case.

In this case, due to the axial symmetry in the model, we can calculate how far the temperature distribution is from an axially symmetric distribution by subtracting a symmetric temperature field

$$(\Delta T(x,y,z))_{\text{sym}} = T(x,y,z) - T_{\text{sym}}(x,y,z) \tag{21}$$

where the symmetric temperature field is an average between the original field and an axially mirrored temperature field:

$$T_{\text{sym}}(x,y,z) = \frac{T(x,y,z) + T(x,y,-z)}{2}. \tag{22}$$

Fig. 10 shows the departure from symmetric temperature field for the diagonal vertical cut through the assembly in the FINIX reference calculation. Only the bottom half of the assembly is shown as the departure from symmetry will be mirrored to the top part of the assembly with a change in the sign

$$(\Delta T(x,y,z))_{\text{sym}} = -(\Delta T(x,y,-z))_{\text{sym}}. \tag{23}$$

As the solution relaxation will eventually smooth out the statistical differences between the top and bottom halves of the assembly, we can expect the small differences to vanish if the number of iterations is increased indefinitely. The remaining differences are currently of the order of 1 K and should have negligible effect on the neutronics solution, which leads us to conclude that the reference calculation has converged to an acceptable level. We cannot, however, conclude that the solution has converged to the correct solution as any symmetric temperature distribution will fulfill the symmetricity condition, but only one will be the solution to the coupled problem.

[2] Due to the diagonal symmetry in the assembly there were only 28 unique lattice positions.

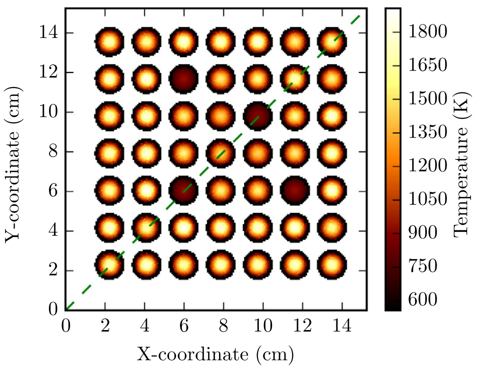


Fig. 6. Horizontal cut at axial midplane of the result temperature field from the FINIX reference calculation. The green line shows the path of the diagonal vertical cut used in the following figures.

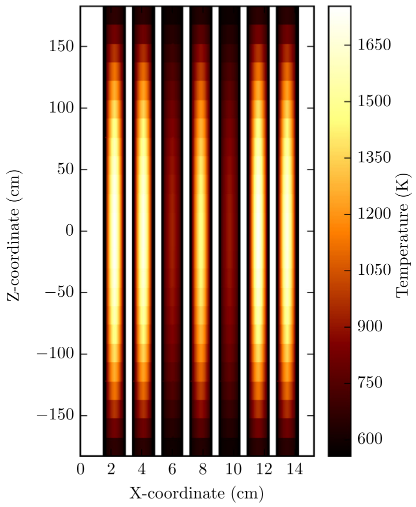


Fig. 7. Vertical cut diagonally through the assembly of the result temperature field from the FINIX reference calculation.

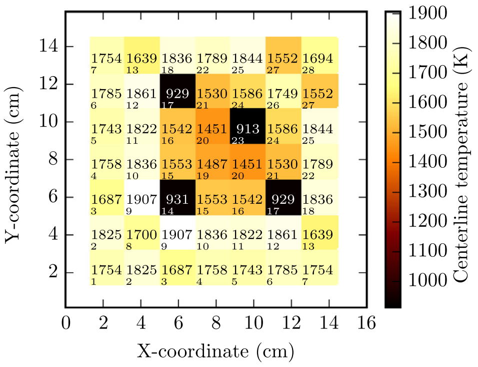


Fig. 8. Fuel centerline temperatures at different lattice positions at axial midplane. Lattice positions shown as small numbers in lower left corner of each position.

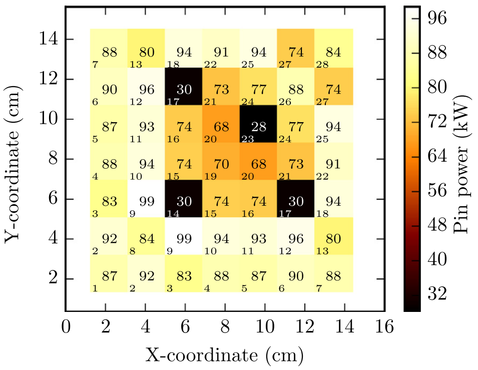


Fig. 9. Total rod powers at different lattice positions. Lattice positions shown as small numbers in lower left corner of each position.

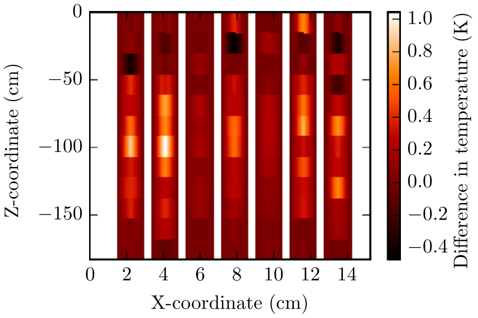


Fig. 10. Difference from a symmetrical temperature distribution, vertical cut diagonally through the bottom half of the assembly (FINIX reference calculation).

10.3. Effect of the fuel behavior solver

For future comparisons of multi-physics calculations done with different fuel behavior solvers, it will be of interest to estimate the differences resulting from the choice of the fuel behavior solver. In this paper the two solvers were the internally coupled FINIX solver and the externally coupled ENIGMA fuel performance code. In this section we'll present the differences in the temperature and power distributions in lattice positions 9 and 14 as well as the differences in the reactivity of the assembly.

The differences in the coupled temperature solution given by the two different fuel behavior solvers is shown in Fig. 11 for the two lattice positions. The coupled solution with ENIGMA yields slightly lower fuel temperatures throughout the whole geometry. In the hottest rod (position 9) we see a maximum temperature difference of approximately 48 K between the two solvers. The largest difference is located closer to the axial ends of the rod than to the axial midplane. In the burnable absorber rod (position 14) the difference follows an axially cosine-like distribution with only small variations in the radial direction.

The effect of these small temperature differences on the power distribution can be estimated by running a separate neutronics simulation using each temperature field and comparing the power distributions in the simulations. The axial symmetry was utilized in tallying the power distribution. The relative differences in the power distribution in the two lattice positions are plotted in Fig. 12. The local differences are small, staying below ±0.5% for the most part. The effect on the system reactivity is small but noticeable: The slightly smaller fuel temperatures of the ENIGMA solution yield a 47 ± 1 pcm larger reactivity than the FINIX solution. In this simulation the codes shared a common boundary condition for the cladding outer temperature. If a coolant temperature boundary condition was used instead, additional differences resulting from different heat transfer models between cladding and coolant might be seen.

10.4. Effect of detailed radial temperature distribution

In this section we calculate several result variables using either the accurate radial temperature fields or a radially averaged effective temperature in each axial zone. The effective temperatures were calculated separately for each unique lattice position, which means that the temperature distribution was still much more detailed than that typically used in assembly level calculations.

For each individual rod and axial elevation, we calculate the weighted volume average (Eq. (12)) using two different weighting functions: $w(r) = 1$, which is equal to the normal volume average of the temperature in the radial direction and $w(r) = 1/\sqrt{T(r)}$, which was suggested by Goltsev et al. Goltsev et al., 2000 and should give a larger weight to low temperature areas. Furthermore, we calculate the simplified Rowlands' effective temperature using Eq. (13) for each rod and axial elevation. Finally, we run a separate Monte Carlo simulation to calculate the local collision rate weighted (Eq. (19)) and local ^{238}U radiative capture rate weighted (Eq. (20)) temperatures in each axial layer of each rod[3]. These effective temperatures are used in a separate Monte Carlo simulation and the results are compared to a Monte Carlo simulation using the detailed temperature distributions.

As the fuel temperature strongly affects the resonance absorption of neutrons by ^{238}U, we are interested in the effect of the fuel temperature distribution on the capture rate by ^{238}U as well as the reactivity of the assembly. The reactivity of the system in the reference calculation was 5316 ± 0.5 pcm and the ^{238}U capture rate was 8.18×10^{16} s^{-1}. The differences in these two integral parameters introduced by the effective temperatures are presented in Table 3. The effective temperatures reproduce these integral parameters reasonably well, although statistically significant differences can be seen both between the reference simulation and the effective temperature simulations and between the effective temperature models themselves. The magnitude of the error in the system reactivity varies between 23 and 70 pcm, whereas the magnitude of the relative error in the ^{238}U capture rate varies between 0.05 and 0.20%. For both result variables, the ^{238}U capture rate weighted effective temperature model reproduces the reference results with the smallest error. The largest error in the reactivity of the system can be seen with the simple volume averaged effective temperature model whereas the largest error in the capture rate is seen with the Goltsev effective temperature model.

To quantify the differences in local reaction rates, the fission rate of ^{235}U, the radiative capture rate by ^{238}U and the radiative capture rate by Gadolinium isotopes were tallied with 24 axial bins and 50 radial bins of equal volume in each fuel rod. The axial symmetry was utilized in tallying the reaction rate distributions. We will present the local differences in the ^{235}U fission and ^{238}U capture rates for the hot rod (lattice position 9) and the local differences in the capture rate by Gd isotopes for the adjacent burnable absorber rod (lattice position 14).

Table 4 shows the maximum local relative difference in the reaction rates (either positive or negative, whichever is larger) as well as the relative differences in rod total reaction rates compared to the reference calculation. To visualize the spatial distribution of the difference, the relative difference in the local reaction rates for two effective temperature models (volume averaged and radiative capture weighted) compared to the reference solution are plotted in Figs. 13,14 for the capture and fission rates of the hottest rod (lattice position 9) and in Fig. 15 for the Gd capture rate of the burnable absorber rod (lattice position 14). A solid black curve has been plotted into each of the figures to indicate the curve of equal temperature between the effective and realistic temperature cases. This equal temperature curve is closer to the pellet surface for the radiative capture weighted effective temperature as is expected due to the ^{238}U radiative capture occurring predominantly near the pellet surface. From the other effective temperature models, the Rowlands and the collision rate weighted temperatures performed qualitatively similarly to the volume averaged effective temperature model in each of the cases, whereas the Goltsev effective temperature performed similarly to the ^{238}U capture rate weighted temperature model for all three reaction rates.

Looking at the relative difference in the capture density distribution we can see that both (and actually all) of the effective temperature models tend to overestimate the capture by ^{238}U near the fuel pellet surface and underestimate it in the inner parts of the pellet. This can be attributed to the combination of two facts: First, the pellet surface temperatures are considerably higher in the effective temperature models, which translates to wider resonance peaks in cross-sections and increased energy-width of resonance absorption by ^{238}U. Second, the strong resonance self-shielding effect of ^{238}U effectively prevents the neutrons at resonance energies penetrating deep into the fuel pellet. At the axial ends of the fuel rod, the volume averaged temperature distribution slightly overestimates the capture rate, whereas the capture weighted temperature fields slightly underestimate the capture rate at the axial ends. The smallest maximum local error is achieved with the Goltsev effective temperature model although the Goltsev model results in the largest rod total error in the capture rate. The smallest difference in the rod total capture rate is achieved with the ^{238}U capture rate weighted effective temperature.

The fission rate distributions calculated with the effective temperatures exhibit small relative differences compared to the reference case. It is interesting to see the totally opposite behavior of the two effective temperature models: The volume averaged temperature distribution slightly underpredicts the fission rates in the axial midplane and overpredicts them in the ends of the rod. The ^{238}U capture weighted temperature distribution, however, slightly overpredicts the fission rates in the axial midplane, while underpredicting them in the ends of the rod. The smallest local errors are obtained with the collision rate weighted effective temperature, while the rod total fission rate is best reproduced by the Goltsev effective temperature.

The predictions for the radiative capture of neutrons by the Gadolinium in the burnable-absorber rod (Fig. 15) exhibit a behavior similar to that observed in the fission rate distributions. The volume averaged temperature distribution again underpredicts the capture rates at axial midplane and overpredicts them in the ends of the rod, while the ^{238}U weighted temperature distribution overpredicts the capture rates in the axial midplane and underpredicts them in the ends of the rod. The collision weighted effective temperature is the best to reproduce both the local and rod total capture rates. It is interesting to note that the ^{238}U capture weighted temperature gives the largest rod total difference in the capture by the Gd-isotopes.

All in all, there does not seem to be one effective temperature model that would be the best choice for reproducing all of the different local and integral reaction rates although the ^{238}U radiative capture rate weighted temperature reproduced the rod-total and

[3] These temperatures could also be tallied already during the coupled calculation.

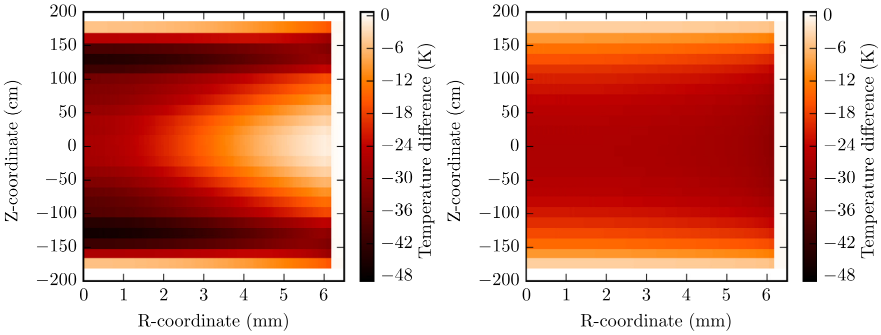


Fig. 11. Absolute differences in the temperature distribution between Enigma and FINIX (ENI–FIN). **Left:** Rod 9. **Right:** Rod 14.

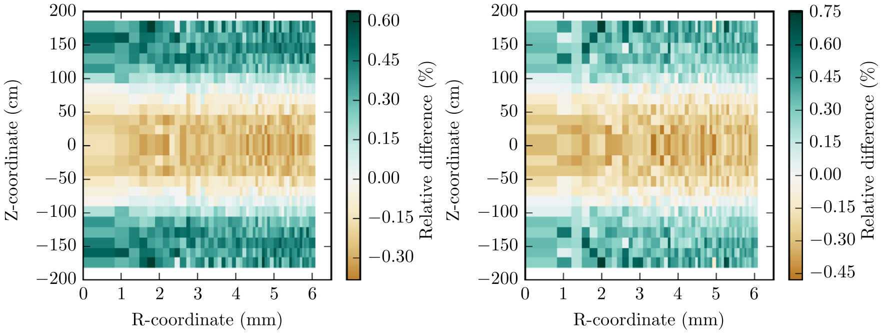


Fig. 12. Relative differences in the power distribution between simulations with Enigma and FINIX (ENI–FIN)/FIN. **Left:** Rod 9. **Right:** Rod 14.

Table 3
Differences in the values of two integral parameters of the assembly between the reference temperature model and several effective temperature models (effective - reference). The one sigma statistical uncertainty is below 1 pcm for the reactivity difference and below 0.0040% for the relative difference in capture rate in all cases.

$T_{\mathrm{eff.}}$ model	Difference in reactivity (pcm)	Difference in ^{238}U capture rate (%)
Vol. avg.	−70	0.17
Goltsev	50	−0.20
Rowlands	−53	0.11
Col. wgt.	−37	0.14
^{238}U capt. wgt.	23	0.05

assembly-total radiative capture rates by ^{283}U very well. The local relative differences in the radiative capture rate by ^{238}U in lattice position 9 range from −4.6% to +8.4% using the volume averaged temperature model. With the radiative capture weighted effective temperature model, the local relative differences were between −4.4% and +8.1%. The differences in fission rates and capture rates by the gadolinium isotopes were smaller, staying below $\pm 1\%$ with most effective temperature models. In each of the cases the effects may accumulate or lead to other differences with increasing burnup: Radiative capture by ^{238}U will produce fissile plutonium, ^{235}U fissions will use up the fissile nuclide and radiative capture by the Gd-isotopes will burn off the burnable absorber affecting the local neutron absorption. The effects of these differences during fuel life are hard to predict and should be estimated in a separate study focusing on coupled burn up calculations.

10.5. Accelerating the convergence with UFS

Since each coupled iteration will take valuable CPU time, the number of iterations required for the solution should be reduced as much as possible. In this section, we will describe the evolution of the iteration based convergence criteria during solution convergence and estimate, whether the convergence of the solution might be accelerated using the Uniform Fission Sites (UFS).

Table 4
Relative differences (percent) in the local reaction rates between the reference simulation and various effective temperature simulations. Maximum local difference and difference in rod total reaction rate presented.

	Rod 9 ^{238}U (n, γ)		Rod 9 ^{235}U (n,f)		Rod 14 Gd (n, γ)	
$T_{\mathrm{eff.}}$ model	Local max	Total rod	Local max	Total tod	Local max	Total rod
Vol. avg.	+8.4	+0.20	+0.8	−0.014	+0.8	+0.024
Goltsev	+7.6	−0.25	−0.8	−0.002	−0.9	−0.019
Rowlands	+8.3	+0.15	+0.9	−0.016	+1.0	+0.022
Col. wgt.	+8.5	+0.18	+0.4	−0.006	+0.5	−0.014
^{238}U capt. wgt.	+8.1	−0.05	−0.5	−0.004	−0.6	−0.035

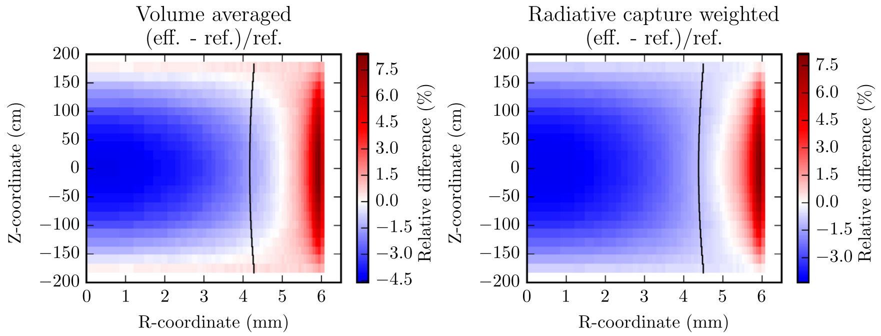


Fig. 13. Relative differences in the ^{238}U radiative capture reaction rates for rod 9 between the effective temperature and realistic temperature profiles **Left:** volume averaged temperature. **Right:** ^{238}U radiative capture weighted temperature. Solid black line indicates the line of equal temperature between the effective and realistic models.

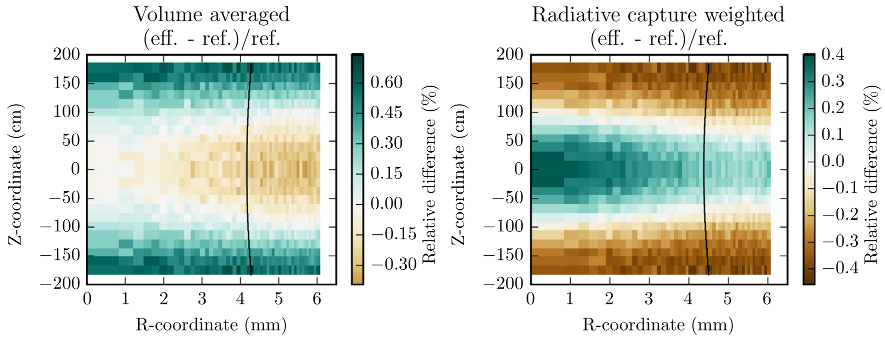


Fig. 14. Relative differences in the ^{235}U fission reaction rates for rod 9 between the effective temperature and realistic temperature profiles **Left:** volume averaged temperature. **Right:** ^{238}U radiative capture weighted temperature. Solid black line indicates the line of equal temperature between the effective and realistic models.

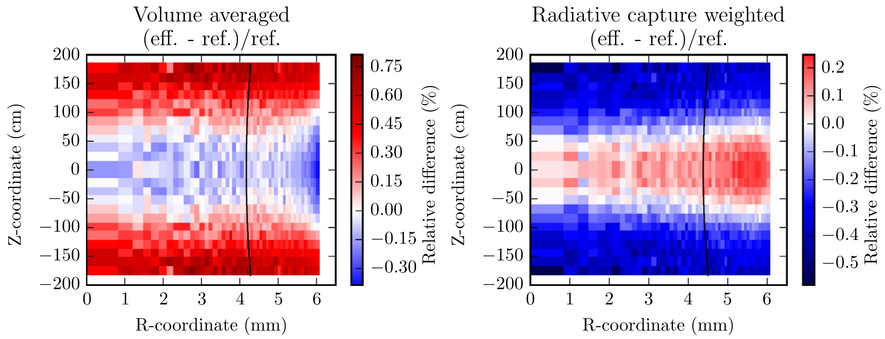


Fig. 15. Relative differences in the gadolinium radiative capture reaction rates for rod 14 between the effective temperature and realistic temperature profiles **Left:** volume averaged temperature. **Right:** ^{238}U radiative capture weighted temperature. Solid black line indicates the line of equal temperature between the effective and realistic models.

The uniform fission site (UFS) method implemented in Serpent is very similar to that developed for the MC21 code Kelly et al., 2012. It can be applied to distribute the fission source more evenly throughout the problem geometry to enhance the statistics in low fission rate regions with the expense of statistics in high fission rate regions. The method works by superimposing a regular cartesian mesh on the problem geometry and tallying the flux, the collision rate or the fission rate in each of the mesh cells during the inactive cycles of the simulation. The number of fission neutrons emitted in interactions during the active cycles are then adjusted based on the local UFS tally value. The statistical weights of the fission neutrons are also adjusted to conserve the total fission neutron weight emitted from each cell.

Using the UFS method should yield better statistics in low-power regions and worse statistics in high-power regions compared to a simulation without the use of the UFS-method. This should bring the statistical uncertainties in the power tallies to a more uniform level throughout the geometry, especially decreasing the maximum statistical uncertainty of the individual power tally bins. The effect on the convergence of the coupled solution, however, is not straightforward. While the method should decrease the statistical uncertainty in the power tally bins with the lowest power, these low-power areas, by definition, are not very interesting from the fission power – fuel temperature coupling point of view. Instead the areas with the highest fission power should experience the strongest feedback from fuel temperature and thus be more important from the coupling point of view.

To obtain a quantitative estimate of the effects of the UFS method on the solution convergence, 100 independent simulations were ran with or without UFS. Each simulation consisted of 20 coupled calculation iterations with 20 million active neutron histories in each neutron transport solutions. In the simulations with UFS, the source biasing was based on the fission rates tallied on a cartesian mesh that divided the system axially into 24 bins (the number of axial zones in the calculation) during the inactive cycles of each iteration. There was only a single horizontal mesh bin in each of the axial layers, which means that the UFS method should mostly even out the difference in statistics between the axial midplane and the assembly ends.

Based on the 100 independent simulations, mean values and standard deviations of the mean were calculated for the local convergence criterion (Eq. (8)) as well as the ℓ^2-norm based convergence criterion (Eq. (9)) as a function of the iteration number. These results are presented in Fig. 16 and Fig. 17. The convergence criteria were calculated for the temperature distribution as well as the relaxed and unrelaxed power distributions.

The results show that the convergence criteria calculated for the unrelaxed power distribution reach a stable level in three iterations and only undergo slight variations after this. To understand this behavior, we'll note that the differences between subsequent solutions for the power distribution stem from two sources: First, updating the fuel behavior solution means that the accurate solution of the neutron transport problem will be slightly different. Second, the power distribution obtained here by a neutron transport simulation is only a stochastic estimate of the accurate solution and will contain some statistical uncertainty. If the solution converges, the first part of the difference will get smaller with iterations but the second part will stay at a constant level. The behavior of the convergence criteria calculated for the unrelaxed power distribution thus tells us that, in this case, it takes approximately three iterations for the coupled solution to converge to a level, where the effect of further convergence on the unrelaxed power distribution is smaller than the effect of statistical uncertainty in each unrelaxed power solution.

If we weren't using the stochastic approximation based relaxation, there would be no great use in iterating the coupled calculation more than three times if we did not increase the amount of neutrons simulated on each iteration thus decreasing the statistical uncertainty in the tallied power distributions. Due to the applied relaxation to the power distribution, the convergence criteria calculated for the relaxed power distribution decrease in an orderly fashion, which is to be expected due to the fact that on iteration n of the simulation, the new fission power solution has a $1/n$ weight while the previous relaxed solution has a $(n-1)/n$ weight (see Eqs. (4) and (5)). This means that the difference between two subsequent relaxed power distributions should decrease in a $1/n$ fashion, which seems to be the case. The temperature distribution is calculated based on the relaxed fission power distribution and it is thus reasonable, that its iteration based convergence criteria follow a similar $1/n$ trend as those for the relaxed power distribution.

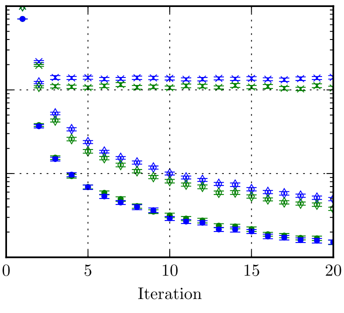


Fig. 16. Evolution of the convergence criterion based on the momentary (×) or relaxed (◊) power distributions and the temperature distribution (o). Blue: Reference, Green: With UFS. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

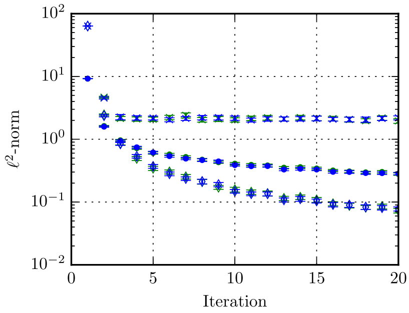


Fig. 17. Evolution of the ℓ^2-norm based convergence criterion based on the momentary (×) or relaxed (◊) power distributions and the temperature distribution (o). Blue: Reference, Green: With UFS. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The effects of UFS on the local convergence criterion can be judged from Fig. 16. Although using the UFS method leads to lower local convergence criteria for the power distribution, the effect on the temperature distribution is negligible. The evolution of the ℓ^2-norm based convergence criterion (Fig. 17) confirms that the UFS-method does not increase the solution convergence in this case in a significant manner. However, since the use of UFS leads to lower local convergence criteria for the power distribution and does not affect the ℓ^2-norm based convergence criteria adversely the use of UFS can be recommended. The use of UFS also leads to better statistics in low fission-rate areas, which should be taken in account, when choosing whether or not to use the method.

11. Known limitations and future work

The fuel behavior interface format has been written specifically to couple Serpent with codes that use the 1.5-dimensional geometry model, where each rod is divided into axial zones, with axially uncoupled radial nodalization in each zone. This means that the axial temperature distribution will always follow a step-profile if the fuel behavior interface is used. If interpolation of the temperature distribution is needed also in the axial direction, or if the coupled code does not use the 1.5-dimensional geometry model, one of the other multi-physics interface formats of Serpent may be better suited for the task. Separate formats exist for example for interpolation of point-wise data, for data based on a regular mesh and for unstructured mesh based data.

The fuel behavior interface only accounts for the radial thermal expansion of the fuel rods as the thermal strain is only brought into Serpent via the changes in the radial coordinates of the nodes. Changes in the fuel rod affecting the axial direction are not brought into the neutron transport simulation. This omission includes the decrease in local fuel density due to axial thermal expansion of the fuel pellets.

Since the modified local material densities are calculated based on the changes in the coordinates of the radial nodes (Eq. (2)) the closing of the gas gap due to thermal expansion of the fuel pellet as well as due to the swelling of the pellet with increasing burnup would increase the material density in the gas gap. This does not consider the fact that the gas density should be equal at different axial elevations and also consider the gas volumes in the plenum (s) of the rod. Moreover, increasing the material density above the density used to calculate the majorant cross sections for the neutron tracking will yield unphysical results. Currently the local density factor (Eq. (2)) is artificially limited to unity, i.e. the material densities cannot exceed their initial values. In the current simulations this mostly affects the density of the fill gas in the fuel rod, which should not have major effects on the neutronics solution. Two possible ways to tackle the larger than unity density factors would be to either recalculate the majorant cross sections at some point of the calculation or to initially calculate the majorant cross sections with elevated material densities.

Extending the methodology to assembly colorset level has been tested in 2D calculations, moving to larger and larger systems requires good parallel scalability from the code as more and more neutron histories have to be simulated to keep the level of statistical accuracy reasonable in an increasing number of tally regions. No scalability study has been made for the current Open-MPI parallelization at the moment. The current implementation of the coupling can utilize the standard hybrid OpenMP-MPI parallelization in Serpent 2 regarding the neutron transport part of the execution. The data transfer related to the multi-physics interfaces as well as the processing of the updated input fields is, however, done solely by a single MPI task. This is also the case with executing the FINIX solvers for the internal coupling, the execution is done only by MPI-task 0 and is furthermore not OpenMP parallelized. Moving into larger systems consisting of thousands or tens of thousands of fuel rods, this will create a bottleneck for the efficiency of the parallel execution of the code system. The obvious solution will be to use an approach similar to that, which Serpent uses in burnup calculations. In systems with hundreds or thousands of depletable materials the parallel efficiency is greatly enhanced by dividing the materials evenly between the MPI tasks and using an OpenMP parallel loop in each task to call the depletion solver for the individual materials. The same approach can be applied to the FINIX solver definitions. As the solvers for different fuel rods are independent, there will be no risks of race conditions. For the external coupling, the parallelization of the coupled solver execution has to be done by the user, but the parallelization of the file I/O can be improved in the future by associating each interface file with a certain MPI-task that will handle the file I/O as well as the processing of the fields.

One area of future development for the coupled calculation routines of Serpent should be the optimization of the coupled iteration, for example, by calculating the first neutronics solutions quickly with a small neutron population and increase the population size and the statistical accuracy on each iteration as suggested by Dufek and Gudowski (2006) to speed up the convergence of the coupled solution.

The capability for the code to provide accurate reference solutions for comparing different methods for calculating the effective fuel temperature should provide the means for further interesting studies on obtaining an optimal effective fuel temperature for conserving certain parameters of interest. The optimal effective fuel temperature is, of course, application dependent and in some cases, such as for the reproduction of ^{238}U local capture rates, no single effective temperature may perform very well. The possible advantages of using Monte Carlo based reaction rate weighted effective temperatures should also be further estimated.

Traditionally, Monte Carlo simulations have been made with a flat temperature profile inside the fuel rods. It will be interesting to estimate the effect of a realistic fuel temperature distribution on the traditional and novel applications of Monte Carlo calculation such as the effect on group constant generation and the associated burn up calculation as well as in coupled transient Monte Carlo simulations.

In parallel to studying the coupling between fuel behavior and neutronics, direct multi-physics modeling of assemblies and small reactor cores can be pursued. For that purpose, the fuel behavior solvers should be complemented, or even completely replaced, by thermal–hydraulics solvers to correctly model the full neutronics – fuel behavior – coolant behavior coupling.

12. Conclusions

The multi-physics capabilities of the Monte Carlo code Serpent have been extended by implementing an interface for bringing in fuel behavior solutions from coupled codes as well as by implementing new routines for coupled calculations that control the program flow, code-to-code communication as well as solution relaxation in Serpent.

The fuel behavior interface has been designed specifically for coupling Serpent with traditional fuel performance codes using the so-called 1.5-dimensional geometry representation, where the fuel rods are divided into very loosely coupled axial zones in which the radial heat transfer is solved separately for each zone. The interface can be used to bring in the radial temperature distribution, as well as the radial deformation of the fuel rod and cladding into Serpent. Serpent will then use the temperature data to accurately model the effects of local material temperatures on the interaction physics on-the-fly by using the TMS temperature treatment technique. The changes in the local material densities as well as the radial changes in fuel rod geometry can also directly be accounted for in the neutron transport simulation.

Additional subroutines now also allow Serpent to run coupled calculations without the need to restart Serpent for each iteration. Data can be exchanged between Serpent and other solvers either via direct memory access (internally coupled solvers) or by reading and writing files (externally coupled solvers) after each neutron transport simulation. Moreover, the fission source distribution can be passed between the iterations to decrease the number of inactive source cycles needed to ensure source convergence. Serpent automatically relaxes the fission power distribution using a stochastic approximation based relaxation scheme to enhance the convergence of the coupled solution.

The new iteration capabilities as well as the fuel behavior multi-physics interface were tested by calculating the coupled neutronics fuel-behavior solution for a full length BWR-assembly at power. The effect of coolant temperature distribution was disregarded in the test simulation. The coupled solution was obtained using both the internal coupling with the FINIX fuel behavior module and the external coupling with the ENIGMA fuel performance code.

The effect of the radial representation of the temperature distribution on local reaction rates as well as on some integral parameters was quantified by running neutronics simulations with either the detailed temperature distributions obtained from the coupled solution or with effective temperature fields. Two novel effective temperature models, namely, the local collision rate and local ^{238}U radiative capture rate weighted temperatures were described and tested. While all of the tested effective temperature fields reproduced the integral parameters well, large differences could be seen in the local radiative capture rates by ^{238}U. Local differences could also be observed in the local fission rates as well as the local radiative capture rate by the gadolinium isotopes in the burnable absorber rods. The effect of these local differences on the material compositions in burnup calculations should be studied separately.

Finally, the possibility to enhance the rate of convergence of the coupled solution by using the Uniform Fission Source (UFS) method was studied. While the use of UFS resulted in slightly lower values for the local convergence criterion calculated from the relaxed power distribution, the effect on the temperature field as well as on all of the ℓ^2-norm based convergence criteria were negligible. The use of UFS can be recommended in future calculations due to the better performance judged by the local convergence criterion as well as the fact that UFS leads to better statistics in any tallies in low-fission rate areas.

The new subroutines in Serpent that have been written specifically for coupled calculations should make it easier to conduct multi-physics analyses using Serpent as the neutronics tool in a more routine manner. These routines are not limited to the fuel behavior interface, but work with the various interface formats supported by Serpent and allow coupled calculations with a wide range of coupled solvers.

Acknowledgments

This work has been funded by the NUMPS project of the Academy of Finland as well as the KÄÄRME, PALAMA and MONSOON projects under the Finnish National Research Programme on Nuclear Power Plant Safety, SAFIR-2014 and SAFIR-2018. Some of the calculations presented above were performed using computer resources within the Aalto University School of Science "Science-IT" project.

References

Aufiero, M., Fiorina, C., Laureau, A., Rubiolo, P., Valtavirta, V., 2015. Serpent-OpenFOAM coupling in transient mode: simulation of a Godiva prompt critical burst. Proc. M&C + SNA + MC 2015, Nashville, TN, USA.

Daeubler, M., Ivanov, A., Sjenitzer, B.L., Sanchez, V., Stieglitz, R., Macian-Juan, R., 2015. High-fidelity coupled monte carlo neutron transport and thermal-hydraulic simulations using Serpent 2/SUBCHANFLOW. Ann. Nucl. Energy 83, 352–375. http://dx.doi.org/10.1016/j.anucene.2015.03.040. URL: http://www.sciencedirect.com/science/article/pii/S0306454915001747.

de Kruijf, W.J.M., Janssen, A.J., 1996. The effective fuel temperature to be used for calculating resonance absorption in a 238UO2 lump with a nonuniform temperature profile. Nucl. Sci. Eng. 123, 121–135. http://dx.doi.org/10.13182/NSE94-108. URL: http://www.ans.org/pubs/journals/nse/a24217.

Dufek, J., Gudowski, W., 2006. Stochastic approximation for Monte Carlo calculation of steady-state conditions in thermal reactors. Nucl. Sci. Eng. 152, 274–283. http://dx.doi.org/10.13182/NSE06-2. URL: http://www.ans.org/pubs/journals/nse/a2581.

Dufek, J., Hoogenboom, J.E., 2014. Description of a stable scheme for steady-state coupled Monte Carlo – thermal-hydraulic calculations. Ann. Nucl. Energy 68, 1–3. http://dx.doi.org/10.1016/j.anucene.2013.12.017. URL: http://www.sciencedirect.com/science/article/pii/S0306454913006683.

Goltsev, A., Davidenko, V., Tsibulsky, V., Lekomtsev, A., 2000. Computational problems in the calculation of temperature effects for heterogeneous nuclear reactor unit cells. Ann. Nucl. Energy 27 (2), 175–183. http://dx.doi.org/10.1016/S0306-4549(99)00067-5.

Ikonen, T., 2013. FINIX Fuel Behavior Model and Interface for Multiphysics Applications. Code Documentation for Version 0.13.9, VTT-R-06563, VTT Technical Research Centre of Finland, Available at: http://virtual.vtt.fi/virtual/montecarlo/download/VTT-R-06563-13.pdf.

Ikonen, T., Tulkki, V., Syrjälahti, E., Valtavirta, V., Leppänen, J., 2013. FINIX-fuel behavior model and interface for multiphysics applications. Proc. 2013 Fuel Performance Meeting TopFuel, Charlotte, USA.

Ikonen, T., Loukusa, H., Syrjälahti, E., Valtavirta, V., Leppänen, J., Tulkki, V., 2015. Module for thermomechanical modeling of LWR fuel in multiphysics simulations. Ann. Nucl. Energy 84, 111–121. http://dx.doi.org/10.1016/j.anucene.2014.11.004. URL: http://www.sciencedirect.com/science/article/pii/S0306454914005842.

Ikonen, T., Syrjalahti, E., Valtavirta, V., Loukusa, H., Leppänen, J., Tulkki, V., 2015. Multiphysics simulation of fast transients with the FINIX fuel behaviour module. Proc. TopFuel 2015, Zurich, Switzerland.

Ivanov, K. et al., 2013. Benchmark for uncertainty analysis in modelling (UAM) for the design, operation and safety analysis of LWRs, volume I: Specification and support data for neutronics cases (phase I), OECD/NEA.

Ivanov, A., Sanchez, V., Stieglitz, R., Ivanov, K., 2014. Internal multi-scale multi-physics coupled system for high fidelity simulation of light water reactors. Ann. Nucl. Energy 66, 104–112. http://dx.doi.org/10.1016/j.anucene.2013.12.003. URL: http://www.sciencedirect.com/science/article/pii/S0306454913006476.

Kelly, D.J., Sutton, T.M., Wilson, S.C., 2012. MC21 analysis of the Nuclear Energy Agency Monte Carlo performance benchmark problem. Proc. Physor 2012, Knoxville, TN, USA.

Kilgour, W.J., Turnbull, J.A., White, R.J., Bull, A.J., Jackson, P.A., Palmer, I.D., 1991. Capabilities and validation of the ENIGMA fuel performance code. Proc. ANS/ENS Int. Topical Meeting on LWR Fuel Performance, Avignon, France.

Knebel, M., Mercatali, L., Sanchez, V., Stieglitz, R., Macian-Juan, R., 2016. Validation of the Serpent 2-DYNSUB code sequence using the Special Power Excursion Reactor Test III (SPERT III). Ann. Nucl. Energy 91, 79–91. http://dx.doi.org/10.1016/j.anucene.2016.01.005. URL: http://www.sciencedirect.com/science/article/pii/S030645491630007X.

Leppänen, J., 2013. Modeling of non-uniform density distributions in the Serpent2 monte carlo code. Nucl. Sci. Eng. 174, 318. http://dx.doi.org/10.13182/NSE12-54. URL: http://www.ans.org/pubs/journals/nse/a17107.

Leppänen, J. et al., 2014. Unstructured mesh based multi-physics interface for CFD code coupling in the Serpent2 Monte Carlo code. Proc. Physor 2014, Kyoto, Japan.

Leppänen, J., Mattila, R., 2016. Validation of the serpent-ares code sequence using the MIT BEAVRS benchmark – HFP conditions and fuel cycle 1 simulations. Ann. Nucl. Energy 96, 324–331. http://dx.doi.org/10.1016/j.anucene.2016.06.014. URL: http://www.sciencedirect.com/science/article/pii/S0306454916304121.

Leppänen, J., Viitanen, T., Valtavirta, V., 2012. Multi-physics coupling scheme in the Serpent 2 Monte Carlo code. Trans. Am. Nucl. Soc. 107, 1165.

Leppänen, J., Hovi, V., Ikonen, T., Kurki, J., Pusa, M., Valtavirta, V., Viitanen, T., 2015. The numerical multi-physics project (NUMPS) at VTT technical research centre of Finland. Ann. Nucl. Energy 84, 55–62. http://dx.doi.org/10.1016/j.anucene.2014.10.014. URL: http://www.sciencedirect.com/science/article/pii/S0306454914005532.

Leppänen, J., Pusa, M., Viitanen, T., Valtavirta, V., Kaltiaisenaho, T., 2015. The serpent Monte Carlo code: status, development and applications in 2013. Ann. Nucl. Energy 82, 142–150. http://dx.doi.org/10.1016/j.anucene.2014.08.024. URL: http://www.sciencedirect.com/science/article/pii/S0306454914004095.

Rossiter, G., 2011. Development of the ENIGMA fuel performance code for whole core analysis and dry storage assessments. Nucl. Eng. Technol. 43, 489.

Rowlands, G., 1962. Resonance absorption and non-uniform temperature distributions. J. Nucl. Energy Parts A/B. Reactor Sci. Technol. 16 (4), 235–236. http://dx.doi.org/10.1016/0368-3230(62)90294-X.

Salino, V., Hébert, A., 2016. On steady-state multiphysics stability and related in-core fuel management capabilities in DONJON5. PHYSOR 2016, Sun Valley, ID, USA.

Tuominen, R., Valtavirta, V., Peltonen, J., Leppänen, J., 2016. Coupling Serpent and OpenFOAM for neutronics – CFD multi-physics calculations. Proc. Physor 2016, Sun Valley, ID, USA.

Valtavirta, V., 2015. Uncertainty underprediction in coupled time-dependent Monte Carlo simulations with Serpent2. Proc. ANS MC2015, Nashville, TN.

Valtavirta, V., Tulkki, V., Leppänen, J., Viitanen, T., 2013. The universal fuel performance code interface in Serpent 2. Proc. 2013 Fuel Performance Meeting TopFuel, Charlotte, USA.

Valtavirta, V., Ikonen, T., Viitanen, T., Leppänen, J., 2014a. Simulating fast transients with fuel behavior feedback using the Serpent2 Monte Carlo code. Physor 2014, Kyoto, Japan.

Valtavirta, V., Viitanen, T., Leppänen, J., 2014b. Internal neutronics temperature coupling in Serpent2. Nucl. Sci. Eng. 177, 193–202. http://dx.doi.org/10.13182/NSE13-3. URL: http://www.ans.org/pubs/journals/nse/a35681.

Viitanen, T., 2015. Development of a Stochastic Temperature Treatment Technique for Monte Carlo Neutron Tracking. Aalto University (Ph.D. thesis).

Viitanen, T., Leppänen, J., 2012. Explicit treatment of thermal motion in continuous-energy Monte Carlo tracking routines. Nucl. Sci. Eng. 171, 165–173. http://dx.doi.org/10.13182/NSE11-36. URL: http://www.ans.org/pubs/journals/nse/a13604.

Viitanen, T., Leppänen, J., 2014. Target motion sampling temperature treatment technique with elevated basis cross section temperatures. Nucl. Sci. Eng. 177, 77–89. http://dx.doi.org/10.13182/NSE13-37. URL: http://www.ans.org/pubs/journals/nse/a35637.

Viitanen, T., Leppänen, J., 2015. Temperature majorant cross sections in Monte Carlo neutron tracking. Nucl. Sci. Eng. 180, 209–223. http://dx.doi.org/10.13182/NSE14-46. URL: http://www.ans.org/pubs/journals/nse/a37098.

Viitanen, T., Tulkki, V., 2012. Combining reactor physics and fuel performance calculations. Proc. TopFuel 2012, Manchester, UK.

Publication VI

V. Valtavirta and J. Leppänen. Estimating the effects of homogenized fuel temperature in group constant generation using Serpent 2. *Annals of Nuclear Energy*, 105, pp. 79–94, http://dx.doi.org/10.1016/j.anucene.2017.03.007, 2017.

Annals of Nuclear Energy 105 (2017) 79–94

Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

Estimating the effects of homogenized fuel temperature in group constant generation using Serpent 2

Ville Valtavirta [*], Jaakko Leppänen

VTT Technical Research Centre of Finland Ltd., P.O. Box 1000, FI-02044 VTT, Finland

ARTICLE INFO

Article history:
Received 8 November 2016
Received in revised form 27 February 2017
Accepted 6 March 2017

Keywords:
Monte Carlo
Multi-physics
Burnup
Fuel temperature
Group constant

ABSTRACT

We extend the multi-physics capabilities of the Serpent 2 Monte Carlo code to coupled burnup calculations by implementing the Stochastic Implicit Euler depletion scheme with thermal feedback. We use these new multi-physics capabilities for the verification of the traditional way of generating group constants using an effective flat fuel temperature profile during the burnup calculation. We investigate the effects of this approximation on the generated nuclide compositions, group constants as well as the results on the simulation of the initial cycle of the EPR reactor using the ARES core simulator.

The main findings state that while the use of an effective temperature model leads to significant differences in the radial nuclide distributions, the assembly wide homogenized group constants are reproduced fairly well and the effects on the simulation of the EPR initial cycle are modest, although interesting axial and radial power redistribution can be observed due to the slower speed of gadolinium burnout when effective fuel temperatures were used. The results indicate that better results for the full core calculations could be obtained by using a separate effective temperature for the burnable absorber rods in the burnup calculation and by considering the fuel temperature history effect in the group constant parametrization.

1. Introduction

In the recent years, the application of Monte Carlo (MC) neutron transport codes to multi-physics problems has been an increasingly popular choice as the development of computational resources has made it feasible. At the same time, multi-physics applications have become a standard part of the development of various MC codes such as MC21 (Griesheimer et al., 2013; Gill et al., 2014, 2015), OpenMC (Herman et al., 2015; Ellis et al., 2015), MONK (Richards et al., 2013) and Serpent (Leppänen et al., 2012, 2015a). Coupling continuous energy MC methods with state-of-the-art fuel performance, thermal hydraulics or CFD-solvers is attractive as it makes it possible to obtain truly high-fidelity coupled solutions to reactor modeling problems. While the computational cost of these solutions usually prohibits their use in everyday full core modeling, they are well suited for verifying the methodology of computationally cheaper methods that are utilized routinely in reactor analysis. Here we apply the new multi-physics capabilities of Serpent 2 in a similar manner.

In this paper, we describe the new coupled burnup capabilities of Serpent 2 and use them to estimate the effects of simplifications concerning the fuel temperature distribution that are usually made in the burnup calculations associated with group constant generation, spatial homogenization being one of the main applications of Serpent (Leppänen et al., 2016, 2014; Leppänen and Mattila, 2016): We couple Serpent 2 with the fuel performance code ENIGMA (Kilgour et al., 1991) for the 2D, infinite lattice, burnup calculations that make up the history part of the group constant generation calculations. We compare the results of the accurate coupled calculation to the results of a group constant generation calculation, where the burnup calculation uses the traditional flat fuel temperature profile approximation. This analysis combines the two major development areas of Serpent 2, namely spatial homogenization (Leppänen et al., 2016) and multi-physics applications (Leppänen et al., 2012, 2015a).

The coupled depletion study continues the analysis of the effects of using effective fuel temperatures that we started in Valtavirta et al. (2017) for a steady state system with no depletion. The previous study showed that using an effective fuel temperature instead of a realistic radial fuel temperature profile significantly affects the radial reaction rate profiles especially regarding neutron capture by ^{238}U. The effective temperature models overestimate the (n, γ)

[*] Corresponding author.
E-mail address: ville.valtavirta@vtt.fi (V. Valtavirta).

http://dx.doi.org/10.1016/j.anucene.2017.03.007
0306-4549/© 2017 Elsevier Ltd. All rights reserved.

reaction rate of ^{238}U near the surface of the pellet, which may lead to an overestimation of the rim-effect in the burnup calculation.

We will first present background information regarding the computational tools, the flat fuel temperature profile approximation in assembly burnup calculations and some recent developments regarding MC burnup calculations with thermal feedback. Second, we will describe the implementation of the SIE algorithm for coupled MC neutronics – fuel behavior burnup calculations in Serpent 2. For the application of the coupled burnup methodology we will generate group constants for the reactor simulator code ARES for the simulation of the EPR initial cycle as described by (Sengler et al., 1999). We will investigate the effects a realistic fuel temperature distribution will have on the nuclide compositions of depleted fuel, the group constants generated using those nuclide compositions and the results of the core simulator calculations using the generated group constants.

2. Computational tools

2.1. The Serpent Monte Carlo code

The Serpent[1] Monte Carlo code (Leppänen et al., 2015b) is a relatively young continuous energy Monte Carlo reactor physics burnup code with recent applications in radiation shielding, multi-physics and fusion neutronics. The code is currently used in more than 160 universities and research organizations for reactor physics applications ranging from homogenized group constant generation to burnup calculations and the modeling of small research reactor cores to multi-physics calculations with couplings to various thermal–hydraulics, CFD and solid-mechanics tools. Serpent has been developed at VTT Technical Research Centre of Finland since 2004 and the current development version, Serpent 2, has notably diversified the applications of the code. The two main areas of development for the new code version have been spatial homogenization and multi-physics applications, both of which are shortly described as this study lies in the boundary between the two.

2.1.1. Spatial Homogenization with Serpent 2

Serpent has the capability to produce homogenized group constants used by core simulator and transient analysis codes based on nodal diffusion methods. This includes homogenized reaction cross sections, scattering matrices, diffusion coefficients, discontinuity factors, time constants and production and absorption cross sections for fission product poisons ^{135}Xe and ^{149}Sm and their precursors. The calculation routines are based on standard Monte Carlo cell flux and surface current tallies, and two deterministic solvers used to obtain B_1 leakage-corrected cross sections and discontinuity factors for geometries homogenized without reflective boundary conditions. Full description of the methodology is found in Leppänen et al. (2016), and not repeated here.

2.1.2. Multi-Physics Capabilities of Serpent 2

Serpent 2 has been designed for multi-physics applications. During the development of the code in the 2010's, several multi-physics specific features have been designed and added to the code. These include.

- On-the-fly treatment of temperature effects on cross sections using the TMS treatment (Viitanen, 2015).
- On-the-fly temperature interpolation of $S(\alpha,\beta)$[2] data (Viitanen and Leppänen, 2016).
- On-the-fly treatment of non-uniform material density distributions (Leppänen, 2013).
- Universal multi-physics interface for separation of temperature and density data from geometry definition and for easy I/O of relevant fields (Leppänen et al., 2012).
- File based and POSIX based communication for external coupling of state-of-the-art solvers without the need to restart Serpent for each iteration (Valtavirta et al., 2017).
- Internal, separately licensed, fuel behavior module FINIX (Ikonen et al., 2015) for multi-physics applications.
- STL (Leppänen, 2015) and unstructured mesh (Leppänen and Aufiero, 2014) based geometry models for importing complex geometries.
- Dynamic simulation mode for uncoupled and coupled transient applications (Valtavirta et al., 2016).

These capabilities have been used for various applications over the years, but the modeling of the coupling between neutronics and fuel temperature has been one of the main applications at VTT (Valtavirta et al., 2014b, 2013, 2014a).

We will not discuss these features in detail as they are well described in their related articles, but we will summarize that Serpent 2 is capable of modeling materials with arbitrarily refined or even continuous temperature and density distributions in steady-state, burnup and time dependent simulations. Furthermore, Serpent is able to exchange data with external solvers during the coupled calculation Picard-iteration using any of the various multi-physics interface file formats.

The interface format used in this study to exchange data between Serpent and the fuel performance code ENIGMA is specifically designed for data exchange between Serpent and fuel performance codes using the so-called 1.5 dimensional geometry representation, i.e. axisymmetric (r, z) coordinate system with the fuel rods consisting of loosely coupled axial zones with a separate radial fuel behavior solution in each axial zone. Using this interface format Serpent will read in radial temperature and strain distributions at various axial layers of different fuel rods and output the fission power distribution tallied with a user specified axial and radial binning (Valtavirta et al., 2017, 2013).

2.2. ARES

Group constants generated by Serpent were used with the ARES core simulator (Mattila, 2003). ARES is a 3D nodal diffusion code developed at the Finnish Radiation and Nuclear Safety Authority (STUK), and based on the two-group analytical function expansion nodal model (AFEN). ARES can be used to perform fuel cycle simulations for square-lattice BWR and PWR cores, and the capabilities relevant for this study include critical boron search and estimation of various feedback coefficients.

The parametrization of group constants in ARES is based on the combination of tabular interpolation and polynomial expansions (Mattila, 2002). Tabulated data is used for burnup points, control rod insertion branches and moderator temperature histories. Variation in fuel and coolant temperature, coolant density and boron concentration, as well as boron history effects are accounted for by using a second order polynomial consisting of 23 terms. These terms cover the independent variations, but also several cross-terms. The use of Serpent for generating the full set of cross sections for ARES simulations was demonstrated in Leppänen and Mattila (2016), where the procedure is discussed in detail.

The group constant tabulation in ARES is divided into *branch variations* and *history variations*: The branch variations take into account momentary changes in the operating conditions that directly affect the interaction probabilities in the node. The history variations take into account conditions that persist for an extended

[1] For a complete description of the code and the latest news, see the project Web site: http://montecarlo.vtt.fi

[2] i.e. thermal scattering

period during the fuel life and affect the nuclide composition of the burnt up fuel. The history variations are handled by Serpent by running separate burnup calculations with the given history parameters to obtain the representative nuclide compositions at different burnup points. The branch variations can then be calculated by simply performing restarts to the given states at the given burnup points. Based on the group constants calculated by Serpent, the parametrization using a 23 term second order polynomial is created for the group constants, which will be used by ARES to evaluate the group constants at the local conditions of each node. The history and branch variations used for generating the group constants in this study are detailed in Section 5.5.

2.3. ENIGMA

The external solver coupled with Serpent for this study is the fuel performance code ENIGMA for steady-state and transient conditions, developed by Nuclear Electric and BNFL in the United Kingdom Kilgour et al., 1991. While the version of the code used in this study is based on ENIGMA v.5.9b, it has been modified at VTT over the years, mostly for model extension and additions for purposes concerning for example cladding material properties, fission gas release and gadolinia-doped fuel rods. The most relevant extension to this work makes it possible to read in the radial power distribution from a separate file in order to bring in the fission power profile from an external neutronics solver. The extension was initially written for modeling the radial power distribution in burnable absorber rod accurately during the absorber burn up, but the routines have also been utilized in previous multiphysics calculations with Serpent 1 and ENIGMA (Viitanen and Tulkki, 2012; Valtavirta et al., 2013, 2017).

The geometry model of ENIGMA follows the 1.5-dimensional model, where the fuel rod is divided into a number of axial zones, in which the radial heat equation is solved alongside with rod mechanics. The axial coupling between the different zones comes from the coolant temperature boundary condition, rod internal pressure as well as fission gas transport in the gas gap. A good overview of the history, structure and applications of ENIGMA as well as the recent development of the UK National Nuclear Laboratory version can be found in Rossiter (2011).

3. Background

3.1. Simplifications in the history part of group constant generation

As described in Section 2.2, group constant generation for ARES for a single history variation includes running a burnup calculation to obtain the correct nuclide compositions for each point in the burnup history that the group constants are generated for. Three simplifications are typically made, when the fuel temperature is chosen for this burnup calculation:

1. The radial temperature distribution in a fuel rod spanning some hundreds of Kelvins between pellet surface and centerline is collapsed into a single, effective, radially constant temperature (Fig. 1).
2. All of the different fuel rods in an assembly are set to use the same effective temperature, thus suppressing the variation of temperatures between different lattice positions and fuel types (From gray lines to cyan (o) in Fig. 2). This is a significant simplification when some of the rods in the assembly contain burnable absorber.
3. The effective temperature is set to stay constant throughout the fuel life, regardless of the changes in the fuel temperatures as a function of burnup (From cyan (o) to red dashed line in Fig. 2).

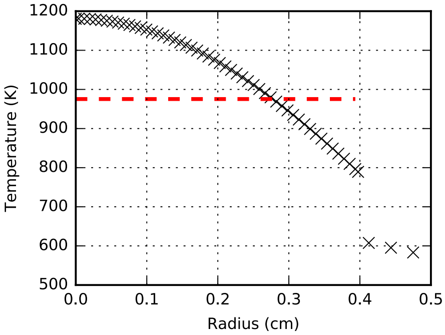


Fig. 1. Simplification by collapsing the radial fuel temperature distribution (black x) to a single pin-wise effective fuel temperature (red dashed line). (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

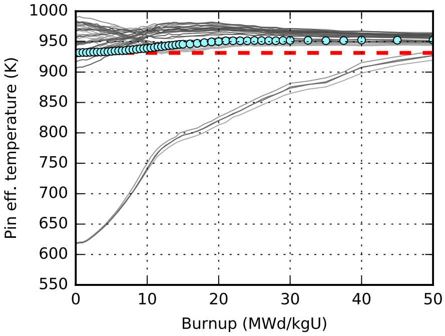


Fig. 2. Simplification by collapsing the pin-wise effective temperatures (gray lines) to an assembly wise effective temperature (cyan o). A further simplification is typically made by assuming this assembly wise effective temperature is constant throughout the burnup calculation (red dashed line). This assembly contains burnable absorber pins that have a low effective temperature at low burnups. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

Transforming the radial fuel temperature distribution into a single effective temperature has been studied previously in some depth. While an effective temperature that conserves some global properties of the fuel rod, such as the total resonance absorption rate in the rod, can be typically calculated based on the radial temperature distribution(Rowlands, 1962; Goltsev et al., 2000; Kruijf and Janssen, 1996), a single effective temperature will, in any case, lead to the overestimation of the temperature at the pellet surface and the underestimation of the temperature at the pellet inner parts. This leads to the overestimation of the width of the resonances in the nuclear cross sections for the regions close to the pellet surface, which leads to an increase in ^{238}U radiative capture and thus in ^{239}Pu production. In the inner parts of the pellet the effect is reversed.

The choice of a single effective temperature for the whole assembly overestimates the temperatures at burnable absorber rods, at least during the beginning of the fuel life. This will, again, lead to the increased breeding of fissile ^{239}Pu in the burnable absorber rods. Keeping the assembly fuel temperature constant throughout the fuel life is less of an approximation if the power level is assumed to be constant, which is a major approximation in itself. While the radial fuel temperature distribution changes during the fuel life due to the decrease in the fuel thermal conductivity associated with increasing burnup and the slow closing of the gas gap between the pellet and the cladding, there are only small changes in most of the rod effective temperatures (see Fig. 2) during the irradiation. The burnable absorber rods are naturally an exception, their effective fuel temperature changing by hundreds of Kelvins during irradiation.

We can now estimate the effects of these simplifications by modeling the accurate fuel temperatures explicitly with Serpent. This can be achieved by using the TMS temperature treatment technique to deal with the complex fuel temperature distributions and the fuel behavior multi-physics interface of Serpent to bring in the realistic fuel temperature profile for each rod from a fuel performance code. One additional implementation is required, however, namely a burnup scheme that can handle feedback, not only from neutronics, but also from the changing fuel temperature.

3.2. Coupled burnup schemes for Monte Carlo simulations

Inclusion of the thermal feedback to the burnup calculation should bring about some changes to the burnup calculation scheme itself. One could naturally use the existing explicit and predictor–corrector burnup schemes present in Serpent, simply apply a Picard-iteration to each transport solution until a converged neutron transport – thermal–hydraulics solution has been obtained and then use that neutron transport solution to solve the Bateman equations. The stability and convergence of the traditional explicit or predictor–corrector burnup schemes might prove to be an issue as the algortihms are known to be subject to instabilities in certain situations (Dufek and Hoogenboom, 2009; Dufek et al., 2013b; Kotlyar and Shwageraus, 2013).

More advanced methods for coupled burnup calculations have been developed in the recent years. The Stochastic Implicit Euler (SIE) method was proposed by Dufek et al. as a solution to the numerical instabilities of the normal explicit Monte Carlo burnup algorithms (Dufek et al., 2013a). An extension of the SIE algorithm using thermal–hydraulic feedback was described shortly thereafter (Dufek and Anglart, 2013). The SIE method relies on obtaining the depletion solution for each depletion step using the implicit Euler method, i.e. by using the end-of-step (EOS) reaction rates to deplete the nuclide composition throughout the step. The Stochastic part of the algorithm comes from the fact that each burnup step can be iterated multiple times by re-calculating the EOS neutron transport solution after the depletion and re-depleting the system with EOS reaction rates relaxed over all iterations of the burnup step using the stochastic approximation (Robbins-Monro) based relaxation.

Kotlyar and Shwageraus (2014) improved on the use of EOS reaction rates by formulating two different approaches using middle-of-step (MOS) reaction rates in combination with the Robbins-Monro algorithm. Both algorithms, referred to as stochastic implicit mid-point (SIMP) methods, solved the coupled neutronics – depletion – thermal hydraulics problem. One of the algorithms iterated on the nuclide densities while the other iterated on the neutron flux solution.

Recent studies by Kotlyar and Shwageraus (2016a) have indicated that the accuracy of the SIE method can be quite poor in cases with rapid spectral changes resulting in significant errors in the prediction of certain nuclide concentrations. Improved methods for coupled burnup calculations have been suggested in the form of the Stochastic Semi-Implicit Substep (SUBSTEP) method and its extended version by Kotlyar and Shwageraus (2016a,b). The SUBSTEP method divides each burnup step into multiple sub-steps to allow for variation of nuclide densities and reaction rates during the depletion solution. The reaction rates for each sub-step are calculated without further neutron transport solutions by establishing a functional relation between the reaction rates and the nuclide densities alongside with the coupled fields such as coolant or fuel temperature and density based on neutron transport solutions at beginning-of-step and end-of-step.

As the SIE method without thermal feedback (as described in Dufek et al. (2013a)) has been implemented in Serpent previously (Dufek and Valtavirta, 2014), we'll apply it in this study despite its obvious pitfalls. For this study we will be comparing the results of two burnup calculations both using the SIE method, which means that we should be able to discern the effect of the temperature simplification based on the differences between the two calculations. As the existing implementation did not contain the thermal feedback, it had to be updated according to Dufek and Anglart (2013) to account for the thermal feedback.

4. Implementation of the coupled burnup scheme

The Stochastic Implicit Euler scheme with thermal–hydraulic feedback was implemented in Serpent as described by Dufek and Anglart (2013). The original description included the (thermal–hydraulic) solution for the coolant nuclide field $\mathbf{N}_C$ as the solution coupled to the neutron transport solution. In this study we held the coolant conditions constant and replaced the feedback with the solution for the fuel behavior $\mathbf{T}_F$, which includes the temperature and thermal expansion distributions in each fuel rod. The coupled burnup scheme is described in Table 1 in a notation as similar to the original description as possible.

The burnup scheme starts from the initial nuclide density and fuel behavior solution for the steady state at zero burnup (step 1). The initial estimate for the transmutation cross sections and the neutron flux is obtained with a transport solution of the zero-burnup system (step 2). The first estimate for the nuclide concentrations at each burnup point is obtained using the explicit Euler method (step 4). The further iterations will always use the implicit Euler method (step 9). The first estimate for the fuel behavior solution at each burnup point is also obtained using the power distribution from the previous burnup point (step 5). After the initial estimate for the coupled solution at the burnup point has been obtained, it can be updated using several iterations of the SIE iteration loop (steps 7–10). On each iteration a new transport solution for the end-of-step system is obtained (step 7), after which the neutron flux, the power distribution and the transmutation cross sections are relaxed using the stochastic approximation based solution relaxation (step 8), which in this work amounted to simple averaging of each distribution over the iterations. After the new relaxed distributions have been obtained for the current burnup point, the depletion solution and the fuel behavior solution are updated (steps 9 and 10). For step 10, the relaxed power distribution is passed to the fuel behavior solver.

When a specified number of iterations has been simulated, the nuclide concentrations, the fuel behavior solution and the neutron transport solution will be considered to be the representative solution for the current burnup point (steps 12–14) and the algorithm will move to the next burnup point by using the explicit Euler method to provide a prediction for the next depletion solution (step 4).

5. Group constant generation for the EPR initial core

The coupled burnup methodology was used to generate group constants for the ARES core simulator for the initial cycle of the EPR initial core. To estimate the effects that the simplified fuel temperature distribution have on the group constant generation, we generated the group constants for the fuel assemblies in two ways. The first way (reference model) used the coupled

Table 1
The Stochastic Implicit Euler method with relaxation of the neutron flux and fuel behavior feedback. The thermal–hydraulic solution for the coolant $\mathbf{N}_\mathrm{C}$ in place in Dufek and Anglart (2013) has been replaced with the fuel behavior solution $\mathbf{T}_\mathrm{F}$ obtained using the fuel behavior solver based on a solved neutron flux/ power distribution $\mathbf{F}(\phi_i)$.

```
input: $\mathbf{N}_{\mathrm{F},0}, \mathbf{T}_{\mathrm{F},0}$
$\phi_0 \leftarrow \phi_B(\mathbf{N}_{\mathrm{F},0}, \mathbf{T}_{\mathrm{F},0})$
for $i \leftarrow 0, 1, \ldots$ do
  $\mathbf{N}^{(0)}_{\mathrm{F},i+1} \leftarrow \exp[\mathbb{M}(\phi_i)\Delta t]\mathbf{N}_{\mathrm{F},i}$
  $\mathbf{T}^{(0)}_{\mathrm{F},i+1} \leftarrow \mathbf{F}(\phi_i)$
  for $n \leftarrow 1, 2, \ldots, c$ do
    $\phi^{(n)}_{i+1} \leftarrow \phi_B(\mathbf{N}^{(n-1)}_{\mathrm{F},i+1}, \mathbf{T}^{(n-1)}_{\mathrm{F},i+1})$
    $\overline{\phi}^{(n)}_{i+1} \leftarrow \sum_{j=1}^{n} \phi^{(j)}_{i+1}/n$
    $\mathbf{N}^{(n)}_{\mathrm{F},i+1} \leftarrow \exp[\mathbb{M}(\overline{\phi}^{(n)}_{i+1})\Delta t]\mathbf{N}_{\mathrm{F},i}$
    $\mathbf{T}^{(n)}_{\mathrm{F},i+1} \leftarrow \mathbf{F}(\overline{\phi}^{(n)}_{i+1})$
  end for
  $\mathbf{N}_{\mathrm{F},i+1} \leftarrow \mathbf{N}^{(c)}_{\mathrm{F},i+1}$
  $\mathbf{T}_{\mathrm{F},i+1} \leftarrow \mathbf{T}^{(c)}_{\mathrm{F},i+1}$
  $\phi_{i+1} \leftarrow \overline{\phi}^{(c)}_{i+1}$
end for
```

methodology to execute the group constant generation for each history case in three parts:

1. Solve an initial guess for the temperature distribution in the assembly at zero burnup by running a coupled Serpent–ENIGMA simulation in steady state. The initial fuel behavior solution was calculated using 10 coupled calculation iterations.
2. Run the coupled burnup calculation using the detailed radial temperature fields provided by ENIGMA for each fuel rod based on the detailed power distribution tallied by Serpent. Each burnup step used 16 iterations of the SIE algorithm.
3. Using the nuclide compositions from the burnup calculation, generate the group constants using flat assembly-wide fuel temperatures based on the ARES group constant parametrization.

The uncoupled calculation sequence (effective model) was very similar:

1. Run the coupled burnup calculation using a flat fuel temperature profile of 900 K. Each burnup step used 16 iterations of the SIE algorithm.
2. Using the nuclide compositions from the burnup calculation, generate the group constants using flat assembly-wide fuel temperatures based on the ARES group constant parametrization.

There were only three variations in the coupled burnup calculation compared to the uncoupled one:

- Optimization mode 1 was used for Serpent instead of optimization mode 4. This means that the microscopic cross sections were not reconstructed on a unionized energy grid and no pre-calculated macroscopic cross sections were used for the different materials in the simulation.
- The temperature dependence of the cross sections for fuel and cladding was handled on-the-fly with TMS instead of using pre-processed cross section data.
- The temperature distributions for fuel and cladding were provided by the ENIGMA fuel performance code instead of using flat temperature distributions.

No differences are expected to be seen because of the first two variations as the choice of the optimization mode in Serpent does not affect the physics of the solution and is mainly used to modify the trade-off between memory consumption and calculation time (see next paragraph) and the TMS temperature treatment is known to give reaction rate estimates equivalent with the use of NJOY pregenerated cross sections within statistical accuracy (Viitanen, 2015).

The higher optimization modes of Serpent (3 and 4) include the pre-calculation of material wise macroscopic cross section on the unionized energy grid, which greatly speeds up the cross section lookup. When the materials no longer have a specific temperature associated with them, as is the case with the realistic fuel temperature distributions, the material-wise macroscopic cross section also becomes a distributed quantity and cannot be pre-calculated. Optimization mode 2 can be used with multi-physics calculations to speed up the simulation as it involves reconstructing the microscopic cross sections of each nuclide on a single unionized energy grid. However this increases the memory consumption which can become a factor especially in calculations with many transport nuclides as was the case here. In order to utilize also the computer nodes with only 16 GB of RAM, optimization mode 1 was used for the multi-physics calculation.

5.1. The EPR initial core

The EPR is a Gen III pressurized water reactor with a large core consisting of 241 fuel assemblies of the 17x17 type with an active fuel height of 420 cm. Four units are currently being built: Single units to Olkiluoto, Finland and Flamanville, France as well as two units to Taishan, China. The reason for choosing the EPR initial core for this study is threefold: First, investigating the effect of the fuel temperature simplifications will be especially interesting if the core contains fuel with co-mixed burnable absorber such as the Gd-rods in the EPR core. Second, as a Finnish research organization we are naturally interested in the EPR reactor due to the unit being constructed in Finland. Last, a detailed description of the EPR initial core has been published by Sengler et al. (1999) and while the core design has no doubt evolved since 1999, the description is sufficient for this study and publicly available for future comparisons.

The EPR initial core consists of four fuel assembly types, two of which contain fuel rods with co-mixed burnable absorber (Gd_2O_3). The assembly types will be addressed in this paper as R00, Y20, B00 and B12. The letter corresponds to the fuel enrichment of the non-absorber rods as well as the color used for plotting the assemblies in the core maps (red, yellow, blue). The number corresponds to the number of burnable absorber rods in the assembly. The BA configuration for the assemblies Y20 and B12 are shown in Fig. 3. All of the assemblies exhibited a 1/8-symmetry. The fuel enrichment and gadolinia content are listed in Table 2 for each fuel type. It bears mentioning that the fuel in burnable absorber rods actually uses depleted uranium (enrichment 0.25 wt.%). The fuel density was not given in the core description so it was set to 10.307 $\mathrm{g\,cm^{-3}}$ for all fuel types – a simplification that should be acceptable since we are only interested in the differences between the two calculation schemes.

The core is radially surrounded by a heavy reflector, which improves the neutron economy of the reactor. The exact composition or geometry of the heavy reflector are not included in the core description, although it is indicated that the reflector is made of stainless steel and contains water channels for its cooling. During full power operation, the reactivity of the core is controlled by adjusting the boron concentration in the coolant. In this study, the control rods were assumed to be always out of the core, which meant that no group constants needed to be generated for the control rod insertion branches. This is a reasonable approximation based on the published core design, which states that "…[in the

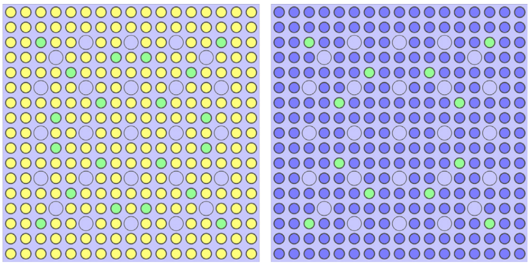


Fig. 3. Burnable absorber configurations in the Y20 (**left**) and the B12 (**right**) assembly types.

Table 2
The uranium enrichment and the gadolinia content for the different fuel types in the EPR initial core.

Fuel type	UO_2 Enrichment wt.%	Gd_2O_3 content wt.%
Red	1.9	–
Yellow	3.0	–
Blue	3.6	–
BA-rod	0.25	8.0

EPR the] control rods are nearly totally withdrawn at full power.” (Sengler et al., 1999) (p.86).

5.2. ARES core model

Due to the lack of details regarding, e.g. the radial reflector and the structures below and above the active core some features of the full core model had to be modeled based on various assumptions and simplifications. The main assumptions concerned the exact geometry and composition of the radial heavy reflector, as well as the areas beyond the heavy reflector. The spacer grids made of Zircaloy were also omitted from the neutronics model in both Serpent and ARES calculations. The omission of the spacer grids should not affect the comparison as the omission was made for both calculations sequences.

The heavy reflector was modeled as a combination of moderator and type 304 stainless steel with mass fractions of 0.1 for moderator and 0.9 for steel. In Sengler et al. (1999) the composition of the heavy reflector is given simply as “water plus steel”. The geometry of the heavy reflector was chosen to form a single assembly width ring outside the outermost fuel assemblies (see Fig. 5). The single assembly width makes the heavy reflector easy to model in the nodal code while being relatively close to the average thickness of the heavy reflector given in Sengler et al. (1999), the single assembly width being 21.504 cm compared to the average thickness of reflector: 19.4 cm.

The regions on top of and below the active fuel were loosely adapted from the BEAVRS benchmark (Horelik et al., 2013): On top of the active fuel, a 21 cm plenum region was added with a 0.0646 cm radius inconel “screw” contained in an otherwise empty cladding. Below the active fuel, a 1 cm high Zircaloy bottom plug (radius equal to cladding outer radius) was added followed by a 20 cm “bottom plate” consisting of type 304 stainless steel pins (radius equal to cladding outer radius) in coolant.

The radial core map used with ARES can be seen in Fig. 4. It should be noted that this core map has a 1/8 symmetry and is thus slightly different than the initial core loading pattern in Fig. 3 of Sengler et al. (1999). The asymmetry concerns row 2 of the core map in the reference. The ARES nodalization was based on the fuel assembly lattice. The horizontal node widths corresponded to the assembly pitch 21.504 cm. In the axial direction, the active length of the fuel (420 cm) was divided into 20 nodes resulting in an axial node height of 21 cm. One additional axial node was added both below and above the active fuel to represent the bottom and top reflector regions. As the width of the heavy reflector was chosen manually, it corresponds to a single node width. An additional ring of radial water reflector nodes was included after the heavy reflector nodes.

While the full core solution cannot be considered to be truly representative for the EPR reactor due to the assumptions made in building the core model, the model will be perfectly applicable to estimating the effects of the simplifications in the group constant generation on the fuel-cycle simulator results.

5.3. Solution transfer between Serpent and ENIGMA

A small wrapper program handled the solution transfer between Serpent and ENIGMA passing the tallied radial power distribution in each rod to ENIGMA input-files alongside with the information regarding the burnup step length that was needed by ENIGMA to correctly predict the effects of the irradiation on the fuel and cladding thermal and mechanical properties. The wrapper program also read the radial temperature distribution and radial node displacement in the fuel rods from ENIGMA output files and wrote them back to a multi-physics interface that was read by Serpent. The restart capability of ENIGMA was utilized in the coupling, meaning that the ENIGMA solution could be continued from the previously solved burnup point without having to solve the whole irradiation history on each execution. Serpent used the coupled calculation capabilities to signal with the wrapper code at the correct points of the execution of the coupled burnup algorithm (steps 5 and 10 in Table 1).

Due to the 1/8 symmetry in all of the fuel assemblies, there were only 39 unique fuel rods in each assembly. The fuel behavior was solved separately for each of these 39 fuel rods. This included tallying the radial fission power distribution in six rings of equal area for each rod[3]. The radial power distribution was transferred to ENIGMA using a separate power depression file and the fuel temperature and radial node displacement results were read from the ENIGMA *.op8* output-file and transferred to Serpent at 52 radial nodes in the fuel and 3 nodes in the cladding.

5.4. Depletion parameters

The 1/8 symmetry of the assemblies was also utilized in the division of the fuel into depletion zones. Each unique lattice position was considered as a separate depletion region. The pellet of the burnable absorber rods was divided in standard manner into 10 radial rings of equal area, while the pellet of the pure UO_2 rods was divided into two depletion zones to capture the formation of the high burnup structure on the outer rim of the pellet based on studies on the radial subdivision of non-BA rods previously made for the group constant generation in Leppänen and Mattila (2016). The outer depletion zone (rim region) in the non-BA rods contained the outermost 0.3 mm of the pellet. The burnup calculation included 1606 nuclides, 354 of which had cross section data and were included in the transport calculation.

Depletion history consisted of 53 steps: Two short steps of 0.05 MWd/kgU to the burnup of 0.1 MWd/kgU were followed by a step of 0.40 MWd/kgU to the burnup of 0.50 MWd/kgU. After this, 0.5 MWd/kgU steps were taken until reaching the burnup of

[3] The number was chosen as a compromise between radial fidelity and statistical accuracy.

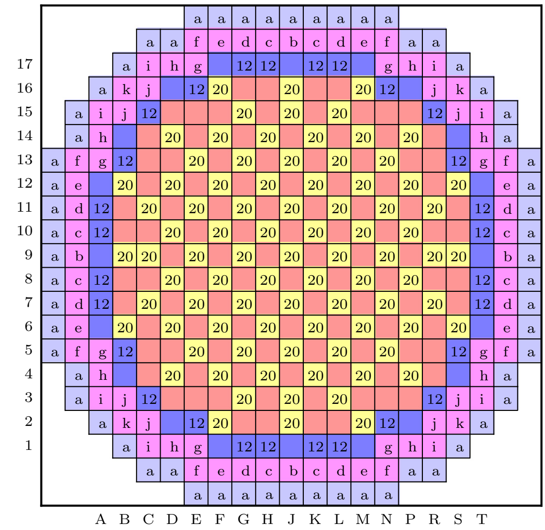

a a a a a a a a a
a a f e d c b c d e f a a
17 a i h g 12 12 12 12 g h i a
16 a k j 12 20 20 20 12 j k a
15 a i j 12 20 20 20 12 j i a
14 a h 20 20 20 20 20 20 h a
13 a f g 12 20 20 20 20 20 12 g f a
12 a e 20 20 20 20 20 20 20 20 e a
11 a d 12 20 20 20 20 20 20 20 12 d a
10 a c 12 20 20 20 20 20 20 12 c a
9 a b 20 20 20 20 20 20 20 20 20 20 b a
8 a c 12 20 20 20 20 20 20 12 c a
7 a d 12 20 20 20 20 20 20 20 12 d a
6 a e 20 20 20 20 20 20 20 20 e a
5 a f g 12 20 20 20 20 20 12 g f a
4 a h 20 20 20 20 20 20 h a
3 a i j 12 20 20 20 12 j i a
2 a k j 12 20 20 20 12 j k a
1 a i h g 12 12 12 12 g h i a
a a f e d c b c d e f a a
a a a a a a a a a
A B C D E F G H J K L M N P R S T

Fig. 4. Core loading pattern with burnable absorber configurations indicated with arabic numerals. Red, yellow and blue colors indicate 1.9, 3.0 and 3.6 wt.% U-235 enrichment, respectively. Radial reflector nodes in the ARES model are plotted in pink (heavy steel reflector) and light blue (water reflector). Different reflector node types are indicated with different lowercase letters. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

15.0 MWd/kgU, after which the step size was increased to 1.0 MWd/kgU until the burnup of 30.0 MWd/kgU was reached. Further four 2.5 MWd/kgU steps were taken to reach the burnup of 40.0 MWd/kgU followed by two 5.0 MWd/kgU steps to reach the maximum burnup of 50.0 MWd/kgU.

For the burnup calculation, the SIE-burnup scheme was used with a constant number of 16 iterations for each burnup step. Each inner iteration consisted of 5 active cycles with 125000 histories per cycle resulting in 10 million active neutron histories per burnup step. The same number of active neutron histories was used also in Leppänen and Mattila (2016). For the first transport solution, a flat fission source distribution was used as an initial guess with 40 inactive cycles to allow for source convergence based on the Shannon entropy of the fission source. On subsequent transport solutions, the initial fission source was taken from the last criticality cycle of the previous transport solution. The number of inactive cycles was set to 20 for the first iteration of each burn-up point and to zero for the subsequent inner iterations. Based on a group ($N = 5$) of comparison calculations for one of the history cases, skipping the inactive cycles on the subsequent inner iterations did not produce a statistically significant effect on the final nuclide concentrations.

5.5. Homogenization of the fuel assemblies

The fuel assemblies were homogenized in infinite lattice using reflective boundary conditions. Nine separate coolant temperature and boron history cases were run to generate the full set of group constants needed in the simulation of the initial cycle. These history cases are listed in Table 3. For the history cases 1, 4 and 7, the group constants had to be generated for 21 branch variations listed in Table 4. For the other history cases the group constants were generated only for the nominal branch (number 0). The branch calculations were conducted using the restart capability of Serpent at 15 burnup points at 0.0, 0.1, 1, 3, 5, 7, 9, 10, 11, 12, 15, 20, 30, 40 and 50 MWd/kgU. Each branch calculation used 10 million active neutron histories with 125 000 histories per cycle. 40 inactive cycles were included for each calculation to allow for the fission source convergence. (See Table 5).

Table 3
The nine history cases used in the group constant generation.

#	H_m (K)	H_{bo} (ppm)	Branches
1	565	0	21
2	565	650	1
3	565	1300	1
4	583	0	21
5	583	650	1
6	583	1300	1
7	610	0	21
8	610	650	1
9	610	1300	1

Table 4
The 21 branch variations used in the group constant generation.

#	B (ppm)	T_f (K)	T_m (K)	void (%)
0	0	900	583	–
1	650	900	583	–
2	1300	900	583	–
3	0	600	583	–
4	650	600	583	–
5	1300	600	583	–
6	0	1200	583	–
7	0	900	565	–
8	650	900	565	–
9	1300	900	565	–
10	0	900	610	–
11	650	900	610	–
12	1300	900	610	–
13	0	1200	610	–
14	0	900	583	7.5
15	650	900	583	7.5
16	0	900	583	15
17	1300	900	583	15
18	0	1200	583	15
19	0	900	565	7.5
20	0	900	610	15

Table 5
Comparison of calculated parameters for the EPR core between the initial core description and the two Serpent-ARES simulations.

	Sengler et al. (1999)	Serpent-ARES Reference	Serpent-ARES Effective
Critical boron at BOL (no Xe135)	960 ppm	973 ppm	969 ppm
Critical boron at BOL (with Xe135)	688 ppm	645 ppm	643 ppm
Cycle length	468 d	462 d	460 d
Axial peaking at BOL	1.4	1.39	1.39
Axial peaking at EOL	1.25	1.24	1.24

5.6. Homogenization of the reflector regions

The top and bottom reflectors were homogenized in a radially infinite 3D assembly lattice mimicking the loading pattern at the core center using a checkerboard pattern of the 1.9 and 3.0 wt.% enriched assemblies (R00 and Y20). For the homogenization, the bottom reflector water conditions were set to the inlet conditions (565 K), while the top reflector used the average outlet conditions (602 K). The active fuel region used the average core coolant temperature (583 K) and a fuel temperature of 900 K.

The radial reflector nodes were homogenized using a 2D full core model at the active core level (Fig. 5). The heavy reflector

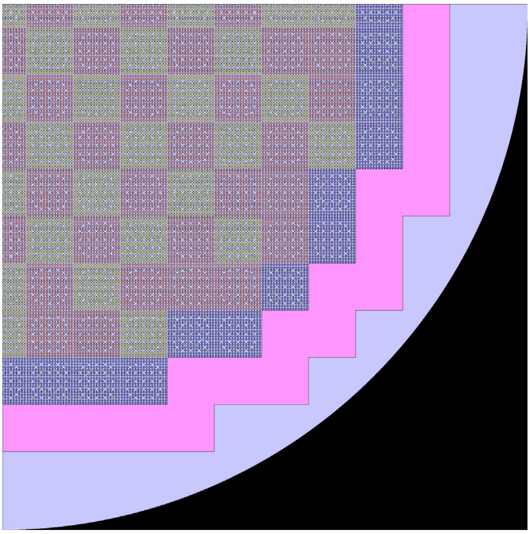


Fig. 5. Serpent geometry plot from the 2D full core model. 1/4 core presented. The simplified model for the heavy radial reflector can be seen in pink. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

was divided into multiple different node types depending on how they were connected with adjacent fuel nodes (see Fig. 4) while the water reflector was homogenized as a single area. The active fuel regions of the core used the same conditions as in the homogenization of the axial reflectors and the radial heavy reflector and water reflector used a temperature of 575 K.

The moderator densities used in the group constant generation were calculated based on the temperature dependent water density obtained from the NIST steam tables for the operating pressure (15.5 MPa) and the moderator boron concentration.

6. Results

The group constants were generated on the Triton cluster of Aalto University School of Science. The simulations used mixed MPI/OpenMP parallelization with 12 OpenMP threads per MPI-task and either two or four MPI-tasks depending on the simulation. The simulations were executed in two parts by first running the burnup calculation and then performing the branch calculations using the restart capabilities of Serpent that automatically runs all of the branch variations during a single simulation. The uncoupled burnup calculations were run with 2 MPI tasks and all other simulations used 4 MPI tasks.

Typical running times (wall clock time) for the different simulations on a typical node (ProLiant SL390s G7, 2x6 core Intel Xeon X5650 processor) were 6 h for each uncoupled burnup history (2 MPI tasks), 10 h for each coupled burnup history (4 MPI tasks) and 12 h for the set of 315 restart calculations (4 MPI tasks) needed to simulate the 21 branch variations in 15 burnup points. Recalculation of the whole set of burnup histories and restarts for either of the two simulation cases could be typically achieved in three days with small variations coming from the overall cluster load.

The difference in the calculation time between the uncoupled and coupled burnup calculation comes almost completely from the fact that the coupled burnup calculation used a lower optimization mode of Serpent. There are three major optimizations lacking from the coupled calculation compared to the uncoupled one: First, the material wise macroscopic cross sections could not be precalculated due to the fact that the temperature was not constant in each material. Second, in order to save memory and make us of even the lowest memory nodes in the cluster, the cross sections of different nuclides were not reconstructed on an unionized energy grid. Last, the spectrum collapse method for calculating the one-group transmutation fluxes and cross sections needed for the depletion solution was not used due to the fact of the non-constant material temperatures. With the spectrum collapse method, only the flux energy spectrum on the unionized energy grid has to be tallied for each depletion region and the one group transmutation cross sections can be collapsed from the continuous energy cross sections as a post-processing step using the tallied flux spectrum. As the temperatures of the burnable materials are non-constant they do not have a material-specific cross section that could be used in the spectrum collapse calculation. Without the spectrum collapse method all of the one group transmutation cross sections have to be tallied during the neutron transport.

The execution of the wrapper code and ENIGMA solvers took less than 2 s per iteration compared to the 60 s combined time of for neutron transport, depletion calculation and processing by Serpent in the coupled burnup calculation.

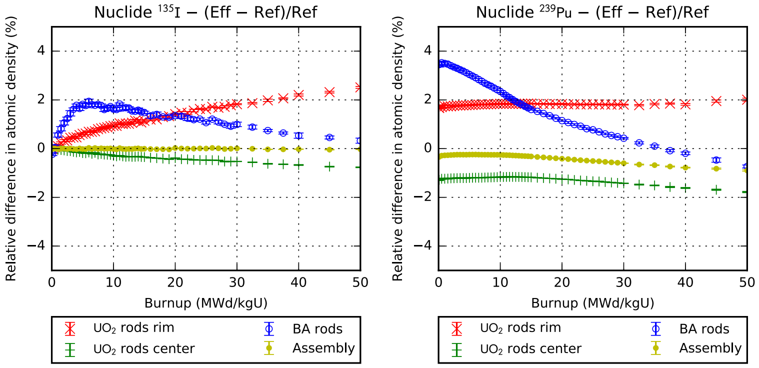


Fig. 6. Relative differences in the concentrations of two nuclides of interest. The trends seen in these two nuclides reflect the general trends seen in other fission products (trends of ^{135}I) and minor actinides (trends of ^{239}Pu). Assembly type Y20. 2σ errorbars. **Left**: ^{135}I a representative non-absorbing fission product with a small total cross section and the precursor of ^{135}Xe, an important parasitic absorber. **Right**: ^{239}Pu an important fissile nuclide transmuted from ^{238}U.

6.1. Nuclide concentrations

As we discussed in Section 3.1, the use of effective temperatures can lead to differences in the local and global nuclide concentrations. In this section we will examine the differences in the nuclide inventories between the effective temperature burnup calculation and the reference calculation. Any observed differences might be of interest to anyone conducting 2D lattice burnup calculations to obtain nuclide inventories of burnt up fuel for purposes such as decay heat, shielding or radiological source term calculations. Due to the large amount of data to analyze (four assemblies, nine burnup histories, hundreds of nuclides) we will present data regarding the burnup history 5 for the Y20 assembly type. This burnup history and assembly type correspond to the node with the maximum power during the initial cycle (node J9 at axial level 7/20 from core bottom). In order to be able to present the results with some estimates of the associated statistical uncertainties, 10 independent burnup calculations were conducted and the sample mean as well as the sample variance of the mean are shown here.

Figs. 6–8 show the relative differences in the concentrations of several interesting nuclides as a function of burnup: The nuclides shown here are ^{135}I representing the fission products and ^{239}Pu representing the actinides produced by transmutation (both in Fig. 6), the two important burnable absorber isotopes ^{155}Gd and ^{157}Gd (Fig. 7) and and finally the two initial heavy metal nuclides ^{235}U and ^{238}U (Fig. 8). Due to its short half–life of 6.6 h and a small transmutation cross section, ^{135}I serves as a good indicator of the fission power level in the analysis. In addition to presenting the relative differences in assembly total nuclide concentrations (yellow dot) we present some of the local differences by looking at the relative differences in the two radial depletion zones of the non burnable absorber rods, namely the outer rim area (red x) and the inner pellet area (green +). We also include the relative differences in total nuclide concentrations in the BA rods (blue circles). We will first note the general conclusions regarding the differences and then provide the details supporting these conclusions in the following subsections.

Using the effective temperature for the burnup simulation overestimates the importance of the surface regions of the fuel pellet due to an increased actinide production and a subsequent increase in fission power, and thus the amount of fission products, in the rim region. The same applies to burnable absorber rods before the burnout of gadolinium: The overestimation of the breeding of fissile ^{239}Pu in BA rods leads to the overestimation of the fission power in the BA rods. The gadolinium burnout is predicted to be slightly slower, when the effective temperature model is used. For the whole assembly, the effective temperature model results in a slightly smaller amount of actinide production and a slightly larger usage of ^{235}U. The total amount of fission products is predicted well. These remarks apply to all of the assembly types and history variations.

6.1.1. Effects on the spatial nuclide distribution

The effective temperature model leads to a constant overestimation of the production of ^{239}Pu (approximately 2 %) and minor actinides in the outer areas of the fuel pellet (Fig. 6, right) as well as a constant underestimation of the actinide production in the inner areas of the pellet. The plutonium production in the BA rods is overpredicted by 3.5 % at the beginning of the burnup history. The trends are also representative of the other plutonium isotopes as well as the minor actinides. The effects on actinide production are explained by the fact that the effective temperature model uses too high fuel temperatures near the pellet surface, as well as in the BA-rods, which leads to an overestimation of the energy–width of the resonances in the ^{238}U radiative capture cross section.

Due to the increased production of fissile actinides, a larger proportion of the fission power is also being generated in the surface areas of the pellet, as well as in the BA-rods, when the effective temperature model is used. This can be seen from the fact that the concentrations of ^{135}I (Fig. 6, left) as well as other fission products (not shown) are overpredicted in the pellet rim area and underpredicted in the pellet central area.

6.1.2. Effects on gadolinium burnout

As the gadolinium burnout from the Y20 assemblies greatly affects the nodal power peaking during the initial cycle it is of great interest to estimate the differences that the effective temperature model may bring to the gadolinium burnout. Fig. 7 shows the relative differences in the momentary concentrations of the two important gadolinium isotopes in the burnable absorber rods. As the relative differences peak in the end of the gadolinium burnout, when the absolute concentrations are small, these figures cannot be used to gauge the effects of the differences on the neutron absorption by the two isotopes. Instead, we will calculate the macroscopic capture cross section for each isotope in the assembly and compare the differences in the macroscopic neutron capture cross section of the Gd isotopes during the burnout. Left side of Fig. 9 shows the macroscopic cross section of each isotope and their sum from 0 to 12 MWd/kgU. As the absolute values of macroscopic cross sections are rarely informative, we have normalized the cross sections by dividing them with the sum of the two isotopic cross sections at zero burnup. We can see that initially ^{157}Gd makes up almost 80 % of the neutron captures between the two isotopes, but from 5 MWd/kgU onwards ^{155}Gd is the main neutron absorbing isotope.

The right side of Fig. 9 shows the difference in the macroscopic capture cross section of the two isotopes as well as in their combined cross section between the effective temperature and the reference calculation. We can see that the effective temperature calculation results in a slightly slower burnout of both isotopes. The differences in the ^{157}Gd burnout reach their maximum at 7 MWd/kgU while the differences in the ^{155}Gd burnout peak at 10 MWd/kgU. The combined effect reaches approximately 0.35 % of the initial Gd reactivity value at the burnup of 9 MWd/kgU. Overall, the differences in the Gd burnout are small but distinguishable. Similar trends could be observed in the B12 assembly type.

6.1.3. Assembly total nuclide inventory

The effective temperature model reproduced the total amount of fission products well. This can be partly attributed to the fact that if the average energy deposition per fission is similar between the two simulations, the fission rates and thus the fission product production rates should also be similar. However, some differences can be seen in certain fission product isotopes that have a significant removal term from transmutation. For example the effective temperature calculation predicts slightly smaller ^{135}Xe concentrations and slightly larger ^{136}Xe concentrations. This difference is attributed to the fact that due to a smaller assembly fission cross section in the effective temperature calculation (see next section), a higher flux is needed to achieve the same fission rate, i.e. the same power level. This increased flux will also lead to a base increase in transmutation reaction rates. Similar differences can be seen between the concentrations of ^{103}Rh and ^{104}Pd as the rhodium isotope has a large radiative capture resonance near 1.26 eV and almost all of the produced ^{104}Rh decays to ^{104}Pd with a half-life of 42 s.

Overall, an effective temperature of 900 K leads to a slightly smaller amount of produced actinides. However, setting the flat temperature to a higher value should produce more actinides in general due to increased resonance absorption by ^{238}U as well as by other actinides. To test this proposition we conducted the

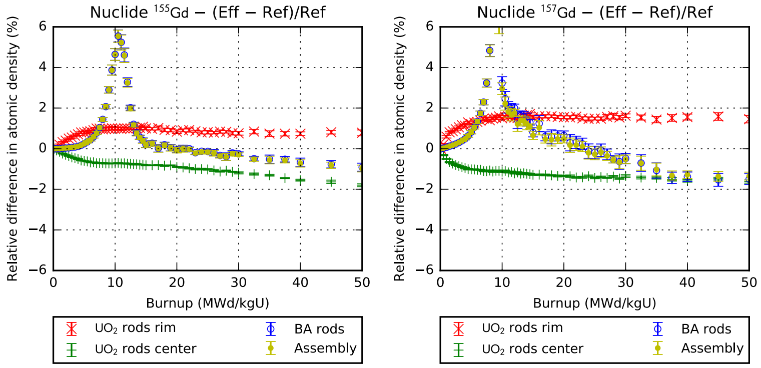


Fig. 7. Relative differences in the concentrations of the two Gd-isotopes with the largest reactivity effect. Assembly type Y20. 2σ errorbars. **Left**: ^{155}Gd. **Right**: ^{157}Gd.

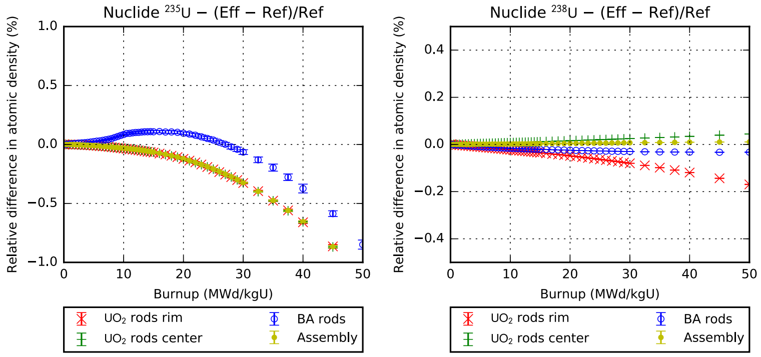


Fig. 8. Relative differences in the concentrations of the two initial Uranium isotopes. Assembly type Y20. 2σ errorbars. **Left**: ^{235}U the main power producing nuclide in the beginning of the burnup history. **Right**: ^{238}U a major resonance absorber and the fertile nuclide transmuted to ^{239}Pu through radiative neutron capture.

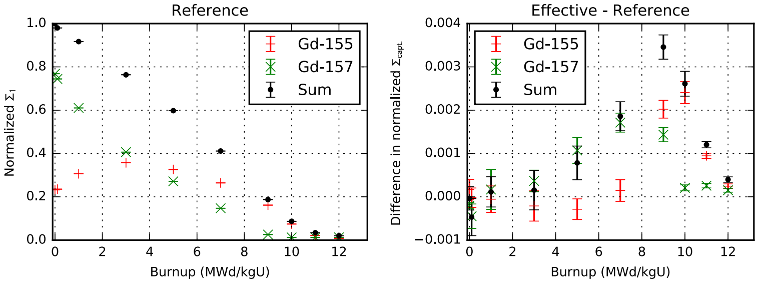


Fig. 9. Gadolinium burnout in the Y20 assembly type. 2σ errorbars. **Left**: Normalized capture cross sections of ^{155}Gd, ^{157}Gd and ^{155}Gd + ^{157}Gd in the reference simulation. **Right**: Differences in the normalized capture cross sections of the different Gd-isotopes between the effective and reference simulation.

calculations a second time for assembly type Y20, history 5, this time using a slightly higher effective temperature of 950 K. We can compare Fig. 6 showing differences in ^{135}I and ^{239}Pu between the 900 K simulation and the reference to Fig. 10, which shows the differences in the case of the additional simulation using a flat fuel temperature of 950 K. We can see that the increase in the effective fuel temperature leads to a slightly stronger overestimation of actinide production and fission power in the rim region, but also results in a larger actinide production overall in a manner that predicts the assembly total ^{239}Pu inventory more accurately throughout the burnup history. This result indicates that a well chosen effective temperature may be able to conserve the total actinide production[4]. However, the increased effective temperature performs worse regarding the gadolinium burnout (see Fig. 11) with the maximum difference in the normalized capture cross section increasing from 0.0035 to almost 0.0050. The results suggest that there is no single optimal effective temperature that would conserve all of the reaction rates in the assembly geometry.

[4] The problem of obtaining such a temperature without needing to execute the detailed coupled calculation, however, still stands.

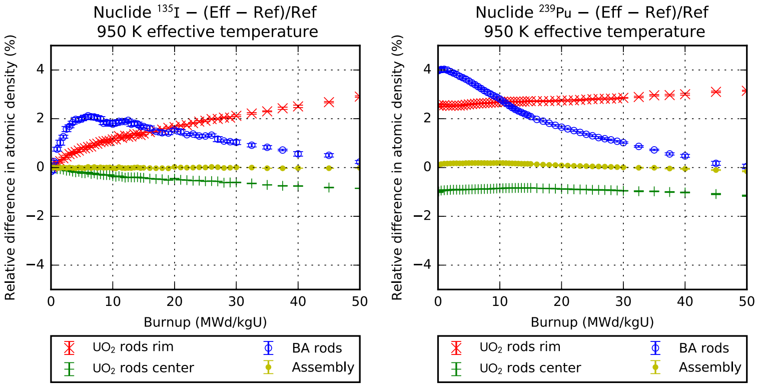


Fig. 10. Relative differences in the concentrations of two nuclides of interest between the **950 K** effective temperature model and reference. Assembly type Y20. 2σ errorbars. **Left**: ^{135}I. Comparison to Fig. 6 (left) shows that increasing the effective temperature to 950 K leads to slightly larger differences in the spatial nuclide distribution (between rim and center). **Right**: ^{239}Pu. Comparison to Fig. 6 (right) shows that increasing the effective temperature to 950 K leads to a more accurate prediction of the assembly-wide actinide concentration.

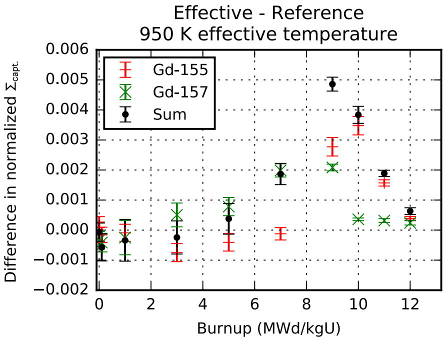


Fig. 11. Absolute differences in the gadolinium burnout between the **950 K** effective simulation and reference simulation for the assembly type Y20. Comparison to Fig. 9 shows that increasing the effective temperature to 950 K increases the differences in the gadolinium burnout.

6.2. Group constants

Prior to the analysis of the group constants we look at the differences in the development of the assembly reactivity and the flux levels required to reach the power level used in the calculations. These parameters are shown in Fig. 12, and while they are not passed to ARES they are nevertheless helpful in assessing the effects of the effective temperature calculation. There is a slight underestimation of the assembly reactivity near 10 MWd/kgU burnup, which coincides to the differences seen in the burnout of the gadolinium from the burnable absorber rods. This reactivity underestimation cannot be seen in the B00 or R00 assembly types, which indicates that it is indeed associated with the differences in the gadolinium burnout. The maximum k_∞ for the Y20 assembly type is reached at 11 MWd/kgU. The reactivity difference decreases until 15 Mwd/kgU, after which it increases steadily. The reason for the decreased reactivity at high burnups is the decreased amount of actinides in the assembly when using the effective temperature of 900 K.

To gain further insight into the components of the reactivity difference we can look at the homogenized capture and fission neutron production cross sections that are shown in Fig. 13. The difference in the reactivity near 10 MWd/kgU seems to be associated with a negative difference in the thermal and fast fission neutron production cross sections at the time combined with a slight overestimation of the thermal capture cross section consistent with the slower burnout of gadolinium. At high burnups, the difference in the fission neutron production cross section increases for both groups as does the difference in the thermal capture cross section. Using an effective temperature of 950 K predicts the behavior of the assembly better at high burnups but has higher discrepancies during the gadolinium burnout (not shown).

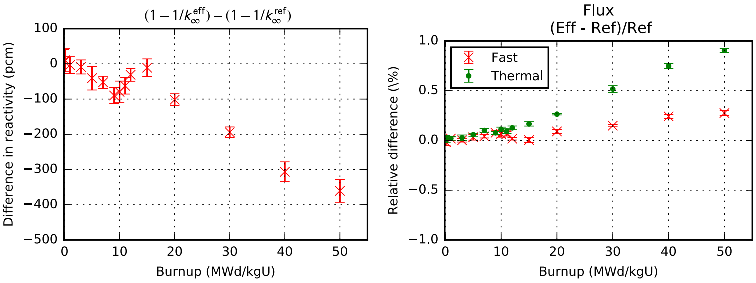


Fig. 12. Relative differences in two parameters obtained from the group constant generation for assembly Y20, history 5, nominal branch. 2σ errorbars. **Left**: Reactivity of the assembly in the infinite lattice geometry. **Right**: Thermal and fast flux.

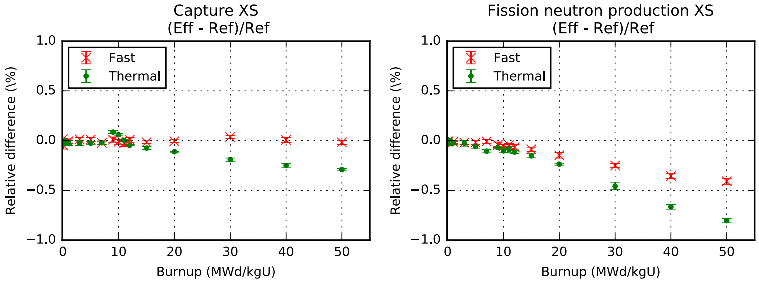


Fig. 13. Relative differences in generated homogenized group constants for assembly Y20, history 5, nominal branch. 2σ errorbars. **Left**: Capture cross section (calculated as the difference of absorption cross section and fission cross section). **Right**: Fission neutron production cross section.

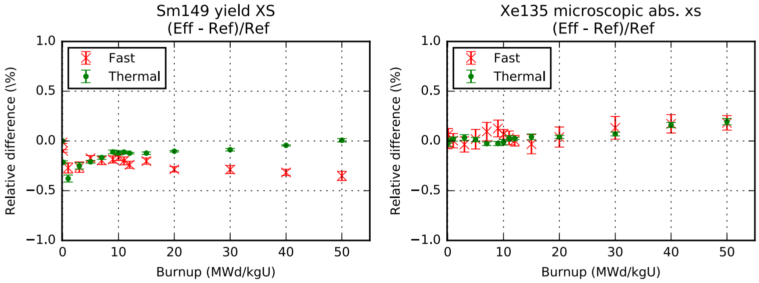


Fig. 14. Relative differences in two generated homogenized group constants for assembly Y20, history 5, nominal branch. 2σ errorbars. **Left**: Microscopic yield for ^{149}Sm. **Right**: ^{135}Xe microscopic absorption cross section.

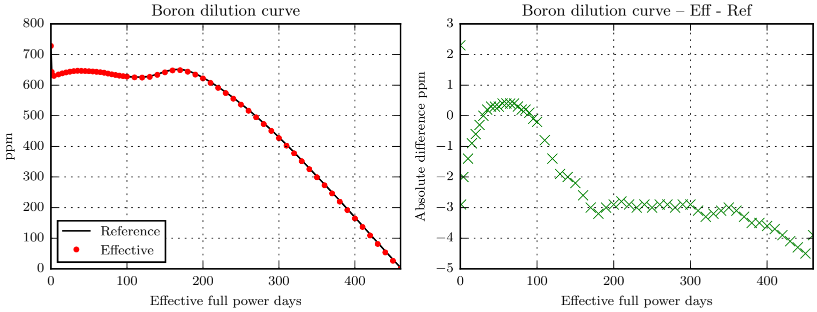


Fig. 15. Boron dilution curves for the EPR initial cycle using the two methods for group constant generation. **Left**: Critical boron for using the group constants from the two fuel temperature models. **Right**: Difference in critical boron.

For the generated group constants, the largest differences could be seen in the fission cross sections, which were already shown. Fission product yields were predicted well for ^{135}I, ^{135}Xe and ^{149}Pm. However, the yield for ^{149}Sm was underestimated throughout the burnup history (see Fig. 14 left). ^{135}Xe microscopic absorption cross section (Fig. 14 right) serves as a representative example of the relative differences in the microscopic absorption cross sections of the poison nuclides: The differences are small overall, with a slight overprediction of the fast absorption cross section during the Gd burnout and overprediction in both groups increasing with burnup after Gd-burnout.

Generally the effect of the temperature model on the generated group constants is small. However, the differences in the fission cross section at high burnups may lead to differences in the distribution of power between high burnup and low burnup nodes in full core simulations.

6.3. ARES simulation

The 58 step history simulation with ARES from 0 to 462 effective full power days (EFPD) using critical boron iteration and reactivity feedback coefficient calculations took approximately 1 h 45 min on a single Intel i5-5300U (2.3 GHz) laptop CPU core. Comparing some of the results from the reference simulation to the values given in Sengler et al. (1999) show that the results of the Serpent-ARES code system are fairly close to the original estimates despite the approximations concerning the radial and axial reflectors. The objective of this study was not, of course, to reproduce the values given in the core description, but it is nevertheless useful to note the similarities and differences between the calculated and the reported values.

The boron dilution curves for both simulations are shown in Fig. 15. No major differences can be seen in the critical boron

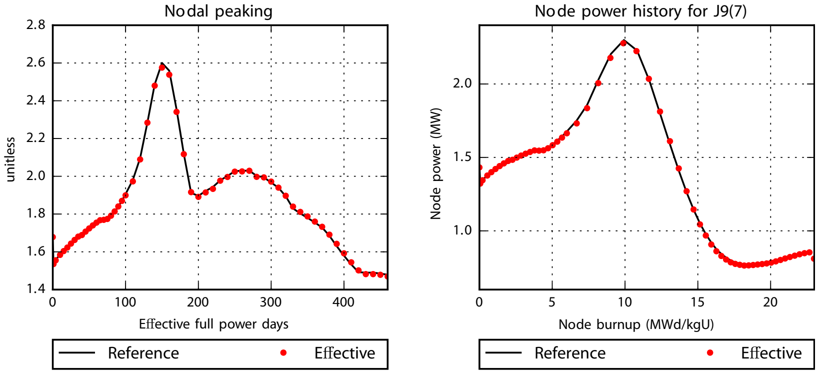


Fig. 16. Right: Nodal power peaking (node power divided by mean node power) during the initial cycle. The maximum peak corresponds to a power of approximately 2.3 MW at the axial level of 7 in the J9 fuel assembly (**Left**). At this point, the node-wise burnup of the peaking node is approximately 10 MWd/kgU, which means that the gadolinium in the node has almost burned out (Fig. 9 left).

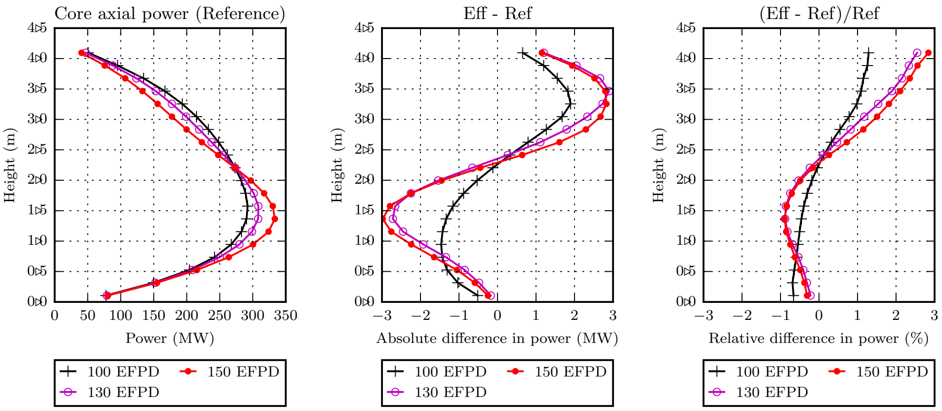


Fig. 17. Axial power distribution in the EPR during node wise power peaking. Different curves indicate different time points in effective full power days (EFPD). **Left**: Power at different axial levels for reference simulation. **Middle**: Absolute difference in the core axial power distribution between effective and reference simulation. **Right**: Relative difference in the core axial power distribution.

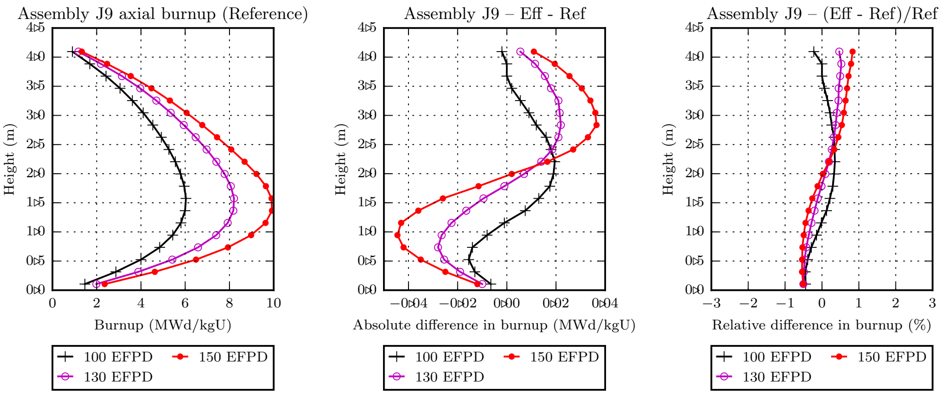
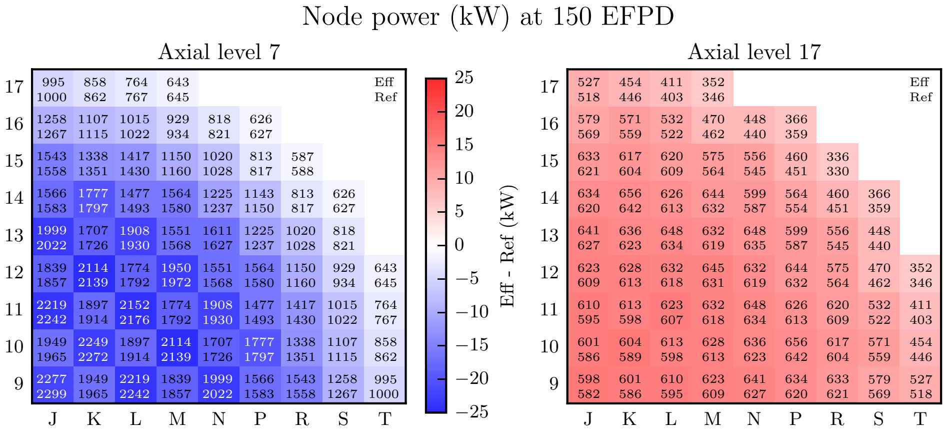


Fig. 18. Axial burnup distribution in the J9 assembly during node wise power peaking. Different curves indicate different time points in effective full power days (EFPD). **Left**: Absolute node wise burnups for reference simulation. **Middle**: Absolute difference in node-wise burnups. **Right**: Relative difference in node-wise burnups.



Node power (kW) at 150 EFPD

Axial level 7

	J	K	L	M	N	P	R	S	T
17	995 1000	858 862	764 767	643 645					Eff Ref
16	1258 1267	1107 1115	1015 1022	929 934	818 821	626 627			
15	1543 1558	1338 1351	1417 1430	1150 1160	1020 1028	813 817	587 588		
14	1566 1583	1777 1797	1477 1493	1564 1580	1225 1237	1143 1150	813 817	626 627	
13	1999 2022	1707 1726	1908 1930	1551 1568	1611 1627	1225 1237	1020 1028	818 821	
12	1839 1857	2114 2139	1774 1792	1950 1972	1551 1568	1564 1580	1150 1160	929 934	643 645
11	2219 2242	1897 1914	2152 2176	1774 1792	1908 1930	1477 1493	1417 1430	1015 1022	764 767
10	1949 1965	2249 2272	1897 1914	2114 2139	1707 1726	1777 1797	1338 1351	1107 1115	858 862
9	2277 2299	1949 1965	2219 2242	1839 1857	1999 2022	1566 1583	1543 1558	1258 1267	995 1000

Axial level 17

	J	K	L	M	N	P	R	S	T
17	527 518	454 446	411 403	352 346					Eff Ref
16	579 569	571 559	532 522	470 462	448 440	366 359			
15	633 621	617 604	620 609	575 564	556 545	460 451	336 330		
14	634 620	656 642	626 613	644 632	599 587	564 554	460 451	366 359	
13	641 627	636 623	648 634	632 619	648 635	599 587	556 545	448 440	
12	623 609	628 613	632 618	645 631	632 619	644 632	575 564	470 462	352 346
11	610 595	613 598	623 607	632 618	648 634	626 613	620 609	532 522	411 403
10	601 586	604 589	613 598	628 613	636 623	656 642	617 604	571 559	454 446
9	598 582	601 586	610 595	623 609	641 627	634 620	633 621	579 569	527 518

Fig. 19. Radial power distribution at 150 EFPD at two axial levels corresponding to the maximum absolute differences in axial power distribution (see Fig. 17 middle). The upper values for each node indicate the node power in kW using the effective temperature generated group constants, whereas the lower value indicates the node power in the reference simulation. The color of the node indicates the absolute difference in node power between the two simulations (effective – reference). As the core is symmetric, only a quarter of the core is shown.

concentrations between the two sets of group constants, although the smaller reactivity of the fuel assemblies at high burnups can be seen as slightly smaller values of critical boron and a shorter cycle length for the effective temperature simulation. The observed 2 ppm to 4 ppm differences in the critical boron are small compared to overall uncertainties in the nodal diffusion methodology. A peak can be seen in the critical boron at 160 EFPD corresponding to the gadolinium burnout at the region of the core with the highest power and thus the highest burnup in the first half of the initial cycle. The nodal power peaking increases by a significant margin (from 1.8 to 2.6) between 100 and 160 EFPD when Gd burns out at maximum power/burnup level (Fig. 16 left). The maximum nodal power is seen after 150 EFPD at axial level 7/20 of the core center assembly J9 (Fig. 16 right).

Some differences can be observed in the axial power shape of the reactor during the gadolinium burnout from the high power nodes. To aid the discussion, the axial power shape for the core has been plotted in the left side of Fig. 17 and the axial burnup distribution in the highest burnup, as well as highest power, assembly J9 has been plotted in the left side of Fig. 18 for 100, 130 and 150 EFPD. The absolute differences in the two axial distributions between the effective and the reference are plotted in the middle axes of the two figures, while the relative differences are presented on the right side of the figure. The effective temperature simulation underestimates the power in the core lower half and overestimates it in the core upper half during this part of the fuel cycle. The maximum relative difference in the axial power distribution reaches approximately 3 %. However, the largest relative differences are at the uppermost axial level, in which the absolute power level is at its lowest. The discrepancy in the axial burnup distribution is more modest but clearly distinguishable. When comparing the differences in the axial burnup distribution (Fig. 18 right) and the axial power distribution (Fig. 17 right) especially at 100 EFPD one should remember that even if there would be no difference in the node burnup during the gadolinium burnout, the effective temperature based group constants underestimate the node reactivity (and thus here the power) due to the differences in the gadolinium burnout (see Section 6.1.2).

To shine light on the differences seen in the axial power and burnup distributions, Fig. 19 presents the radial power distribution at the axial levels with the largest absolute difference in power level at 150 EFPD (see Fig. 17, middle). The figure shows that the largest differences in the node power at axial level 7 can be found in the nodes that contain burnable absorber rods. This can be easily understood on the basis that the effective temperature model led to a slower burnout of gadolinium, which is here seen as a reduced power level in the BA-rod containing nodes compared to the reference simulation. At the top part of the core (axial level 17), where the burnups are more modest, the differences in the gadolinium burnout have not started to show yet.

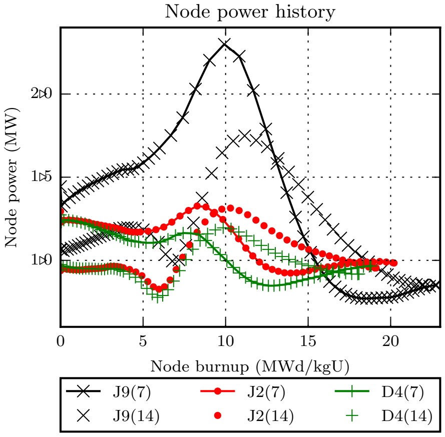


Fig. 20. The power history of six different nodes in the ARES calculation. The assembly type of each node is Y20. The axial level of the node is given in parenthesis. For example, J9(7) refers to the assembly position J9 (see Fig. 4) at the axial level of 7/20.

Overall, small but clearly distinguishable differences can be seen in the axial and radial power distributions between the two ARES simulations. The formation of the observed differences seems to be driven by the slower gadolinium depletion in the group constants generated with the effective temperature model. Using the reference model based group constants, the gadolinium is more rapidly depleted from the lower half of the core, where the fission power (and flux) is peaked in the beginning of the reactor operation.

This decrease in burnable absorber concentrations then leads to even more power being generated in the lower half of the core, which furthers the difference between the two ARES simulations.

It should be noted that the maximum node-wise burnups stay below 24 MWd/kgU during the initial cycle and the differences in the nuclide concentrations and group constants could be seen to grow with burnup reaching their maximum at the final burnup point of 50 MWd/kgU.

7. Future work

7.1. Regarding GC generation with thermal feedback

The full core simulation in this work only covered the initial cycle of the EPR with the maximum node wise burnups staying below 24 MWd/kgU. As can be seen from the results presented in Sections 6.1 and 6.2, the differences in the nuclide concentrations as well as in the group constants steadily increase with burnup. This indicates that some interesting differences might be found when studying the equilibrium fuel cycle in which the node wise burnups reach much higher values. The initial cycle also only contains assemblies with uranium dioxide fuel and similar studies could be applied to one of the MOX fuel cycles of the EPR reactor described in Sengler et al. (1999). The effect of the realistic temperature distributions on the group constant generation might also be somewhat different in BWR applications.

The results also showed that choosing a slightly higher effective fuel temperature of 950 K instead of 900 K increased the accuracy in the predictions for actinide production but decreased the accuracy in predicting the Gd-burnout in the burnable absorber rods. This suggests that the choice of a single effective temperature for all fuel rods in an assembly containing burnable absorber rods will be a difficult task. Instead, better predictions might be obtained by using one effective temperature for the fuel containing burnable absorber and another for the non-BA fuel.

The use of a constant power level throughout the burnup history is another common simplification that was also used in this work. In practice, the power history (and thus the fuel temperature history) of an assembly type can be very different depending on the location of the assembly in the core. Furthermore, the results in Sections 6.1 and 6.2 showed that the fuel temperature history clearly affects the actinide concentrations, and thus the reactivity of the assembly. Fig. 20 shows the power history of six different nodes, all of them the Y20 assembly type, during the initial cycle of the EPR. The average nodal power[5] in the core is 0.88 MW, which corresponds to the power level of the burnup calculations in this study. Looking at Fig. 20 it is easy to question, whether the use of a single, constant, power history (and an associated fuel temperature history) to represent all of these different power histories gives, on average, acceptable results. Using the coupled burnup capabilities, it would be straightforward to verify this by comparing the results of a coupled burnup calculation using the pre-calculated detailed power history to the results from a burnup calculation using a constant power level.

The results of such a comparison would be useful in assessing, whether the fuel temperature should be considered also as a history variation when the group constants are being generated instead of simply as a branch (momentary) variation. Such a change in the group constant generation would also require implementing changes in the group constant parametrization of the reactor simulator.

7.2. Regarding coupled MC burnup calculations with thermal feedback

As discussed in Section 3.2, the SIE burnup scheme does not account for the changes in reaction rates during the burnup step, limiting the accurate step lengths of the algorithm in cases where there are fast shifts in the neutron spectrum, such as in the case of gadolinum burnup. For future MC burnup calculations with thermal feedback, the SIE burnup scheme in Serpent should be replaced with an algorithm that makes it possible to account for the changes in the reaction rates during the burnup step. One potential candidate is the improved SUBSTEP method (Kotlyar and Shwageraus, 2016b), which parametrizes the reaction rates with respect to nuclide concentrations and thermal hydraulic conditions and updates the reaction rates during the burnup step by dividing it to sub-steps for the depletion solver.

The execution of the ENIGMA fuel performance solvers by the wrapper code was not parallelized in this study. Since there is no interaction between the fuel performance solution between the rods, the execution of the solvers could easily be parallelized if the number of required ENIGMA solutions grows large in future applications.

8. Conclusions

The implementation of the Stochastic Implicit Euler algorithm with thermal feedback allows Serpent to execute coupled burnup calculations including the thermal feedback from fuel behavior and/or thermal–hydraulics. These capabilities were utilized in this work for the verification of the traditional way of generating group constants for a core simulator which uses a flat fuel temperature approximation in the burnup calculation. The calculation chain consisted of three parts: First, a burnup calculation to obtain material compositions for each burnup point for each history variation. Second, group constant generation by running restart calculations for the branch variations using the material compositions obtained from the first part. Last, a simulation of the EPR initial cycle by the ARES core simulator using the group constants generated in part two.

For the verification part the calculation chain was executed in two ways: The reference simulation used realistic fuel temperature profiles in the burnup simulation obtained from the externally coupled fuel performance code ENIGMA. The effective temperature simulation used an effective flat fuel temperature of 900 K for all of the burnup calculations. The group constant generation and the simulation of the fuel cycle with ARES were then executed in the same manner for both of the simulations. The main findings from the verification study are as follows: While the use of the effective temperature leads to an overestimation of the actinide and fission product concentrations in the outer rim of the fuel pellet and an associated underestimation of the concentrations in the pellet inner parts, it reproduces the total amounts of fission products very well. The amount of produced actinides can be predicted well if the effective temperature is chosen correctly, however an effective temperature which predicted the actinide production better fared worse in predicting the gadolinium burnout. Small differences were observed in the gadolinium burnout between the reference and the effective simulation with the effective temperature of 900 K leading to a slightly slower gadolinium burnout.

The effects on generated group constants were typically small, although noticeable differences could be seen during the gadolinium burnout. When the effective temperature model predicted the actinide production incorrectly, the differences in the group constants increased at high burnups. The differences in the gadolinium burnout resulted in slight differences in the axial power shape of the ARES solution during the first half of the operation cycle. A slightly shorter cycle length could be observed with the effective temperature simulation with small (2–4 ppm) differences in the critical boron in the beginning of the cycle. Small but clearly distinguishable differences could be seen in the axial and radial power distributions during the gadolinium burnout from the lower part

[5] Core thermal power per number of fuel nodes.

of the reactor core. These differences seem to be driven by the differences in the gadolinium burnout between the effective and reference models and may warrant future study. However, the overall differences between the two simulations were modest.

The observed effect of the fuel temperature history on the actinide content of the fuel assembly, and thus on the homogenized fission cross section of the assembly, suggests that the use of fuel temperature history in the group constant parametrization may provide an opportunity to increase the accuracy of the group constant parametrization for core simulators.

Acknowledgments

This work has been funded by the NUMPS project of the Academy of Finland as well as the MONSOON project under the Finnish National Research Programme on Nuclear Power Plant Safety, SAFIR-2014 and SAFIR-2018. The simulations presented above were performed using computer resources within the Aalto University School of Science "Science-IT" project.

References

Dufek, J., Anglart, H., 2013. Derivation of a stable coupling scheme for Monte Carlo burnup calculations with the thermal-hydraulic feedback. Ann. Nucl. Energy 62, 260.

Dufek, J., Hoogenboom, J., 2009. Numerical stability of existing Monte Carlo burnup codes in cycle calculations of critical reactors. Nucl. Sci. Eng. 162, 307–311.

Dufek, J., Kotlyar, D., Shwageraus, E., 2013a. The stochastic implicit Euler method – a stable coupling scheme for Monte Carlo burnup calculations. Ann. Nucl. Energy 60, 295–300.

Dufek, J., Kotlyar, D., Shwageraus, E., Leppänen, J., 2013b. Numerical stability of the predictor–corrector method in Monte Carlo burnup calculations of critical reactors. Ann. Nucl. Energy 56, 34–38.

Dufek, J., Valtavirta, V., 2014. Time step length versus efficiency of Monte Carlo burnup calculations. Ann. Nucl. Energy 72, 409.

Ellis, M., Forget, B., Smith, K., Gaston, D., 2015. Preliminary coupling of the Monte Carlo code OpenMC and the Multiphysics Object-Oriented Simulation Environment (MOOSE) for analyzing Doppler feedback in Monte Carlo simulations. In: Proc. M&C + SNA + MC 2015, Nashville, TN.

Gill, D., Aumiller, D., Griesheimer, D., 2014. Monte Carlo and thermal-hydraulic coupling via PVMEXEC. In: Proc. Physor 2014, Kyoto, Japan.

Gill, D., Griesheimer, D., Aumiller, D., 2015. Numerical methods in coupled Monte Carlo and thermal-hydraulic calculations. In: Proc. M&C + SNA + MC 2015, Nashville, TN.

Goltsev, A., Davidenko, V., Tsibulsky, V., Lekomtsev, A., 2000. Computational problems in the calculation of temperature effects for heterogeneous nuclear reactor unit cells. Ann. Nucl. Energy 27, 175–183.

Griesheimer, D.P., Gill, D.F., Nease, B.R., Sutton, T.M., Stedry, M.H., Dobreff, P.S., Carpenter, D.C., Trumbull, T.H., Caro, E., Joo, H., Millman, D.L., 2013. MC21 v. 6.0 – a continuous-energy Monte Carlo particle transport code with integrated reactor feedback capabilities. In: Proc. SNA+MC 2013, Paris, France.

Herman, B.R., Forget, B., Smith, K., 2015. Progress toward monte carlo–thermal hydraulic coupling using low-order nonlinear diffusion acceleration methods. Ann. Nucl. Energy 84, 63–72.

Horelik, N., Herman, B., Forget, B., Smith, K., 2013. Benchmark for evaluation and validation of reactor simulations (BEAVRS), v1.0.1. In: Proc. M&C 2013, Sun Valley, ID.

Ikonen, T., Loukusa, H., Syrjälahti, E., Valtavirta, V., Leppänen, J., Tulkki, V., 2015. Module for thermomechanical modeling of LWR fuel in multiphysics simulations. Ann. Nucl. Energy 84, 111–121.

Kilgour, W.J., Turnbull, J.A., White, R.J., Bull, A.J., Jackson, P.A., Palmer, I.D., 1991. Capabilities and validation of the ENIGMA fuel performance code. In: Proc. ANS/ENS Int. Topical Meeting on LWR Fuel Performance, Avignon, France.

Kotlyar, D., Shwageraus, E., 2013. On the use of predictor–corrector method for coupled Monte Carlo burnup codes. Ann. Nucl. Energy 58, 228–237.

Kotlyar, D., Shwageraus, E., 2014. Numerically stable monte carlo-burnup-thermal hydraulic coupling schemes. Ann. Nucl. Energy 63, 371–381.

Kotlyar, D., Shwageraus, E., 2016a. Stochastic semi-implicit substep method for coupled depletion monte-carlo codes. Ann. Nucl. Energy 92, 52–60.

Kotlyar, D., Shwageraus, E., 2016b. Sub-step methodology for coupled monte carlo depletion and thermal hydraulic codes. Ann. Nucl. Energy 96, 61–75.

Kruijf, W.J.M.D., Janssen, A.J., 1996. The effective fuel temperature to be used for calculating resonance absorption in a 238UO2 lump with a nonuniform temperature profile. Nucl. Sci. Eng. 123, 121–135.

Leppänen, J., 2013. Modeling of non-uniform density distributions in the Serpent 2 Monte Carlo code. Nucl. Sci. Eng. 174, 318.

Leppänen, J., 2015. CAD-based geometry type in Serpent 2 – application in fusion neutronics. In: Proc. M&C + SNA + MC 2015, Nashville, TN.

Leppänen, J., Aufiero, M., 2014. Development of an unstructured mesh based geometry model in the Serpent 2 monte carlo code. In: Proc. Physor 2014, Kyoto, Japan.

Leppänen, J., Hovi, V., Ikonen, T., Kurki, J., Pusa, M., Valtavirta, V., Viitanen, T., 2015a. The Numerical Multi-Physics project (NUMPS) at VTT Technical Research Centre of Finland. Ann. Nucl. Energy 84, 55–62.

Leppänen, J., Mattila, R., 2016. Validation of the Serpent–ARES code sequence using the MIT BEAVRS benchmark–HFP conditions and fuel cycle 1 simulations. Ann. Nucl. Energy 96, 324–331.

Leppänen, J., Mattila, R., Pusa, M., 2014. Validation of the Serpent–ARES code sequence using the MIT BEAVRS benchmark initial core at HZP conditions. Ann. Nucl. Energy 69, 212–225.

Leppänen, J., Pusa, M., Fridman, E., 2016. Overview of methodology for spatial homogenization in the Serpent 2 Monte Carlo code. Ann. Nucl. Energy 96, 126–136.

Leppänen, J., Pusa, M., Viitanen, T., Valtavirta, V., Kaltiaisenaho, T., 2015b. The Serpent Monte Carlo code: status, development and applications in 2013. Ann. Nucl. Energy 82, 142–150.

Leppänen, J., Viitanen, T., Valtavirta, V., 2012. Multi-physics coupling scheme in the Serpent 2 Monte Carlo code. Trans. Am. Nucl. Soc. 107, 1165.

Mattila, R., 2002. Kiehutusvesireaktorin Vaikutusalamalli CROSS, Technical Report. VTT Technical Research Centre of Finland.

Mattila, R., 2003. Three-dimensional analytic function expansion nodal model. In: Proc. ANFM 2003, Hilton Head Island, SC.

Richards, S.D., Davies, N., Armishaw, M.J., Dobson, G.P., Wright, G.A., 2013. Parallelisation of MONK with coupling to thermal hydraulics and gamma heating calculations for reactor physics applications. In: Proc. SNA+MC 2013, Paris, France.

Rossiter, G., 2011. Development of the ENIGMA fuel performance code for whole core analysis and dry storage assessments. Nucl. Eng. Technol. 43, 489.

Rowlands, G., 1962. Resonance absorption and non-uniform temperature distributions. J. Nucl. Energy Parts A/B. Reactor Sci. Technol. 16, 235–236.

Sengler, G., Fort, F., Schlosser, G., Lisdat, R., Stelletta, S., 1999. Epr core design. Nucl. Eng. Des. 187, 79–119.

Valtavirta, V., Hessan, M., Leppänen, J., 2016. Delayed neutron emission model for time dependent simulations with the Serpent 2 Monte Carlo code – first results. In: Proc. Physor 2016, Sun Valley, ID, USA.

Valtavirta, V., Ikonen, T., Viitanen, T., Leppänen, J., 2014a. Simulating fast transients with fuel behavior feedback using the Serpent 2 Monte Carlo code. In: Proc. Physor 2014, Kyoto, Japan.

Valtavirta, V., Leppänen, J., Viitanen, T., 2017. Coupled neutronics–fuel behavior calculations in steady state using the Serpent 2 Monte Carlo code. Ann. Nucl. Energy 100, 50–64. Part 2.

Valtavirta, V., Tulkki, V., Leppänen, J., Viitanen, T., 2013. The universal fuel performance code interface in Serpent 2. In: Proc. 2013 Fuel Performance Meeting TopFuel, Charlotte, USA.

Valtavirta, V., Viitanen, T., Leppänen, J., 2014b. Internal neutronics temperature coupling in Serpent 2. Nucl. Sci. Eng. 177, 193–202.

Viitanen, T., 2015. Development of a Stochastic Temperature Treatment Technique for Monte Carlo Neutron Tracking Ph.D. thesis. Aalto University.

Viitanen, T., Leppänen, J., 2016. New interpolation capabilities for thermal scattering data in Serpent 2. In: PHYSOR 2016, Sun Valley, ID, USA.

Viitanen, T., Tulkki, V., 2012. Combining reactor physics and fuel performance calculations. In: Proc. TopFuel 2012, Manchester, UK.

Aalto-DD 66/2017
VTT SCIENCE 150

9 789526 073774

ISBN 978-952-60-7377-4 (printed)	978-951-38-8530-4 (printed)
ISBN 978-952-60-7376-7 (pdf)	978-951-38-8529-8 (pdf)
ISSN-L 1799-4934	2242-119X
ISSN 1799-4934 (printed)	2242-119X (printed)
ISSN 1799-4942 (pdf)	2242-1203 (pdf)

Aalto University
School of Science
Department of Applied Physics
www.aalto.fi