
Simulation of Neutron Multiplicity Measurements using Geant4

Open Source Software for Nuclear Arms Control

Zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation von M.Sc. Moritz Kütt aus Frankfurt am Main

Tag der Einreichung: 14. Juni 2016, Tag der Prüfung: 7. Juli 2016

Juli 2016 – Darmstadt – D 17

1. Gutachten: Prof. Dr. Franz Fujara
2. Gutachten: Prof. Dr. Alexander Glaser



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Physik
Institut für Festkörperphysik

Simulation of Neutron Multiplicity Measurements using Geant4
Open Source Software for Nuclear Arms Control

Genehmigte Dissertation von M.Sc. Moritz Kütt aus Frankfurt am Main

1. Gutachten: Prof. Dr. Franz Fujara
2. Gutachten: Prof. Dr. Alexander Glaser

Tag der Einreichung: 14. Juni 2016

Tag der Prüfung: 7. Juli 2016

Darmstadt – D 17

Bitte zitieren Sie dieses Dokument als:

URN: urn:nbn:de:tuda-tuprints-56213

URL: <http://tuprints.ulb.tu-darmstadt.de/5621>

Dieses Dokument wird bereitgestellt von tuprints,

E-Publishing-Service der TU Darmstadt

<http://tuprints.ulb.tu-darmstadt.de>

tuprints@ulb.tu-darmstadt.de



Die Veröffentlichung steht unter folgender Creative Commons Lizenz:

Namensnennung – Weitergabe unter gleichen Bedingungen – 4.0 International

<http://creativecommons.org/licenses/by-sa/4.0/>

Zusammenfassung

Die vorliegende Arbeit beschreibt die Notwendigkeit der Nutzung von Open Source Software für Verifikationsaufgaben in der nuklearen Rüstungskontrolle und stellt die Entwicklung einer solchen offenen Software zur Simulation von Neutronenmultiplizitätsmessungen vor. Verifikation dient der Sicherstellung der Einhaltung internationaler Verträge. Technische Verfahren spielen dabei eine zentrale Rolle, sowohl zur Vermeidung von Proliferation als auch bei der Verifikation zukünftiger Abrüstung.

Auch 26 Jahre nach Ende des kalten Krieges stellen Kernwaffen noch immer eine weltweite, ernsthafte Bedrohung dar. Neun Staaten besitzen insgesamt rund 15350 Kernwaffen, die Mehrheit davon mit höherer Sprengkraft als die in Nagasaki und Hiroshima eingesetzten Waffen. Sowohl die beabsichtigte Nutzung, wie in den Militärdoktrinen vieler Länder vorgesehen, als auch die unabsichtliche Explosion durch technische oder menschliche Fehler hätten weitreichende Folgen. Um das Ziel einer atomwaffenfreien Welt zu realisieren, können neue internationale Vereinbarungen getroffen werden oder existierende Vereinbarungen ausgeweitet werden. In beiden Fällen wäre dies mit einer Steigerung der Nutzung von Verifikationstechnologien verbunden.

Diese Arbeit identifiziert, bezogen auf diese Messverfahren und Technologien, ein neues Problem, welches in anderen Untersuchungen typischerweise nicht berücksichtigt wird. Die Ergebnisse technischer Verifikationsmaßnahmen spielen oft eine entscheidende Rolle in der Entscheidungsfindung von Staaten und internationalen Organisationen. Neben der inhaltlichen Validität der gewonnenen Daten ist es daher ebenso wichtig, dass alle beteiligten Akteure den Ergebnissen vertrauen können, die zugrunde liegenden Annahmen transparent dargestellt werden und dass alle Vertragspartner, möglicherweise auch weitere gesellschaftliche Akteure, Zugang zu den verwendeten Technologien haben. Im folgenden wird dabei ein Fokus auf Software als Teil von Verifikationstechnologien gelegt, für die die genannten Aspekte ebenso gelten. Existierende eingesetzte Software unterliegt oft verschiedenen Einschränkungen mit Bezug auf die genannten Anforderungen, Software ist häufig proprietär oder steht unter Exportkontrolle.

Dies kann auch an einem konkreten Beispiel deutlich gemacht werden. Für den Forschungsreaktor in Arak hat der Iran ursprünglich die Nutzung von Uran mit natürlichem Anreicherungsgrad geplant, ein solcher Reaktor weist jedoch eine besonders hohe Proliferationsgefahr auf. Im Rahmen des Joint Comprehensive Plan of Action (JCPOA) hat der Iran mit anderen Staaten unter anderem vereinbart, den Reaktor auf andere Brennstoffe umzurüsten. Typischerweise werden für Simulationsrechnungen von neuen Reaktorkonfigurationen Monte Carlo Neutronentransportprogramme verwendet, das mit Abstand gängigste Tool ist Monte Carlo N-Particle (MCNP), eine amerikanische Software. Die proprietäre und exportkontrollierte Software steht Wissenschaftlern aus dem

Iran jedoch nicht zur Verfügung. Sollte den Diskussionen auf Basis von Resultaten dieses Programmes erfolgen, erschwert dies eine internationale, offene Diskussion über die Umrüstung. Zwei weitere Beispiele werden ausführlich in der Arbeit behandelt.

Zur Lösung des beschriebenen Problems wird vorgeschlagen, dass Software, die im Kontext von nuklear Rüstungskontrolle verwendet wird, folgende drei Kriterien erfüllen sollte:

1. Freier Zugang zum Programm, ohne Einschränkungen jeglicher Art
2. Weitergabe des Programms muss vollständigen Quellcode einschließen
3. Veränderungen am Programm (und deren Weitergabe) sind erlaubt

Die genannten Kriterien sind abgeleitet von den Definitionen für Freie Software [Sta10] und Open Source Software [OSI], wie sie in den letzten zwei Jahrzehnten im Bereich allgemeiner Softwareentwicklung diskutiert wurden. Software, die in nuklearer Rüstungskontrolle eingesetzt wird, erfüllt häufig diese Kriterien nicht. Um dies zu ändern, ist es entweder notwendig, den Zugang zu Software und Quellcode zu öffnen, oder alternative Möglichkeiten zu entwickeln, sofern die erste Option nicht möglich ist.

Die Nutzung der genannten Kriterien bietet eine Reihe von Vorteilen für in der nuklearen Rüstungskontrolle eingesetzte Software. Durch die Offenlegung des Quelltextes haben alle Akteure die Möglichkeit, Software sowohl auf etwaige Betrugsroutinen, als auch auf unentdeckte Fehlfunktionen zu überprüfen. Dadurch wird das Vertrauen in die Verifikationsergebnisse erhöht. Verifikation wird durch offenen Quellcode auch transparenter - alle einer Software und den damit produzierten Resultaten zugrundeliegenden Annahmen können eingesehen und geprüft werden. Außerdem gibt es durch die genannten Kriterien für alle beteiligten Staaten und internationalen Organisationen, sowie für weitere Interessierte die Möglichkeit, Zugang zu verwendeter Software zu bekommen. Dies dient erhöhter Partizipation in wichtigen Prozessen, vereinfachter Ausbildung von neuen Expertinnen und Experten sowie einer Demokratisierung des Rüstungskontrollprozesses.

Um die Umsetzbarkeit dieses Vorhabens darzulegen, und gleichzeitig einen technischen Beitrag zu leisten, wurde die Software "Open Neutron Multiplicity Simulation" (ONMS) entwickelt, welche die Simulation von Neutronenmultiplizitätsmessungen ermöglicht. Solche Messungen erlauben es, in nicht-destruktiven Verfahren die in Proben enthaltene Menge an Plutonium zu bestimmen. Dazu wird über Koinzidenzmessungen die Rate an Spontanspaltungen in der Probe ermittelt. Die geradzahigen Plutoniumisotope haben deutlich höhere Wahrscheinlichkeiten für einen Zerfall über Spontanspaltung. Bei bekannter Isotopenzusammensetzung kann daher aus der Spontanspaltungsrate die Plutoniummasse ermittelt werden.

Für eine Simulation solcher Messungen sind eine Reihe von Schritten notwendig. Zunächst müssen die Quelldefinitionen für Proben adäquat modelliert werden. Dies beinhaltet korrekte Zerfallsraten sowie korrekte Wahrscheinlichkeitsverteilung der Zahl bei

einzelnen Spaltungen freigesetzten Neutronen. Ebenso notwendig ist eine korrekte Behandlung von (α, n) Reaktionen, die bei oxidischen Proben eine signifikante Rolle als Neutronenquelle spielen. Zur Simulation des Teilchentransports durch komplexe Detektorgeometrien ist es notwendig, eine gute Repräsentation von Neutronenwechselwirkungen mit verschiedenen Materialien zur Verfügung zu haben. Abschließend muss die Pulsfolge detektierter Neutronen durch entsprechende Methoden analysiert werden, um daraus auf die Masse rückschließen zu können.

Alle diese Funktionen wurden in der vorgestellten Software ONMS abgebildet. ONMS arbeitet auf der Basis von Geant4, einem Software Framework für Monte Carlo Teilchentransportsimulationen in unterschiedlichsten Anwendungsbereichen, insbesondere auch in der Hochenergiephysik. Es bietet aber auch die notwendigen Routinen für den Neutronentransport im Energiebereich bis zu 20 MeV, sowie speziell Neutronenstreuung in thermischen Regionen (bis rund 4 eV). ONMS ist die erste Anwendung, die die Monte Carlo Routinen von Geant4 nutzt. Fast alle anderen, existierenden Simulationsprogramme nutzen entsprechende Funktionen basierend auf der Software MCNP. Daher haben sie die schon zuvor genannten Einschränkungen mit Bezug auf Verfügbarkeit von Software und Quellcode.

Speziell für ONMS entwickelt wurde eine neue Routine für Teilchenquellen. Sie nutzt aktuelle Datensätze für Spontanspaltung, und ermöglicht die automatische Berechnung der durch (α, n) Reaktionen emittierten Neutronen basierend auf durch den Nutzer vorgegebenen Probenmaterialien. Weiterhin wurde die Analysefunktion für Neutronenpulsfolgen von Grund auf neu konzipiert, basierend auf der Methodik elektronischer Schieberegister. Die Funktion erlaubt die Berechnung der Masse sowie die Bestimmung der drei zentralen Größen Singles, Doubles und Triples, die Momente der emittierten und detektierten Multiplizitätsverteilung. Über den kompletten Entwicklungsprozess hinweg wurde darauf geachtet, dass die Software den oben genannten Kriterien entspricht. Sie ist als Open Source Software verfügbar, ebenso sind alle erforderlichen Datenquellen frei zugänglich.

Die Software wurde anhand von verschiedenen Datensätzen validiert. Zunächst wurden dafür experimentelle Messungen herangezogen, die von Malte Götsche im Frühjahr 2012 durchgeführt wurden. Diese Messungen beinhalten drei metallische und fünf oxidische Proben. Simulationen der Messungen mit ONMS zeigen in fast allen Fällen sehr gute Übereinstimmung mit gemessenen Werten, sowohl für die Momente Singles, Doubles und Triples, als auch bei detektierten Plutoniummassen.

Weiterhin wurden Daten aus einem umfassenden Satz von speziellen Vergleichsrechnungen verwendet, die als sog. "ESARDA Neutron Multiplicity Benchmark" zusammengestellt wurden. Die Daten umfassen einerseits theoretische, idealisierte Probendefinitionen mit dazugehörigen, simulierten Neutronenpulsfolgen, sowie Spezifikationen von realen Proben und gemessene Neutronenpulsfolgen. Diese Daten wurden von einer Reihe von Arbeitsgruppen analysiert, und deren Ergebnisse verglichen. Die Analysen umfassten zwei Teile, zum einen die Analyse der zur Verfügung gestellten Neutronenpulsfolgen, zum anderen die vollständige Simulation der Messung inklusive Mon-

te Carlo Teilchentransport. Beide Teile des Benchmarks wurden ebenfalls mit ONMS durchgeführt. Die Ergebnisse der Analyse der Neutronenpulsefolgen sind sehr gut, Abweichungen für einzelne Proben vom Mittelwert der Benchmarkergebnisse liegen im Bereich von wenigen Prozent. Für die vollständigen Monte Carlo Simulationen werden auch gute Ergebnisse erzielt. Insgesamt sind die Abweichungen größer, insbesondere bei Proben mit großer Plutoniummasse, jedoch im Rahmen typischer Unsicherheiten der Messmethode.

Die Validierungsrechnungen lassen den Schluss zu, dass die Software grundsätzlich funktioniert und eine mögliche Alternative zu existierenden Routinen bietet. Weitere Schritte zur kontinuierlichen Verbesserung der Software können die Nutzung anderer Wechselwirkungsdaten sowie die Implementierung von verbesserten Routinen zur Neutronenpulsfolgenanalyse, wie sie von einigen Autoren vorgeschlagen wurden, umfassen. Mit Bezug auf die anfangs genannten grundlegenden Probleme der Softwarenutzung in nuklearer Rüstungskontrolle ist in Zukunft eine vermehrte Anwendung der Open Source Kriterien sinnvoll.

Contents

Zusammenfassung	i
Acknowledgments	xi
1 Introduction	1
I Open Source meets Nuclear Arms Control	7
2 Arms Control and the Use of Software	9
2.1 History of Nuclear Arms Control	9
2.2 Software Applications in Nuclear Arms Control?	11
2.3 Three Detailed Examples of Software Use	14
2.3.1 Information Barriers	14
2.3.2 Neutron Multiplicity Counting	15
2.3.3 Plutonium Production Potential of Nuclear Reactors	16
2.4 Issues and Limitations using Software for Arms Control	17
3 The Case for Open Source Software	21
3.1 Background on Open Source Software	21
3.2 Benefits of the approach	26
3.3 Challenges and Criticism	28
3.4 Current Situation of Open Source Software in Nuclear Arms Control	30
II Simulation of Neutron Multiplicity Measurements	33
4 Neutron Multiplicity Measurements	35
4.1 Overview of the physical effects to be considered	37
4.2 Neutrons from Fission	40
4.3 Neutron Detection and Common Detectors	46
4.4 Reconstruction of the Plutonium Mass	49
4.5 Role of α Decays	55

4.6	Neutron Yield from (α , n) Reactions	56
4.7	Energy Spectrum of Produced Neutrons	58
5	Description of the Simulation Software	63
5.1	Geant4 and Neutron Particle Transport	65
5.2	Global Functions of Open Neutron Multiplicity Simulation	67
5.3	Material Decay Source	70
5.4	Pulsetrain Analysis	77
6	Validation of Open Neutron Multiplicity Simulation	85
6.1	Experimental Data - Description of Measurements	87
6.2	Validation Results and Discussion	88
6.3	Description of the ESARDA Benchmark	92
6.3.1	Samples of Phase I and II	93
6.3.2	Phase III and IV	95
6.4	ESARDA Benchmark - Validation results	99
6.4.1	Pulsetrain Analysis - Phase II and IV	99
6.4.2	Monte Carlo Simulations - Phase I and III	105
7	Conclusion and Outlook	111
III	Appendix	117
A	JENDL/AN-2005 Cross Sections	119
B	Example GDML file	123
C	Usage of Open Neutron Multiplicity Simulation	127
C.1	Command Line Options	127
C.2	Messenger Options	127
C.3	Example Macro file	130
C.4	Formats of result files produced by ONMS	131
D	Additional results for dead time calculations	133
	Bibliography	139

List of Figures

3.1	Number of authors that contributed to selected open source projects. . . .	27
4.1	Example neutron pulsetrains, taking different effects into account.	38
4.2	Multiplicity distributions of ^{240}Pu spontaneous ^{239}Pu induced fission. . . .	45
4.3	^3He neutron interaction cross sections from JEFF 3.1.	47
4.4	^1H neutron interaction cross sections from JEFF 3.1.	47
4.5	Rendering of a Geant4 model of the Active Well Coincidence Counter. . . .	49
4.6	Transformation of a neutron pulsetrain into R+A and A distributions . . .	51
5.1	Basic library structure of ONMS.	64
5.2	Energy spectrum of neutrons from (α, n) reactions in the sample “PuO2-10”. . .	77
5.3	Arbitrary pulsetrain (a) / pulsetrain after quantization (b)	79
5.4	Algorithm for simulating dead time effects on pulsetrain.	80
5.5	Standard algorithm for pulsetrain analysis.	81
6.1	Rendering of a Geant4 model of the Plutonium Scrap Multiplicity Counter. . .	87
6.2	ONMS simulation results compared to experimental data.	90
6.3	ONMS simulation results compared to experimental data, including dead time.	91
6.4	ONMS simulation results for sample plutonium masses.	92
6.5	Generic drawing of the containers for samples epIII+IV-c4, -c5 and -c6. . .	98
6.6	Pulsetrain analysis for ESARDA benchmark phase II, no dead time.	101
6.7	Pulsetrain analysis including different assumptions for dead time.	102
6.8	Pulsetrain analysis for ESARDA benchmark phase IV.	104
6.9	Results of ONMS simulations for ESARDA benchmark phase I.	106
6.10	Results of ONMS simulations for ESARDA benchmark phase III.	108
D.1	Pulsetrain analysis, ESARDA b. phase II, dead time $0.5\mu\text{s}$, non-updating. . .	134
D.2	Pulsetrain analysis, ESARDA b. phase II, dead time $0.5\mu\text{s}$, updating. . . .	135
D.3	Pulsetrain analysis, ESARDA b. phase II, dead time $2\mu\text{s}$, non-updating. . .	136
D.4	Pulsetrain analysis, ESARDA b. phase II, dead time $2\mu\text{s}$, updating.	137



List of Tables

2.1	Existing and possible tasks for software in arms control regimes.	12
3.1	Open source software examples with arms control relevance.	30
4.1	Decay half-life of selected isotopes from different references.	41
4.2	Decay branching ratio for spontaneous fission from different references. . .	42
4.3	Selected reference values for half-life / spontaneous fission branching. . .	44
4.4	Important kinetic energy levels of α -particles emitted by various isotopes.	55
5.1	Central ONMS classes and short descriptions.	67
5.2	Source events implemented in <code>ONMSMaterialDecaySource</code>	71
5.3	Parameters for pulsetrain analysis.	78
6.1	Characteristics of the samples measured by M. Göttsche.	88
6.2	Short summary of all the samples defined in the ESARDA benchmark. . . .	94
6.3	Isotopic compositions for different samples of phase III and phase IV. . . .	97
6.4	Parameters for the container of epIII+IV-c4, epIII+IV-c5 and epIII+IV-c6.	98
A.1	Isotopes included in JENDL/AN-2005.	119
C.1	Command line options for ONMS.	127
C.2	Messenger Commands	128



Acknowledgments

I have been very fortunate to be a member of the IANUS group (“Interdisziplinäre Arbeitsgruppe Naturwissenschaft, Technik und Sicherheit”) of the Technische Universität Darmstadt throughout most of my time at the university. Clearly, I learned a lot from all members of the group, in discussions, seminars and the weekly group meetings. Having the opportunity to take part in the group’s activities has proved very fruitful to sharpen my thinking about physics and political issues, science and society.

I am extremely grateful to my advisor Franz Fujara, who I consider a role model as academic researcher and teacher. This project would not have been possible without his backing from the beginning. Besides having always time to discuss my research, he gave me critical support for my teaching activities, a lecture series on Fukushima and a rather unconventional course project where students were supposed to hack measurement electronics. A special thanks goes to Matthias Englert. He was essential in helping me with my research project from the beginning. We had many interesting discussions, co-authored papers and presentations, and worked together on grant proposals which clearly clarified my ideas about this project.

Thanks also go to all of the other IANUS members I was able and glad to work with over the years, among them Wolfgang Liebert, Brigitte Schulda, Friederike Frieß, Martin Ziegler, Regina Hagen, Dirk Ipsen, Egbert Kankeleit, Christoph Pistner, Alfred Nordmann, Michael Deneke and Richard Beecroft, who as my flatmate many years ago told me first about the group. IANUS has been an absolutely unique environment of academic work, and I will miss it.

A lot of input for the technical part of this dissertation came from Malte Götsche. I have to thank him for providing experimental results, enabling contact to many other experts and continuously giving me new insights in the physics of neutron multiplicity measurements. I would also like to thank the members of the Carl-Friedrich von Weizsäcker Center for Science and Peace Research for welcoming me during many visits over the last couple of years.

Significant progress and the final steps were made during the thirteen month I spent at the Program On Science and Global Security (the Program), in summer 2014 and from October 2015 to June 2016. From the expertise of all members in the group I have benefited enormously, every day of work was an inspiration and triggered many new ideas. I am particularly grateful to Alexander Glaser. From early on he agreed to become an external advisor for this dissertation. He was crucial to allow me to come to the Program as a visiting student. With his assistance, I was able to secure funding for both visiting time periods. He showed outstanding support for all aspects of my work, which continued also during the times I have been in Germany.

I also thank all the other members of the Program. Sébastien Phillipe was not only sharing an office with me, but also made all the times in the library more pleasant, and was up for discussions in times of crisis. Finishing this work would have been impossible without Zia Mian. During the writing process, he gave me motivation and continuously made sure that I kept to my schedule and wrote page after page, reminding me to focus, especially when other things offered interesting distractions.

On a more formal note, I would like to express my gratitude to all institutions that partly funded my work: The department of physics of TU Darmstadt to fund Franz Fujara to give me a research assistant position during the first four years; the IANUS group and the Program which supported my travels to numerous conferences and meetings and allowed me to meet many interesting people in our field; the German Academic Exchange Service (DAAD) and the German Academic Scholarship Foundation who subsequently financed my research periods at the small college on the US east coast; and the mathematics department of TU Darmstadt, where I found a job at a time when other sources of income were uncertain.

While writing this dissertation, I received very helpful comments on different drafts from Stephen Croft, Malte Göttsche, Robert J. Goldston, Zia Mian, Benjamin Reimold, Ryan Snyder, and Mark Walker. I thank them all, and clearly, I am responsible for all errors and mistakes remaining.

I am deeply grateful towards my parents who made many things possible for me. They gave me all the support I needed during studies that took longer than expected, during several times of being abroad, and much more. I also thank my sister for her continuous support. I probably should also apologize for not trying to explain more about my work to them whenever I was asked.

I am delighted to end this by thanking my partner Martha Kandziora. She has agreed to be apart from me while I was abroad multiple times. I am also very grateful that after earning her doctoral degree much faster than I would ever been able to, she waited for me so that we can soon explore the world together.

1 Introduction

The central issue addressed in this thesis is the role of trust, transparency and participation in technical verification of nuclear arms control. This involves a general analysis of these topics, and the development of an open source software application to be used for verification tasks in nuclear disarmament verification and safeguarding of fissile materials.

Nearly all existing arms control regulations rely on verification mechanisms as ways to proof states' compliant behavior with international agreements. Future regulations might include the same and new verification mechanisms. Often, the task of developing and improving arms control verification is seen as a political issue to be solved primarily by diplomats, politicians and experts in international relations, but it is also a technical problem. From early on, technical aspects have been seen as central parts of approaches to control or eliminate the danger of nuclear weapons. In the Franck Report, written by members of the Manhattan project before the first nuclear weapon was dropped, the authors stated that “any international agreement on prevention of nuclear armaments must be backed by actual and efficient controls” [Fra+45].

Until today, no international treaty or codified multilateral effort is in place which would set the stage for complete, global elimination of nuclear weapons. Nine countries¹ still own an estimated number of 15350 nuclear weapons [KN16], each one of them capable of destroying a medium sized city, many of them orders of magnitude more powerful. So-called strategic nuclear weapons sit on intercontinental ballistic missiles, are ready to be loaded onto long-range bomber aircraft and cruise the oceans on submarines. Additionally, tactical nuclear weapons are stored in military bases around the world, even on foreign soil². There seems to be an imminent threat of usage and resulting disaster, caused either by intentional explosions as proposed in several current military doctrines, or by accidental detonations due to technical or human failures. At the same time, the spread of nuclear weapons to additional countries remains a risk. Nuclear proliferation in the past resulted in constant additions of countries to the pool of nuclear weapon states. More than 26 years after the end of the cold war, the often promised “peace dividend” has not payed off. And as time goes by, the group of women and men that remember living in a world without nuclear weapons dwindles, eventually all will die.

At the same time, some positive developments are underway. The “Humanitarian Initiative” can be seen as a refreshing new step forward towards prohibition of nuclear

¹ United States, Russia, France, United Kingdom, China, Israel, India, Pakistan and North Korea.

² US tactical nuclear weapons are held in Belgium, Germany, Italy, the Netherlands and Turkey under NATO's nuclear sharing policy [NK11].

weapons. In 2013 and 2014, three international meetings were held in Norway, Mexico and Austria, organized by the respective foreign ministries. Countries met to discuss effects of nuclear weapons and possible catastrophic humanitarian consequences of any use, intentional or accidental, of nuclear weapons. The discussions led to the draft of the “Humanitarian pledge for the prohibition and elimination of nuclear weapons”. It was tabled as a resolution in the United Nations General Assembly in 2015 and has received 139 votes in favor [UNG15]. The pledge includes the proposition to negotiate a treaty which would ban possession of nuclear weapons. In the May 2016 meeting of the Open Ended Working Group of the United Nations General Assembly, tasked with “Taking Forward Multilateral Nuclear Disarmament Negotiations”, a proposal to start such negotiations by the year 2017 was tabled by nine countries, all members of Nuclear Weapon Free Zones³ [Arg+16].

The goal of a world free of nuclear weapons will not be possible without comprehensive verification. Very likely the importance and necessary capabilities of verification for existing and future nuclear arms control, understood here as both the attempt to prevent proliferation and to limit and finally remove existing nuclear weapons, will continue to grow. Such an increased need for verification technologies also increases the demand for new technological developments. Three particular aspects of verification are often neglected or overlooked by researchers working on verification technologies: First the role of trust in relation to the used technology, second the question of transparency of the used tools and lastly the question of accessibility of technology to all parties involved and a broader audience.

These issues are addressed in this thesis, with a general focus on software that can underpin verification instruments and methods and allows for simulations of physical effects. Used as part of verification technologies for international treaties, it is clear that software results often form an important basis for decision making. Increased software use also goes along with the continuous increase in information technologies. Software plays important roles in different phases of the life of verification technologies. It is used in the beginning as part of the design and testing process of measurement devices. After measurements are taken, software is used for data analysis and distribution. Independent from actual measurements, software can play a role in simulating future developments or events that should be covered by verification but where actual real tests are undesirable, as for example a nuclear weapons test explosion.

To increase trust, transparency and participation in nuclear arms control, employed software should fulfill criteria similar to those definitions proposed by the movements for Open Source Software and Free Software. Software should be distributed including source code, to allow other users to verify its functionality. There should be no restrictions on access to software used for arms control verification. Many applications currently used in nuclear arms control do not fulfill the listed criteria, in contrast they

³ Nuclear Weapon Free Zones are formed by groups of countries and ban nuclear weapons in certain areas. The proposal was tabled by Argentina, Brazil, Costa Rica, Ecuador, Guatemala, Indonesia, Malaysia, Mexico and Zambia.

are often characterized by strict access controls (e.g. export limitations) and are mostly of proprietary nature. Changing these characteristics would contribute to removing secrecy, mistrust, lack of transparency and exclusive arrangements. The open source community, which developed alongside the above mentioned movements, is an example for the successful integration of knowledge sharing in a different area. Transferring some of the achievements and benefits from the open source community to nuclear arms control would be beneficial.

Looking at recent literature, this work is neither the first to publish open source software that is or could be used for arms control applications (for example OpenMC, [RF13]), nor the first to emphasize the benefits (for example [Whi01]). However, to my knowledge, it is the first broader study based on both discussions of the open source issue, as well as providing a piece of software which has been developed from the ground up using an open source philosophy.

The software developed is a new application to carry out simulations of neutron multiplicity measurements, “Open Neutron Multiplicity Simulation” (ONMS). It is the first of its kind to be available as open source software and the first to use Geant4 as a backend for the required Monte Carlo particle transport. Most, probably all, of the existing tools rely on the export-controlled and proprietary tool Monte Carlo N-Particle Code (MCNP) for the Monte Carlo particle transport. During the development of ONMS it was ensured that necessary components, including the required nuclear data sets, were chosen from openly available sources. Beyond obvious benefits of the open source standard, the software also implements several additional features, especially with regard to the particle source for simulations.

Using neutron multiplicity measurements can reveal the mass of plutonium in a given sample. Such methods could be used during several arms control verification tasks. Safeguards preventing nuclear non-proliferation can use the measurements to provide material accountability for fissile materials. During future disarmament efforts, the same method could be utilized to help authenticating an object to be a nuclear warhead. A commonly discussed method to do this is the “Attribute Approach”, where several different attributes of a warhead would be defined, one of which very likely would be an item’s plutonium mass. Simulations of neutron multiplicity measurements help developing new detectors and allow to simplify checks for possible ways of cheating measurements. As stated above, no open source software for such a task is currently available. In addition, availability for existing tools is limited. In one of the descriptions of the benchmark calculations used to validate ONMS, the authors writes that “few codes are available today [...] and they are not available to everybody” [PS06, p. 2].

The dissertation is organized in two parts. The first part addresses the issue from a policy perspective, and is split in chapter 2 and 3. Chapter 2, “Arms Control and the Use of Software” describes the use of software as part of arms control verification in detail, categorizes applications and illustrates possible issues and limitations based on three examples. In chapter 3, “The Case for Open Source Software”, the open source approach

is presented. It is explained how it could be employed for arms control verification and what the benefits and challenges would be.

The second part covers all details related to the development and testing of Open Neutron Multiplicity Simulation. It starts with chapter 4, “Neutron Multiplicity Measurements,” which describes the physics of neutron multiplicity measurements. The chapter includes the steps of the calculation required to retrieve the plutonium mass from a measurement and detailed discussions on the role of (α, n) reactions for neutron multiplicity measurements. Chapter 5, “Description of the Simulation Software”, focuses on the software itself and the implementation of physical processes and models. The last chapter of the second part, chapter 6, “Validation of Open Neutron Multiplicity Simulation”, discusses testing and validation of the code. It is based both on experimental results and a large benchmark exercise for neutron multiplicity simulations. The dissertation ends with concluding remarks, which also give an outlook about possible subsequent research.

From a broader, historical perspective, the thoughts of this thesis go in line with ideas that have been raised seven decades ago. It is important to strengthen international agreements through technical verification, as first outlined in the Franck report mentioned above. Sharing of knowledge is a fundamental idea of the open source approach, a similar idea can be traced back in the early history of nuclear weapons, too. The “Report on the International Control of Atomic Energy”, known as the Acheson-Lilienthal Plan, concludes with the following sentences: “When the plan is in full operation, there will no longer be secrets about atomic energy. We believe that this is the firmest basis of security; for in the long term there can be no international control and no international cooperation which does not presuppose an international community of knowledge” [Lil+46, p. 60]. Applying open source criteria to software (and possibly other means of verification) would be a step towards supporting an international community of knowledge on nuclear arms control.

The motivation to contribute as a scientist to a field of public interest has been motivated by Bertrand Russels ideas of the responsibility of scientists and their moral obligations as citizens [Rus60], and the relationship between science, society and atomic energy as voiced by Albert Einstein: “We scientists recognize our inescapable responsibility to carry to our fellow citizens an understanding of the simple facts of atomic energy and its implications for society. In this lies our only security and our only hope – we believe that an informed citizenry will act for life and not for death” [Ein46]. The work for this dissertation has been carried out mostly as part of the Interdisciplinary Research Group on Science, Technology and Security (IANUS) at the Technische Universität Darmstadt and partly in the Program for Science and Global Security at Princeton University. Both groups have a longstanding tradition of contributing to public policy issues by doing scientific research, not only limited to the field of nuclear arms control. The idea of helping with scientific and technical work to eliminate nuclear weapons was part of the overall motivation for the research project described in this thesis. Besides scientific challenges, there was an ongoing struggle to please scientists and policy experts alike, fulfilling different requirements of the different peer groups and scientific communities.

I hope that this work is able to meet these standards. At the same time, I hope that some of the findings and results can contribute to nuclear disarmament and a safer world in the future.



Part I.

**Open Source meets
Nuclear Arms Control**



2 Arms Control and the Use of Software

The goal of this chapter is to outline the role of software in nuclear arms control, and to characterize problems and challenges. At first, a short, general history of arms control is presented, which could be skipped by readers familiar with the topic. The section is followed by a broad overview of actual use cases of software in arms control, sorted by different classes and categories. Then, three more detailed examples shed additional light on the previously defined categories. Beginning with these examples, several issues and limitations related to the use of software in arms control are identified. The list of issues and limitations will be expanded in a general way at the end of this chapter.

2.1 History of Nuclear Arms Control

Ideas on how to limit the use of nuclear weapons and their spread have been raised for a long time. Already during the Manhattan Project, several scientists realized what radically changing effects nuclear weapons would have on warfare and the international community. It is very remarkable that important points, which are valid until today, were raised at a time before the first nuclear weapon was tested and before they were actually used in war. The scientists discussed among their fellows, and addressed high level politicians to possibly stop the use on the Japanese cities of Nagasaki and Hiroshima. Leo Szilard wrote internally on this issue and made attempts to contact presidents Roosevelt and Truman. Niels Bohr proposed the idea of negotiating an agreement to share control of nuclear weapons among the United States, United Kingdom and Russia to Churchill and Roosevelt. In June 1945, James Franck chaired a committee on social and political implications of atomic energy which produced the so-called Franck report. The report predicted a nuclear arms race, and also argued against the use of weapons on Japanese cities [Fra+45; Wit09]. It also discussed the possibility of international arms control, including the necessity of technical verification methods.

After the war, the ideas on international control were followed up. Based on the Acheson-Lilienthal report [Lil+46], the United States introduced the Baruch Plan in 1946 at the newly founded United Nations, which proposed international cooperation for nuclear energy and the elimination of nuclear weapons from arsenals [FGL02, p. 504]. The first resolution ever passed by the United Nations General Assembly called for the elimination of nuclear weapons of national arsenals [UNG46]. Contrary to these and other attempts, no agreement was reached, nuclear weapons remained in

US arsenals, and in the seven decades since, eight additional states acquired their own weapons.

In parallel to proliferation, efforts to regulate nuclear weapons were continued, some of them successful. Until today, a complex international regime for nuclear arms control has developed. Often considered “the cornerstone” of this regime, the Nuclear Non-Proliferation Treaty (NPT) entered into force in 1970 [UNO12; UNO15]. It defines two categories of treaty members: States who tested a nuclear weapon before January 1st, 1967 are “Nuclear Weapon States”, other member states are “Non-Nuclear Weapon States”. The latter are not allowed to acquire nuclear weapons, while the first have an obligation to disarm their arsenals over time. States of both categories should cooperate on the civilian use of nuclear energy. Testing of nuclear weapons has been regulated since 1963 on the surface, in the atmosphere and underwater by the Limited Test-Ban Treaty. In 1995 the Comprehensive Test-Ban Treaty was concluded, banning underground tests as well. Lacking some states’ ratification, however, it has not yet entered into force. The nuclear arms control regime further includes bilateral measures by the Soviet Union and the United States to limit their arsenals, most importantly the “New START” treaty which regulates strategic arsenals and the Intermediate-Range Nuclear Forces Treaty (“INF”) which bans missiles. Many states also formed Nuclear Weapon Free-Zone’ based on multilateral treaties. Today, such zones cover the whole southern hemisphere.

Nevertheless, the most important component of the regime is missing: A clear way forward towards a world free of nuclear weapons. A first step would be the regulation of fissile materials used in weapons. The so-called Fissile Material Cut-Off Treaty, which would ban the production of such materials, has been on the agenda for international negotiations for more than two decades. Substantive negotiations have not started due to a stalemate in the Conference on Disarmament, the assigned negotiating arena. In a more comprehensive version, such a treaty might also ban possession of fissile materials in general [Fei+14, 143ff.]. Regarding the weapons itself, no international agreement has been reached, or even put forward for negotiations. The NPT includes a general provision for disarmament, but does not prohibit possession. Such a prohibition including provisions for phases towards that goal would be included in a “Nuclear Weapons Convention”, which has been proposed by several NGOs as a model treaty text similar to the Biological Weapons Convention and the Chemical Weapons Convention [III07].

In recent years, many states met to discuss the humanitarian impact of the use of nuclear weapons in conferences in Norway, Mexico and Austria. These meetings resulted in the “Humanitarian Pledge”, calling for a rather simple treaty banning all weapons [UNG15]. This would be a different approach compared to a comprehensive convention, as concrete steps to disarm would be negotiated only after such a “ban treaty” entered into force. The treaty itself would focus on setting the normative goal. A proposal to start such negotiations by the year 2017 was tabled by nine countries of different Nuclear Weapon Free Zones in an international meeting in 2016 [Arg+16]. Even given that proposal, numerous calls for a Nuclear Weapons Convention, and endorsements of the

Humanitarian Pledge - there is much more work to be done to achieve the goal of a world free of nuclear weapons.

Looking back in history, while ideas on arms control started among scientists, the issue quickly shifted, probably exclusively, to the field of international relations and foreign policy, dominated and carried out by state leaders, diplomats and other political personnel. The general public has had some influence on the developments, especially during times of strong peace movement activities. Scientists, science, engineers and technology today mainly enter the picture with relation to verification and monitoring of treaty obligations. Most successful international treaties rely on such verification mechanisms. The most obvious reason for verification is to detect if states are failing their obligations or trying to cheat. But there are other functions: Verification and monitoring can create and preserve trust and confidence that every member is complying. Showing ways to ensure compliance, verification also can help building support for treaties and motivate states to join a treaty. It is also important for the scope of treaties - in most cases they include only provisions that are deemed verifiable (cf. for example [Gay86]). Serious negotiations on a Comprehensive Test Ban Treaty only started after viable verification means had been demonstrated. Experts and scientists played an important role in developing these tools, and also formed the “Group of Scientific Experts” to help negotiations with technical advice [Dah13].

Early ways to carry out verification originated in the military. It often consisted of so-called “National Technical Means”, where states would use their own military and civilian intelligence assets to observe other countries actions. On-site inspections, investigations on other countries soil were added over time [Gay86]. In parallel, international organizations were founded that would be tasked with verification, for example the International Atomic Energy Agency and the Preparatory Committee for the Comprehensive Test Ban Treaty Organization.

2.2 Software Applications in Nuclear Arms Control?

The importance of software use in arms control applications goes beyond common reliance on software in a computerized world. Of course, there are many applications that use general purpose software, standard software that an organization inevitably will be using, such as word processing tools, web browser, means for email and other communication or simply operating systems. More relevant, though, are applications that are more closely involved with treaty verification. Technical verification consists of complex measurement systems and networks, and is carried out by groups of specially trained scientists. Similar to many other fields, the use and reliance on software constantly increases with new emerging information technologies. A significant part of the work is carried out using software, sets of instructions and algorithms to be executed by computing hardware like personal computers or signal processing units. For many tasks, there is no way to avoid using software. Software programs are also often designed specifically for a particular task and implement representations of particular

Table 2.1.: Existing and possible tasks fulfilled by software in current and future arms control regimes, ordered by treaty regime and main use of software.

	Non-Proliferation / Safeguards	Comprehensive Test-Ban	Fissile Material Cutoff Treaty	Nuclear Disarmament Agreement
Particle Transport	Development of non-destructive assay methods for fresh/spent fuel analysis		Development of non-destructive assay methods for material analysis	Development of non-destructive assay methods for warhead authentication
Criticality Calculation	Generate neutron flux-spectra for depletion calculation		Generate neutron flux-spectra for depletion calculation, nuclear archaeology	
Depletion Calculations	Proliferation potential of reactors	Estimate fission product release	Estimate past/current fissile material production capabilities	Fission product tagging for warhead identification
Spectrum Analysis	Identify items (spent/fresh fuel), determine material compositions	Detection of radionuclides	Identify items (spent/fresh fuel), determine material compositions	Identify items (warheads) and respective material compositions
Correlation Analysis	Neutron correlation / multiplicity measurements for plutonium mass estimates		Neutron correlation / multiplicity measurements for plutonium mass estimates	Neutron correlation / multiplicity measurements for plutonium mass estimates
Atmospheric Transport Modeling	Kr-85 detection (clandestine and declared reprocessing)	Radionuclide detection	Kr-85 detection (clandestine and declared reprocessing)	
Fuel Cycle Simulation	Material balancing & accounting		Material balancing & accounting, past fissile material production	
Fluid dynamics	Isotope separation modeling	Enrichment cascade analysis	Nuclear archaeology	
Waveform Analysis		Discriminate explosion / earthquake		
Image Identification	Find clandestine facilities using satellite imagery		Find clandestine facilities using satellite imagery	
Image Change Detection	Find clandestine facilities, track operational status of existing ones	Detect crater / sinkings after explosions	Check operational status of facilities	Chain-of-custody for nuclear warheads
Geographic Information System	Combine data from different sources	Reconstruct possible locations of explosions		
Virtual Reality	Inspector training, data visualization	Training for on-site inspections	“Virtual” archaeology	Improvement of verification process
Encryption	Data transmission and authentication	Data transmission and authentication	Data transmission and authentication	Data transmission and authentication
Other Cryptographic Methods			Hashed declarations	Hashed declarations

physical or technical contexts. As states rely on arms control verification, and therefore the software involved, the role of software should be carefully analyzed.

To emphasize the importance of software in arms control, this section provides a comprehensive overview over the different use cases. There are different ways to group and categorize such applications. Table 2.1 lists many of the common tasks dealt with by software in nuclear arms control. The tasks are sorted according to the possible treaties they would support and the physical or technical function they fulfill. The first column considers the existing methods to control nuclear proliferation, including the NPT and the related safeguards regime. The next column lists applications used to verify the absence of nuclear tests. The last two columns go beyond existing agreements and obligations. While they give an overview over proposed technologies, they are also partly speculative. As long as no international agreement is reached, the future selection of related verification technologies remains an open question. Similarly, new technological developments in the future might extend the potential for new software, making agreements more likely to be negotiated.

Beyond the ordering scheme selected for the table, software packages can be sorted into categories based on the purpose they have related to verification. It is useful to define five different classes, which partly overlap. In the first three of them, applications are included that deal with data from real, physically existing, measurements. The last two classes involve various ways of simulations.

Data Processing: Software is used to process data either in real time (during measurements), or some later time after the data have been acquired. Such processing happens in close relation to the physical quantity measured, e.g. discriminating measured signals from background, reducing noise in signals.

Data Transmission and Display: After measurements, data acquired by technical equipment and human inspectors have to be distributed to international organizations and member states. The second category of software includes applications that can be used for this task. Such software is responsible of ensuring completeness and authenticity of the data transmitted, and appropriate ways to display data.

Data Analysis: Depending on the question asked, measured and processed data have to undergo subsequent data analysis, combining the measurements with data sets from other sources.

Simulation of Measurement Systems: It is often cheaper to run several simulations instead of testing hardware during the development of verification technologies. Simulations make it possible to run training exercises and analyze events that are very undesirable in reality, for example a test of a nuclear weapon. Lastly, simulations also provide for more comprehensive tests of resilience against cheating.

Technology Assessment: This category covers a wider range of tools than the other four. Beyond explicit verification tasks, software is used to analyze risk and possibilities of current and future technological developments. It can be used to understand how an emerging reactor system performs with regard to possible proliferation

risks, or how technological developments could allow for completely new verification approaches. Such an analysis is often an important input for a state's or an international organization's policy. It enables and helps diplomacy and treaty negotiations, and probably shows new ways for peaceful developments.

2.3 Three Detailed Examples of Software Use

2.3.1 Information Barriers

In many practical cases, verification technologies will measure or produce information that can not be revealed due to treaty reasons or due to an inspected states' national security concerns. To overcome such problems, the development and construction of so-called information barriers has been proposed. They will be mostly used for war-head verification applications, but other uses are also possible. The information to be protected could be of numerous types, among them gamma spectra, x-ray images or the spatial distribution of neutron count rates. To protect the information, information barriers turn complex data input into binary outputs, e.g. a green and a red light. Both actors involved in a verification, the inspected party and the inspecting party, have to trust the information barrier. On the one hand, it is important that the information barrier only shows a green light if the object under inspection really is what it is supposed to be. This includes the necessity of high confidence that no cheating could be done by the inspected party. On the other hand, the inspected party wants to have high confidence that no additional information beyond red or green light is released to the inspectors.

Important developments with regard to information barriers were carried out during the Trilateral Initiative, in which the Soviet Union, the United States and the International Atomic Energy Agency tried to develop ways to verify classified forms of fissile material [Int97]. During this initiative, different groups worked on information barriers, making a list of general requirements [Ful99] as well as discussing issues related to the authentication of software used [Wol+01]. Further research on information barriers was made as part of an initiative between the United Kingdom and Norway [Cha+10; All+13]. This project recently published some of the source code of the developed software [UKN15].

An information barrier has a high reliance on software, mainly in the areas of data processing, data transmission and analysis. Software typically is developed for particular cases, since off the shelf applications rarely exist. All past projects struggled to create trusted devices, even if they incorporated joint software development among inspected and inspecting party. Mistrust among an inspected state and an inspecting state is also influenced by the hardware used, and it is necessary to avoid any manipulation in this area. Cheating bears high cost for at least one and possibly both sides involved. If for example a nuclear weapon state would manipulate an information barrier, the verification could reveal higher numbers of weapons than actually dismantled. The weapon

state would be able to keep some weapons beyond their declaration and gain a strategic advantage, because the other side would believe that they are dismantled.

Some of the issues of information barriers were explored in a course taught in 2015 at Technische Universität Darmstadt, where students were tasked to cheat a given information barrier system. The course has been summarized together with a course taught in parallel in Princeton, New Jersey, where groups of students built information barriers [KPG15]. The experiences of both courses led to the attempt to build a new, partly open source, information barrier, results of which will be published in [KGG16].

2.3.2 Neutron Multiplicity Counting

Neutron Multiplicity Counting is a measurement technique for non-destructive assay of a material. In a verification exercise, it is a way to estimate the mass of plutonium in a complex sample. Such measurements could be applied for warhead verification, and to ensure non-proliferation of states using peaceful nuclear power by inspecting plutonium in fresh or spent fuel, or separated plutonium stockpiles. While such a system would use software for data processing, the main application of software here is the simulation of the measurement system. This has to be done during the development of respective detector systems, and for the evaluation of possible ways of physically cheating such systems. It is faster and cheaper to simulate different kinds of cheating attempts and assess a verification system's outcome than to build a physical system and investigate all its vulnerabilities.

To simulate neutron multiplicity counting systems, Monte Carlo particle transport routines are used to track neutrons from a simulated source through a detector. In a next step, the simulated detector signals are analyzed with specific post-processing codes. A commonly used tool that includes functionality for both steps is MCNPX-PoliMi [PPM03; Poz+12]. This software package is a special version of the more general Monte Carlo particle transport code Monte Carlo N-Particle (MCNP) [Goo+13]⁴. Development of both codes mainly took place in the United States and is subject to export controls which limit distribution. These codes are proprietary tools, although under certain circumstances it is possible to request access to source code. While it is more important that states trust the actual hardware than the software used to build it, limited access might create a lack of transparency among actors during the design phase of a detector. Transparent, detailed simulations would allow open discussions among all parties to prevent possible cheating and prevent misuse. Also, limited access to simulation tools limits the number of experts involved to a small community - thus possibly constraining good ideas for detector improvements, too, and reducing opportunities for capacity building.

⁴ Until recently, two different codes existed, MCNP and MCNPX. With version 6 of MCNP, they have been merged to form a single application.

2.3.3 Plutonium Production Potential of Nuclear Reactors

This example shows how software is used to assess a future technology. For about a decade, Iran has been building the IR-40, an indigenous research reactor in Arak, the IR-40. Its stated purpose is a research reactor and the production of medical isotopes. The reactor's original design was also well suited for weapon grade plutonium production. It featured the use of natural uranium as fuel and heavy water as moderator and coolant. Similar designs have been used by several other countries for weapon programs (e.g. the Dimona reactor in Israel, the CIRUS reactor in India or the Khushab reactors in Pakistan). Therefore, the Iranian plans raised suspicions in the international community. Several proposals to modify the reactor making it less proliferation-prone were made in the past. They typically involved some of the following options: Increasing the uranium enrichment, changing the moderator material and / or reducing the core size (among others [WFK12], [AG15]).

Modification of the reactor towards a design less prone to proliferation was agreed upon as part of the Joint Comprehensive Plan of Action (JCPOA), commonly known as "Iran Deal", which was concluded in 2015. Several initial design parameters are included in the JCPOA, but further design work has to be carried out to make it a working reactor. Final designs have to be approved by all states involved in the JCPOA[Cou16].

To evaluate possible changes to the design prior to construction, simulation software has to be used to calculate reactor parameters (e.g. criticality, various safety coefficients) and the change in inventory (e.g. plutonium production). Reactor parameters can be calculated using Monte Carlo particle transport routines. However, the most common used software is MCNP, whose issues and limitations have been described in the previous subsection. For the change in inventory, results of the Monte Carlo particle transport are coupled with reactor burnup models. There is a large variety of software for burnup calculations, typically export controlled in the country of origin. None of them is publicly available, as all major codes are available via code distribution centers⁵, which ensure that export control procedures are followed. While it is possible to gain access to the source code of some programs, most are proprietary developments. Depending on the states involved, tools to be used for the Arak reactor conversion might not be shared with Iranian scientists. Therefore, even if results derived of software inaccessible to the other actor might be shared, the question arises on how to effectively collaborate on a design among several states.

Many of past works carried out in the IANUS group have been using this type of software, and the issue of plutonium production in nuclear reactors. With MCMATH, a locally developed depletion code exists[Pis06; Küt11]. It was used for many different applications, among them the calculation of the plutonium production potential of fusion reactors and spallation neutron sources [Eng09] or the role of fast reactors

⁵ Radiation Safety Information Computational Center (RSICC) at Oak Ridge National Laboratory distributes software on the American continent, while the Nuclear Energy Agency of the Organisation for Economic Co-operation and Development (OECD/NEA) distributes software in Europe.

for the US-Russian Plutonium Management and Disposition Agreement [KGE14]. MC-MATH has been developed since more than ten years, using the proprietary software Mathematica and MCNP, hence it suffers similar limitations as previously described.

2.4 Issues and Limitations using Software for Arms Control

In nuclear arms control, different use cases for software exist. Several of them are very particular and custom tailored for specific tasks. The use of technology influences how treaties are negotiated and verified, how states might shape their nuclear ambitions in the future, and even how states decide on sanctions and other measures, possibly war. However, as already the three detailed examples showed in part, there are several issues that affect to the use of software in arms control. These issues, and also limitations on existing software and software use in general will be outlined in the following. With an increased use and reliance on software, the impact of issues and limitations will become stronger.

Independent of the use in arms control verification, software has to deliver valid and correct results. The physical models that are implemented in data processing tools or simulations have to be accurate and optimized for the particular questions they answer. This requirement is most likely not an issue for the specially made software examples listed above, nevertheless important.

Particularly for the use in arms control verification, software results have to be trusted. This issue is probably the most important. All actors involved in a verification exercise or the assessment of a new technology have to be convinced that the results revealed actually represent a given or future physical reality. Without trust of states and other actors in the correctness of software, there is basically no value in the verification results or the findings of technology assessment at all. There has to be confidence that software was not manipulated intentionally to achieve a certain outcome and that no one tried to cheat. Unintentional malfunctions of code or simple software bugs can also undermine trust.

For software used in arms control, it is important that the ways and means to come to results are as transparent as possible. This ensure the ability to reproduce the results in the future by interested parties. If the software used has been proprietary or has limited access, a future reproduction of results could become impossible if the vendor or provider of the software is not around anymore. One can draw a parallel between scientific experiments and verification challenges. Scientific results are considered valid only if they can be reproduced independently by a third party. The same is required for results of arms control verification - if they can not be independently examined, they might not be accepted. In the example of the Arak Reactor conversion, the importance of such transparency becomes very clear. Both the Iranians and the other parties in the agreement have an interest in reproducible models of reactor conversion proposals,

before they will agree on a particular design and build it. With export-controlled or proprietary tools, it is very difficult to achieve transparency levels that would be acceptable to both parties.

Software tools have to be available to all parties involved in an agreement. Clearly it is important that every state party to a treaty should be able to participate in verification, and have access to all the required tools including software. In a broader context, also participation of other actors might be preferential, including international organizations but also society at large. Large participation could significantly reduce errors made, be a way to employ societal verification for many tasks and introduce democratic means to nuclear arms control in general.

Regarding these issues, the majority of the currently used software has at least some, if not all, of the following three limitations. First, most software tools available for the applications mentioned in table 2.1 are proprietary. This results in distribution of only the binary form of an application, even if that is done sometimes free of cost. Binary software is executable on a computer and can yield results, but it is not easily possible for a human to understand the underlying algorithm. While there has been no proven case of this occurring, this method of distribution could be a way to cheat on states and actors using clandestine methods, which would for example hide a specific measurement result. Without the so-called source code – the human readable “recipe” of a binary application – it is difficult to detect such modifications, and also harder to remove bugs and improve software.

Second, some applications are under strict export controls in several countries, often due to possible dual-use capabilities. While it is claimed that the control is beneficial to prevent proliferation, it could have negative implications for verification. Whenever software is used in a verification setting, such a limitation can have negative effects as not all parties involved might be able to access the same software. Sometimes, access is limited not only due to export controls, but because of other reasons to a small group of people, like the members of an organization. Reasons for such a limitation might be patent or copyright issues, or just a reluctance to make a tool publicly available.

As a last limitation, the cost of software might be an issue. In several cases, especially for software that also has commercial application outside of nuclear arms control, the software comes at relatively high costs. These can range from several \$100 up to prices in the order of \$10000. Although these are not unusually high numbers compared to other necessary items in the context of nuclear arms control, they still could prevent certain actors from getting involved, e.g. developing countries or non-governmental actors.

All the listed issues and limitations have typically not been addressed, but definitely have an impact on current agreements and might limit future verification efforts. The proprietary and export control aspects of many used tools limit the access to software and can reduce trust among partners. It is difficult to imagine how a “global community of knowledge”, as proposed in the Acheson-Lilienthal report [Lil+46], could be

achieved without changing that situation. The next chapter proposes how this community could be formed using open source approaches.



3 The Case for Open Source Software

Software used in arms control would benefit greatly if the main paradigm of development and distribution of software would change more towards an approach often referred to as “Open Source Software”. For software with arms control purpose that would require software to fulfill the following three criteria:

1. No restrictions for access to the program.
2. Distribution of the program must include the full source code.
3. Modifications of the program are allowed to anybody.

The criteria are derived from standard definitions of Free Software [SLG02] and Open Source Software [OSI]. To argue for a broader application of the approach, this chapter will discuss these criteria in detail, starting with a first section on general history of open source software, explaining the criteria’s origin. The idea of such criteria seems to be contrary to many current approaches in the world of nuclear weapons, which are often characterized by secrecy, compartmentalization and the urge to keep things as private as possible. Open source software in contrast supports traditions of sharing, collaboration and open exchange. In a second section, benefits based on these differences will be discussed with regard to the issues listed in the previous chapter. After that, a section tackles aspects often raised as a critique of open source approaches. In a fourth section, a brief overview of the current situation of open source software in nuclear arms control will be given.

3.1 Background on Open Source Software

Codified definitions of open source software similar to the three criteria defined above are relatively new. However, many of the ideas behind open source software have been prevalent throughout the history of computing. With early mainframe computers, software often was provided for free bundled with hardware, the latter being very expensive. New tools and improvements of existing ones were often shared among users and developers, saving time and money. Sharing programs was a defining norm for the forming culture of “hackers”, for example at the Artificial Intelligence Laboratory / Massachusetts Institute of Technology. The researchers (or hackers) had a set of shared values, which later Stephen Levy phrased as “Hacker Ethics” [Lev75]. An earlier example of sharing is the “Project for Advancement of Coding Techniques” (PACT), established in 1953 by the competing companies Lockheed, Douglas, RAND and other defense contractors. The companies realized that they could improve their work faster if

they shared common routines and tools used in computing, instead of having to develop everything from scratch [Web04, p. 22].

While the community of computer users and the number of machines was small, sharing was common practice. In the 1980's, Personal Computers (PC) became common, and allowed for a significant increase in computing infrastructure and usage. They also made it possible to have computers in private homes. Together with the further spread of the hardware, serious commodification of many software tools began. Companies started selling software, and hence saw unlimited sharing as a threat to their business model. A very famous example is a letter by young Bill Gates when he realized that many people were copying his program without paying. The program of concern was an interpreter for the programming language BASIC, which Gates offered for sale. In a letter from 1976, entitled "Open Letter to the Hobbyists", he complained about the loss in income due to copying. According to the letter, such a lack in income would prevent quality software from being written. A couple of years later, in 1983, AT&T decided to commercialize its operating system, Unix, developed by the Bell Laboratories of AT&T. Previous versions had been shared relatively widely, and often for free⁶. Generally, in the 1990's most commonly used software was sold as proprietary product and not shared freely. Many members of the industry agreed with Gates' letter, seeing the tradition of sharing as a threat to the motivation to produce good programs. Many of the companies whose main business is software were founded at that time. Out of the top ten companies listed in the 2015 Forbes "The World's Biggest Public Companies" ranking under the category "Software / Programming", seven were founded between 1970 and 1985 [For15].

Around the same time, activities working into the opposite direction started as well. Beyond the original behavior of sharing, attempts were made to codify rules to protect and promote free and open source software. A key actor was Richard Stallman, a hacker at the MIT Artificial Intelligence Laboratory. In 1983, he proposed the "GNU's Not Unix" (GNU) project, a free unix-like operating system, and a year after he left MIT to focus on writing free software. In 1985, he founded the "Free Software Foundation" (FSF) as an organization to promote his ideas of free software. He published the GNU Manifesto based on the statement: "If I like a program I must share it with other people who like it" [Sta10, p. 27]. The manifesto also defines "Free Software", based on the following four freedoms to the user of a program:

Freedom 0: The freedom to run the program, for any purpose.

Freedom 1: The freedom to study how the program works, and adapt it to your needs. [...]

Freedom 2: The freedom to redistribute copies so you can help your neighbor.

Freedom 3: The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. [...]

[Sta10, p. 3]

⁶ AT&T's main reason to share earlier versions without charge was to stay in compliance with court rulings regarding its monopoly on the US phone market [Web04].

The criteria defined at the beginning of this chapter are partly based on these freedoms. To ensure freedom 1 and 3, access to source code of software is required. If those would be directly transferred to software for arms control purposes, mainly freedom 1 is crucial. Studying the function of a program is what parties involved in a verification would need to do to figure out if they could trust the software. Freedom 2 and 3 ensure the inclusiveness and participation of many in verification activities. Freedom 0, however, needs to be treated with care. It is not included specifically in the above listed criteria for open source arms control software. Sometimes, the tools used for verification purposes have so-called dual-use properties, making them also usable for weapons development. Clearly, such use would be in contradiction with the goal of a nuclear weapon free world.

The FSF developed the GNU Public License (GPL), a general copyright license which can be used by any software developer to ensure these four freedoms. While the GNU project did release many useful tools, it did not provide the central part of an operating system - a kernel. In 1991, Linus Torvalds created the first version of Linux, a Unix-like operating system kernel, which filled this gap. Torvalds shared the source code with others and asked for contributions. Licensing for the Linux kernel was soon changed to use the GPL. In a sense it replaced the missing operating system part of the GNU project. Today, Linux is the basic operating system for numerous applications, running everything from large servers down to cell phones and small devices as internet routers or TV set-top boxes. The way the Linux kernel was developed is also a major example for the process of open source development. Beyond sharing of software as in earlier days, it is structured in a way to allow many developers to collaboratively work on a very complex tool, often on a voluntary basis. Another example for successful open source software is the Apache web server, which today is one of the most important programs to deliver websites to users.

While many shared basic ideas promoted by the Free Software Foundation, major critique addressed the “viral” property of the GPL. The license requires that the program is only used together with other programs using similarly free licenses, possibly “freeing” other programs with more restrictive copyright requirements by requiring their users to switch the license. According to the GPL, it is not possible to use an application or even parts of it in a new developed proprietary tool. This has been seen as a problem. In 1998, several important actors of the free software movement met and wrote a new definition for free software, and created a new brand: “Open Source Software”⁷. To support the brand and the definition, the Open Source Initiative was founded [OSI; FF01]. The definition of open source software specified ten criteria for software to be considered open source and was based on Bruce Perens’ “Debian Free Software Guidelines”. The criteria include free access to software, guaranteed access to source code and the right to modifications. It explicitly states that the licenses for open source

⁷ Throughout this work, free software and open source software are used interchangeably. While there is a personal preference of the author for the first term, neither side of the debate related to moral issues of both terms seems to be better or worse suited for the application of related criteria to arms control applications.

software should not discriminate persons or groups and also put no restrictions on the fields of endeavor. The open source definition was seen by many to be more attractive for businesses compared to the more political ideas of the FSF [FF01].

Beyond the historical development, some particular aspects of the process of developing and maintaining open source software are relevant for the following analysis. Often, references to open source software actually go beyond properties of applications and programs and include procedural aspects in the description, talking about an “open source process” or even an “open source movement”. Very often people unfamiliar with the open source process ask: How can it happen that open source software is created? As there are no direct economic benefits, because you can not exclusively sell open source software as a product, what gives incentives to produce such software?

Typical motivations listed include mainly four points (cf. for example [BR03; FF01; HK03; Fel+05]): First, developers engage into software development because of the intrinsic utility of a program for themselves, because they need something that could solve a specific problem. Second, there can be an intrinsic motivation of people. They develop software knowing that it might be shared later and benefit other people. The third point can help increase this intrinsic motivation - as part of sometimes very large, typically international projects, developers have the sense of being a member of a community. Lastly, open source development can also have economic motivations. Although there is no immediate return, there are often delayed benefits. Many developers have received future job opportunities through publicly posted work. Publishing code as open source software shows the skills and talent of a programmer to possible employers. Clearly, not all of these motivations apply to everyone. Sometimes, it is also just part of a persons job to develop open source software, depending on the policy of the employer.

The open source characteristic is not only a constant status, but also has procedural aspects, addressing a very common problem of software development. Because of the complexity of programs and the inherent need to keep an overall design structure, the software development process can not be accelerated by just adding more developers. This has been described already in 1975 by Frederick Brooks in his book “Mythical man month” [Bro75]. One of the big challenges of developing software, especially with a number of developers, is the conflict between global concepts and the necessity to split the work into smaller tasks. Eric Raymond characterizes two ways of software development, cathedral or bazaar style [Ray00]. A cathedral is build by a master, who makes all important design decisions – for Raymond this has been the traditional process of proprietary software development. In contrast, he sees open source software as being developed “bazaar-style” - many people can work on the same piece and discussing issues without any hierarchical structure. While this is not true for all open source projects, as some only consist of single developers or have a hierarchical structure, Raymond’s description gives a good characterization of the process of volunteer collaboration in general. Steven Weber writes that the 90’s actually revealed and proved that there are two possibilities for production models of software, the open source model and the proprietary model [Web04, p. 94]. Other authors go even further, discussing

the removal of a direct wage relation between product and producer under the title “Hacking Capitalism” [Söd08].

Included in the work by Eric Raymond is the famous quote “given enough eyeballs, all bugs are shallow” [Ray00, p. 8]. It describes the expectation that open source software projects can produce software of better quality, because they have the possibility to tap into the resources of large numbers of developers. Open source software thus has the potential to create a community of developers possibly larger than any company or closed organization could achieve. Although this is not guaranteed to happen, it is clearly visible for many successful projects.

Consisting of many voluntary members, open source projects naturally suffer sometimes from internal conflicts and have to overcome specific challenges. Good projects not only have a stable developer base, but also a clearly defined decision making structure. Such a structure can have different forms - sometimes projects are organized in hierarchies surrounding leading developers. For example, Linus Torvalds until today has the final say over changes to the Linux kernel. Alternatively, they are more spread out to groups of core developers. Independent of the structure, basically all projects make use of different technological helpers (e.g. version control systems) to foster the efficiency of the process. Typically, the decision making structure also prevents decline of software quality and the addition of intentional changes that would undermine functionality or open up security loopholes. While it is often assumed that changing the source code of an open source project would be as easy as modifying entries in the Wikipedia, this is mostly not true. Projects implement numerous cross-checking mechanisms to make sure that changes to the source code only can be published after they have been checked by other developers.

Due to the voluntary nature of open source projects there can be no strong enforcement mechanisms, groups are hold together by one important characteristic of the possibility to share changes: In case of insurmountable differences, projects can undergo a process called “forking”, where some developers take an existing, freely available code base and create a new separate project. The fork and the original software then compete for users and developers, reducing resources available for both. However, this only occurs very rarely. Mostly, it is tried to overcome conflicts among developers and to achieve compromises before a fork could happen.

Concluding this background one can state that today, open source or free software is used very widely. It is common not only for individuals, but also for companies to get involved in the development of open source software. This can take the form of paying developers or releasing formerly proprietary software under open source Licenses. Even one of the most outspoken critics of open source model in the past, Microsoft [Web04, p. 126], recently joined the movement by publishing several of its products as open source software, for example with the release of the .NET framework [Lan14]. The idea of open source software and the characteristic of the development process has not reached the field of nuclear arms control yet – in the following, several of the benefits and advantages of open source approaches to that field will be discussed.

3.2 Benefits of the approach

Using open source software for nuclear arms control can directly address the issues raised in the previous chapter. It can increase trust, transparency and participation, enable societal verification and lead to broader capacity building.

Trust among all actors involved is increased if the source code of software is accessible together with the software itself. This allows for inspection of a software's functionality, carrying out a "verification of the verification". Instead of trusting proprietary software applications like black boxes, states and international organizations could thoroughly study how the software works. This inspection could even be carried out by a greater audience; basically everyone interested would be able to access open source verification software and check its functions for possible backdoors. A perfect example for the need to open, intensive scrutiny are information barriers, as described in the previous chapter. Both the inspected and inspecting party need to know what the machine is doing, and most likely see its source code. This already can be achieved by jointly developing the software and using only open source software whenever external tools are required. But there does not seem to be a convincing reason why that should not be opened up to a larger audience. If many experts would confirm the functionality of such a tool, trust could be even greater.

Interrelated to the issue of trust is the issue of transparency. Access to source code not only allows one to verify that there has not been an attempt to cheating, but also to show the underlying principles used to obtain a certain result. Such principles could be basic assumptions about physics, as well as the physical model used in a simulation. With regard to the examples from the previous chapter, this might be beneficial in cases like the Arak reactor conversion. Transparency would allow different parties not only to discuss "this is how we think the reactor should be modified", but also "this is why we think it is the case", by showing model and method for the simulation of a converted reactor model. Transparency is also a fundamental requirement for correct scientific behavior ensuring reproducibility of results. Using open source software simplifies repeating measurements or simulations because it provides all the main assumptions and models that are part of the software.

Open source criteria applied to arms control have the benefit of increasing participation. Instead of a policy of compartmentalization, openness in sharing developments enables connection between communities. Proprietary tools always have the risk that due to access restrictions and proprietary licenses, development of new products could be made risky and difficult. For similar issues in bio-medical research, [HE98] called this the "tragedy of the anticommons". In contrast to this risk, open source licenses would guarantee re-usability of changes in the future.

Increased sharing is a benefit for the arms control community itself. By reusing software and being able to adapt it to new tasks, developers can possibly save time and achieve better results. In addition, the open source aspect of software also has implications

beyond the arms control community. It is possible to attract more developers, engineers and people with a multitude of backgrounds. Under the right circumstances, why not even have disarmament hackers? Most likely, such an increase in the number of people involved will create more ideas on how to solve specific verification problems. This could be done in an organized fashion, for example using crowd sourcing approaches ([Sur05], [Bra13]). Clearly, this is only viable if the software has been available to the “crowd”, and if everyone is allowed to modify the code.

In successful open source projects, a trend of an increase in numbers is sometimes triggered just by releasing software as open source software. Figure 3.1 shows the number of authors over time in some software projects related to arms control applications. The figure is based on the number of author contributing per year to projects that are hosted by the development platform GitHub. While it not possible to generalize from such a small number of cases, there is at least an indication that open source projects might create participation options for larger numbers of developers.

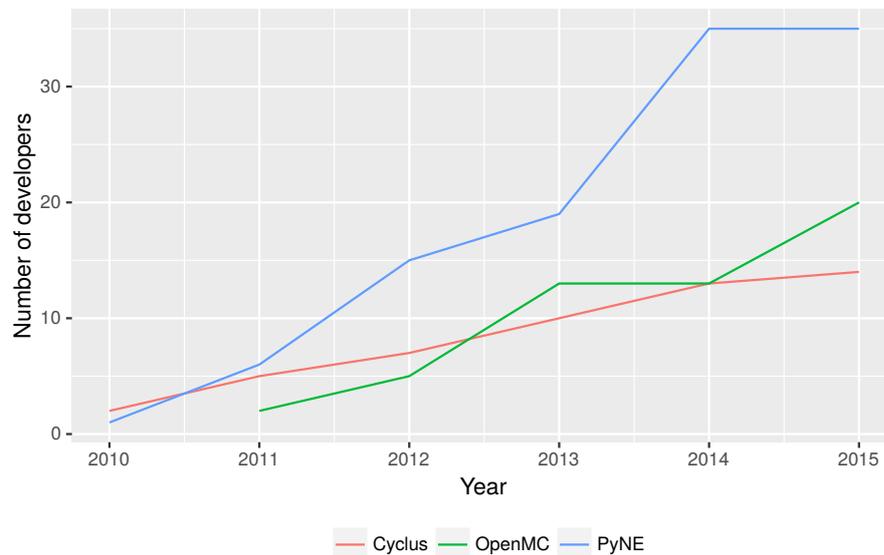


Figure 3.1.: Number of authors that contributed to selected open source projects.

Besides increasing the developer base through openness, a move in this direction also could enhance capacity building through an increased user base. To give arms control experts easy access to software tools by releasing them as open source software might broaden their interest, and also draw previously not involved people to the field. Even further, beyond the group of experts, software developers and specialists, open source software makes it possible to involve the society at large in arms control verification. Giving access to the general public allows everyone to participate in certain steps of arms control verification (“Societal Verification” [Rot+93; HH12]), but also to learn and gain knowledge on an important issue. An example closely related to nuclear arms control is the project “Safecast”, which developed a world-wide radiation monitoring network based on voluntary participants. They constructed relatively cheap

radiation detectors using open source hardware designs, and set up an open database for the collected data. This helped especially the Japanese population to get a better understanding of the radiation exposure after the Fukushima nuclear accident. Future developments might allow for similar projects of citizens helping nuclear arms control. Giving access to tools allows everyone interested to know what is going on. Generally, releasing verification tools using open source criteria would be an important step towards more democratic arms control regimes.

3.3 Challenges and Criticism

Of course, applying open source criteria to arms control does not come without cost and there are challenges that should be considered. A point often raised is that sharing source code would give an adversary the chance to cheat because she or he can know how the software works. It is true that if there exist a way to cheat, it is detected more easily if the source code is open. For example, if an inspected state would know that an information barrier shows a green light for a specific combination of fissile materials that are not a weapon, it might use such a combination during an inspection. More complicated cheats are possible, too. However, having no access to software source code as in proprietary tools is by no means a guarantee that no ways to cheat can be found. In both cases, there is a competition between an attacker and a defender to find loopholes, and either exploit them or close them in time. But only open source software is open for scrutiny by everyone, and does not rely on “security through obscurity” by not allowing an adversary to understand the workings of a program. Especially for critical applications such as arms control, there is well-grounded reason to believe that the broader group of analysts will give advantages to those working for a trusted piece of software over those trying to cheat (cf. [Sch99; Sch14]).

Cryptography has a similar problem - weaknesses of encryption could be exploited by an unwanted listener. To overcome this problem, it is a widely accepted principle that security of an encryption algorithm should only require the key to be secret, not the cryptographic algorithm used. Already in 1883, this has been stated as the Kerckhoffs’ principle, named for a Dutch military cryptography specialist [Ker83]. The best algorithms indeed are public ones, where the algorithms have undergone numerous checks by independent experts. Of course, this could only be done if the algorithm (or source code) is published along with the software itself.

Kerckhoffs’ principle also addresses another challenge. As stated previously, not all of the information and data generated during an arms control inspection or prepared for negotiations might be shared among inspecting and inspected party as well as the general public, because it would be considered sensitive or sharing would be prohibited by international treaties. As an example, sharing weapon design information with a non-nuclear weapon state is prohibited under the Nuclear Non-Proliferation treaty. That point has been often raised as an argument against the use of free software for arms

control purposes. Doing so often mixes two different things - data and the data processing software, or in terms of encryption message and cryptographic algorithm. If there is a necessity to protect sensitive information, the information should be separate from the algorithm. But such a separation should rely on an algorithm that can be published - following Kerckhoffs, it should be possible to design an algorithm that can be shared.

While openness can help increase participation, it is also possible to reduce participation, mainly because of the voluntary nature and free-riding effects. This can lead to crucial malfunctions or even to the death of a project because of limited and decreasing numbers of developers and maintainers. One of the more recent examples for such a development is the bug called “Heartbleed”, which was found 2014 in the OpenSSL library [Cox14]. The SSL protocol is used for encryption between websites and web servers, e.g. to protect personal data from eyes of third-parties. The bug allowed attackers to steal data from affected web servers, and has been part of the software for many years. Discussions on the causes of the bug clearly point out that the voluntary participation for a security critical tool failed to gather adequate support [Pag16]. At the time of the bug, the project had only a small number of developers contributing actively to the code. Large companies, while relying on the functions of the software, did not contribute resources. They chose to do so after the bug was discovered [Lin14]. The same effect could be an issue for open source projects that are employed for nuclear arms control. However, releasing a program as an open source code does not necessary imply reliance on a completely random community. Of course, institutional contributions can continue or be added. Verification tools are almost always sponsored by states and international organizations. There is no reason to believe that their motivation will decrease when software is open.

Possible usage areas of software pose another challenge. Software tools can have dual-use characteristics, they could be employed for arms control purposes, and at the same time might be useful for development of the weapons itself, or optimization of related processes. Clearly, open source software that has such dual-use characteristics would raise concerns as the easy access could eventually contribute to further weapons proliferation. These characteristics are the main reasons why software access is regulated by states and the international community, using export controls. Export controls typically are based on international agreements, as the Wassenaar Arrangement or the Export Control Guidelines of the Nuclear Suppliers Group in case of nuclear weapons. To be effective, these have to be transferred into domestic export laws by member states, resulting in a variety of characteristics, but still limiting access. This issue is relatively difficult to resolve, especially if cooperation and openness should be favored over secrecy. One the one hand, it should be prevented that software is used for malicious purposes or help weapons development. On the other hand, such a limitation of sharing could create mistrust, and probably also prevent or slow down successful progress on disarmament and arms control. If possible, software to be used in verification should be developed in ways to prevent other use cases, even if openly shared.

3.4 Current Situation of Open Source Software in Nuclear Arms Control

As shown before, the issues related to the use of software with regard to trust, transparency and participation play an important role for arms control applications, probably more important than for software applications in other fields. However, the possible use of open source software to address these issues has been discussed only in very few cases in the past⁸. Probably the first to discuss the issue, Greg White presented a paper that lists open source software and hardware as the preferred option for the development of information barriers as part of the Trilateral Initiative [Whi01]. 10 years later, [MBH11] promotes the use of open source software for geospatial information systems (GIS) for use in nuclear safeguards. Further publications that explicitly discuss open source can be found in the field of civilian nuclear energy production - [VW07] and [WVV07] discuss the use of open source software for reactor safety applications.

Beyond the scholarly debate, the idea of open source software is exemplified by the development of software. Several examples of open source software related to arms control issues are listed in table 3.1. Some of the listed examples have been particularly developed as open source software. However, none of the authors claim that a main reason of publishing their software as open source was primary benefits for arms control as described in this work.

Table 3.1.: Open source software examples that could be used for nuclear arms control purposes.

Code	Purpose	Citation
OpenMC	Neutron transport Monte Carlo code	[RF13]
Cyclus	Nuclear fuel cycle simulation	[HWG11]
PyNE	General purpose tool for nuclear engineering	[Sco+12]
Geant4	General purpose Monte Carlo framework	[Ago+03]

There are not only individual examples of open source applications, but signs of organizations engaging more broadly in this field. Several US national laboratories have specific organization pages on GitHub, the largest platform for publishing and developing open source software:

- Sandia National Labs: <https://github.com/sandialabs>
- Los Alamos National Lab: <https://losalamos.github.io/>
- Pacific Northwest National Lab: <http://pnnl.github.io/>

⁸ Independent of the above mentioned reference, a substantive set of articles can be found that has open source in the title. However, these works typically deal with open source information, information available in the open, e.g. satellite images from Google Earth, newspaper articles or Twitter messages. Those articles are not related to the issue of this work (software), and hence not discussed here.

-
- Lawrence Livermore National Lab: <http://software.llnl.gov/>

On these pages, the labs list some of the software packages they developed, and make them available for download and modifications.

The Preparatory Commission for the Comprehensive Test Ban Treaty Organization seems to have realized some of the benefits of open source software, which it states publicly: “It was therefore decided to migrate the IDC applications software (i.e. data acquisition, forwarding, automatic processing and interactive analysis) to open source systems.” [Com10]. At the same time, while source code and software sharing is prevalent inside the CTBTO and among the national data centers of member states, there is no real engagement with the broader public (for example, the CTBTO has no public GitHub page or similar open source repository).

While the listed references and examples show that there are some activities in the area of arms control with regard to open source software, it also shows a lack of broader application of the open source idea. With the development of an open source application for simulations of neutron multiplicity measurements, the feasibility for the approach will be shown. The software will also give users and experts an option that is different from the commonly used proprietary simulation tools. Further expansion of open source software use should be possible, and this chapter should have shown its necessity, too.



Part II.

**Simulation of Neutron
Multiplicity Measurements**



4 Neutron Multiplicity Measurements

This chapter describes the physical background needed to carry out adequate simulations of passive neutron multiplicity measurements. Passive neutron multiplicity measurements are used to estimate spontaneous fission rates of samples, in most cases to determine their plutonium masses. While all important aspects will be mentioned, a special focus will be put on the description of (α , n) reactions. For the treatment of these reactions, special routines that were developed as part of ONMS. Because of this reason, sections covering these aspects are discussed at the end of this chapter.

The chapter is in parts based on the good overviews over method, application and theory which can be found in the literature, e.g. in chapter 6 of the addendum to the comprehensive book “Passive Non-Destructive Assay of Nuclear Materials” [Rei07; Ens+07a], also in the “Application Guide to Neutron Multiplicity Counting”, [Ens+98]. Among more recent sources, a detailed description can be found in [Göt15].

A neutron multiplicity measurement or neutron multiplicity counting is a non-destructive assay (NDA) technique, based on neutron emission and detection. It extends the functionality of total neutron counting and neutron coincidence counting methods. Detecting neutrons has the advantage that they travel relatively far through matter, as they do not interact via electromagnetic forces. Often it is possible to carry out passive measurements, where the initial source of neutrons is based on radioactive decays in the sample. If the natural activity of a sample is not strong enough to acquire the desired information, active interrogation can be applied by using external neutron or gamma emitters to induce reactions in the sample [RE08]. All detection can be done without changing the material itself, in contrast to destructive assay methods, which for example include dissolving parts of the material for chemical analysis⁹. NDA methods are beneficial when destructive assay is not possible, e.g. for measuring fabricated fresh reactor fuel rods or nuclear warheads during warhead verification, where the fissile material most likely would be inaccessible or destructive assay would be considered too intrusive.

Measurement devices specifically designed for neutron multiplicity measurements have been developed since the 1980's (cf. [KS84; LKE90; Men+93; Gua93; Rei07]). It was possible to use older coincidence counters by employing modified analyses. Some older detector designs are today commonly used for passive neutron multiplicity analysis, as

⁹ Technically, active interrogation changes a sample by inducing fission reactions inside. However, acceptable neutron count rates are already achieved with very small changes. For example while 1 g of ^{235}U consists of about 2.6×10^{21} atoms, fissioning only 1000 atoms/second could be enough to carry out adequate measurements.

for example the “Active Well Coincidence Counter” (used in passive mode), which will be described later.

The most common property to be derived via passive neutron multiplicity measurements is the mass of plutonium in a sample. The different plutonium isotopes have different spontaneous fission rates, isotopes with the even atomic mass numbers A have rates several orders of magnitude above those with odd A . To derive the total mass of the sample, the sample composition needs to be known, in case of plutonium its isotopic vector. Isotopics can be measured using other technologies, for example gamma spectroscopy. It is possible to measure plutonium masses in a variety of chemical compositions, e.g. metal, oxide, fluoride, even in liquids. Especially for plutonium with impurities, multiplicity counting methods are beneficial over coincidence or total counting approaches, because of additional information generated in the process.

Samples can originate from every step of processing nuclear fuel, weapons production or other research activities related to plutonium. According to [RE08, p. 46], multiplicity measurements are used in several countries “for NDA of impure Pu metals, oxides, mixed oxides, residues, wastes”. Reasons for measurements range from material accountability and safeguards verification in plutonium handling facilities like reprocessing plants or weapon production facilities to the verification of excess weapons plutonium. Warhead verification and authentication is possible, too. Some examples of possible applications will be listed in the following.

The “Plutonium Scrap Multiplicity Counter” (PSMC), which is one of the detector modeled in later chapters, was designed for IAEA verification during Japanese reprocessing efforts. It is supposed to measure “plutonium scrap”, impure samples of plutonium that result from various chemical processes during reprocessing and MOX fuel fabrication. The samples can contain plutonium metal or oxide, but also fluorine, beryllium or other matrix materials [Men+93; Ens+98]. A similar approach, but more recent, is described by [Nak+10]. The approach proposes to use neutron multiplicity measurements to analyze process waste (“sludge”) of the conversion of plutonium-uranium nitrate to MOX powder. Resulting MOX powder samples can be analyzed using neutron multiplicity measurements, in Japan for example using the PSMC detector type, too. It is also foreseen to measure full fabricated MOX fuel elements with neutron multiplicity measurement methods [Nak+14].

As stated earlier, the method can not only be used to safeguard reactor materials. After the end of the cold war, the United States and Russia declared some of their weapons plutonium to be excess for its original purpose. In several inspections in the United States, the IAEA used neutron multiplicity measurements for inventory verification of excess weapons materials [LFL97].

Beyond material accountability, it has been proposed to use neutron multiplicity measurements for verifying full warheads and fissile material in warhead components. In one of the first publications on this idea, [Fet+90] describes the use of neutron radiation for warhead verification. A comprehensive overview of measurement technologies

for warhead verification is given in [GK14]. In this context, neutron multiplicity measurements are employed for the so-called “Attribute Approach”. The approach identifies nuclear weapons based on measurement results for a predefined set of physical properties, called attributes. One attribute usually is a mass range for the fissile material inside a weapon.

Development of several systems using neutron multiplicity measurements for fissile material mass estimates took place during and after the Trilateral Initiative (cf. chapter 3, page 30 of this work, also [Whi12]). One system was built in cooperation among Russian and US scientists - the AVNG, spelled out as “attribute verification system, with IBs for plutonium with classified characteristics, built on the neutron multiplicity counting and high-resolution gamma-spectrometry” [Raz+10]. In US laboratories, over time three different systems were built. A first version was the Fissile Material Transparency Technology Demonstration (FMTTD, [Bou+01]), followed up later by the Next Generation Attribute Measurement System (NG-AMS, [She+05]) and the 3rd Generation Attribute Measurement System (3G-AMS, [War+12]). Besides neutron multiplicity detectors, all had various other components, e.g. for gamma spectroscopy and for the data processing and analysis part to form information barriers.

Neutron multiplicity measurements can also be employed to detect the mass of Curium. Curium is produced during irradiation of fuel in a reactor. For high reactor burnup times, curium isotopes are the major source of neutron radiation in spent fuel, and directly determining the plutonium content is not possible anymore. Such high burnups are common for commercial nuclear power applications. Combined with the “curium ratio technique”, this method can also determine the plutonium content in this spent fuel. As part of this technique, the ratio between curium and plutonium of a stream of spent fuel would be measured using different technologies, for example destructive assay methods of small samples. From the ratio and the curium content determined by neutron multiplicity measurements of larger samples, the related plutonium content could be determined [Lee+08; MM94].

To measure uranium, typically active approaches are used, where additional neutron sources create neutrons for fission inside of the sample material. Only if the detector efficiency is high enough, uranium samples can be measured using just the passive approach. At the Joint Research Centre in Ispra, Italy, the “High Efficiency Passive Counter” has been developed and tested for large samples of low enriched uranium [LPT09].

4.1 Overview of the physical effects to be considered

The spontaneous fission rate cannot be measured directly. It can be derived by counting the number of emitted particles, neutrons or gammas. For the approach discussed here, neutrons are preferred over gamma emissions. They have a much higher probability of leaving the samples, and most gammas result from other decays than spontaneous

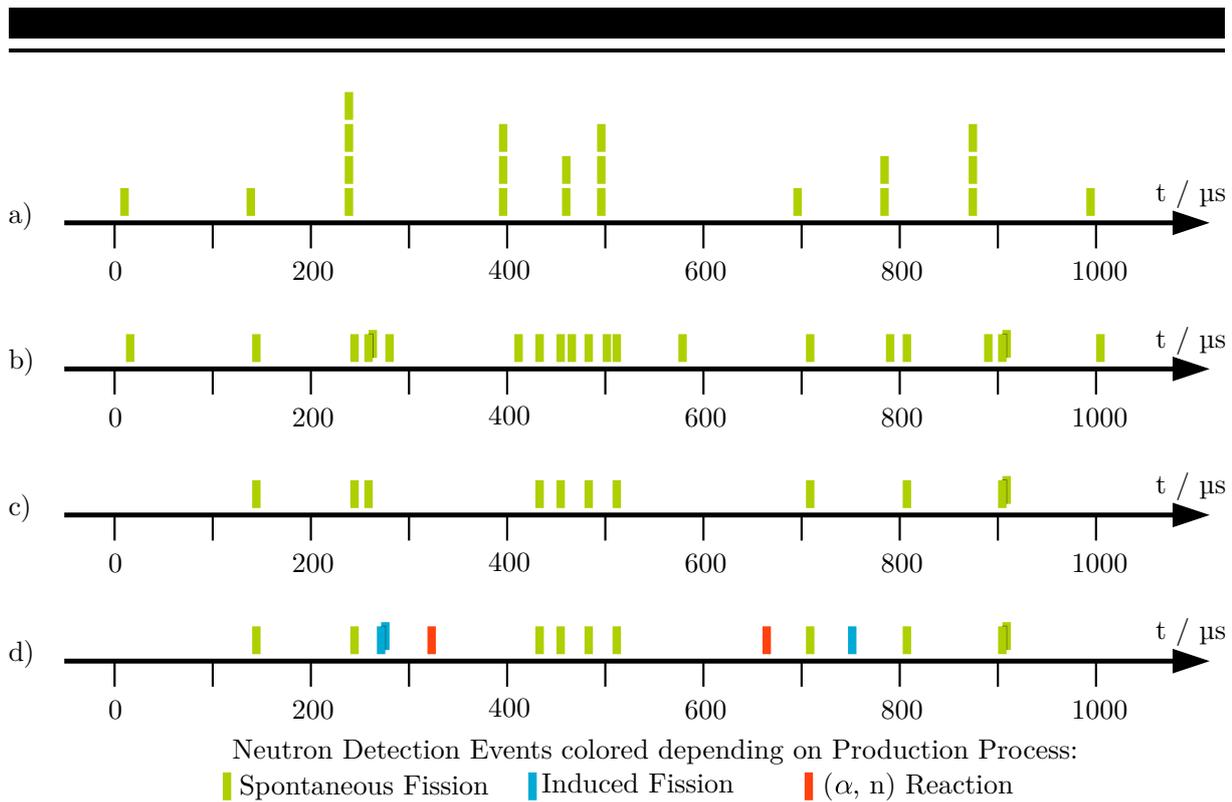


Figure 4.1.: Example neutron pulsetrains, taking different effects into account. Each rectangle stands for a neutron detection event. a) shows a pulsetrain as it would be emitted by a spontaneous fission source, b) the same after moderation, c) with detection efficiency < 1 . d) includes also neutrons from other reactions.

fission. Figure 4.1 shows examples of several detected neutron events, so-called “pulsetrains”, depicting the effect of different physical processes involved in measurements. The pulsetrains typically come from multiple individual detection units inside of a detector, e.g. multiple ^3He tubes. The simplest case (a) shows neutrons as they would be emitted by a spontaneous fission source with a total fission rate F . Single fission events have different numbers of neutrons that are emitted. The average number of neutrons per fission is typically noted as $\bar{\nu}$, hence the total neutron rate from spontaneous fission is

$$r_{n,SF} = F \cdot \bar{\nu} \quad (4.1)$$

Most detectors rely on thermal neutron counting, so moderation is required, leading to (b), where neutrons from single spontaneous fission events get stretched out over time. Detectors have a limited detection efficiency, some neutrons get lost outside of the detector or absorbed in detector materials other than the “sensor” (c). More neutrons are added to the pulsetrain by additional effects (d): α -particles from decay can undergo

(α, n) reactions and produce additional neutrons, with a rate $r_{n,\alpha}$. The relation between spontaneous fission neutrons and the (α, n) reaction rate is defined as

$$\alpha \equiv \frac{r_{n,\alpha}}{F \cdot \bar{\nu}} \quad (4.2)$$

Any neutron in the sample can cause an induced fission, again producing more neutrons, as shown in (d), too. This increase is described by the multiplication M . All these effects are discussed in this chapter, preparing for the explanation of the developed simulation software in the following chapter.

With only the effects in part (b) of figure 4.1, total neutron counting could be an adequate method to derive the spontaneous fission rate. To use it also for (c), it is necessary to know the detector efficiency as a function of the positions of neutron emission, or have a detector built in a way that its efficiency can be considered equal for all positions in possible samples. If only one effect in (d) is unknown, neutron coincidence counting can be applied. The method results in two measured quantities, often called “reals” and “accidentals”, allowing to solve for one effect of (d) and the spontaneous fission rate. This would be the case for example for pure metal samples, where only multiplication but no (α, n) reactions occur, or generally for small samples, where nearly no multiplication occurs.

Only the introduction of neutron multiplicity measurements allowed for the measurement of a third variable. Mostly, this is used to solve for the multiplication, the second effect depicted in figure 4.1. However, there are many more variables or functions that could influence the results. The following list is reproduced as listed in [Ens+07b]:

- Energy spectrum of (α, n) reactions
- Spatial in-homogeneity of multiplication (induced fission rate varying by position)
- Spatial and energy differences in detection efficiency
- Neutron capture in the sample
- Detector die-away time

For all these unknowns, it is necessary to make assumptions and characterize detectors with calibration samples. For example, as has been mentioned before, a flat efficiency profile is a goal for detector development, and often samples are relatively small compared to the overall detector geometry. It is possible to check detectors for their efficiency profile using calibration samples. For the following calculations it is assumed that there is no variation in detection efficiency depending on source position or energy, which is reasonable for many detectors. Similarly, the other variables and functions are typically small or reduced by careful detector and measurement design. A comprehensive description of the assumptions and preparations made for the other effects and variables is given in [Ens+07b]. Some of them will also be discussed later in this chapter.

4.2 Neutrons from Fission

For passive neutron measurements, spontaneous fission events are the most important neutron source. Several isotopes can decay by spontaneous fission, most with very low branching ratios compared to other forms of decay. However, as neutrons have low interaction probabilities within typically measured samples, already spontaneous fission rates of a few hundred to thousand fission reactions per second can lead to significant neutron flux leaving a sample.

Probably the most important figure for neutron multiplicity counting is the rate of spontaneous fission reactions of ^{240}Pu per gram and second, calculated via:

$$F_{Pu240} = N \cdot BR \cdot \frac{\ln 2}{T_{1/2}} \quad (4.3)$$

where N is the number of ^{240}Pu atoms in 1 g, BR the decay branching ratio for spontaneous fission and $T_{1/2}$ the total half-life of ^{240}Pu .

For simulations of mixed samples, it is also necessary to calculate fission rates for all other isotopes undergoing spontaneous fission. Collecting data for total half-life and spontaneous fission branching ratio was more complex than expected when starting this work. Starting out with values taken from a textbook for nuclear engineering [SF02], it became quickly clear that even among very recent references there is a relevant divergence in data sets. This is especially true for data references often quoted in relation to neutron multiplicity measurements, as in the “Application Guide to Neutron Multiplicity Counting”, [Ens+98, p. 43], the “Passive Nondestructive Assay of Nuclear Materials” (PANDA), [Rei07, pp. 11-15]¹⁰, and “A Good Practice Guide for the use of Modelling Codes in Non Destructive Assay of Nuclear Materials”, [Cha+09].

Table 4.1 collects half-life data from different sources, table 4.2 for branching ratios for a selected list of isotopes. Besides the mentioned references, data from the following sources has been included:

- Current versions of the Evaluated Nuclear Structure Data File, which are compiled by the International Network of Nuclear Structure and Decay Evaluators. The files used were downloaded from a website of the IAEA (https://www-nds.iaea.org/ensdf_base_files/) in the version from December 08, 2015.
- Two major evaluations of nuclear reaction data commonly used for Monte Carlo simulations, ENDF-B in version VII.1 and JEFF in version 3.1.1. Although their main content deals with induced reactions, they also have data files for decay data. According to the textual information included in most files of ENDF-B VII.1 (data set with MF=1, MT=451 according to ENDF file format [HT09]), they are compiled based on recent ENSDF data, JEFF 3.1.1 radioactive data comes from a variety of sources (cf. [KBM09]).

¹⁰ These two sources actually list the same values.

Table 4.1.: Comparison of decay half-life of selected isotopes from different sources. All values are given in years. Cells with “-” indicate that a value was not listed in the respective reference. Uncertainties are included if available.

Data Source / Nuclide	ESARDA “Good Practice Guide” [Cha+09]	Shultis / Faw Textbook [SF02]	Passive NDA of Nuclear Materials [Rei07]	Articles from <i>Nuclear Data Sheets</i> *	JEFF 3.1.1 [San+09]	ENDF B-VII.1 [Cha+11; NNDCa]	ENSDF [NNDC]
²³⁵ U	-	7.04×10^8	7.04×10^8	$7.04(1) \times 10^8$	$7.038(5) \times 10^8$	$7.038(5) \times 10^8$	$7.04(1) \times 10^8$
²³⁸ U	4.469×10^9	4.47×10^9	4.47×10^9	$4.468(6) \times 10^9$	$4.468(3) \times 10^9$	$4.468(3) \times 10^9$	$4.468(6) \times 10^9$
²³⁷ Np	-	2.14×10^6	2.14×10^6	$2.144(7) \times 10^6$	$2.14(1) \times 10^6$	$2.144(7) \times 10^6$	$2.144(7) \times 10^6$
²³⁸ Pu	87.768	87.7	87.74	87.7(1)	87.7(3)	87.7(1)	87.7(1)
²³⁹ Pu	24112	24100	24100	$2.441(3) \times 10^4$	24113(11)	$2.411(3) \times 10^4$	$2.411(3) \times 10^4$
²⁴⁰ Pu	6555	6569	6560	6561(7)	6563(5)	6561(7)	6561(7)
²⁴¹ Pu	14.35	14.35	14.35	14.329(29)	14.33(4)	14.290(6)	14.290(6)
²⁴² Pu	376200	376000	376000	$3.75(2) \times 10^5$	$3.735(11) \times 10^5$	$3.735(11) \times 10^5$	$3.75(2) \times 10^5$
²⁴¹ Am	433.65	433.6	433.6	432.6(6)	432.8(7)	432.6(6)	432.6(6)
²⁴² Cm [†]	-	0.44627	0.44627	0.44572(55)	0.4461(2)	0.4461(2)	0.44572(55)
²⁴⁴ Cm	-	18.11	18.1	18.1(1)	18.0(1)	18.11(3)	18.1(1)
²⁵² Cf	2.64675	2.638	2.656	2.645(8)	2.645(8)	2.645(8)	2.645(8)

* All the values in this column are taken from the most recent publications in the *Nuclear Data Sheets*. In detail: ²³⁵U [BT14a], ²³⁸U, ²³⁸Pu [BT15], ²³⁹Pu [BT14b], ²⁴⁰Pu [SB08], ²⁴¹Pu, ²⁴¹Am [Nes15], ²⁴²Pu, ²⁴²Cm [Ako02], ²⁴⁴Cm [Ako03], ²⁵²Cf [Nic05].

† Cm-242 decays in less than a year, sources typically give value in days. It has been converted to years for this table by dividing by 365.25.

Table 4.2.: Comparison of decay branching ratio for spontaneous fission for selected isotopes. Values are given as fractions (not in %). Cells with “-” indicate that a value was not listed in the respective reference.

Data Source / Nuclide	ESARDA “Good Practice Guide” [Cha+09]*	Shultis / Faw Textbook [SF02]	Passive NDA of Nuclear Materials [Rei07]	Articles from Nuclear Data Sheets [†]	JEFF 3.1.1 [San+09]	ENDF B-VII.1 [Cha+11; NNDCa]	ENSDF [NNDC]
²³⁵ U	–	2×10^{-9}	2.01×10^{-9}	$7.0(2) \times 10^{-11}$	$7.2(21) \times 10^{-11}$	$7.2(21) \times 10^{-11}$	$7.0(2) \times 10^{-11}$
²³⁸ U	5.45×10^{-7}	5.4×10^{-7}	5.44×10^{-7}	$5.45(7) \times 10^{-7}$	$5.46(10) \times 10^{-7}$	$5.46(10) \times 10^{-7}$	$5.45(7) \times 10^{-7}$
²³⁷ Np	–	2.1×10^{-14}	2.13×10^{-12}	$<2 \times 10^{-12}$	–	–	$<2 \times 10^{-12}$
²³⁸ Pu	1.84×10^{-9}	8.1×10^{-10}	1.85×10^{-9}	$1.9(1) \times 10^{-9}$	$1.86(6) \times 10^{-9}$	$1.86(6) \times 10^{-9}$	$1.9(1) \times 10^{-9}$
²³⁹ Pu	4.4×10^{-12}	4.4×10^{-12}	4.4×10^{-12}	$3.1(6) \times 10^{-12}$	$3.1(6) \times 10^{-12}$	$3.1(6) \times 10^{-12}$	$3.1(6) \times 10^{-12}$
²⁴⁰ Pu	5.7×10^{-8}	5×10^{-8}	5.62×10^{-8}	$5.7(2) \times 10^{-8}$	$5.7(2) \times 10^{-8}$	$5.7(2) \times 10^{-8}$	$5.7(2) \times 10^{-8}$
²⁴¹ Pu	5.74×10^{-15}	5.7×10^{-15}	5.81×10^{-15}	$<2 \times 10^{-16}$	–	–	$<2.4 \times 10^{-16}$
²⁴² Pu	5.5×10^{-6}	5.5×10^{-6}	5.5×10^{-6}	$5.50(6) \times 10^{-6}$	$5.50(6) \times 10^{-6}$	$5.50(6) \times 10^{-6}$	$5.50(6) \times 10^{-6}$
²⁴¹ Am	4.13×10^{-12}	4.1×10^{-12}	2.9×10^{-12}	$3.6(9) \times 10^{-12}$	$4.3(4) \times 10^{-12}$	$4.3(4) \times 10^{-12}$	$3.6(9) \times 10^{-12}$
²⁴² Cm	–	6.8×10^{-8}	6.75×10^{-8}	$6.2(3) \times 10^{-8}$	$6.1(1) \times 10^{-8}$	$6.1(1) \times 10^{-8}$	$0.6(30) \times 10^{-7}$
²⁴⁴ Cm	–	1.3×10^{-6}	1.33×10^{-6}	$1.37(3) \times 10^{-6}$	$1.38(4) \times 10^{-6}$	$1.38(4) \times 10^{-6}$	$1.37(3) \times 10^{-6}$
²⁵² Cf	0.03092	0.0309	0.0315	0.03092(8)	0.03092(8)	0.03092(8)	0.0309(80)

* Reference had only fission rate and half-life, from which the table values were calculated.

[†] All the values in this column are taken from the most recent publications in the *Nuclear Data Sheets*. In detail: ²³⁵U [BT14a], ²³⁸U, ²³⁸Pu [BT15], ²³⁹Pu [BT14b], ²⁴⁰Pu [SB08], ²⁴¹Pu, ²⁴¹Am [Nes15], ²⁴²Pu, ²⁴²Cm [Ako02], ²⁴⁴Cm [Ako03], ²⁵²Cf [Nic05].

- A manual compilation of the most recent issues of *Nuclear Data Sheets*, covering the listed isotopes

Nuclear Data Sheets is often considered a reference standard, and form the basis for the ENSDF data set, which are updated accordingly with a short delay. Among most other sources, several differences can be found. Even though the data evaluation ENDF-VII.1 claims that it is based on ENSDF data, spontaneous fission branching ratios differ (e.g. for ^{238}Pu $1.9(1) \times 10^{-9}$ from ENSDF compared to $1.86(6) \times 10^{-9}$ from ENDF). For ^{240}Pu , there are also differences, which can lead to significant differences in neutron source rates. *Nuclear Data Sheets* lists a half-life of 6561(6)yr. PANDA and the ESARDA Good Practice Guide list values inside the uncertainty range of the previous value, however the latter is at the lowermost end (6555yr). Outside of the error margin are the values of [SF02], listing a half-life of 6569yr and a branching ratio of 5×10^{-8} . The latter is more than 10% off all the other values. Many publications related to neutron multiplicity measurements cite a spontaneous fission rate for ^{240}Pu of 473 fissions $\text{g}^{-1} \text{s}^{-1}$ (e.g. [Ens+07b]). Given current evaluations, this is too low. Based on values from *Nuclear Data Sheets*, the spontaneous fission rate can be calculated to be

$$F_{Pu240} = 479(17) \text{ fissions g}^{-1} \text{ s}^{-1} \quad (4.4)$$

The relative uncertainty is about 3.5%, with the main contribution coming from the uncertainty specified for the branching ratio of spontaneous fission. As will be discussed later, the measured effective mass of ^{240}Pu is calculated by dividing the measured spontaneous fission rate by this value. Hence, the relative uncertainties of the natural spontaneous fission rate translate directly into relative uncertainties of the measured plutonium mass.

For this work the data sets from the issues of *Nuclear Data Sheets* have been used¹¹. They are summarized again in table 4.3, together with the resulting fission rate per gram per second. Additionally listed is the average neutrons produced per spontaneous fission $\bar{\nu}$ and the resulting neutron rate per gram and second of each isotope. Also for $\bar{\nu}$, different sources list different values. Differences are not listed here. It has been more difficult to find reference values, displayed are those that can be calculated from table 4 of [VHW10], or directly taken from table 5 of the same source. The software library described in [VHW10] is later used for calculations of fission events.

It is sometimes discussed that for safeguards and verification purposes, using a set of nuclear data that is consistent over time can be more important than using a physically accurate data set [Cha+09, p. 41]. For example to verify that items have not been

¹¹ Some exceptions relate to additional isotopes not listed in tables 4.1, 4.2 and 4.3, which were included in developing related functions for ONMS (^{232}U and ^{233}U). For both, spontaneous fission branching ratio listed was not consistent with ENSDF files, nor the spontaneous fission half-life listed in the respective journal article. The latter is very likely an editorial error. In both cases, the values from ENSDF files were implemented in the code.

Table 4.3.: Selected reference values for half-life, spontaneous fission branching ratio and average neutrons per spontaneous fission. Fission and neutron rates have been calculated based on the other listed values.

	$T_{1/2}^*$ years	Branching Ratio [*]	fission rate [†] fissions $\text{g}^{-1} \text{s}^{-1}$	$\bar{\nu}^\ddagger$	neutron rate neutrons $\text{g}^{-1} \text{s}^{-1}$
²³⁵ U	$7.04(1) \times 10^8$	$7.0(2) \times 10^{-11}$	$5.60(16) \times 10^{-6}$	1.86	$1.041(30) \times 10^{-5}$
²³⁸ U	$4.468(6) \times 10^9$	$5.45(7) \times 10^{-7}$	0.006 778(88)	1.99	0.013 49(17)
²³⁷ Np	$2.144(7) \times 10^6$	2×10^{-12} §	$5.205(17) \times 10^{-5}$	2.05	0.000 106 71(35)
²³⁸ Pu	87.7(1)	$1.9(1) \times 10^{-9}$	1204(63)	2.187	$2.63(14) \times 10^3$
²³⁹ Pu	$2.441(3) \times 10^4$	$3.1(6) \times 10^{-12}$	0.0070(14)	2.16	0.0152(29)
²⁴⁰ Pu	6561(7)	$5.7(2) \times 10^{-8}$	479(17)	2.154	1031(36)
²⁴¹ Pu	14.329(29)	2×10^{-16} §	0.000 765 9(16)	2.25	0.001 723 3(35)
²⁴² Pu	$3.75(2) \times 10^5$	$5.50(6) \times 10^{-6}$	801.5(97)	2.149	1722(21)
²⁴¹ Am	432.6(6)	$3.6(9) \times 10^{-12}$	0.46(11)	3.22	1.47(37)
²⁴² Cm	0.445 72(55)	$6.2(3) \times 10^{-8}$	$7.60(37) \times 10^6$	2.54	$1.931(93) \times 10^7$
²⁴⁴ Cm	18.1(1)	$1.37(3) \times 10^{-6}$	$4.102(93) \times 10^6$	2.72	$1.116(25) \times 10^7$
²⁵² Cf	2.645(8)	0.030 92(8)	$6.134(24) \times 10^{11}$	3.757	$2.3046(92) \times 10^{12}$

* Values from the most recent publications in *Nuclear Data Sheets*. In detail: ²³⁵U [BT14a], ²³⁸U, ²³⁸Pu [BT15], ²³⁹Pu [BT14b], ²⁴⁰Pu [SB08], ²⁴¹Pu, ²⁴¹Am [Nes15], ²⁴²Pu, ²⁴²Cm [Ako02], ²⁴⁴Cm [Ako03], ²⁵²Cf [Nic05].

† Atomic masses required in the calculation where taken from the 2012 Atomic Mass Evaluation [Aud+12; Wan+12].

‡ Calculated from [VHW10].

§ For calculations, the maximal possible value was used (the < from the source was dropped).

changed over time, it is more important to base measurements on the same assumptions than on changing data sets. However, this is not taken into account in this thesis. A higher emphasis here was given to more accurate data sets. If in the future the developed software should be applied for tasks where other data sets were used in the past, the respective data could be easily changed by knowledgeable users.

In most samples, the neutrons can undergo absorption processes. If they are captured in a radiative capture (n, γ) process, this leads to a decrease in neutron population. In samples of fissionable materials / actinides, absorption can cause induced fission, which would increase the neutron population. Beyond radiative capture and induced fission, there are ($n, 2n$) and ($n, 3n$) reactions. Because of high threshold energies, the latter two only rarely occur for spontaneous fission neutrons and are typically not accounted for in theoretical descriptions of the source for neutron multiplicity counting. These effects are summarized as the sample multiplication M . A given sample will emit $M \cdot x$ neutrons if x neutrons start in the sample by decay processes¹².

¹² These decay processes usually also include neutrons are produced by α -decay of materials and subsequent (α, n) processes, as discussed in later sections.

The energy distribution of neutrons produced in fission reactions is usually described by a Watt spectrum, based on empirically derived parameters a and b :

$$W(a, b, E) = \sqrt{\frac{\pi b}{4a}} \cdot \frac{e^{-\frac{b}{4a}}}{a} \cdot \sinh(\sqrt{bE}) \quad (4.5)$$

In this equation, E is the energy of the outgoing neutrons. The equation was taken from [VHW10], where also parameters can be found in tabulated form.

Most simulations of neutron reactions only require knowledge of the average number of neutrons produced per fission event $\bar{\nu}$. However, neutron multiplicity measurements make use of the known distribution of produced neutrons. The latter is also called a “Multiplicity Distribution”, or sometimes just “multiplicity”. To derive the spontaneous fission rate after detection, knowledge of this multiplicity is employed. Good knowledge of the multiplicity is not only important for the mathematical analysis after a measurement, but also to define particle sources for Monte Carlo simulations. Again, in the following the data of [VHW10] is used, which lists the same multiplicities as [SM08]. Figure 4.2 shows distributions for spontaneous fission of ^{240}Pu and induced fission of ^{239}Pu . These multiplicities are the most frequently used distributions in the analysis of neutron multiplicity measurements.

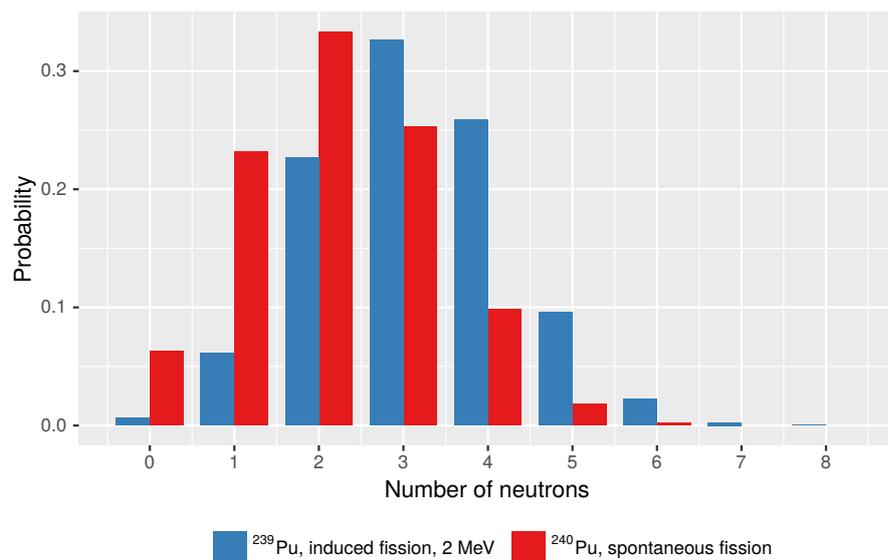


Figure 4.2.: Multiplicity distribution of neutrons resulting from induced fission of ^{239}Pu and spontaneous fission of ^{240}Pu .

Multiplicity distributions are often described by factorial moments. The first three factorial moments m_1 , m_2 , and m_3 can be calculated as follows for a probability distribution $P(n)$:

$$m_1 = \sum_{n=0}^{\infty} nP(n) \quad (4.6)$$

$$m_2 = \sum_{n=0}^{\infty} n(n-1)P(n) \quad (4.7)$$

$$m_3 = \sum_{n=0}^{\infty} n(n-1)(n-2)P(n) \quad (4.8)$$

All sums only need to be evaluated up to the biggest n with $P(n) > 0$. Whenever such moments are discussed in the following, they are based on normalized probability distributions, where

$$1 = \sum_{n=0}^{\infty} P(n) \quad (4.9)$$

4.3 Neutron Detection and Common Detectors

The majority of existing detectors for neutron multiplicity measurements are based on ^3He tubes to register incoming neutrons. ^3He neutron detection is most efficient for thermal neutrons. Other detection materials for thermal neutrons are possible, as well as detection in the epithermal (e.g. [Tam+10]) or fast neutron energy range (e.g. [PVE07]). Detection in ^3He is beneficial for several reasons. The incoming neutron gets absorbed, avoiding cross talk with other detectors, and the tubes show nearly no signal from gammas. However, recent increases in neutron detection applications lead to a world-wide ^3He shortage and drastically increased prices. [Uni11] discusses the issue and also lists possible replacement technologies. Still, no large trend is visible to replace existing ^3He based neutron multiplicity detectors.

Inside of ^3He tubes, an electrical field is established between a central wire and the tube casing. Typical voltages applied are between 1500 V to 1680 V [Ens+98]. The typical reaction of a neutron with gaseous $\text{He}3$ is a capture process that leads to the emission of a proton and a triton:



The reaction has a q -value of $763.7560(32) \text{ keV}^{13}$, carried away as kinetic energy of the proton and the triton. Both resulting particles ionize additional atoms of the ^3He gas.

¹³ Calculated using atomic mass data and conversion factor to keV as in [Aud+12; Wan+12].

Due to the electric field between central wire and casing, ions and electrons move into opposite directions. They create a charge pulse, which can then be further amplified and processed by the detector electronics to form a neutron pulse.

Figure 4.3 shows neutron interaction cross sections in ^3He . The cross section for the above described reaction is shown in green. It is increasing with falling energy, showing proportionality to $1/v$. In the MeV energy range, the elastic interaction cross sections becomes more important. Not shown in the figure are the interaction cross sections for the (n, γ) reaction and deuteron production. The latter has a very high threshold value, the first is orders of magnitude smaller than the shown range. Because neutrons released from fission are in the fast neutron energy range, detectors based on ^3He tubes require slowing down of neutrons by moderation to achieve high detection efficiency.

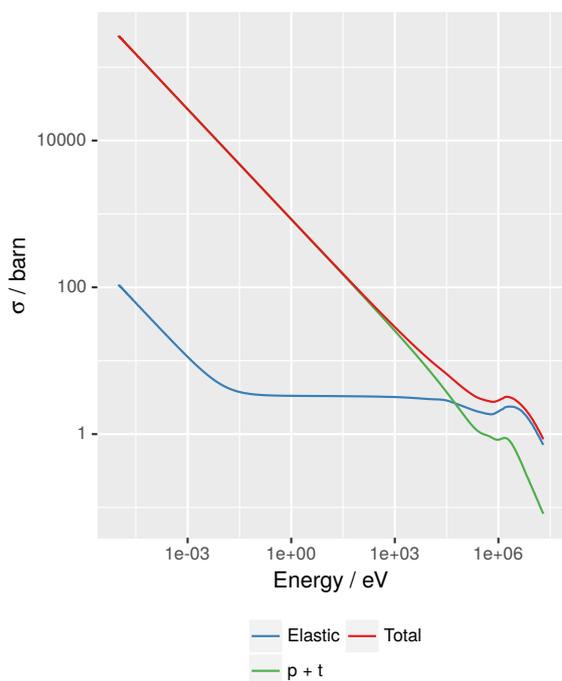


Figure 4.3.: ^3He neutron interaction cross sections from JEFF 3.1.

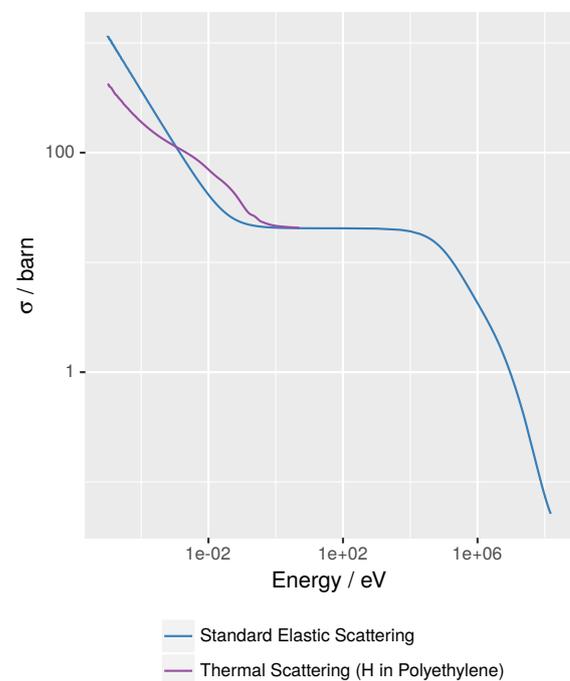


Figure 4.4.: ^1H neutron interaction cross sections from JEFF 3.1.

Moderation typically is done using polyethylene $(\text{C}_2\text{H}_4)_n$, which can at the same time form a structural component of a detector. Neutrons entering the polyethylene undergo elastic scattering processes mostly with the hydrogen atoms of the polyethylene. It takes multiple scattering events for the neutrons to be slowed down to thermal energies. Some neutrons get absorbed, forming deuterated polyethylene, reducing the detector efficiency. The number of neutrons absorbed varies with the detector, but can be as large as around a fourth of all neutrons emitted from a sample. The overall process of slowing down takes time, on the order of $10 \mu\text{s}$ to $100 \mu\text{s}$ before neutrons are absorbed or have left the detector. The average life time is called die-away time τ , a characteristic

of the detector. Smaller die-away times are beneficial, because it reduces the chance of overlapping neutron detection of subsequent fission events, and allows for shorter coincidence windows.

While most higher energy scattering can be described well on a nuclear level, for thermal energies also interaction effects with molecules have to be taken into account. This phenomenon is known from solid state physics, where neutrons are used to study atomic structures, because thermal neutrons have a deBroglie wavelength of the order of the dimensions of inter-atomic spacing dimensions and a kinetic energy in the order of collective vibrational modes of solids [FP13, p. 3]. Figure 4.4 shows the difference between 1H cross-sections that only take nuclear interaction into account (“Standard Elastic Scattering”) and those that are corrected for thermal scattering effects with the structure of polyethylene.

Most detectors have a cylindrical geometry, and employ several rings of 3He tubes. A model of the previously mentioned Active Well Coincidence Counter is shown in figure 4.5. This detector has two rings of 3He tubes. Compared to other detectors, it has a rather low detection efficiency. Sometimes, detectors employ $4-\pi$ geometries, surrounding samples on all sides. This is especially the case if it is intended to measure large and/or low emission samples (e.g. nuclear waste drums). Generally, while designing detectors, it is important to maximize efficiency and minimize the die-away time of neutrons inside the detector [RE08, p. 46]. As will be discussed later, high efficiency plays a major role as some results include the cube of the efficiency.

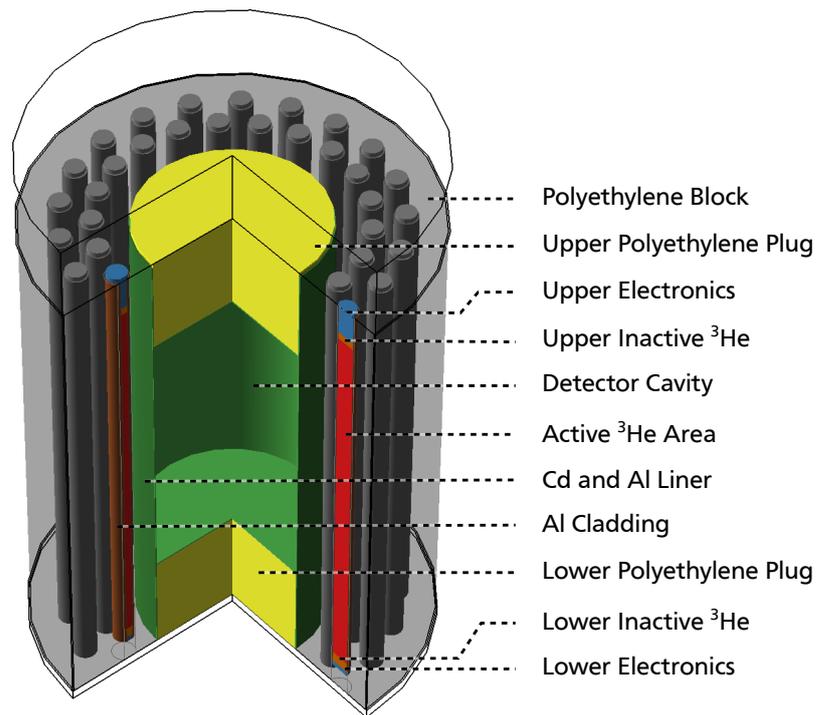


Figure 4.5.: Rendering of a Geant4 model of the Active Well Coincidence Counter, which is also commonly used for passive measurements.

4.4 Reconstruction of the Plutonium Mass

After a sample has been measured, experimentalists or simulators are left with a neutron pulsetrain, which is influenced by the effects described in the beginning of this chapter (cf. figure 4.1). An analysis of this pulsetrain is carried out inside of detector electronics using shift registers. If the pulsetrain has been recorded, the latter step can instead be done with a computer program.

Central references for the discussion of the shift register approach in this and the following sections are [Ens+98], [Ens+07b] and [Göt15]. This section deviates partly from their notation when it seemed to improve readability. In the shift register approach, two multiplicity distributions are defined, usually called “R+A” or “foreground” and “A” or “background”. The goal is to apply a method which stores the correlated events in the “foreground” distribution and uncorrelated events in the “background” distribution. In electronics, every incoming pulse is stored for a time period in the shift register, which acts as a coincidence window. A typical length is $G = 64\mu\text{s}$ (“gate length”), hence a shift register driven by a 2 MHz clock has 128 positions to store events. The pulses are delayed from entering the shift register by a very short time (“predelay”), with $PD = 2\mu\text{s}$ to $5\mu\text{s}$.

To construct the two distributions, each incoming neutron is considered a trigger. Immediately when a neutron is detected, the number of neutrons in the shift register is stored in the “R+A” distribution table. The number of neutrons detected in this detection window includes correlated and uncorrelated neutrons with regard to the trigger neutron. A second detection window is opened by the trigger neutron and a very long delay. The number of neutrons in that detection window is stored in the “A” distribution table, and it is assumed that all these events are uncorrelated neutrons with regard to the trigger neutron. The long delay L typically has a length in the order of ms, an often used value is $L = 4096 \mu\text{s}$.

Figure 4.6 shows this process in a simplified way for 5 trigger neutrons, based on the pulsetrain already shown in this chapter¹⁴. From the resulting distributions, three values are calculated, called Singles (S), Doubles (D), and Triples (T):

$$S = \sum_{i=0}^{n_{max}} A(i) = \sum_{i=0}^{n_{max}} RA(i) \quad (4.11a)$$

$$D = S \cdot (f_1 - b_1) \quad (4.11b)$$

$$T = S \cdot \frac{1}{2} \cdot (f_2 - b_2 - 2(f_1 - b_1)b_1) \quad (4.11c)$$

In these equations, f_n and b_n are the factorial moments of the “R+A” and “A” distribution respectively. Singles represent the sum of all trigger neutron events, or simply of all detected neutron events, and relate to the total number of emitted neutrons simply by the detector efficiency ε . Doubles and Triples represent the detected subsequent factorial moments of the combined multiplicity distribution of source neutrons (including spontaneous fission, (α, n) reactions and multiplication). Similar to S , they differ from the theoretical distributions due to detection effects, depending on efficiency to higher powers and “gate fractions”, the latter will be discussed later in this section.

The theoretical values for S , D and T can be derived with a model that is a basic version of the different processes happening in a multiplicity sample. It takes into account the three properties that are the measurement goal, spontaneous fission rate F , multiplication M , and relation between neutrons from spontaneous fission to neutrons from (α, n) reactions α . Detailed derivations of the model can be found in [Boe85]. The basic model requires several assumptions:

- “Point Model”, where the sample is thought to be concentrated in a single point. This is equivalent to the assumption of spatially homogeneous detector efficiency and multiplication

¹⁴ Coincidence counting methods also use such a shift register approach. In contrast to multiplicity measurements, only the total sums of the number of neutrons in the immediate and delayed detection windows are stored in two scalars.

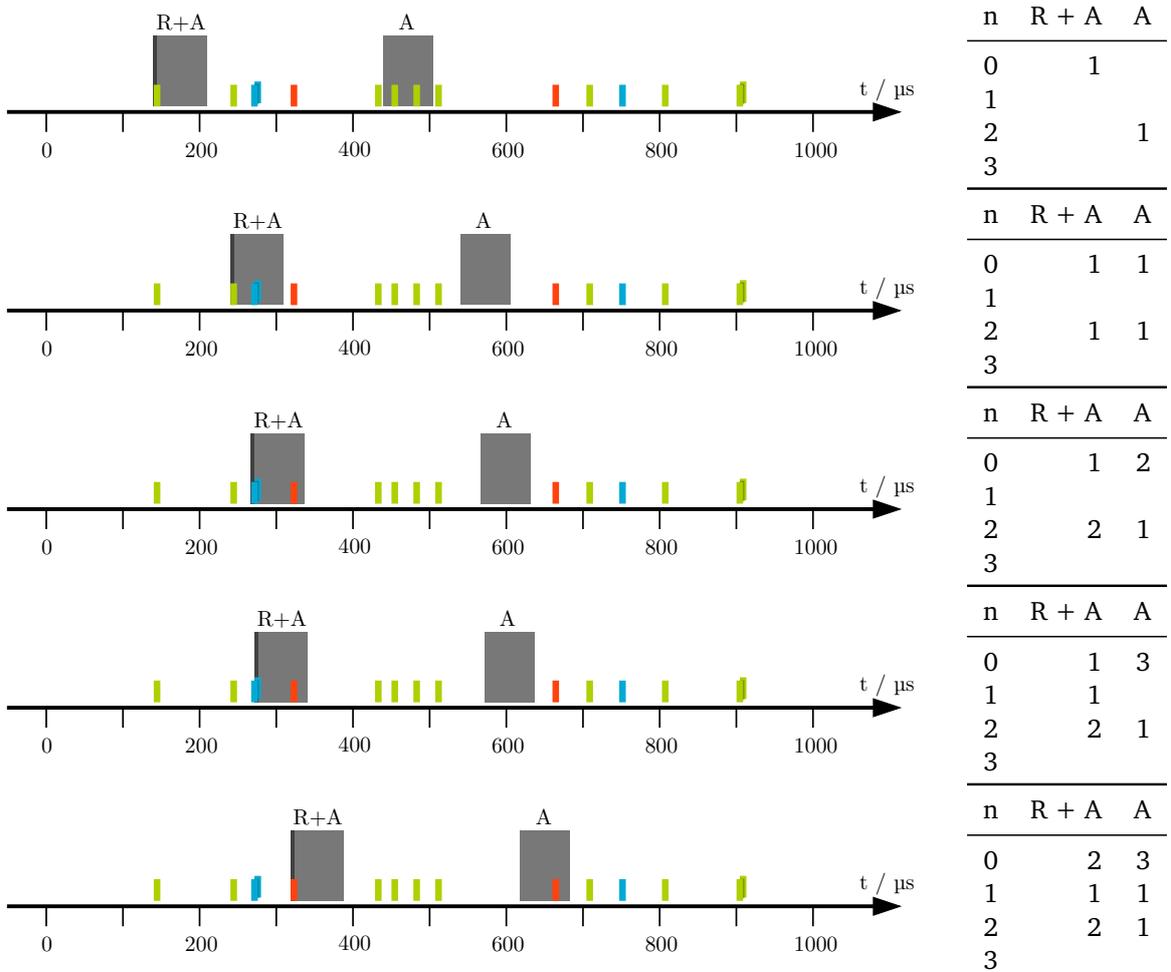


Figure 4.6.: Transformation of a neutron pulsetrain into R+A and A distributions. To achieve better visualization, a short gate length ($32\ \mu\text{s}$) and short long delay ($300\ \mu\text{s}$) was used. In real measurements, both length of the gate and space between gates would be larger.

- Superfission concept: Neutrons from a decay, (α , n) reactions and subsequent multiplication are assumed to occur at the same time.
- Similar energy distributions for spontaneous fission and (α , n) reactions
- Neutron capture (without fission) in sample is negligible
- There is no correlation between neutron energy and neutron number for single fission events
- Die-away time can be described by a single exponential function

Several extensions to the model have been developed over time, which relax requirements for some of the assumptions (e.g. [Cro+12], [GK15]). This section will focus on the non-extended version. The spontaneous fission rate of a sample can be described as

$$F = m_{pu240} \cdot F_{Pu240} \quad (4.12)$$

where F_{Pu240} is the spontaneous fission rate of ^{240}Pu as described above and m_{pu240} is the effective ^{240}Pu mass in the sample.

Given the weight fractions of ^{238}Pu as f_{238} , ^{240}Pu as f_{240} , ^{242}Pu as f_{242} , the effective mass relates to the total plutonium mass via

$$m_{pu240} = m_{pu} (2.52 \cdot f_{238} + f_{240} + 1.68 \cdot f_{242}) \quad (4.13)$$

For the superfission concept, where neutrons from a spontaneous fission decay and (α , n) reactions are assumed to occur at the same time, a “superfission rate” can be defined, adding the rate of neutrons from (α , n) reactions to the spontaneous fission rate:

$$R_{SuF} = F + r_{n,\alpha} = F(1 + \alpha \nu_{sf,1}) \quad (4.14)$$

These superfissions will itself have a multiplicity distribution. The derivation of this distribution is shown in [Boe85]. The expressions for the factorial moments of the mul-

tiplicity distribution, the number of neutrons emitted per superfission, are the following:

$$\nu_1 = \frac{M}{1 + \alpha \nu_{SF,1}} \nu_{SF,1} (1 + \alpha) \quad (4.15a)$$

$$\nu_2 = \frac{M}{1 + \alpha \nu_{SF,1}} \left(\nu_{SF,2} + \frac{M-1}{\nu_{IF,1}-1} \nu_{SF,1} (1 + \alpha) \nu_{i,2} \right) \quad (4.15b)$$

$$\begin{aligned} \nu_3 = \frac{M}{1 + \alpha \nu_{SF,1}} & \left(\nu_{SF,3} \right. & (4.15c) \\ & + \frac{M-1}{\nu_{IF,1}-1} (3 \nu_{SF,2} \nu_{IF,2} + \nu_{SF,1} (1 + \alpha) \nu_{IF,3}) \\ & \left. + 3 \left(\frac{M-1}{\nu_{IF,1}-1} \right)^2 \nu_{SF,1} (1 + \alpha) \nu_{IF,2}^2 \right) \end{aligned}$$

In this equation, $\nu_{SF,1/2/3}$ are the first three factorial moments of the multiplicity distribution of spontaneous fission of ^{240}Pu , $\nu_{IF,1/2/3}$ are the first three factorial moments of the multiplicity distribution of induced fission of ^{239}Pu . $\nu_{SF,1}$ is equal to $\bar{\nu}$ of the spontaneous fission of ^{240}Pu , similarly for $\nu_{IF,1}$. Combining R_{SuF} with the moments of equation (4.15) gives equations for Singles, Doubles and Triples.

$$S = R_{SuF} \cdot \varepsilon \cdot \nu_1 \quad (4.16a)$$

$$D = R_{SuF} \cdot \frac{\varepsilon^2 f_d}{2} \cdot \nu_2 \quad (4.16b)$$

$$T = R_{SuF} \cdot \frac{\varepsilon^3 f_t}{6} \cdot \nu_3 \quad (4.16c)$$

Now, the detector effects have to be taken into account. For Singles, the value is multiplied by the detector efficiency. For Doubles, the value is multiplied by the square of the detector efficiency, and the Doubles gate fraction f_d . Similarly, Triples include the Triples gate fraction f_t , and a cube of the efficiency. The gate fractions describe the fact that only a fraction of correlated neutrons can be detected in a finite gate length, other neutrons might not be measured because they are detected later. A simple approximation to calculate gate fractions, valid if the die-away of neutrons in the detector can be described as a single exponential function, is the following:

$$f_d = e^{-\frac{t_{predelay}}{\tau}} (1 - e^{-\frac{G}{\tau}}) \quad (4.17)$$

In the equation, $t_{predelay}$ is the time for the pre-delay before the shift register gate is opened, and t_{gate} the gate length. τ is the die-away time in the detector. The Triples

gate fraction in this approach would be $f_t = f_d^2$. Often, the assumption of single exponential die away is not true. Taking D_∞ as the Doubles and T_∞ as the Triples of such an infinite gate length, the gate fraction rather would be:

$$f_d = \frac{D}{D_\infty} \quad (4.18a)$$

$$f_t = \frac{T}{T_\infty} \quad (4.18b)$$

While infinite gates are impractical, one can however compare relatively long gate times to the standard time used for the measurement. Using such long gates, the gate fractions can be estimated using known samples.

After equations (4.14) and (4.15) are inserted in equations (4.16), and the system of equations is solved for F , M , and α , it is possible to use values of S , D and T that result from measurement and pulsetrain analysis, equations (4.11), to eventually calculate the plutonium mass. Eliminating F and α leads to a cubic equation for M .

$$0 = a + bM + cM^2 + M^3 \quad (4.19a)$$

$$a = \frac{-6T \nu_{sf,2}(\nu_{if,1} - 1)}{\varepsilon^2 f_t S (\nu_{sf,2} \nu_{if,3} - \nu_{sf,3} \nu_{if,2})} \quad (4.19b)$$

$$b = \frac{2D(\nu_{sf,3}(\nu_{if,1} - 1) - 3\nu_{sf,2} \nu_{if,2})}{\varepsilon f_d S (\nu_{sf,2} \nu_{if,3} - \nu_{sf,3} \nu_{if,2})} \quad (4.19c)$$

$$c = \frac{6D \nu_{sf,2} \nu_{if,2}}{\varepsilon f_d S (\nu_{sf,2} \nu_{if,3} - \nu_{sf,3} \nu_{if,2})} - 1 \quad (4.19d)$$

M can be calculated by solving the cubic equation (4.19a). This value can be used to solve the system for F and α :

$$F = \frac{\frac{2D}{\varepsilon f_d} - \frac{M(M-1)\nu_{if,2}S}{\nu_{if,1} - 1}}{\varepsilon M^2 \nu_{sf,2}} \quad (4.20)$$

$$\alpha = \frac{S}{F \varepsilon \nu_{sf,1} M} - 1 \quad (4.21)$$

From this, the effective ^{240}Pu mass and the total plutonium mass can be calculated according to equations (4.12) and (4.13).

4.5 Role of α Decays

Most materials that undergo spontaneous fission mainly decay via α -decay, some via β -decay. Probabilities for the latter two decay modes usually are many orders of magnitude higher than for the spontaneous fission event. The largest branching ratio for spontaneous fission among commonly measured isotopes is that of ^{252}Cf , where 3.092(8)% of the decays are fission events. Table 4.4 shows typical α -decay energies.

Table 4.4.: Important kinetic energy levels of α -particles emitted by decay of various isotopes. Intensities are normalized to all α -reactions of the respective isotopes.

Isotope	Energy / MeV	Intensity / %	
^{238}Pu	5.4563(3)	28.98(10)	[BT07, p. 746]
^{238}Pu	5.49903(20)	70.91(10)	
^{239}Pu	5.1055(8)	11.94(7)	[BT14a, p. 241]
^{239}Pu	5.1443(8)	17.11(14)	
^{239}Pu	5.15659(14)	70.77(14)	
^{240}Pu	5.12368(23)	27.10(10)	[BT06, p. 2676]
^{240}Pu	5.16817(15)	72.80(10)	
^{242}Pu	4.8582(10)	23.4(6)	[BT15, p. 227]
^{242}Pu	4.9023(10)	76.5(6)	
^{241}Am	5.44280(13)	13.1(3)	[Bas06, p. 2381]
^{241}Am	5.48556(12)	84.8(5)	

As ionizing particles, the range of α -particles with energies as those mentioned in table 4.4 in matter is very short. The particles would never reach a He-3 tube in a neutron multiplicity counter. However, neutrons produced through α -particles in (α, n) -reactions can contribute significantly to the neutron source term in cases where low-Z materials are present. Examples for low-Z materials in close range to fissile materials could be oxygen, fluorine or beryllium in different applications. The actual reaction rate is relatively small, about one in a million α -particles will undergo an inelastic interaction that releases a neutron. But as discussed above, typical branching ratios for α -decay and spontaneous fission show inverse properties - spontaneous fission occurs only in the order of one in a million decays and hence the neutron rates from (α, n) reactions and spontaneous fission have approximately the same order of magnitude. Thus, adequate treatment of (α, n) reactions has to be included in any simulation of neutron multiplicity counting. This includes an estimate of the neutron yield, the number of neutrons produced per α -particle, for an energy range up to several MeV. Also the energy distribution of the resulting neutrons should be adequately reproduced in simulations. These two effects will be discussed in the following.

4.6 Neutron Yield from (α , n) Reactions

There are mainly two ways to provide neutron yield data, which will be described in this section. First, one can rely on experimental data of experiments that directly measured the neutron yield. Second, it is possible to calculate yield as a function of energy during slowing down from measured inelastic cross sections and α particle stopping powers of materials.

Two comprehensive measurements of several sample materials were carried out by [BC79] and [WS82], mainly aiming for thick target yields. Thick targets do not have to be large, only bigger than the typical range of α -particles. Then, nearly all α particle of an external particle beam will be stopped inside, only very few will leave the sample again because of being scattered backwards. [BC79] used a Van-de-Graaff generator to produce α -particles of different energies, and bombarded targets inside a graphite sphere. Neutrons were detected using eight BF_3 neutron detectors placed in the sphere, and several low-Z samples were measured. In [WS82], measurements based on a slightly more complex setup have been carried out. They also include a thorough evaluation and description of uncertainties. The measurement has been done with selected α -particle energies. Afterwards, a cubic function has been fit to the results, from which the complete range from 3.0 MeV to 10 MeV has been interpolated in 0.1 MeV steps. Beyond these comprehensive works, [BB73] measured a very pure sample of ^{238}Pu oxide, one of the few references that can be found for measurements of fissile material oxides.

Besides direct experiments, neutron yields can also be calculated if cross section data for the interaction are available (cf. [FOK68], [LP77], [Hea+89]). While such data might be a result of measurements similar to those mentioned above, they could also come from other sources. Typically, many different cross sections are used by these references to calculate yield.

In the following, the relevant steps for such a calculation will be discussed. Notation and derivation for neutron yield is similar to those given in [LP77; Wes79; CM13]. For a single isotope i undergoing (α , n) reactions, the chance of creating dY neutrons during every small step dx that the particle travels can be described as:

$$dY_i = n_i \sigma_i(E(x)) dx \quad (4.22)$$

n_i describes the number density, σ_i the reaction cross section for (α , n) reactions of the isotope. To get the actual yield, one has to integrate the reaction cross section over the distance traveled by the α particle in matter, R .

$$Y_i = \int_0^R n_i \sigma_i(E(x)) dx \quad (4.23)$$

The total distance depends on the initial energy and the energy loss due to electromagnetic interactions on the way. The effect of such interactions is typically summarized as the stopping power of an element. It is possible to describe infinitesimal energy loss by stopping power using the following relation:

$$dE = -S_e(E)dx \quad (4.24)$$

As an effect mostly dominated by atomic interaction, stopping powers are equal for different isotopes of the same element. Hence in the equation, S_e is the elemental stopping power of the element e . Combining both equations, the neutron yield per single α particle for a given isotope can be calculated:

$$Y_i(E_\alpha) = n_i \int_{E_{\alpha\text{-decay}}}^0 -\frac{\sigma(E')}{S_e(E')} dE' \quad (4.25)$$

Here, the index e is chosen with respect to the isotope i . Equation (4.25) gives the yield only for a single α -decay energy $E_{\alpha\text{-decay}}$, and a single isotopic cross section and elemental stopping power. To calculate the yield of a material mixtures, several summations have to be carried out. First, one has to calculate the total stopping power S of the mixture using the Bragg additivity rule, which simply sums elemental stopping power over all elements e in the mixture (denoted as set Mix)¹⁵.

$$S(E) = \sum_{e \in \text{Mix}} w_e S_e(E) \quad (4.26)$$

Second, isotopic yields have to be summed over all isotopes undergoing α -decay (denoted as set AD), all α -decay energies of each such isotope a (denoted as $E_{\text{level},a}$) and finally over all isotopes i in the mixture that have a non-zero chance to undergo (α, n) reactions (denoted as set AN). All yield integrals need to take into account the total stopping power for the mix as previously defined.

$$Y = \sum_{a \in AD} n_a \frac{\ln(2)}{T_{1/2,a}} \left(\sum_{E_{\alpha\text{-decay}} \in E_{\text{level},a}} f_{E_{\alpha\text{-decay}}} \left(\sum_{i \in AN} n_i \int_{E_{\alpha\text{-decay}}}^0 -\frac{\sigma_i(E')}{S(E')} \cdot dE' \right) \right) \quad (4.27)$$

¹⁵ The Bragg additivity rule is a simplification and can introduce small errors in the order of several percent depending on the compound. These errors are neglected here. It is possible to correct for these errors by additional correction factors for compound materials. However, the common sets of stopping power data do not provide factors for the most relevant compound PuO_2 [ICR93].

Here, w_α is the weight fraction of the decaying isotopes, $T_{1/2,\alpha}$ its half-life, and f_{E_α} the branching ratio for a decay with α particle energy E_α .

Commonly, the stopping power part related to electronic interaction is described by the Bethe (or Bethe-Bloch) formula. However, at the relevant energies, the low MeV range, the formula is not precise enough, even when various correction terms are applied. For these energy ranges, there exist various tabulated data sets, for example in [Zie77], in the ASTAR database [Ber+05], mainly in form of instructions and data collection in [ICR93] or directly calculated with the SRIM software [ZBZ08]. The tabulations are typically based on parameterized fits of experimental data, often combined with a smooth transition towards the Bethe formula. The transition usually takes place at an energy of about 10 MeV. The second part of stopping power, nuclear stopping, plays only a minor role in the energy range of the radioactive decays as described above, it is not included in the calculation of stopping power for ONMS.

The calculation of yield in the software described in the following chapter is based on a relatively large, current evaluation of cross sections, the JENDL/AN-2005 data set, which has been put together by Murata et al. [MMS06]. It is available in the ENDF-6 file format [HT09]. Generally, the authors used available data and software called mEXIFON to produce their cross sections. For every cross section, the authors have calculated thick target neutron yields using the formula described above, and stopping powers based on [Zie77]. The authors of [MMS06] claim that they have adjusted the cross sections using these yields to closely match yield data from [BC79] and [WS82]. It has been decided to use these cross sections for this work, as they are the most recent comprehensive evaluation, and freely available online. A detailed description of included isotopes and the necessary steps to make the evaluation usable for ONMS is given in appendix A.

4.7 Energy Spectrum of Produced Neutrons

The (α, n) reaction can be described using a classical kinematic description of a scattering process where incoming particles differ from outgoing particles, as given for example in [Fod71]. Incoming particles are the α -particle and the target nucleus t , outgoing a neutron and the residual nucleus r . The difference in mass of incoming and outgoing particles is described as the Q-value of the reaction.

$$Q = (m_\alpha + m_t - (m_n + m_r)) \cdot c^2 \quad (4.28)$$

where m_α and m_n describe masses of α -particle and neutron, and m_t and m_r describe masses of target and residual respectively. In addition to energy generated or lost due to the change in mass, the residual nucleus can be left in an excited state with energy E_{ex} . Both can be summarized to

$$Q_{ex} = Q - E_{ex} \quad (4.29)$$

The reaction details are calculated in the center of mass frame. All x' (or \vec{x}') values denote center of mass frame values. If the target is at rest in the laboratory frame, the velocity of the center of mass in the lab frame is:

$$\vec{u} = \frac{m_\alpha}{m_\alpha + m_t} \vec{v}_\alpha \quad (4.30)$$

Using this and the relations $\vec{v}'_\alpha = \vec{v}_\alpha - \vec{u}$ and $\vec{v}'_t = \vec{u}$, the kinetic energies for α -particle and target in the center of mass frame can be formulated:

$$E'_\alpha = \frac{1}{2} m_\alpha \vec{v}'_\alpha{}^2 = \frac{1}{2} m_\alpha (\vec{v}_\alpha - \vec{u})^2 = \frac{1}{2} m_\alpha \left(\frac{m_t}{m_\alpha + m_t} \right)^2 \vec{v}_\alpha{}^2 = \left(\frac{m_t}{m_\alpha + m_t} \right)^2 E_\alpha \quad (4.31)$$

$$E'_t = \frac{m_\alpha m_t}{(m_\alpha + m_t)^2} E_\alpha \quad (4.32)$$

The resulting kinetic energy of the neutron in the center of mass frame can be described as

$$E'_n = \frac{m_r}{m_n + m_r} (E'_\alpha + E'_t + Q_{Ex}) \quad (4.33)$$

It should be noted that this equation is independent of angle θ between outgoing neutron and incoming α -particle. Scattering in the center of mass frame is isotropic. To get back the energy in the laboratory frame, the following relation for the velocities has to be used:

$$\vec{v}_n{}^2 = \vec{v}'_n{}^2 + \vec{u}^2 + 2\vec{v}'_n \cdot \vec{u} = v_n'^2 + u^2 + v'_n u \cos \theta'_n \quad (4.34)$$

Maximal and minimal velocity (now only considering the magnitude) are those where the angle between the reaction products in the center of mass frame is minimal / maximal, or $\cos \theta'_n = \pm 1$. Inserted in the previous equation:

$$v_{n,\min/\max} = \sqrt{v_n'^2 + u^2 \pm 2v'_n u} \quad (4.35)$$

$$E_{n,\min/\max} = \frac{1}{2} m_n v_{n,\min/\max}^2 = \frac{1}{2} m_n (v_n'^2 + u^2 \pm 2v'_n u) = \frac{1}{2} m_n (v'_n \pm u)^2 \quad (4.36)$$

$$E_{n,\min/\max} = \frac{1}{2} m_n \left(\sqrt{\frac{2}{m_n} \frac{m_r}{m_n + m_r} (E'_\alpha + E'_t + Q_{ex})} \pm \frac{m_\alpha}{m_\alpha + m_t} \cdot \sqrt{\frac{2E_\alpha}{m_\alpha}} \right)^2 \quad (4.37)$$

or with E'_α from equation (4.31) and E'_t from equation (4.32) inserted

$$E_{n,\min/\max} = \frac{1}{2}m_n \left(\sqrt{\frac{2}{m_n} \frac{m_r}{m_n + m_r} \left(\left(\frac{m_t}{m_\alpha + m_t} \right)^2 E_\alpha + \frac{m_\alpha m_t}{(m_\alpha + m_t)^2} E_\alpha + Q_{ex} \right)} \right. \\ \left. \pm \frac{m_\alpha}{m_\alpha + m_t} \cdot \sqrt{\frac{2E_\alpha}{m_\alpha}} \right)^2$$

This describes the minimal and maximal energy of a neutron produced by an α -particle with energy E_α for a single reaction with Q_{ex} , the combination of Q-value and particular residual excitation energy as defined in equation (4.29). The neutron energy in the laboratory frame is evenly distributed between $E_{n,\min}$ and $E_{n,\max}$, because the scattering in the center of mass frame is isotropic, and can be shown as follows. First, equation (4.34) is used to write a function of the neutron energy of the scattering angle in the center of mass:

$$E_n = \frac{1}{2}m_n \left(v_n'^2 + u^2 + v_n' u \cos \theta_n' \right) \quad (4.38)$$

Building the derivative with regard to $(\cos \theta_n')$ yields:

$$dE_n = \frac{1}{2}m_n v_n' u d(\cos \theta_n') \quad (4.39)$$

As due to isotropic scattering, every value $d(\cos \theta_n')$ is equally probable, all dE_n are equally probable, too.

In the following, the neutron energy distribution will be called $f(E_\alpha, E_n, Q_{ex})$, a function that is 0 below $E_{n,\min}$ and above $E_{n,\max}$. Between these limits, it has the value $\frac{1}{E_{n,\max} - E_{n,\min}}$, so its integral is 1.

Typically, different excitation energies are possible for the residual nucleus. Although the ground state is the most likely level, higher levels are reached with non-negligible branching ratios. The reactions of a residual nucleus with N_{ex} excitation levels (including the ground state) can be described by the partial reaction cross section $\sigma_n(E_\alpha)$ for each excitation level $n \in N_{ex}$, where Q_n is equal to Q_{ex} for that particular excitation level (cf. equation (4.29)) the combination of Q-value and the excitation energy $E_{ex,n}$ of this level. Such partial reaction cross sections are for example in the JENDL/AN-2005 cross section data set.

Summing the products of the partial reaction cross sections with the corresponding distributions $f(E_\alpha, E_n, Q_{ex})$ over the different levels allows for the definition of a general cross section:

$$\sigma(E_\alpha, E_n) = \sum_{n=1}^{N_{ex}} \sigma_n(E_\alpha) f(E_\alpha, E_n, Q_n) \quad (4.40)$$

Using this general cross section, the probability of producing a neutron with energy E_n during a small step dx along the trajectory of an α -particle can be calculated similar to equation (4.22):

$$dP(E_\alpha, E_n) = n\sigma(E_\alpha, E_n)dx \quad (4.41)$$

with n the number density of the target atoms and E_α the momentary energy of the particle during the step.

The energy is reduced along the path due to the stopping power $S(E)$, allowing to express the previous equation in terms of infinitesimal energy changes:

$$dP(E_\alpha, E_n) = -n\sigma(E_\alpha, E_n)\frac{1}{S(E)}dE \quad (4.42)$$

To get the distribution for neutron energies of a single α -decay with energy $E_{\alpha\text{-decay}}$, one has to integrate from $E_{\alpha\text{-decay}}$ until the particle is stopped ($E = 0$)¹⁶.

$$P(E_{\alpha\text{-decay}}, E_n) = \int_{E_{\alpha\text{-decay}}}^0 -n\frac{\sigma(E_{\alpha\text{-decay}}, E_n)}{S(E)}dE \quad (4.43)$$

$$(4.44)$$

The total neutron energy distribution of a material can be calculated by summing P over all possible target materials and source α -energies.

¹⁶ Often, integrals are expressed with 0 as lower and $E_{\alpha\text{-decay}}$ as upper bound, implicitly changing the sign of the integrand.



5 Description of the Simulation Software

The main part of this doctoral research project consisted of the development of an application for simulating neutron multiplicity measurements, “Open Nuclear Multiplicity Simulation” (ONMS). Simulations of neutron multiplicity measurements can be divided generally into two steps: A Monte Carlo based neutron transport step, which simulates neutron sources as well as the physical processes and particle transport in a detector geometry, and a second step to analyze the pulsetrain of detected neutron events produced by the first step. The implementation of all parts of the first step and the second step will be described now.

There are several reasons why such simulations are useful. Simulations are very often the first step of any new hardware development. In 1991, [Rei+91] wrote that “Computer calculations employing Monte Carlo codes for neutron transport have been used to optimize the design of ^3He neutron coincidence detector systems” (p. 493). Beyond optimization and development, simulations can also be used to detect weaknesses in detectors, e.g. regarding possible cheating in arms control verification. With validated physical simulation tools, it is often cheaper and easier to simulate many different configurations, instead of taking measurements (cf. for example [GC13], [LPT09], [Nak+14]).

While other tools to do these types of simulations exist, the development of ONMS is worth pursuing for several reasons. In the description of the neutron multiplicity benchmark exercises, the authors mention that “few codes are available today to perform the first step and they are not available to everybody” [PS06, p. 2], indicating a need for more codes and especially for higher availability. The first step mentioned in the preceding quote is the same first step as described above.

ONMS is open source software, and fulfills the three criteria listed in chapter 3. As such, it is an example open source software with an arms control purpose, available to everyone. Future uses include not only simulations, but also its use as a case study for further developing the concept of open source software for arms control applications. Beyond being open source, it also fills some technical gaps that existing codes have, which will be shown in this chapter. Beneficial for the development of ONMS was the fact that from the outset of the project, experimental data were available for validating the code from the doctoral research project of Malte Göttsche, ZNF Hamburg [Göt15]. This was especially interesting, because the experimental data were gathered in the context of an arms control application of the technology, nuclear warhead verification.

ONMS was developed as a Geant4 application mainly in C++ [Str13]. Figure 5 shows a conceptual overview of the C++ components of ONMS. Many parts were developed from scratch, as there were no existing functions available, or existing functions were not of the required quality. Apart from implementing required physical capabilities, the software was also designed in a way that should allow for easy further extension or replacement of individual parts and models. It was also carefully checked that all required tools, libraries and data sets are openly available.

Besides components developed in C++, the source folder of ONMS also includes auxiliary files. There is a folder that holds geometry files for the two detectors and all samples that were calculated as part of this thesis. These files are not required for the code, but could serve as examples for future usage. Another folder contains Geant4 macro files to carry out simple calculations of all samples, and also macro files for visualization.

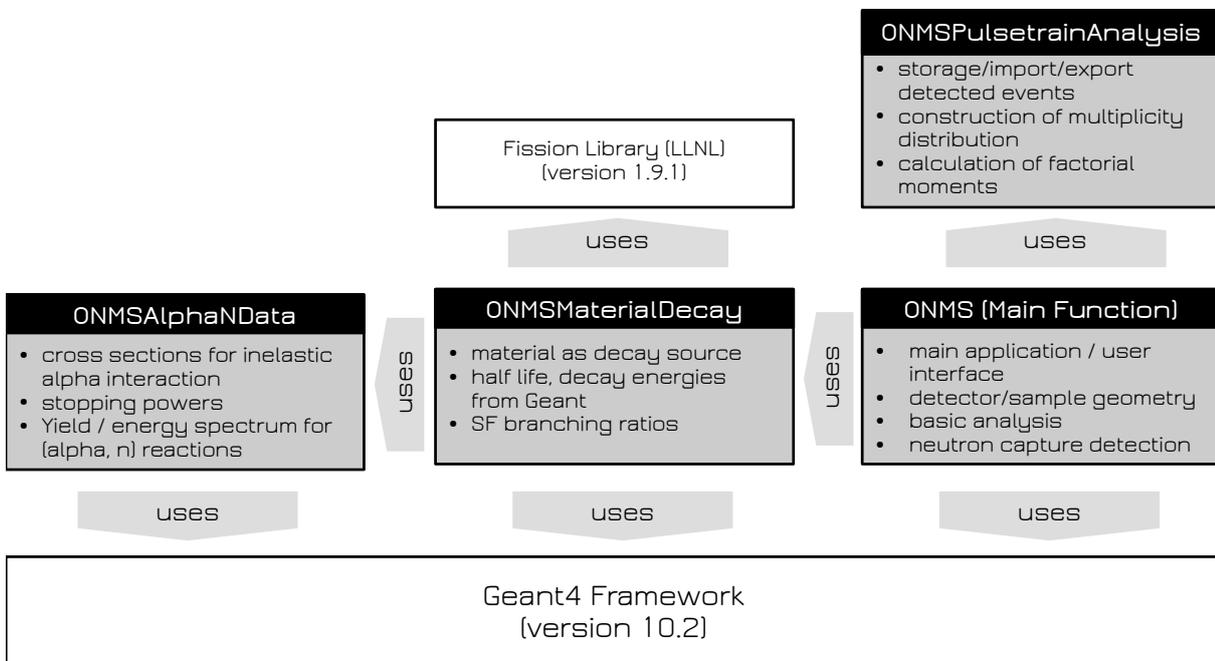


Figure 5.1.: Basic library structure of ONMS. Grey boxes are own developments, white boxes show external tools.

Currently, ONMS consists of 93 different source files and more than 8000 lines of code, excluding the fission library. The software is available at <https://github.com/nuclearfreesoftware/onms>, allowing internet users to download and use it, access its source code and give them the option to add modifications.

The content of this chapter is split in four sections. First, the Geant4 framework will be described in more detail, together with a summary of the Monte Carlo neutron transport capabilities and cross sections used in ONMS. In a second section, the global structure and main functions of ONMS will be described. This is followed by sections three and

four, which are specific descriptions of two sublibraries of ONMS, one for adequately simulating the particle source related to a sample, the other to carry out the analysis of the detected pulsetrain.

5.1 Geant4 and Neutron Particle Transport

Geant4 is a software framework for Monte Carlo simulations of particle interactions in different geometries [Ago+03; All+06]. Compared to common tools used in the arms control realm for Monte Carlo simulations (e.g. MCNP), the approach of Geant4 could be described as less “engineering-centered”. It is itself not an application that can be run individually, but a set of libraries which can be used to build specific applications. Also, it does not have a common standard for “input files” to define simulation problems. Instead, the approach is more driven by basic science and experiments. The framework is developed by the Geant4 collaboration, which consists of more than hundred scientists and developers in different institutions around the world¹⁷. The software is capable of simulating many different particles and can employ numerous physical processes. Often, it is even possible to select one of several concurrent implementations of a process to focus on particular problems and theoretical assumptions. For every use case, a new application has to be developed. The development language of Geant4 is C++ [Str13], a simple Python API exists as well. A detailed description of the development process for Geant4 applications can be found in the User’s Guide for Application Developers [Gea15a]. Background on the physics of Geant4 is given in the Physics Reference Manual [Gea15b].

The main focus of Geant4 applications is on problems in the field of high energy physics. It has been used for example for simulations of ALICE [Hřı+11] and ATLAS [Rim+04], detectors at the Large Hadron Collider at CERN. Some past examples of usage of Geant4 for arms control and nuclear safeguards applications exist, too:

- Simulation of nuclear resonance fluorescence as a non-destructive assay method for nuclear safeguards [Hay+10]
- Simulation of fast neutron detection for non-destructive assay methods [NCP13]
- Simulation of muon tomography of spent fuel containers [Jon+13]

A common theme of these (and other) examples is the use of Geant4 for more “exotic” fields of physics, which consider particles other than neutrons.

The Geant4 framework has been selected as a basis for ONMS as it fulfills the open source criteria defined in chapter 3. The copyright license used by the collaboration has been drafted specially for Geant4 (<http://geant4.web.cern.ch/geant4/license/LICENSE.html>). It is very similar to some common open source licenses, and should fulfill the standard of the Open Source Initiative for open source licenses.

¹⁷ <http://geant4.cern.ch/collaboration/members.shtml>

The main feature directly provided by Geant4 is the transport of neutrons at relevant energies. No modifications have been made to this functionality. Relevant for the problem is the neutron energy range between 0 MeV to 20 MeV. In nuclear engineering, neutrons with energies of the order of MeV would be considered to be in a high energy range, however for Geant4 this is a very low energy¹⁸. Geant4 implements special classes for the treatment of low energy neutrons. While originally being developed for neutron interaction as `NeutronHP` classes, the most recent version of Geant4 (10.2) has a unified version for multiple particles called `ParticleHP`. HP stands for High Precision. If used, these classes load tabulated point-wise nuclear cross section data from a special library and use these to simulate interactions. The Geant4 consortium publishes a neutron cross section library for `ParticleHP` called G4NDL, currently in version 4.5. The data in this library is cited as data from the ENDF/B-VII evaluation from NNDC (“comes largely from ENDF/B-VII”¹⁹).

However, the library only includes isotopes of elements that naturally occur on earth ($Z \leq 92$), hence it does not contain any transuranic actinides, including plutonium²⁰. A different library has been compiled by scientists of the Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT) in Spain [Men+12; Men+14]. They converted many of the major evaluations (ENDF-B/VII.0 from the United States, JEFF-3.1/Europe, BROND-2.2/Russia, JENDL-4.0/Japan, CENDL-31/China) to files compatible with `ParticleHP`, and included also isotopes of elements with $Z > 92$. The libraries are freely available from the website of the International Atomic Energy Agency (<https://www-nds.iaea.org/geant4/>). No specific copyright or usage license is given, but the public availability on the website of an international organization should suffice to fulfill at least the first open source criteria of access without limitations. They are provided and as text files, which are human readable.

Unless otherwise stated, the JEFF-3.1-based cross section library was used for all calculations of this work. Nevertheless, it is easily possible to use other cross section libraries. ONMS will output a warning when there is an indication that another library is used.

As discussed in the physics section, at very low neutron energies, interaction processes between neutrons and molecules also need to be taken into account. The G4NDL library includes elastic thermal scattering cross sections, which are available in their base evaluation. Four are primarily relevant for neutron multiplicity simulations: Hydrogen in polyethylene, graphite, and depending on the problem beryllium oxide and beryllium metal. The respective files had to be copied to the CIEMAT libraries according to the instructions on the website. To use thermal scattering cross sections, a special Physics List has to be written (cf. section 5.2).

¹⁸ Low energy neutrons in the Geant4 context are all neutrons with energy below 20 MeV.

¹⁹ From Readme file of the library.

²⁰ Additionally, using materials with $Z > 92$ leads to a program exception for standard Geant4 applications. To avoid this limitation, it is necessary to set the environment variable `AllowForHeavyElements` to 1, which is automatically done by the ONMS main application.

Comparison tests of these thermal cross sections with MCNPX have been carried out by [Men+14]. They show very good agreement for resulting energy spectra of neutrons interacting with a block of polyethylene, which is the main moderating and structural material of most ^3He based neutron multiplicity counters.

5.2 Global Functions of Open Neutron Multiplicity Simulation

The central process is governed by the main source file `onms.cc` and several classes defined in the main folder. The main function of `onms.cc` instantiates all C++ classes that are required for a Geant4 application. Mostly, the classes were specifically developed for ONMS. The classes are listed in table 5.1 in the order of loading. Details will be discussed in this section.

Table 5.1.: Central ONMS classes and short descriptions.

Class Name	Man- datory [*]	Specifically Developed	Main function
ONMSRunManager	x	x	start / end run
ONMSDetectorConstruction	x	x	material and geometry definition
PhysicsList [†]	x	x / -	simulated particles, physics models and processes
ONMSAnalysisManager	-	x	store detected events in pulsetrain
ONMSTrackingAction	-	x	keep statistics on particle source
ONMSSteppingAction	-	x	store neutron capture events
ONMSEventAction	-	x	run-time information display
ONMSRunAction	-	x	output results at end of each run
ONMSPrimaryGeneratorAction	x	-	sample and create source particles
G4VisManager	-	-	visualization routines
G4UImanager	x	-	user interface
G4UIExecutive	-	-	helper class for user interface

^{*} An instance of each of these mandatory classes listed is necessary for every Geant4 application to be executable.

[†] Not a single class name, several different classes are possible, refer to text for more information.

The `ONMSRunManager` class is derived from `G4RunManager`. It initializes the main functions and controls event loops, which carry out Monte Carlo particle tracking. During program execution, a set of events, typically equivalent to a measurement is summarized as a “run”. In common Monte Carlo codes, users define number of particles to be simulated for every run. The ONMS extensions gives users an additional option: They can define a specific measurement time in the beginning, called “runtime”²¹. From this time, the software calculates the required particles based on the source activity. The

²¹ In the following, this is not the actual time the program takes to run on a computer, but the length of the simulated measurement.

simulation can be started with the messenger command `/ONMS/run/beamOnRuntime`²². The approach removes the necessity for the user to carefully calculate source activities using tables of nuclear decay data and the source material composition before using it as an input for the simulation. This automation makes the process easier and less error-prone.

`ONMSDetectorConstruction` is the class responsible for defining used materials and setting up the geometry of the problem. Although it bears detector in its name, it loads both the description of the actual neutron multiplicity detector and the samples to be measured. It is derived from `G4VUserDetectorConstruction`, which is an abstract base class provided with the Geant4 source. Instead of defining materials and geometry by C++ directives, `ONMSDetectorConstruction` makes use of `G4GDMLParser`, a set of routines that allow to store this information in external XML files. The parser is part of Geant4, and GDML is an open description standard defined by a group at CERN [Chy+06; GDML]. The use of GDML allows users to make changes to material or geometry without knowledge of C++ and without recompilation of the program. The `ONMSDetectorConstruction` class was written in a way that allows the specification of two different GDML files, a global detector description and a sample description, which can be exchanged individually. This reflects typical usage of simulating several different samples using the same detector. The separation of detector and sample files allows users to keep one file for many calculations, while changing only the second, again reducing possible errors and making development easier. One example GDML file of a simple plutonium container is listed in Annex B.

Every Geant4 application needs to have a Physics List, which defines all particles that could be tracked. It also defines processes and models that govern particle interaction with materials in the problem geometry. In Geant4, a process is a method for figuring out when an interaction takes place. For example, it could be a set of interaction cross sections. When a specific interaction has been sampled, the particular model of this interaction is called. The model then simulates the actual interaction, and the production of secondary particles and / or excited states of targets [Gea15b, p. 3]. Three different options for loading a Physics List are provided by ONMS, but only the first should be used for substantive calculations. The others are provided mainly for debugging and testing.

- `ONMS_QGSP_BIC_HP_Thermal` is the default option, including all required physics processes for ONMS, high precision neutron cross section and thermal scattering.
- `QGSP_BIC_HP` is the `QGSP_BIC` version with high precision neutron models, but without thermal scattering.
- `QGSP_BIC` does not include any high-precision neutron models. It is offered as a third option mainly to compare results with the first.

²² Messenger commands are the typical Geant4 user interface. Geant4 provides ways to define new commands. These can be used at runtime to change settings or control program flow.

All are based on the QGSP_BIC Physics List that is one of the pre-defined Geant4 Physics Lists. It defines all particles in the standard model²³. For very high energy hadronic interactions (> 10 GeV) a so-called quark-gluon string model is used, for intermediate energies the Geant4 Binary Cascade. Models and process for low energies, those relevant for ONMS, depend on the specific selection. To use other other Physics Lists, it would be necessary to change the source code and load respective Physics List classes.

The ONMS_QGSP_BIC_HP_Thermal Physics List has a small addition relative to the QGSP_BIC_HP list, as the latter does not yet include thermal neutron scattering. The following lines of code have been added at the function that loads elastic scattering data for neutrons:

```
1  G4NeutronHPThermalScattering* theThermalModel = new
    G4NeutronHPThermalScattering();
2  theThermalModel->SetMaxEnergy(4.0*eV);
3  hel->RegisterMe(theThermalModel);
4  hel->AddDataSet(new G4NeutronHPThermalScatteringData());
```

This code makes sure that thermal scattering data are used for energies below 4 eV. It is not used for all isotopes, but only for those for which cross sections exist and which are named TS_<element>_of_<material>, e.g. TS_H_of_Polyethylene.

The next class loaded, ONMSAnalysisManager, is not based on any Geant4 base class, but was developed completely from scratch. It is a singleton class that manages the global analysis of the simulated problem. It stores information about the emitted source particles and holds an instance of the pulsetrain analysis class which is part of the pulsetrain sublibrary discussed separately in section 5.4. Besides collecting and storing data, methods have been implemented to output results on the screen or in specifically formatted files for further analysis and plotting with other tools.

The loading process is continued by five different “actions”, all classes derived from abstract base classes. The ONMSPrimaryGeneratorAction is the class which generates the source particles for all events. It is based on G4VUserPrimaryGeneratorAction, and mainly serves as an interface between the main program and ONMSMaterialDecaySource. The latter does all calculations for sampling source particles, and is described separately in section 5.3.

The other four classes provide ways to control, monitor and analyze the simulation process of the application at different steps in time. ONMSEventAction is based on G4UserEventAction. It contains a method that is called at the start of each source event to give some informational output during calculations, but does not influence results. ONMSRunAction is based on G4UserRunAction. It has methods that are called at the beginning and end of every run to reset the analysis manager or output its results.

²³ While this might seem to be more than needed for a problem of basically transporting neutrons, it does not affect computation time significantly, as all other secondary particles are “killed” immediately after creation by ONMSTrackingAction and not transported further. This should not affect neutron count rates, as it is very unlikely that a neutron with the common energies in the described problems produces a secondary particle, which in turn would again produce a new neutron.

`ONMSTrackingAction` is based on `G4UserTrackingAction`. It has methods called at the beginning and end of each track. A track in Geant4 is the history of the movement of a single individual particle, either from the initial source or secondary particles produced in reactions. The class is mainly used to store statistics about produced neutrons, both from the user defined source and due to other reactions. `ONMSSteppingAction`, based on `G4UserSteppingAction`, is the most important class for producing results. It is called at every step of a track. If a neutron has “died” in a step (has been absorbed or decayed), the class evaluates where this took place. If it took place in a geometry volume with a name specified by the user (e.g. HE-3-Tube), a detection event is added to the neutron pulsetrain²⁴.

Three more objects instantiated are `G4VisualisationManager`, `G4UIManager` and `G4UI-Executive`, which are all Geant4 default classes that control visualization output and handling of textual input and output for the program.

There are several parameters of ONMS that can be controlled by command line options, e.g. turning thermal neutron scattering cross sections on or off, selecting detector model, selecting sample model and output file names. A full list of command line options can be found in table C.1 in appendix C.

Further settings are given via user interface commands defined in different messenger classes, a common way for Geant4 applications to add interactive elements. A full list of commands is given in appendix C. It is also possible to combine several of these commands in macro files, which can be loaded in different ways. Such macro files can be loaded using the “-b” command line option, which runs them files in “batch”-mode. In this mode, the program terminates after execution of the calculations defined in the macro file. This can be used to script the execution of many different calculations with low user effort. A sample macro file that would be used for the calculations discussed in the following chapters is shown in appendix C.3.

5.3 Material Decay Source

`ONMSMaterialDecaySource` is a sublibrary used for calculation of source particles involved in neutron multiplicity simulations. Early in the development process, the decision to put this in a sublibrary was made as it is a relatively complex part of the total program. The separate library was also designed in a way that allows for using it as a particle source method for programs other than ONMS.

The library development has four main goals. First, the library should allow for an adequate representation of particles coming from spontaneous fission. Second, the library has to take care of the calculation of the (α , n) yield of a source material and produce source neutrons with respective intensities and energy distributions. Third, it should reduce the necessary user input with regard to the construction of the source. Instead of

²⁴ A second volume can be defined as a “lost volume,” for example, the polyethylene block of the detector. Doing so gives information on where neutrons are lost in the geometry.

forcing the user to do precalculations using atomic fractions of materials and tabulated data to specify a source, this is all done during the execution of the program. The user only specifies a material, from which ONMS calculates the quantities and intensities of sources based on existing decay data. And lastly, the source should similarly allow for flexibility to change specific source settings for specific problems in an easy way.

The physical background is described in sections 4.5, 4.6 and 4.7. In terms of programming, `ONMSMaterialDecaySource` consists of a number of classes, and has a structure that is similar to the `G4GeneralParticleSource` (part of Geant4). Implementation of functions for all four goals has been mostly done from scratch, as often no Geant4 functions existed or were inadequate.

Before a simulation is started and source particles can be sampled, the user must provide `ONMSMaterialDecaySource` with three parameters. The user has to specify a material to be used as source material, the material has to be defined in the sample GDML file. As a naming scheme for the rest of this section, the selected material M consists of i_{tot} isotopes indexed with i , where $i \in i_{tot}$. The next selection has to be made regarding geometry and volume of the source. Here, users can either provide a name of a volume defined in the GDML file or select among several different geometrical shapes (point, sphere, cylinder). If the first option is chosen, ONMS automatically extracts related parameters (e.g. radius) from the GDML file and calculates the volume of the source. If the second option is chosen, parameters of the shape and the volume have to be specified manually via messenger commands. `ONMSMaterialDecaySource` stores the source volume only once, so that all individual sources have the same shape and volume. The last main selection to be made is to turn on or off particular decay or reaction types from the list in table 5.2.

Table 5.2.: Source events implemented in `ONMSMaterialDecaySource`.

Reaction	Equation notation	Class name
Neutrons from (α , n) reactions	αn	<code>NMSANSource</code>
Neutrons from spontaneous fission events	SFn	<code>NMSSFSingleIsoDecaySource</code>
γ -particles from spontaneous fission events	SFg	<code>NMSSFSingleIsoDecaySource</code>
α -particles from α -decay	αd	<code>NMSAlphaSingleIsoDecaySource</code>

In the notation of the following equations, all these decay modes should be summarized as the set of source types

$$ST = \{\alpha n, SFn, SFg, \alpha d\} \quad (5.1)$$

For simulations of neutron multiplicity measurements, only the first two sources are needed, hence they are activated by default. The other two were added for additional calculations, but are typically turned off. Besides these three major choices, several minor adjustments can be made. All selections are done via messenger commands. A full list of commands and options is given in appendix C.

Depending on the settings, the full source representation can consist of up to $4 \cdot N_{iso}$ individual sources for a source material consisting of N_{iso} isotopes. During an initialization stage, the intensities of all individual sources are calculated, as well as the total activity of the source.

For a given isotope i , the activity per unit volume of spontaneous fission leading to neutron emission²⁵ is calculated by

$$a_{SFn,i} = n_i \cdot BR_{SF,i} \cdot \frac{\ln 2}{T_{1/2,i}} \quad (5.2)$$

where n_i is the atomic density of isotope i as specified in the source material definition. $BR_{SF,i}$ is the branching ratio for spontaneous fission of isotope i and $T_{1/2,i}$ the total half-life of the isotope. $T_{1/2}$ was taken from the data included with Geant4 in the G4ENSDFSTATE data set in version 1.2.1. This data set is used by Geant4 to define nuclear properties, such as atomic mass and also half-life. The values for half-life are extracted by the Geant4 developers from the most recent ENSDF data set [NNDC]. Except for one isotope, these values match the data collected in table 4.3²⁶. Activities of the reactions SFg and αd are calculated in the same way, using respective branching ratios.

Geant4 provides a second data set which includes half-life - G4RadioactiveDecay. However, several cases could be found where the data for half-life in the most recent version (4.3.1) does not match the data in G4ENDFSTATE, version 1.2.1²⁷. It is not clear why this is the case, as both data sets claim to be based on recent ENSDF evaluations. Hence the latter data set was not used for the calculations described above.

Although Geant4 generally allows for automatic decay of isotopes as source definitions, spontaneous fission processes are not yet implemented. Recently, starting with version 4.3 (November 2015) of the G4RadioactiveDecay data set, decay branching ratios were included in the data files. However, there is no Geant4 defined class that executes such a decay, and method to load decay data in the file G4RadioactiveDecay.cc reads:

```
1 case SpFission:
2 // Not yet implemented
```

For this reason, ONMS implements its own class for source particles coming from spontaneous fission reactions. The SF branching ratios are hard coded in the source file ONMSMaterialDecaySource.cc, based on the data collected from *Nuclear Data Sheets* and listed in the previous chapter (table 4.3).

²⁵ For this source, it is not the intensity of neutrons but of fission events. The number of neutrons is sampled every time a fission takes place.

²⁶ Of the isotopes listed in table 4.3, only ²⁴¹Pu has a half-life that apparently has not yet been updated to the most recent value from *Nuclear Data Sheets*.

²⁷ Discrepancies were found for ²³⁸U, ²⁴¹Pu, ²⁴¹Am and ²⁵²Cf from the list of important isotopes. Only those isotopes listed in table 4.3 were checked, however.

To retain a consistent class structure, also the class for α -decay source particles was newly developed, instead of relying on the Geant4 decay. In this case, branching ratios for α -decays were taken from the G4RadioactiveDecay data set.

The activity of the αn reaction is the neutron yield according to the derivation described in section 4.6. The total activity per unit volume of the source is calculated by summing over all isotopes and active source types ST_{active} :

$$a = \sum_{i=1}^{i_{\text{tot}}} \left(\sum_{s \in ST_{\text{active}}} a_{s,i} \right) \quad (5.3)$$

For a given source volume, the total activity can be easily calculated to

$$A = V \cdot a \quad (5.4)$$

where V is the source volume calculated from shape and size information. When the source has no defined volume (e.g. a ^{252}Cf point source), it is necessary that a total activity is specified manually. Calling that activity A' , individual (total) source activities are then calculated by

$$A_{d,i} = A' \cdot \frac{a_{d,i}}{\sum_{i=1}^{i_{\text{tot}}} \left(\sum_{s \in ST_{\text{active}}} a_{s,i} \right)} \quad (5.5)$$

Clearly, turning on or off specific source types does not influence the activity of the others, but only changes the total activity. The total activity is used by the main code to derive the number of events to be started during a given runtime.

During a simulation run, every time a new source event should take place, the method `GeneratePrimaryVertex` of `ONMSMaterialDecaySource` is called. First, a single internal source is sampled depending on a table that holds activities of all separate sources. Next, the respective `GeneratePrimaryVertex` method of this selected source is called. The latter function samples one or more particles and energies from the specific source. For SF events, multiple particles are created for a single source event, α -decay and (α , n) reaction produce single particles. Without a material defined as source material, `ONMSMaterialDecaySource` will fail to run and raise an error, as it would be unclear which material should be decayed.

Every source event is attributed an absolute event time t_e by sampling uniformly from the specified runtime t_r :

$$t_e = t_r \cdot u_{\text{rng}} \quad (5.6)$$

where u_{rng} is a random number, uniformly selected of the interval between 0 and 1. This sampling method neglects changes in material activity due to radioactive decay during the measurement time. However, the measurement time is very short compared

to typical half-lives of isotopes measured, for example the shortest half-life of common plutonium isotopes is that of ^{241}Pu with 14.329(29) years.

The position of the beginning of the track of an individual source particle depends on the defined source geometry. For point sources, this is the point of the source. For volumetric sources taken from the GDML definitions or specified by the user (e.g. Cylinder, Sphere), starting points are sampled homogeneously inside this volume. The direction of particles is sampled from an isotropic distribution.

After a single source is selected, the respective classes use different approaches to sample particle energies. If activated, the α -decay source, as specified by the class `ONMSAlphaSingleIsoDecaySource`, has separate instances of the class for each isotope that decays via α -decay. If called, an instance simply selects an energy from possible levels according to data stored in `G4RadioactiveDecay` and starts an α -particle with that energy.

The spontaneous fission source is specified by the class `ONMSSFSSingleIsoDecaySource`. Similar to α -decay, several instances can be part of the total source, in case that there are multiple isotopes decaying by spontaneous fission. It does not require additional initialization before the run starts. The class makes use of the `Fission` library, which has been developed by the Lawrence Livermore National Laboratory. The library is available as open source software under a permissive license [LLNLFIS]. This library is employed mainly because it contains a comprehensive collection of the necessary data, mostly from recent other references. The library is described in detail in [VHW10]. The code allows to sample the number of neutrons or γ -particles from a spontaneous fission event. In doing so, it includes tabulated multiplicity distributions of most isotopes decaying by spontaneous fission.

Every time a spontaneous fission takes place, the number of neutrons is sampled from this distribution. For each neutron emitted, the energy is calculated by the library using stored parameters for the Watt spectrum. Similarly, the library can emit γ -particles from a fission reaction, this is used for the SFg decay type. The emitted particles are not correlated - no mechanism is implemented to check that the sums of energy would match the energy released in a fission event, and that the sum of momentum of outgoing particles would be zero. This should not influence the outcome of the program, as it does not rely on such correlation, mainly due to the strong moderation that takes places before neutron detection.

The properties of particles from the α n decay type are computed in the `ONMSANSource` class. It was first envisioned to split the full simulation into two different runs. One would simulate the transport of α -particles and store location and parameters of occurring (α , n) reactions. The second run would transport neutrons, coming both from these stored data-points and additional sources as spontaneous fission. While it was clear that the first run would be computationally expensive, it would be beneficial to accurately simulate (α , n) reactions completely problem independent, even in complex geometries. During development, several problems arose. It turned out that most existing implementation α -particle Monte Carlo transport are not well suited for a treatment

of (α , n) reactions. The originally implemented inelastic interaction in Geant4 is not suitable for such low energies. Although some neutrons are produced by inelastic interaction, the numbers do not match measured data. Even after implementing special (α , n) cross section data sets for inelastic interactions, based on JENDL/AN-2005 data, no reliable values for neutron yield could be achieved. Further investigation showed that this might be due to the combination of multiple elastic interaction steps into the model of a continuous slowing down process. It should be possible to improve this further by simulating every single coulomb scattering step for elastic scattering, however this would increase computation time by another factor of 1000, and as such would not be practical with current computing power.

Instead, a routine was implemented that samples source neutrons based on a fixed activity and from a given energy distribution. Per default, source activity and neutron energy spectrum are calculated by `ONMSANSource` based on external (α , n) cross section data and stopping power data, according to the descriptions in the previous chapter, sections 4.5, 4.6 and 4.7. All the calculations are carried out during the initialization process of `ONMSMaterialDecaySource` and stored for a single run, or even multiple runs, if no parameters are changed in the meantime. Besides the method of precalculated neutron spectra, `ONMSANSource` has two additional methods implemented. One creates source neutrons of fixed energy. The second takes a text file with an energy spectrum as input, which could for example be calculated with external tools such as `Sources4C` [Wil+02].

Unlike other source particle classes, only one instance of the `ONMSANSource` is created in the program, which calculates source particles for all isotopes. The (α , n) activity is calculated according to equation (4.27), multiplied by the source volume. To deal with the two sets of necessary physical data - stopping power of the source material and the (α , n) reaction cross sections for relevant isotopes, an additional library was developed, `ONMSAlphaNData`.

With regard to stopping power, the additional library accesses a list of stopping powers based on the parameterization [Zie77], data for plutonium was added based on [PW81]. The necessary calculations for stopping powers of materials composed of several elements using the Bragg addition method are implemented in the `ONMSAlphaNData` library. The total interaction cross section for (α , n) reactions is read from JENDL/AN-2005 cross section files. They have been converted into a data format similar to G4NDL. This conversion was done for two main reasons: On the one hand, it allows for easy future use of the cross sections with the `Geant4 ParticleHP` class. On the other hand, it was easier to implement simple routines to read G4NDL style files in `ONMS`, instead of implementing a routine that would read ENDF formatted files.

The conversion was carried out with a Python script, based on PyNE, a Python tool set for nuclear engineering with routines for reading ENDF files. In the process of the conversion, several bug fixes for the ENDF routine of PyNE were contributed to the open source project, also some minor errors in the actual ENDF files of JENDL/AN-2005 had

to be corrected to comply fully with the ENDF file format [HT09]. A description of the Python script and the changes made to the cross sections is given in appendix A.

For the pre-generated neutron energy spectrum, `ONMSANSource` in combination with the `ONMSAlphaData` library also needs partial cross sections for the different excited states of the residual nuclei of (α, n) reactions. These cross sections are also part of JENDL/AN-2005, and were also part of the above mentioned file preparation process. Overall, the neutron energy spectrum pre-calculation uses an energy group structure with 1000 bins from 0 MeV to 10 MeV.

Besides generating neutrons as source particles, the `ONMSANSource` class provides two methods to check the correctness of the pre-calculated data, which can be executed using messenger commands. With the first method, it is possible to extract the neutron yield for the given source material as a function of α -particle energy.

The second method allows for extraction of the resulting (α, n) neutron energy spectrum and for storing it in a file. Based on this, the resulting neutron energy spectrum of the “PuO2-10” sample is shown in figure 5.2. The “PuO2-10” sample is one of the samples that were measured by Malte Göttsche and used to validate ONMS. Another energy distribution was calculated using Sources4C [Wil+02]. The small differences visible in the figure are a result of different data sets used in both cases. The data sets used in ONMS are slightly newer, hence it can be assumed that they might contain improved values.

With all the components described above, the `ONMSMaterialDecaySource` class should be a very versatile library for calculating source particles in simulations of neutron multiplicity measurements. It makes use of very recent data sets, and offers a variety of settings to adjust the source. It also has the ability to output most of the initial data as a neutron energy spectrum or isotope specific source intensities.

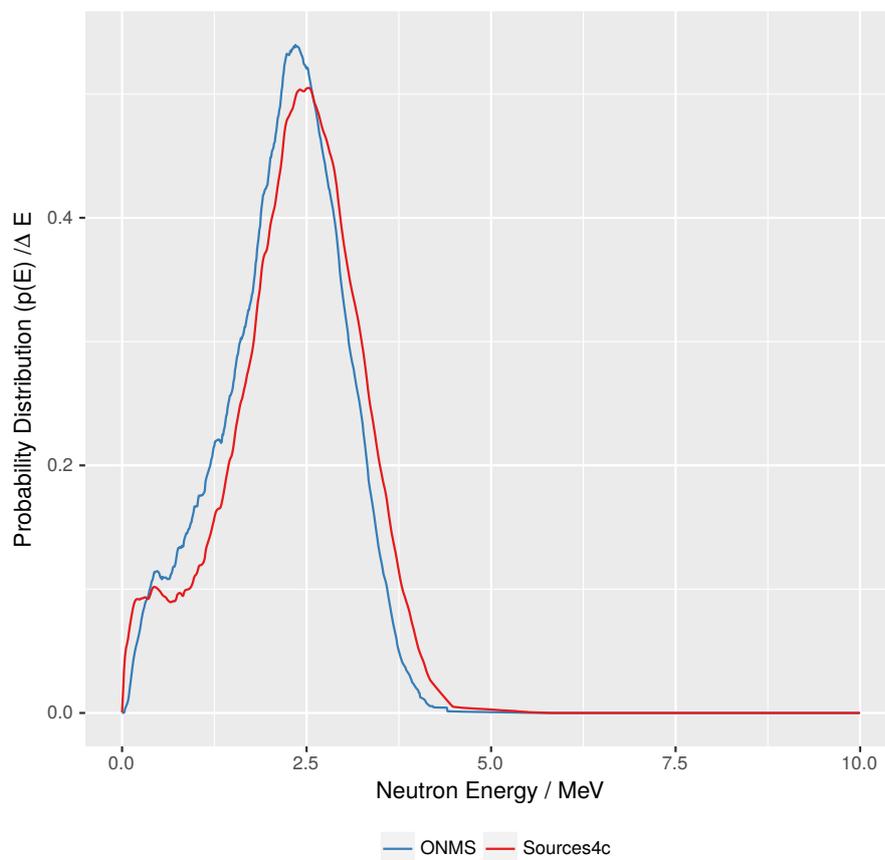


Figure 5.2.: Energy spectrum of neutrons from (α, n) reactions in the sample "PuO2-10", calculated both with Sources4c (green line) and ONMS (blue) line.

5.4 Pulsetrain Analysis

The functions to carry out pulsetrain analysis reside in the subfolder `ONMSPulseTrainAnalysis` and are used as a software library by the main program. Although the library and source files are part of ONMS, it was developed with the possibility to compile it separately. It does not rely on any Geant4 component. A special helper routine is provided, allowing the pulsetrain analysis to be run as a discrete program, without the requirement for a Geant4 installation.

During a simulation, the class `ONMSPulsetrainManager` simultaneously stores a set of detection events. A single detection event is defined in the class `ONMSDetectedEvent` and consists of the following information:

```
1 long long eventtime;
2 long long lifetime;
3 int eventid;
```

`eventtime` is the detection time in terms of global time of the problem, `lifetime` is the time since the neutron has been generated. `eventid` stores the id of the event that

led to the neutron detected. This information is useful for example for deriving gate fractions from pulsetrains.

Detection events can be the result of the continuous creation during a simulation in ONMS, or can be read from external files with special methods defined in `ONMSPulsetrainManager`. As input files both those using a structure reflecting the detection event structure, but also files listing detection event times using the unit shakes (10^{-8} s) are possible. The latter are required to analyze files provided as part of the ESARDA benchmark, used to validate the code. The pulsetrain analysis benchmark and calculation results is discussed in chapter 6. `ONMSPulsetrainManager` also provides ways to change, store or output settings used for the analysis. Possible settings are listed in table 5.3.

Table 5.3.: Parameters for pulsetrain analysis. Respective messenger commands start with `/ONMS/analysis/`.

Variable	Messenger Command Name	Default Value
Predelay	<code>setpredelay</code>	$4.5 \mu\text{s}$
Gate Length	<code>setgate</code>	$64 \mu\text{s}$
Long delay	<code>setlongdelay</code>	$4096 \mu\text{s}$
Register length *	<code>registerlength</code>	128
Dieaway	<code>dieaway</code>	$50 \mu\text{s}$
Dead time (before quantization)	<code>predeadtime</code>	0
Dead time (after quantization)	<code>postdeadtime</code>	0

* This is the length of a virtual shift register. It is only used to calculate the operating frequency (or length of a single register period), based on the gate length. The register period is used if the user selects quantization for the register.

Deriving the R+A and A distribution from the recorded or loaded pulsetrain is done by a function that implements typical shift register logic in software. This way should allow for direct comparison to experimental results from shift register electronics. Many steps of the process can be configured by settings.

First of all, it is checked whether the pulsetrain events are sorted in time. Coming from Monte Carlo simulations, they are typically not sorted, because the timing is sampled over the full simulated measurement length. If they appear not to be in order, they are sorted.

In a real shift register, several quantization processes take place which move events to specific time slots. Pulses can only be measured at times when the register shifts all positions. For example in a register driven by a 2 MHz clock, there can be a pulse only every $0.5 \mu\text{s}$. If a neutron pulse is slightly off that timing, it is registered as it would be on the closest clock pulse. If more pulses (e.g. from different He-3 tubes) arrive in the same shifting cycle, only one can be registered, the others would be lost. Many multiplicity counters add a “Derandomizer Circuit”, which reduces the number of lost neutrons due to overlap in event time. The circuit stores parallel events, usually

with a faster driving clock than the shift register, in parallel buffers, and outputs them sequentially to the shift register. Figure 5.3 shows a simplified quantization process. Quantization in general has two main effects on data: Events are counted later than they were detected, and some events might get lost.

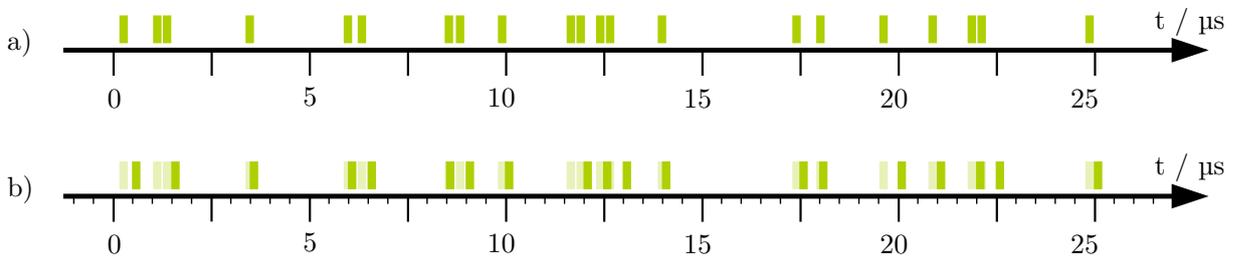


Figure 5.3.: Arbitrary pulsetrain (a) and pulsetrain after quantization (b) to 2 MHz clock. At $1.5 \mu\text{s}$ and $12 \mu\text{s}$, events are lost due to quantization effects. The light green squares in (b) show the positions of events in (a) to help visualization.

ONMS implements a quantization method that can be used if required. It takes quantization frequency and a possible buffer length for storing of overlapping events as an argument. If the buffer length is zero, overlapping events will be discarded immediately, otherwise only if the buffer for such events is full. The method can be applied multiple times to a data set.

Simulations using computers add another quantization effect due to finite precision commonly used for binary representation of numbers. In early versions of ONMS, event times of the pulsetrain were stored as C++ double values. Using double values leads to different quantization effects depending on compiler and processor used, which might change the total outcome. Also, quantization effects are bigger towards the end of simulated measurement times, as the absolute value becomes higher and as such the double precision decreases. To avoid this dependency, the current version works with the long long data type for the event time. The time values are stored as values in ps. As such, this leads to a quantization similar to an imaginary shift register with a 1 THz clock²⁸.

Besides quantization, physical detectors also show dead time effects due to several reasons. After a neutron is absorbed in a He-3 tube, it takes some time until the avalanche of ionized particles has been removed and the tube can detect another signal [Ens+98, p. 29]. After amplification, signals are typically transformed to relatively short pulses. Still, when multiple He-3 tubes are used, overlapping signals could shadow each other, thus leading to dead time effects. Often, detector dead time is characterized summarizing several effects to give a single dead time value. Dead times also can be distinguished between non-updating (non-accumulating) and updating (accumulating) circuits. In the latter case, events that happen shortly after a first event “reset” the dead time, so

²⁸ It would be possible to use arbitrary precision to avoid such quantization, however the listed solution should be sufficient for good results, and is significantly faster.

Figure 5.4.: Algorithm for simulating dead time effects on pulsetrain.

Input:

PulsetrainEvents: An array holding pulsetrain event times

dt: Variable for dead time

dtu: Boolean variable, true for updating dead time

Result:

ProcessedEvents: An array holding remaining events

LostEvents: An array holding events lost due to dead time effects

begin

$deadtimestep \leftarrow 0$

for $i \leftarrow 0$ **to** $Length(PulsetrainEvents)$ **do**

if $PulsetrainEvents[i] > deadtimestep$ **then**

 Append $PulsetrainEvents[i]$ to *ProcessedEvents*

$deadtimestep \leftarrow PulsetrainEvents[i] + dt$

else

 Append $PulsetrainEvents[i]$ to *LostEvents*

if $dtu = True$ **then**

$deadtimestep \leftarrow PulsetrainEvents[i] + dt$

it is extended. When high count rates are measured, long updating dead times might make measurements difficult to impossible.

ONMSPulsetrainManager has implemented a function for simulating dead time behavior. The function can be called twice, once before any quantization effects are taken into account, and once after the quantization functions. Mostly, there is no physical need to do this twice for a single calculation, however this feature gives flexibility to carry out different simulations where the analysis reflects different actual physical measurement systems. An overview of the algorithm to apply dead time is shown in figure 5.4.

After selected effects are applied to the original pulsetrain, the algorithm loops through all events to derive the “R+A” and “A” distribution. The commonly used algorithm is shown in figure 5.5. For each event (“trigger”), it checks the number of events in a time window starting after predelay and of the length of gate and increases the respective entry of the “R+A” distribution by one. Similarly it checks for the number of events in a time window of same length, but starting “longdelay” after the trigger. Depending on the latter number, the respective entry in the “A” distribution is increased by one.

Besides the algorithm in figure 5.5, a similar method has been implemented to carry out the analysis with a simulated shift register “backwards”. Backwards means that instead of searching for all events that fall into the time period of “gate length” after the triggering event (and the predelay), the algorithm searches for all events that fall into a time period of “gate length” before the triggering event. The latter algo-

Figure 5.5.: Standard algorithm for pulsetrain analysis.

Input:

PulsetrainEvents: An array holding pulsetrain event times

pd: Predelay time

g: Gate Length (time)

ld: Long delay time

Result:

R+A-distribution: Array for calculated foreground multiplicity distribution

A-distribution: Array for calculated background multiplicity distribution

begin

Initialize all values of *R+A-distribution* and *A-distribution* with 0

for $i \leftarrow 0$ **to** $\text{Length}(\text{PulsetrainEvents})$ **do**

$j \leftarrow i + 1$

while $\text{PulsetrainEvents}[j] < \text{PulsetrainEvents}[i] + pd$ **do**

$j \leftarrow j + 1$

 counter $\leftarrow 0$

while $\text{PulsetrainEvents}[j] < \text{PulsetrainEvents}[i] + pd + g$ **do**

$j \leftarrow j + 1$

 counter \leftarrow counter + 1

$R+A\text{-distribution}[\text{counter}] \leftarrow R+A\text{-distribution}[\text{counter}] + 1$

$j \leftarrow i + 1$

while $\text{PulsetrainEvents}[j] < \text{PulsetrainEvents}[i] + ld$ **do**

$j \leftarrow j + 1$

 counter $\leftarrow 0$

while $\text{PulsetrainEvents}[j] < \text{PulsetrainEvents}[i] + ld + g$ **do**

$j \leftarrow j + 1$

 counter \leftarrow counter + 1

$A\text{-distribution}[\text{counter}] \leftarrow A\text{-distribution}[\text{counter}] + 1$

rithm is not used by default, but was used for testing purposes, and can be used to compare results to other tools that work with such a method. In contrast to ONMS, which simulates events using a global event time, many other codes only simulate time on a per-event basis. That means that in such codes every spontaneous fission event would occur at $t = 0$ ²⁹. To carry out pulsetrain analysis in this case, two options are commonly used: As a first method, the “R” distribution is determined on a per-event basis, and no A distribution needs to be calculated. As a second option, the resulting events can be randomly distributed over a given measurement time. The latter approach is similar to the default approach used in ONMS. To allow for comparison with the first approach, another way to analyze the pulsetrain is included in ONMS. If this approach (`ResultsEventNoForward`) is selected, the pulsetrain is first split into subgroups based on `eventId`, the id of a source event that led to a specific detected event. For each of these subgroups, an analysis similar to figure 5.5 is carried out separately, the multiplicity results are later combined.

After the algorithm has calculated the distributions, the `ONMSPulsetrainManager` can return an `ONMSMultiplicityResult` object, which holds foreground and background multiplicity distributions. The `ONMSMultiplicityResult` object also includes functions to calculate Singles, Doubles and Triples from these distributions, as well as multiplication M , (α , n) neutron fraction α , spontaneous fission rate F , and effective ²⁴⁰Pu mass.

By default, the gate fractions f_d and f_t needed to solve for M , α and F are calculated based on single exponential dieaways from the specified dieaway time, according to equation (4.17). Besides the default, `ONMSPulsetrainManager` offers two other options. They are based on the application of gates that can be considered infinite (cf. equation 4.18a). For one of the options, the assumption of an infinite gate is achieved by calculating the “R” distribution with the above described `ResultsEventNoForward` method twice, one time for the user defined gate length, a second time counting all neutrons detected after a single event - equivalent to an infinite gate. The latter is possible because for every given `eventId` only a finite number of events can be detected. Doubles and Triples gate fractions are then just the quotients of D or T calculated with defined gate length and D or T calculated with no gate length. As a third option to calculate gate fractions, the standard pulsetrain analysis method is called twice, one time for the user defined gate length and once for a gate length of twice that time³⁰. Again, Doubles and Triples gate fractions can be calculated as the quotients of the results of both calls.

As a first estimate of uncertainties of calculations, `ONMSPulsetrainManager` automatically creates “R+A” and “A” distributions for fractions of a tenth of the total pulsetrain. From this set of distributions, average Singles, Doubles and Triples and the related stan-

²⁹ For example MCNPX uses such an approach. Every spontaneous fission event (or ‘history’) starts at $t = 0$, and detection times recorded are relative to that.

³⁰ Doubling the gate length has been done by other codes, as it is for example described in [PS06].

standard deviations can be calculated. These values are included in the output of ONMS, and can be used as a quick uncertainty estimate.

For any output, `ONMSPulsetrainManager` either writes results to the screen using the messenger command `/NMS/analysis/showresults` or to a file with the command `/NMS/analysis/writeresults <filename>`. It is possible to store pulsetrains for further analysis in files.



6 Validation of Open Neutron Multiplicity Simulation

Every simulation software needs validation, especially relatively complex applications as ONMS, which consist of a number of different components and make use of a large variety of nuclear data sets.

For neutron multiplicity measurements, there are mainly three different ways to generate data which can serve as a standard of comparison for the validation of a given simulation code. All methods have benefits and disadvantages, and overlapping utility. Probably the most obvious is the comparison of results of actual physical measurements of different samples using neutron multiplicity counters to those derived by simulations. Based on simulations that accurately predict detector results, use cases for safeguarding or disarmament verification can be evaluated. It should be noted, however, that measurement results vary depending on the detector and are influenced by detector design and efficiency profile. They also depend on dead time of electronics and neutron die away time, as well as the different pulsetrain analysis methods and shift register electronics employed in detectors. It is important for simulations to either account for all of these effects, or otherwise compare results having the possible differences in mind.

A second way to generate data for validation is to provide theoretical characteristics of a sample using the point model approach. Based on a sample mass, Singles, Doubles and Triples rates can be calculated. The mass can be an assumed mass for a hypothetical sample or measured by different means, e.g. measure a samples using a scale if possible. To calculate Singles, Doubles and Triples via the point model, it is necessary to have good knowledge of characteristics of the detector that is simulated (e.g. efficiency).

Third, results of a given code can be compared to results simulated using similar applications. Clearly, these results are not actual experimental results, but are checks for the implementation of physical models that are the basis of simulation codes. The method is beneficial to characterize the overall quality of simulation codes. Given similar physical models, results should be similar for equal initial parameters and models. Using this way of validation, it is also possible to define entirely virtual samples and initial parameters. This is an easier step, as exact detector and sample geometries are sometimes hard to be obtained. Simulation based on such virtual samples will give indications of code quality, even if they do not resemble exact physical models of detectors or samples. They are also helpful to find differences among different methods used in simulations.

This chapter presents simulations carried out with ONMS comparing to data derived from all three methods. First, there will be a section describing measurements by Malte Göttsche [Göt15], followed by the related ONMS results and a short discussion. After

that, a comprehensive benchmark exercise for neutron multiplicity simulations will be presented, again with results of ONMS and discussion.

The large number of calculations for this chapter were mostly carried out using the high performance computing cluster “Della” of Princeton University³¹. While it is possible to run simulations for neutron multiplicity measurements on common desktop computers, the use of these resources allowed for a much higher number of simulated events, as such improving statistical uncertainty of Monte Carlo simulations. ONMS is not (yet) capable of using multithreading or cluster architectures as e.g. applications based on the Message Passing Interface. Nevertheless, several provisions were made in the code to achieve better parallelization. Simulations can be started with shorter measurement times, and then accumulated for the final result. As such, a single task can be split in smaller pieces, that can be distributed over different machines. This method only works if different random number generator seeds are specified for each run. Otherwise, ONMS uses the same seed each time, which is a deliberate function to make results reproducible - but not practical for the given task. The generation of lists of random seeds can be directly done with ONMS itself, using a specific command line argument. The seeds can be used in macro files for calculations using the `/random/setSeeds` messenger command.

After all calculation parts are finished, they need to be combined to form a consistent result. The pulsetrains that have been written to files by each part can just be concatenated. So generated pulsetrain files can be reread with ONMS to carry out pulsetrain analysis on the full pulsetrain. Apart from pulsetrain analysis, other results generated by ONMS (e.g. the number of simulated source events or the average source neutron energy) are written to `<calculationname>.results` files (cf. appendix C). The aggregation of these values has to be done using other tools, carefully taking into account the type of result. While for example the number of source neutrons just needs to be summed, the average source neutron energy of each run needs to be weighted by the respective source neutron number before combining to a total result.

In addition to splitting single runs into smaller pieces, for every calculation multiple runs were carried out to improve the statistics of the results of random sampling processes. The data of these different runs have to be combined as well using other tools. For all the calculations presented in this section, a set of Python scripts was written, that both generated initial macro files for calculations split into parts and also merged the results together to coherent value sets. The set of scripts was developed with the goal to allow for easy reproducibility of the results presented and recalculations with future versions of ONMS.

³¹ The cluster uses common Intel 64bit processor architecture, different cores of the Intel generations Westmere, Ivybridge and Haswell. The cluster has a total of 3484 cores. Typical simulation times vary depending on the problem. The current version of ONMS allows for the simulation of about 300000 source events per hour computer time.

6.1 Experimental Data - Description of Measurements

Over the full time of the development of ONMS, a set of experimental results were used to validate different steps of the process. The results are based on a measurement campaign carried out from April 23 to May 4 2012 by Malte Götsche at the PERLA laboratory, Joint Research Centre, Ispra. They have been published in [Göt15], additional information was gathered by personal communication with the author. He used the results to validate simulations produced with another application capable of simulating neutron multiplicity measurements, MCNPX-PoliMi.

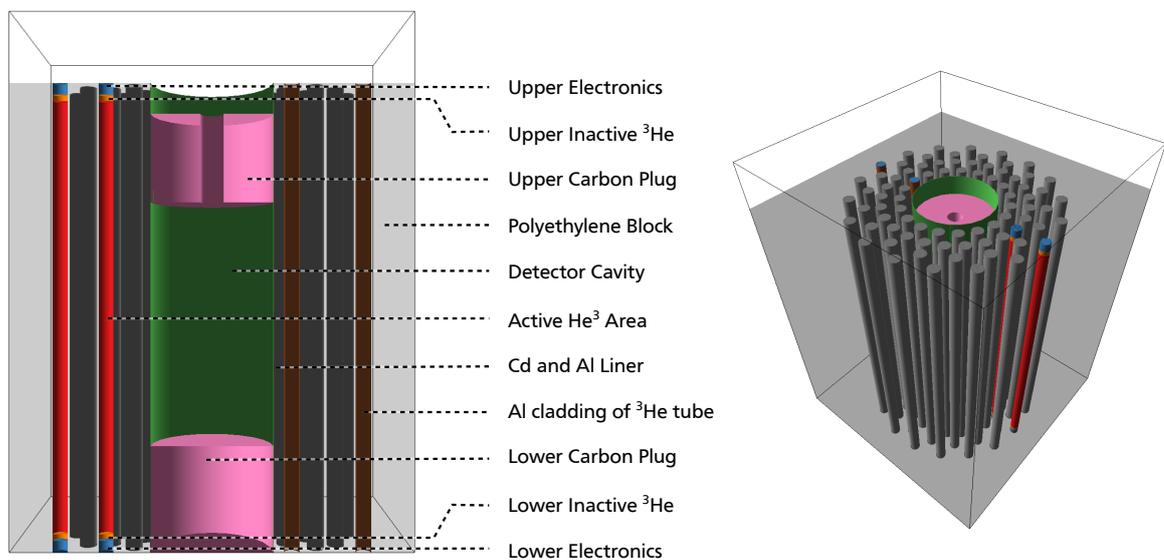


Figure 6.1.: Rendering of a Geant4 model of the Plutonium Scrap Multiplicity Counter. The image has been created directly from the Geant4 GDML file used for the calculations. Every gray tube is a space in the polyethylene where a ^3He tube is installed. Only selected ^3He tubes have been divided in subsections. The top part of the detector is not filled and only shown by its outline.

For the measurements, the so-called “Plutonium Scrap Multiplicity Counter” (PSMC) has been used, figure 6.1 shows a graphical representation of the version that was modeled using ONMS. It uses a cylindrical geometry and consists of 80 ^3He tubes in 4 rings in a polyethylene block with a square ground area. The polyethylene block has a width of 66 cm and a height of 77.1 cm. For the model, polyethylene was defined to have a density of $\rho = 0.955\text{g cm}^{-3}$. To optimize efficiency, tube numbers vary for each ring, the inner-most ring has 19, the next has 25 and the two outer rings have 18 ^3He tubes each. The ^3He tubes have an active height of 71.1 cm and have an outer diameter of 2.54 cm including the aluminum cladding, with a radius of 1.23 cm for the volume containing ^3He . The ^3He is modeled with a density of 0.502 mg cm^{-3} . The cavity of the detector has a diameter of 20 cm, and a height of 40 cm. When characterized

in calibration measurements, the detector shows a very flat axial and radial efficiency profile. The detector was modeled according to an MCNP input file developed by Malte Göttsche [Göt15], which has been received in private communication.

For validation purposes, 3 metal samples (named PM1, PM2, PM3 in the following) and 5 oxide samples (PuO2-10, PuO2-20, PuO2-21, PuO2-22, PuO2-23) have been simulated. Isotopics, plutonium mass and total mass of the different samples are given in table 6.1. The metal samples are enclosed in small aluminum cylinders, which in turn are enclosed in stainless steel containers of cylindrical shape with a thicker top. For the aluminum, the standard Geant4 aluminum density and composition (G4_A1) was used, the composition of the stainless steel (AISI 304) was taken from personal communication with Malte Göttsche. The oxide samples were all enclosed in a simple aluminum cylinder, and typically in the form of powders. The full GDML code of the sample PM1 is shown in appendix B. All simulated GDML files of this validation exercise are part of the code repository of ONMS.

Table 6.1.: Characteristics of the samples measured by M. Göttsche, according to table B.1 and B.2 in [Göt15, pp. 95-96].

Sample	Mass (g)		Density [†] (g cm ⁻³)	Isotopic vector (wt %)					
	Total [*]	Pu		²³⁸ Pu	²³⁹ Pu	²⁴⁰ Pu	²⁴¹ Pu	²⁴² Pu	²⁴¹ Am ‡
PM1	13.7363	12.5	20.06	0.004	95.42	4.529	0.032	0.015	0.245
PM2	20.5016	18.8	19.78	0.004	95.493	4.455	0.033	0.015	0.235
PM3	20.1064	18.9	22.43	0.025	91.358	8.468	0.102	0.047	0.895
PuO2-10	2.25278	1.987	4.243	0.058	86.082	13.27	0.321	0.27	1.483
PuO2-20	5.64953	4.983	5.32	0.058	86.082	13.27	0.321	0.27	1.483
PuO2-21	11.3002	9.967	5.676	0.058	86.082	13.27	0.321	0.27	1.483
PuO2-22	22.5834	19.919	5.671	0.058	86.082	13.27	0.321	0.27	1.483
PuO2-23	23.3142	20.566	5.166	0.084	70.906	26.856	0.691	1.463	4.878

* Total mass has been calculated including either the mass of oxygen atoms for oxide samples or the mass of nickel and copper for the three metal samples.

† Density calculated based on total mass and size of the sample.

‡ Americium content is given in wt % of total plutonium content.

The samples were all placed in the cavity so that the center of the plutonium volume was at the center of the cavity.

6.2 Validation Results and Discussion

For all the measured values described in section 6.1, ONMS simulations were carried out. Figure 6.2 shows the factorial moments for all samples. The red bars depict data as simulated with ONMS, the green bars show measurement results, and the blue bars show simulations carried out with MCNPX-PoliMi. The latter two values were taken from [Göt15]. For all samples, the ONMS simulations were carried out for a simulated measurement time of 1000 s, 10 different runs were calculated for each sample. The

number of simulated particles per sample per run varied from 2.6×10^5 to 6×10^6 , because of the difference in mass and proportions of isotopes undergoing spontaneous fission. Simulations took between 80 and 1680 minutes to calculate per run. For each of the ten different simulation runs, the pulsetrain analysis was carried out and Singles, Doubles and Triples were calculated. The value shown in figure 6.2 is the mean of these 10 values, the error bars show the standard deviation of results from this mean. All calculations were carried out using a pre-delay of $4.5 \mu\text{s}$, a long delay of $4096 \mu\text{s}$ and a gate length of $64 \mu\text{s}$, similar to the experimental settings used in [Göt15]. Overall, ONMS reaches good accordance with the measured values and the simulated values. A general trend of slight overestimation of Doubles and Triples is visible, especially for the metallic fuels. In figure 6.3, the pulsetrains of the same ONMS simulations have been analyzed using an assumed dead time of $0.5 \mu\text{s}$. No simulations with MCNPX-PoliMi are shown, as they have not been carried out using the same dead time. The dead time of the PSMC detector is specified with a shorter value of $0.12 \mu\text{s}$ in [Men+93, p. 11]. Applying dead time improves all results slightly.

For all samples, the plutonium mass has been calculated, the results are shown in figure 6.4. Again, ONMS shows good results, differences to the measured masses are not larger than seven percent. Besides using the Singles, Doubles and Triples from the previous figure, these results include more assumptions. The equation for the ^{240}Pu effective mass requires knowledge of the gate fractions of the detector. For the calculations shown here, a fixed die-away time of $50 \mu\text{s}$ has been assumed, from which gate fractions were calculated based on equation 4.17, yielding a Doubles gate fraction of 0.659 and a Triples gate fraction of 0.435. The total plutonium mass was calculated from the effective plutonium mass based on the given sample isotopic composition.

It is important to note that even small possible systematic uncertainties often dominate the statistical uncertainties for neutron multiplicity simulations. The detector model used here is based on [Göt15], where it has been produced based on coarse design sketches of the detector. While it was tried to ensure that the models were representations of the physical detector and samples as accurate as possible, not all systematic errors can be excluded, partly because of limited data availability.

In his work, [Göt15] carries out comprehensive sensitivity calculations for the PSMC detector and the given samples using simulations with MCNPX-PoliMi. For the detector model, he found uncertainties for Triples of up to 7.5 percent for very small changes of $\pm 4 \text{ mm}$ of the horizontal position of the He-3 tubes in the detector. Smaller, but still significant changes could arise from changes of the vertical position of the tubes and the gas pressure. Changes of 1 cm would influence the final result for Triples of up to 1.5 percent, an error of the same magnitude would be generated by differences in pressure of the tubes of 1 percent. Similar uncertainties are also listed in [PW12]. Although it was carefully checked, geometry definitions could allow for the possibility of smaller systematic errors. Independently, it can be stated that ONMS shows good results for this validation step.

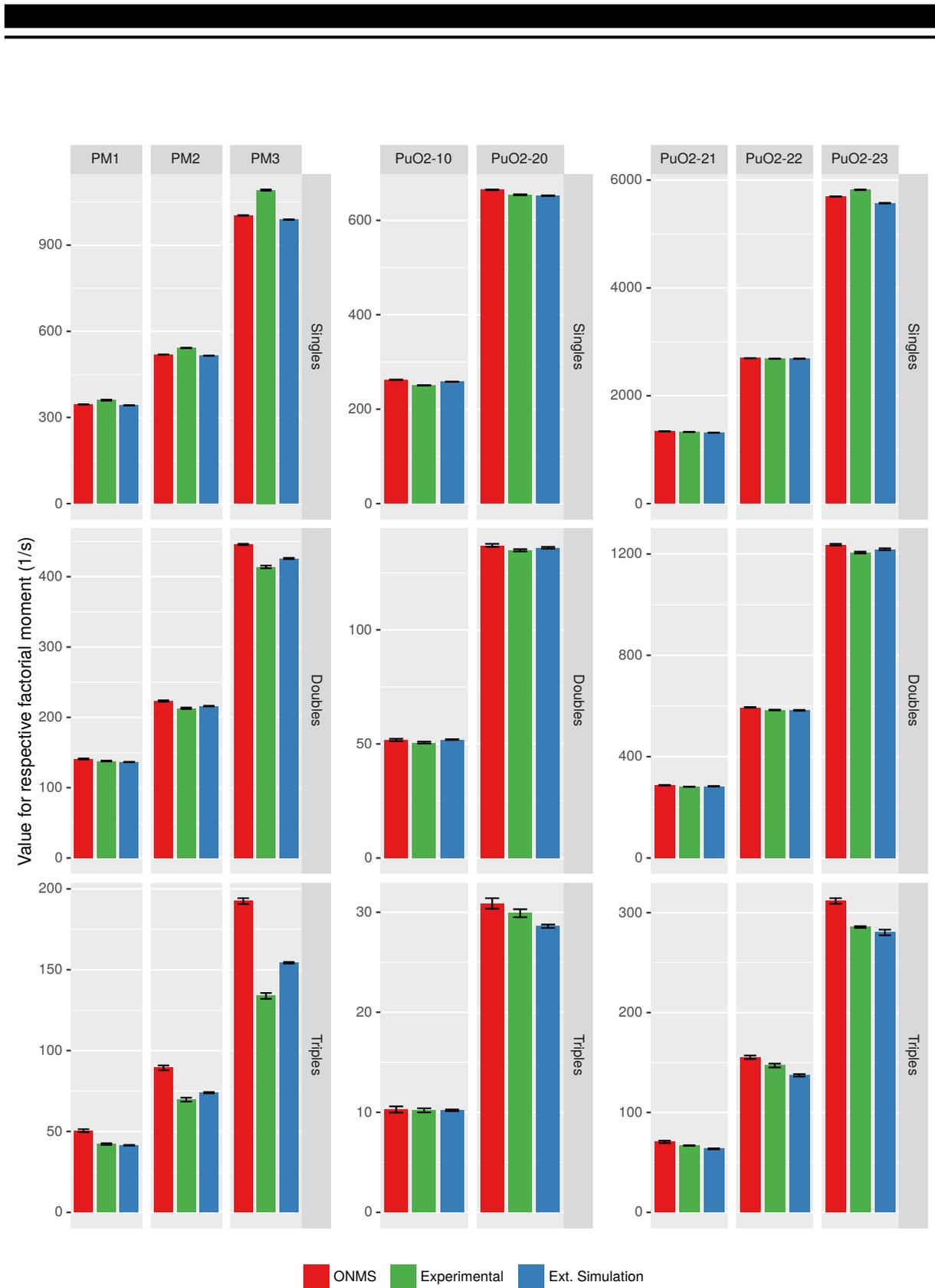


Figure 6.2.: ONMS simulation results for factorial moments compared to experimental data and simulations (“Ext. Simulation”) as listed in [Göt15].

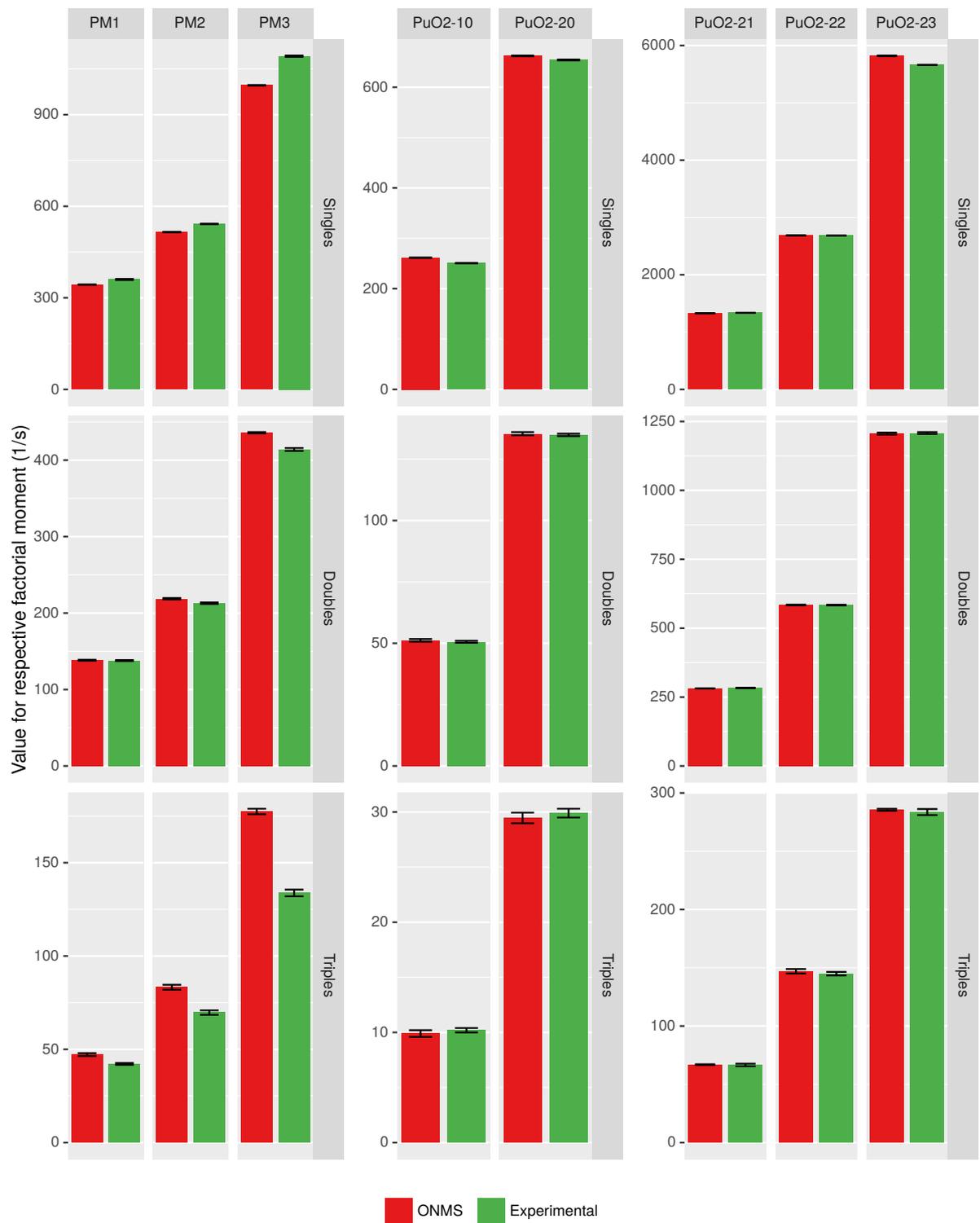


Figure 6.3.: ONMS simulation results for factorial moments including an assumed dead time of $0.5 \mu\text{s}$ compared to experimental data as listed in [Göt15].

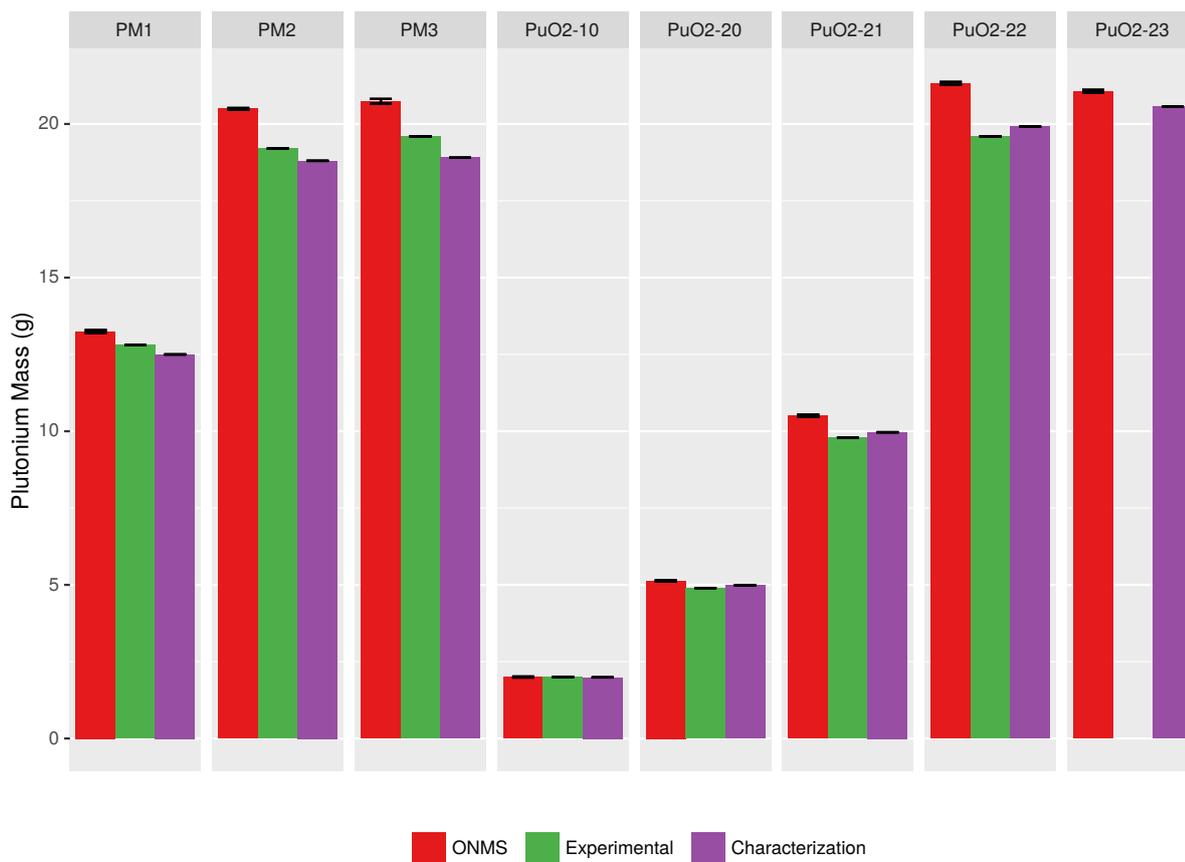


Figure 6.4.: Sample plutonium masses, as simulated with ONMS, measured in [Göt15] and characterized by the ISPRA laboratory. No experimental value is available for PuO2-23.

6.3 Description of the ESARDA Benchmark

The second part of the validation of this chapter is based on a large benchmark exercise which was organized by the Non-Destructive Assay Working Group of the European Safeguards Research and Development Association (ESARDA). Validation based on this exercise is more comprehensive than the calculations and experiments described in the previous two sections.

The description of the benchmarks and results were published in the ESARDA Bulletin, [PS06], [Pee+09]. Unless otherwise noted, all the descriptions and details presented in this chapter were taken from these two sources, as well as from personal communication with Paolo Peerani, one of the authors of the benchmark. The exercise consisted of a total of four phases, and was carried out from 2003 until 2009. The first two phases were based on “theoretical” samples, no actual measurements were carried out, and 11 different groups participated. During phase I, four different simulation codes were employed to carry out full simulations of the samples, starting with Monte Carlo particle transport of neutrons, followed by pulsetrain analysis. For the phase II, a set of neutron

detection pulsetrains for all samples was generated using MCNPX by the authors of the benchmark. These pulsetrains were distributed among the participants and had to be analyzed. For that phase, results from ten different software tools were provided.

The other two phases describe measurements that were carried out for the benchmark, and the results of simulations of these. For phase III, participants again had to carry out full simulations of the samples, including Monte Carlo transport. Five different research groups provided results for this step, coming from six different code systems. Measurements of the samples were carried out in February 2007. Phase IV again consisted of analyzing pulsetrains. For this phase, the pulsetrains were actually generated during the measurement campaign. Besides feeding the neutron detection events to a shift register analyzer, they were also stored using a data acquisition card and a computer. The files generated by this process were shared with all the benchmark participants for pulsetrain analysis.

All phases of the benchmark used or assumed the use of a detector model called “Active Well Coincidence Counter” (AWCC, cf. also figure 4.5). Although it is possible to use this detector in an active mode, it was used in a configuration for passive detection for all phases. It is a relatively small neutron coincidence counter, and was modeled for the use with ONMS according to the MCNP model of appendix A in [Mil12].

The detector contains 42 ^3He tubes, arranged in two rings of 21 tubes, respectively. The tubes have an active height of 50.8 cm, the radius of the ^3He volume is 1.23 cm. They are embedded in a cylindrical block of high density polyethylene, modeled with a density $\rho = 0.955 \text{ g cm}^{-3}$ and an outer diameter of 47.31 cm. In the center of the cylinder is a cavity with a diameter of 22.48 cm. It has a height of 35 cm, placed below and above the cavity are polyethylene plugs. The cavity is lined with a small aluminum cladding (thickness 0.15 cm) and a thin layer of cadmium (0.04 cm). Both use a material composition and density as defined in the Material Database included with Geant4³².

6.3.1 Samples of Phase I and II

A total of 13 different calculations have been described for phases I and II, partly they were based on the same sample, differing only in the assumed samples activity. All samples were given specific identifiers, which are used later for the presentation of results. A short summary of the samples and identifiers is shown in table 6.2. The first sample is an AmLi neutron source. The sample is assumed to be a point-like source, placed in the center of the detector. It has been simulated using three different neutron intensities, which should correspond to count rates in the detector of 10, 100 and 1000 kHz (identifiers epI+II-c1-10, epI+II-c1-100, epI+II-c1-1000). Such count rates can be achieved using a source activity of 26800 for the lowest count rate, 268000 and 2680000 for the other two, respectively. The activities assume a detector efficiency of

³² http://geant4.web.cern.ch/geant4/workAreaUserDocKA/Backup/Docbook_UsersGuides_beta/ForApplicationDeveloper/html/apas08.html

Table 6.2.: Short summary of all the samples defined in the four phases of the ESARDA benchmark. Unless otherwise noted, samples were placed in the center of the detector. Additional information for all samples is given in the text as well as in table 6.3, figure 6.5 and table 6.4.

Sample name	Description	Pu Mass (g)
Phase I and Phase II		
epI+II-c1-10	Point-like AmLi neutron source (26800 n/s), expected singles rate 10 kHz	-
epI+II-c1-100	Point-like AmLi neutron source (268000 n/s), expected singles rate 100 kHz	-
epI+II-c1-1000	Point-like AmLi neutron source (2680000 n/s), expected singles rate 1000 kHz	-
epI+II-c2-10	Point-like ^{252}Cf neutron source (8600 sp.fis./s), expected singles rate 10 kHz	-
epI+II-c2-100	Point-like ^{252}Cf neutron source (86000 sp.fis./s), expected singles rate 100 kHz	-
epI+II-c2-1000	Point-like ^{252}Cf neutron source (860000 sp.fis./s), expected singles rate 1000 kHz	-
epI+II-c3s	Small cubic plutonium metal cylinder, $\rho = 20 \text{ g cm}^{-3}$, $H = D = 8.6 \text{ mm}$	10
epI+II-c3b	Large cubic plutonium metal cylinder, $\rho = 20 \text{ g cm}^{-3}$, $H = D = 39.9 \text{ mm}$	1000
epI+II-c4s	Small cubic cylinder of plutonium oxide powder, $\rho = 2 \text{ g cm}^{-3}$, $H = D = 18.5 \text{ mm}$	8.82
epI+II-c4b	Small cubic cylinder of plutonium oxide powder, $\rho = 2 \text{ g cm}^{-3}$, $H = D = 147.1 \text{ mm}$	4410
epI+II-c5-10	Oxide sample as in c4s, with AmLi source tuned to give $\alpha=10$	8.82
epI+II-c5-100	Oxide sample as in c4s, with AmLi source tuned to give $\alpha=20$	8.82
epI+II-c5-1000	Oxide sample as in c4s, with AmLi source tuned to give $\alpha=100$	8.82
Phase III and Phase IV		
epIII+IV-c1	Point-like ^{252}Cf source with 3781 n/s (1006.4 spontaneous fissions/s) in small stainless steel capsule	-
epIII+IV-c2	Point-like ^{252}Cf source with 497200 n/s (132339.6 spontaneous fissions/s) in small stainless steel capsule	-
epIII+IV-c3	Plutonium metal disk, PuGa alloy, $\rho = 19.05 \text{ g cm}^{-3}$, $R = 16.5 \text{ mm}$, $H = 0.6 \text{ mm}$	9.37559
epIII+IV-c4	Plutonium oxide powder, in special container (cf. figure 6.5), container positioned 10 cm from the bottom of cavity	50.368
epIII+IV-c5	Plutonium oxide powder, in special container (cf. figure 6.5), $\rho = 2.6 \text{ g cm}^{-3}$, container positioned at the bottom of cavity	978.703
epIII+IV-c6	Mixed oxide sample, natural uranium and plutonium, in special container (cf. figure 6.5), $\rho = 0.8 \text{ g cm}^{-3}$, container positioned at the bottom of cavity	164.676

37.3% for the neutron spectrum released by the AmLi source³³. The activities were used as given for the ONMS calculations. AmLi neutron sources generate neutrons by (α , n) reactions, hence no correlated neutrons are to be expected. For the energy spectrum of the neutrons, the “Geiger/van der Zwan” spectrum is recommended. A version of this spectrum as listed in Table 2 of [TL12] was used.

The second sample is a point-like ²⁵²Cf source, again in the center of the detector and with three different activities. Also for this source count rate goals have been specified instead of source activities, with rates of 10, 100 and 1000 kHz to be achieved (sample identifiers epI+II-c2-10, epI+II-c2-100, epI+II-c2-1000). Using a $\bar{\nu}$ of 3.757 and a detector efficiency for ²⁵²Cf spontaneous fission neutrons of 30.95 %, the spontaneous fission rates required to achieve the given count rates are 8600, 86000 and 860000 spontaneous fissions per second. The energy spectrum of the neutrons released by spontaneous fission should be a Watt spectrum with $a = 1.175$ and $b = 1.04$ according to the benchmark specification.

The third and fourth samples are plutonium samples. Both are to be simulated as cubic cylinders ($H = D$) placed in the center of the cavity. No container needs to be simulated. H and D are calculated depending on the different source volumes, which in turn can be derived from density and mass. The third sample is supposed to be plutonium metal with an isotopic composition of 90 wt% ²³⁹Pu and 10 wt% ²⁴⁰Pu, and a density of 20 g cm^{-3} . Two different simulations are to be carried out for this sample, one with a total mass of 10 g (epI+II-c3s) and one with a total mass of 1000 g (epI+II-c3b). The fourth sample is a sample of plutonium oxide powder. The isotopic composition of the plutonium in this case is 2 wt% ²³⁸Pu, 60 wt% ²³⁹Pu, 25 wt% ²⁴⁰Pu, 8 wt% ²⁴¹Pu and 5 wt% ²⁴²Pu, the powder density is supposed to be 2 g cm^{-3} . The first simulation should be done with a total mass of 10 g (epI+II-c4s), the second with a larger total mass of 5000 g (epI+II-c4b).

The fifth sample is a mix of the plutonium oxide sample and an AmLi neutron source. This sample was not simulated using ONMS, it is currently not possible to define two parallel sources. However, it was possible to carry out a pulsetrain analysis on a pulsetrain file that has been produced for that source.

6.3.2 Phase III and IV

Six different samples have been simulated and measured in phases III and IV of the benchmark, in these cases existing real samples formed the bases of calculations. The first two samples are ²⁵²Cf samples, both in small stainless steel capsules. The first has a source strength of 3781 n/s at the time of the measurement, the second a source strength of 497200 n/s. In ONMS, the activity of a source has to be specified in spontaneous fission reactions per second, hence the above values have been converted using

³³ The specification of sources according to their count rates in the detector has been given in [PS06]. This adds additional uncertainties to the benchmark, because it makes assumptions on the detection efficiency that might not necessarily be exactly reflected in simulated models.

$\bar{\nu} = 3.757$, giving activities of 1006.4 and 132339.6 spontaneous fissions per second. The samples have the identifiers epIII+IV-c1 and epIII+IV-c2. The stainless steel capsules have a height of 1 cm, a radius of 0.4 cm, with a wall thickness of 0.13 cm. In the simulation with ONMS, the sources were placed in the center of the cavity.

The third sample (epIII+IV-c3) is a small plutonium metal disk with a radius of 1.65 cm and a thickness of 0.6 mm. The metal of this sample is a PuGa alloy with 1.5% Ga. The sample was placed at the center of the detector cavity. The isotopic composition given in [Pee+09] is the composition at the time of the sample characterization (July 1996). The measurement of the sample took place together with all other samples in February 2007. Over time, the sample underwent radioactive decay and its isotopic composition changed. To carry out adequate simulations, it was necessary to calculate this isotopic change. Given the mentioned dates, a decay time of 10 years and 7 months (3.3372×10^8 s) was assumed. The initial and the resulting isotopic compositions are shown in table 6.3³⁴.

Sample four and five (epIII+IV-c4, epIII+IV-c5) are plutonium oxide samples. The samples have been characterized in November 1987, hence a decay for 6.0737×10^8 s (19 years, 3 month) was necessary to calculate. Initial and decayed isotopic compositions, as well as sample masses, are shown in table 6.3. The plutonium oxide is stored in specific containers, which consist of an inner cylinder and an outer rotationally symmetric body. A generic drawing of such a container is shown in figure 6.5, respective parameters are listed in table 6.4. In case of sample epIII+IV-c4, the container was positioned 10 cm from the bottom of the cavity, for epIII+IV-c5 it was positioned immediately on the bottom. It was assumed that the plutonium oxide powder would fill a cylindrical volume at the bottom of the inner container. The size of this volume for each sample was calculated based on mass and density of the sample.

The last sample for phase III and IV of the benchmark is a sample containing mixed-oxide (MOX) material. The sample was characterized in April 1988, a decay for a time of 5.9395×10^8 s, or 18 years and 10 months was assumed. Isotopic composition of the plutonium as characterized and as used for simulations is listed in table 6.3. The uranium in the mix has the isotopic composition of natural uranium (^{234}U : 5.7×10^{-3} wt%, ^{235}U : 0.7204 wt%, ^{238}U : 99.2739 wt%)³⁵. In [Pee+09], uranium and plutonium masses are given, from which a fraction of plutonium in the MOX of 20 wt% can be calculated. For the calculation of the volume of the sample, a density of 0.8 g cm^{-3} and the given plutonium mass was used³⁶. The sample is enclosed in a container similar to the previous samples. Figure 6.5 shows a generic model, and table 6.4 the relevant parameters. It was modeled that the plutonium would fill the inner container from

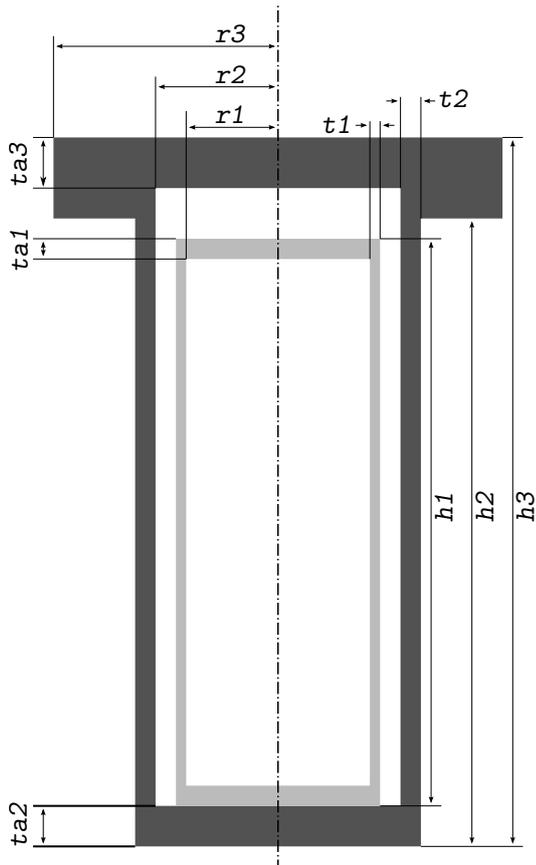
³⁴ For the composition of the sample after decay, only plutonium isotopes and ^{241}Am were taken into account. Spontaneous fission products are clearly negligible, also other actinides only occur with small fractions (less than 0.1 wt%) and have minor contributions to the source.

³⁵ This composition was taken from the values in PyNE for the material "Uranium, Natural (NU)".

³⁶ [Pee+09] also lists a total sample mass, however the difference between the sum of plutonium and uranium is too high to be only coming from the oxygen atoms - hence the plutonium mass was selected as a reference value.

Table 6.3.: Isotopic compositions for different samples of phase III and phase IV. The sample epIII+IV-c3 is a plutonium metal sample, epIII+IV-c4 and epIII+IV-c5 are plutonium oxides, epIII+IV-c6 is a MOX sample. The rows marked as "original" show the values as specified in [Pee+09], the rows marked as "decay" show the values after the respective decay times (refer to text). The latter values have been used for the simulations.

Sample	Total Mass g	Pu Mass g	Density g cm ⁻³	²³⁸ Pu wt%	²³⁹ Pu wt%	²⁴⁰ Pu wt%	²⁴¹ Pu wt%	²⁴² Pu wt%	²⁴¹ Am wt% of Pu
epIII + IV-c3 original	9.77753	9.455	19.05	0.13	75.66	21.49	1.95	0.77	1.86
epIII + IV-c3 decay	9.77737	9.37559	19.05	0.120589	76.2777	21.6478	1.17741	0.776507	2.62612
epIII + IV-c4 original	58.9274	51.455	2.6	0.199	70.955	24.583	3.288	0.975	1.02
epIII + IV-c4 decay	58.9258	50.368	2.6	0.174608	72.4463	25.0625	1.32058	0.996007	1.02
epIII + IV-c5 original	1145.02	999.825	2.6	0.199	70.955	24.583	3.288	0.975	1.02
epIII + IV-c5 decay	1144.99	978.703	2.6	0.174608	72.4463	25.0625	1.32058	0.996007	1.02
epIII + IV-c6 original	955.188	168.151	0.8	0.17	66.54	28.02	3.26	2.01	0.81
epIII + IV-c6 decay	955.183	164.676	0.8	0.149594	67.9076	28.5545	1.33598	2.05235	0.81



	epIII+IV-c4	epIII+IV-c5	epIII+IV-c6
r1	17.4	29.15	41.3
t1	1.6	2.6	3.05
r2	26	38.5	52
t2	1	1.5	2
r3	43	55	67.5
h1	156	220	270
ta1	5	5	5
h2	168	242	293
h3	189	266	332
ta2	6	8	10
ta3	11	11	13

Figure 6.5.: Generic drawing of the PERLA containers used for samples epIII+IV-c4, epIII+IV-c5 and epIII+IV-c6. The respective values are shown in the table on the right.

Table 6.4.: Parameters for the containers of samples epIII+IV-c4, epIII+IV-c5 and epIII+IV-c6. The values correspond to the figure on the left. All values are given in mm.

the bottom to the given volume. The total container was placed at the bottom of the detector cavity.

6.4 ESARDA Benchmark - Validation results

Although the described ESARDA benchmark was split in four phases, the discussion of results here is organized in two parts, combining similar phases. The first part describes the results obtained by the phases II and IV, arguably the easier parts of the benchmark as they only required the analysis of provided pulsetrains. ONMS is capable of doing such an analysis by skipping the Monte Carlo transport part and loading pulsetrains from files. The second part of this chapter describes results for phases I and III, which were full Monte Carlo simulations of given samples.

6.4.1 Pulsetrain Analysis - Phase II and IV

Phase II of the ESARDA benchmark consisted of analyzing the artificially created pulsetrains, which were provided to all participants (13 files). The files contained between approx. 5 million and 19 million neutron events each. From these files, Singles, Doubles and Triples were calculated by ten benchmark participants. Similar analyses were repeated using the functionality of ONMS. To compare further capabilities, benchmark participants also calculated the results based on four different dead time assumptions, leading to five different sets of calculations:

- no dead time
- dead time of $0.5\ \mu\text{s}$, non-updating
- dead time of $0.5\ \mu\text{s}$, updating
- dead time of $2\ \mu\text{s}$, non-updating
- dead time of $2\ \mu\text{s}$, updating

Of the ten benchmark participants, not all participants were able to give results for all five assumptions. ONMS is able to do pulsetrain analyses for all five assumptions, hence all 65 analyses were carried out. All used a predelay of $4.5\ \mu\text{s}$, a long delay of $4096\ \mu\text{s}$ and a gate length of $64\ \mu\text{s}$. The values for predelay and gate width have been specified in the benchmark descriptions [PS06]. The value for the long delay is a commonly used value, different values were used by different participants, varying from $1000\ \mu\text{s}$ to $4096\ \mu\text{s}$.

As no measurements were taken in phase I and II of the ESARDA benchmark, the benchmark authors provided theoretical calculations of the results based on the point model. The alpha ratio α is calculated based on the sample specifications. The pulsetrains were produced using a Monte Carlo detector model using MCNPX, from the same calculations efficiency and sample multiplication were extracted.

For the Singles rate estimated from the pulsetrain, very good agreement among all benchmark participants and ONMS has been achieved. This is expected, as the Singles ideally should represent the number of events in the analyzed file. Small discrepancies

occur due to different treatments of the end of the pulsetrain. Some of the last events have to be neglected in the analysis, as it is not possible to analyze for events in the long delay window. Depending on the size of the long delay, different numbers are removed from the total count rate.

The main results for double and triple rates for the case without dead time are shown in figure 6.6. To improve the representation, all results have been normalized based on the point-model calculations, which is represented by an orange line at $x = 1$. The x-axis for every sample has been scaled to include the ONMS result and most of the benchmark participants. Sometimes, the point-model value is visible far off the values resulting from pulsetrain analysis, due to reasons as described above. In red, the results as calculated with ONMS are shown. The error bars for ONMS depict the standard deviation of individual results of ten equal splits of the single pulsetrain file.

The benchmark results are shown in blue, different dots in a single plot represent different benchmark participants. The uncertainties shown are those that were reported in [PS06]³⁷ In the tables included in [PS06], the origin of the ten different sets of results is only listed as a participant number. For the plots in figure 6.6, benchmark participant “two” of the paper is left out - this result often was up to twice the value of the average.

Also missing from the figure are the results for epI+II-c1-10, epI+II-c1-100 and epI+II-c1-1000. These samples were all AmLi samples, which only emit uncorrelated neutrons, hence one would expect Doubles and Triples to be zero. No benchmark participant that provided results was able to reproduce this value, also ONMS is not able to reproduce the point model values of zero. Instead they all show very different values, both positive and negative. For ONMS, this happens due to the methods used to generate the foreground and background multiplicity distributions based on shifting gate windows. To achieve zeros for Doubles and Triples, both distributions would need to be exactly equal. However it is clear that due to the random nature of radioactive decay, some $64\mu\text{s}$ windows might include slightly more events than others, leading to differences in the distributions, which in turn lead to non-zero Doubles and Triples.

For all other samples shown in figure 6.6, the pulsetrain analyses of ONMS show very good agreement with other codes. The resulting values for Doubles and Triples are for all cases inside the respective set of the results by different benchmark participants. It is clearly visible from the plots that in general better results are achieved for samples with lower count rates. The large samples 'epI+II-c3b' and 'epI+II-c4b' as well as the samples with high count rates 'epI+II-c2-1000' and 'epI+II-c5-1000' show the largest spread among benchmark participants³⁸. Nevertheless, for these samples ONMS shows good results. Given the large variety of samples, it should be clear that the pulsetrain analysis method of ONMS clearly works well.

³⁷ Depending on the calculation method and interpretation of the benchmark, it is possible that participants either gave the standard deviation of subsets, or the standard error of the mean. This might explain some of the differences in the size of the uncertainties.

³⁸ For these samples, the x-axis range is also the largest in absolute values.

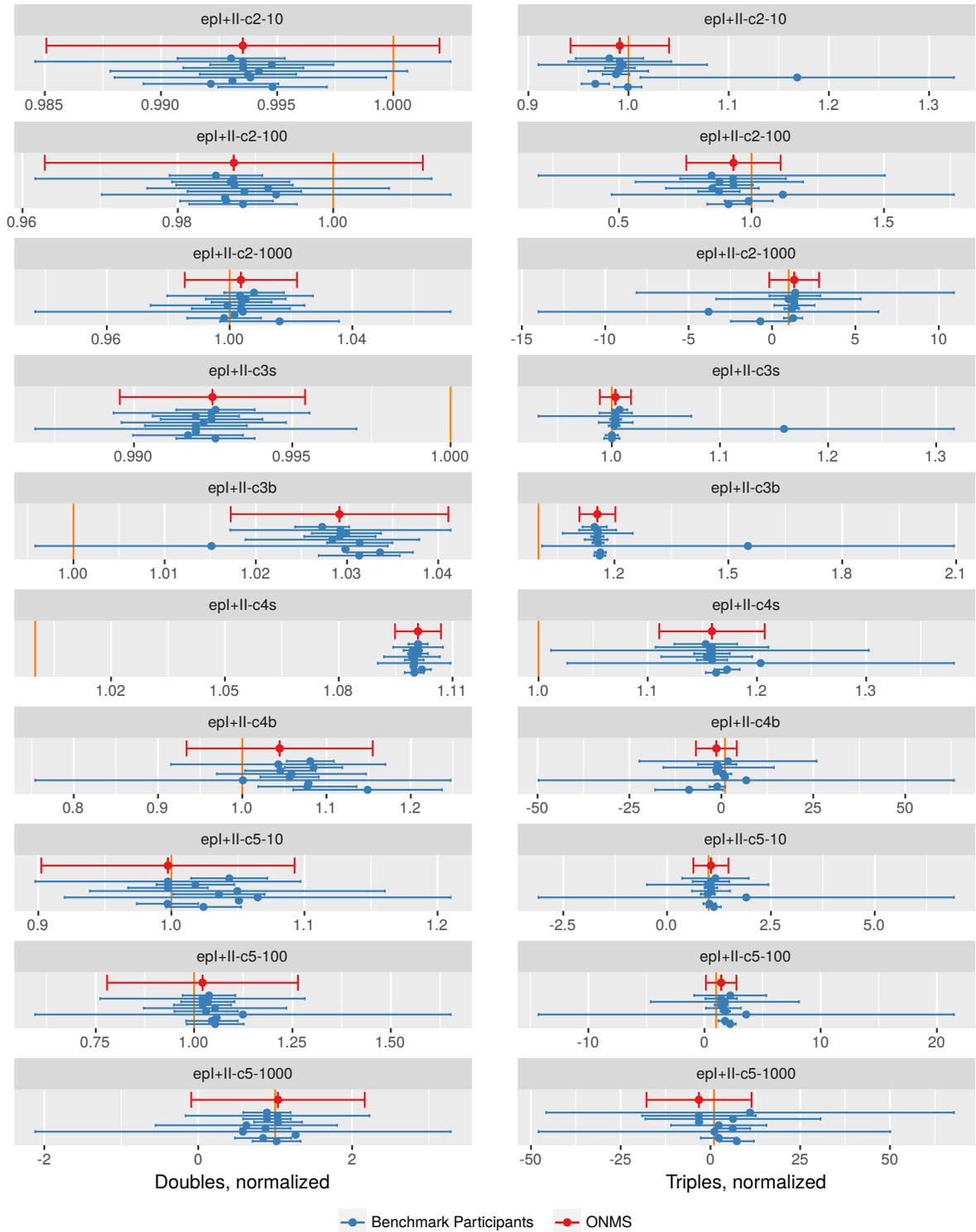


Figure 6.6.: Pulstrain analysis for ESARDA benchmark phase II, no dead time. Results are normalized with regard to the point model results given in [PS06] (orange line, if visible in shown range).

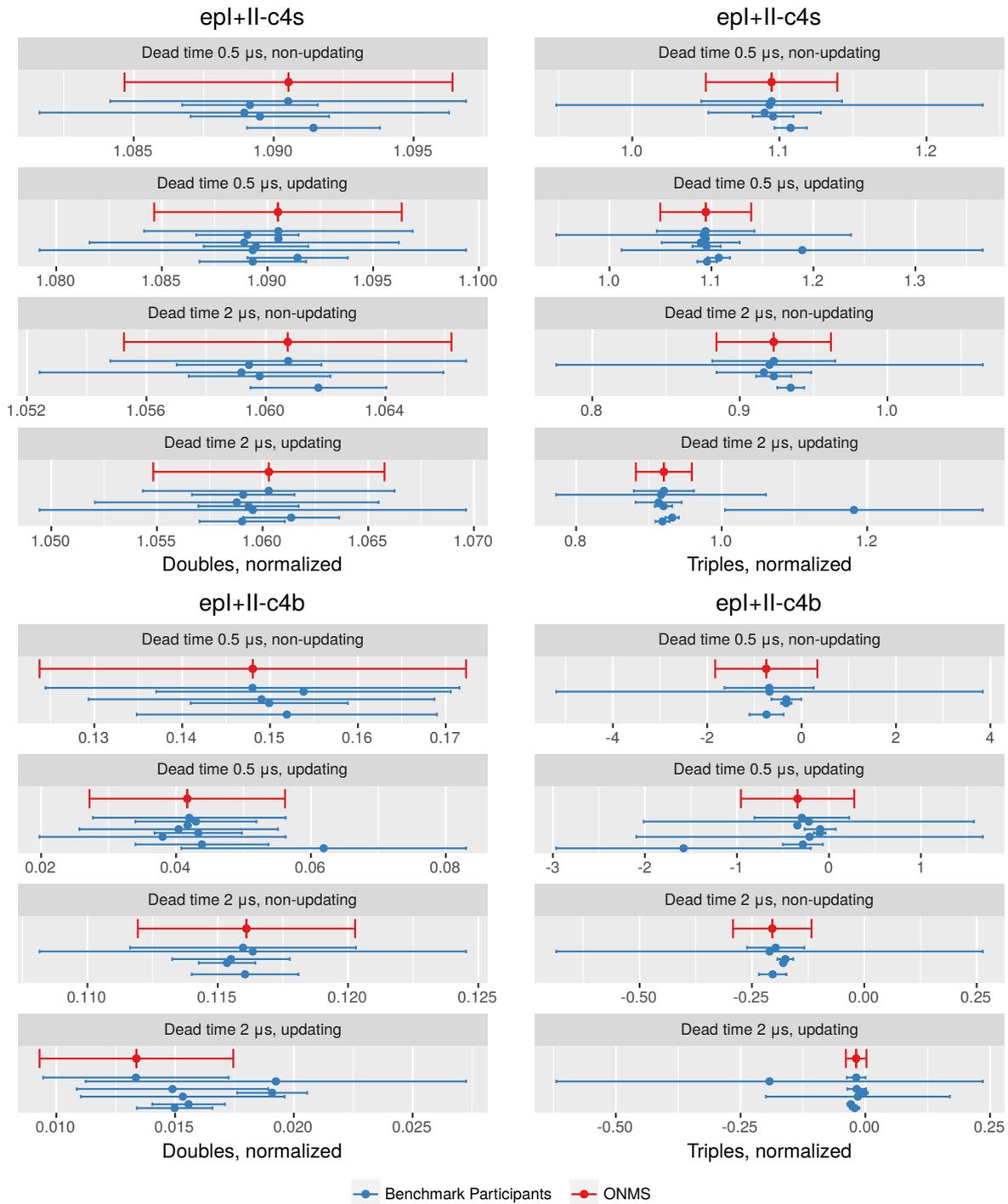


Figure 6.7.: Pulsetrain analysis including different assumptions for dead time for the samples epl+II-c4s and epl+II-c4b. Dead times are given in μ s. Results are normalized with regard to the point model results given in [PS06]. ONMS calculations are shown in red, Benchmark participants in blue.

Results using different dead time assumptions are shown in figure 6.7. Here, only those results for the small and large version of the fourth sample (epI+II-c4s, epI+II-c4b) are shown. Full plots of all samples for all four different dead time assumptions are shown in appendix D. Again, the pulsetrain analysis routine of ONMS gives good results for all different types of dead times. Clearly visible in the picture are also the differences in dead time effects between smaller and larger samples. For the epI+II-c4s sample, a relatively small plutonium mass with a low rate of neutron emission, results are still relatively close to the point model ($x = 1$). The much larger epI+II-c4b sample, however, shows very large distances to the point model results. The count rate of this sample is so large that many counts are removed from the pulsetrain due to dead time effects, especially for updating (accumulating) dead times. These effects reduce the Singles rate and subsequently Doubles and Triples rates.

The pulsetrains provided for the ESARDA benchmark phase IV are results from actual measurements, making them more realistic test objects. For each of the six different samples, ten files have been provided, each the result of a measurement of approximately 100 s. The files contain between 81 000 and 15×10^6 samples.

The results for double and triple rates of phase IV are shown in figure 6.8, again Singles are not shown as very good agreement of values is achieved between all benchmark participants and ONMS. The points displayed in the figure are the averaged values of the analysis of the ten individual files. Error bars for ONMS results in this case are the standard deviations of the ten calculations from the mean. As each sample was measured ten times in this phase, the statistical uncertainty of the ONMS results has been estimated calculating the standard deviation of the individual values, as recommended in the benchmark specifications in [Pee+09]. According to the report, this method was also applied by the majority of the benchmark participants³⁹.

Similar to phase II, ONMS shows very good agreement with the results of the benchmark participants, being inside of the set of results for most calculated values. The double rates typically vary by less than 0.5 %, except for the samples epIII+IV-c4 and epIII+IV-c6, where they vary about 1 %. Triples rate variation is about a factor of ten higher. The benchmark states that “the difference between the results is small for most practical purposes” [Pee+09, p. 7]. As the ONMS results completely fall in the field of benchmark participants, one can assume that the statement would apply for them, too.

As no mass estimates for either of the two phases were given by the benchmark participants and discussed, no calculations of mass have been carried out with ONMS. As discussed earlier, this would require additional assumptions, e.g. detector gate fractions.

In conclusion it can be stated that the ONMS pulsetrain analysis part works very well compared to other existing tools. As part of the article describing the benchmark, it was also discussed to introduce a standardized algorithm to carry out such analysis. This could be beneficial, especially for applications in nuclear arms control, as it would allow

³⁹ Some participants carried out additional splitting of the pulsetrains.

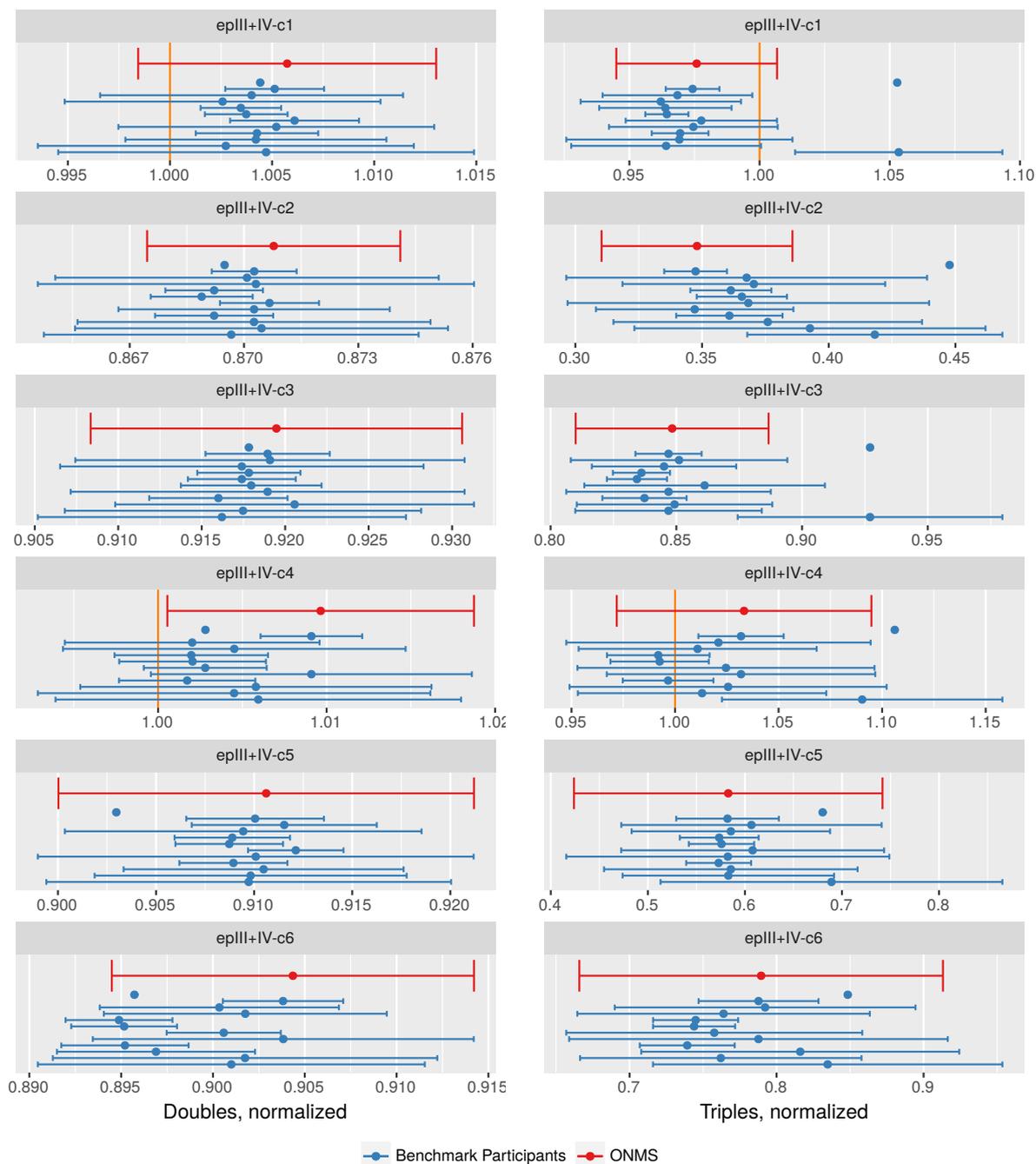


Figure 6.8.: Pulsetrain analysis for ESARDA benchmark phase IV. Results are normalized with regard to the point model values given in [Pee+09] (orange line, if visible in shown range). ONMS calculations are shown in red, Benchmark participants in blue.

data consistency and easier comparison of results. Here the open source characteristic of ONMS might be useful, as it allows for very easy implementation of new methods, and also provides a transparent way to analyze the currently used method.

6.4.2 Monte Carlo Simulations - Phase I and III

For phases I and III, full Monte Carlo simulations have been carried out for the neutron emissions by the samples and their tracking through the detector. This was followed by a pulsetrain analysis of the detected events. Much less participants in the ESARDA benchmark carried out these exercises. It should be noted that ONMS is the only application based on a Monte Carlo transport capability different from MCNP. All participants of the two described phases of the benchmark use MCNP as the Monte Carlo transport model of their application, adding only additional source routines, data sets and analysis functions.

The statistical uncertainties calculated with ONMS (and other codes if given) for most of the samples of these phases are relatively large. This is the case even though for all samples the number of particles simulated were corresponding to typical real measurement times⁴⁰. Therefore, it should not be concluded that the uncertainties are results of too small numbers of simulated particles (statistical uncertainty), but more likely characteristic for the methodology to create the measured multiplicities and derive Singles, Doubles and Triples rates from the neutron pulsetrain.

The results for phase I are shown in figure 6.9. In this case all three important values, Singles rate, Doubles rate, and Triples rate are shown in the graph. Variations in the single rate are possible because samples' source terms had to be individually simulated by ONMS (and every benchmark participant). Not shown in the figure are again the AmLi samples (epI+II-c1-10, epI+II-c1-100, epI+II-c1-1000). The Singles rates as simulated with ONMS for these three samples only show very small deviations from the point model and the other benchmark participants. For Doubles and Triples, variation is very large between all calculations, due to the reasons discussed in the previous subsection. Also not shown are results for the samples containing a combined PuO₂ and AmLi source (epI+II-c5-10, epI+II-c5-100, epI+II-c5-1000), as ONMS does not have a function to combine the two types of sources required for that sample in a simulation. All results do not include any dead time correction. The ONMS results were calculated by simulating ten runs for each sample. The simulated measurement time for each run was either assumed to be long enough to generate more than 20 million neutron events for the larger samples or 1000 s. The values shown for ONMS are the average values obtained by pulsetrain analysis of the ten runs, the uncertainties are based on the standard deviation of these values. No uncertainties were given in [PS06] for the benchmark participants, hence no uncertainties are included in the plot.

⁴⁰ Often, Monte Carlo Simulations simulate only small fractions of the particles actually physical present, e.g. in a reactor.

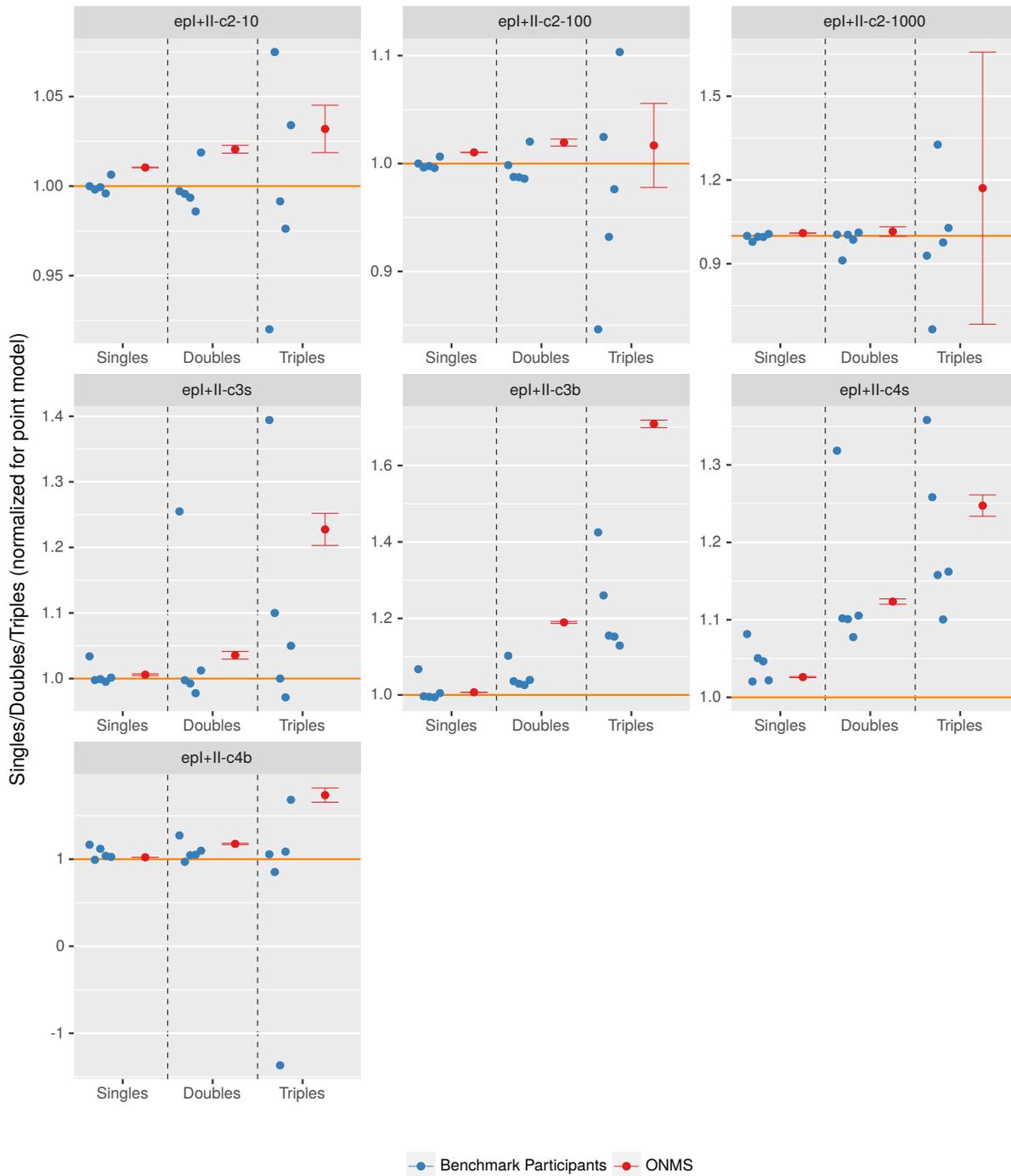


Figure 6.9.: Results of ONMS simulations for ESARDA Benchmark phase I in comparison with benchmark participants' results. The error bars for the ONMS values are show the standard deviation of ten independent calculations for each sample. Results are normalized with regard to the point model values given in [PS06] (orange line).

Clearly visible is the fact that the discrepancies are much higher among benchmark participants compared to the previously shown results of pulsetrain analysis. Closest agreement among benchmark participants and ONMS is reached for the three different ^{252}Cf samples (epI+II-c2-10, epI+II-c2-100, epI+II-c2-1000). This is reasonable, as these samples do not have any neutrons resulting from (α, n) reactions or multiplication. Nevertheless, differences for Triples can go up to 30 % in the epI+II-c2-1000 case. The rising spread in results and increasing uncertainty among these samples directly relates to the count rate. For the lowest count rate, less than one neutron is detected in a foreground gate of $64\mu\text{s}$ length, which allows for good detection of correlated and uncorrelated neutrons. In contrast, with the strongest source nearly 65 neutrons are detected in the same gate length, making differentiation between individual spontaneous fission events significantly harder.

Throughout, the plutonium samples show larger differences among the benchmark participants themselves as well as ONMS. Good agreement between benchmark participants, ONMS and point model can be found for the Singles rate. For the small oxide sample (epI+II-c4s), deviations of all simulations from the point model are largest. This is most likely due to a small underestimation of the sample multiplication used for the calculation. With regard to the double count rate, ONMS result show good agreement for all plutonium samples, they are close to the average of the benchmark participants, with the exception of the large metallic sample (epI+II-c3b). This sample is also the sample with by far the highest multiplication. The ONMS results are about 10 % higher than the highest value of a benchmark participant.

The Triples rates for the plutonium samples show a large spread among the simulated results, for one sample a benchmark participant is even off by approx. 200 %. For the two oxide samples and the small metal sample (epI+II-c4s, epI+II-c4b), ONMS simulations yield a Triples rate that is slightly higher than the average, which is a reasonable result. For the small metal sample, the value is inside the range of benchmark participants, but about 10 % higher than all except for one.

For the big metal sample, the ONMS overestimates Triples 20 % higher than biggest value for metal sample. Such a value means that the multiplicity distributions derived from the pulsetrain are slightly shifted towards higher moments. Other studies found that this is a common problem also for other simulation codes [MS10] [Mil+11]. The authors of these studies suggested that it could probably be resolved by a modified $\bar{\nu}$ value for ^{239}Pu . At the current stage, no such modification has been implemented in ONMS.

Figure 6.10 shows the results for simulations of the ESARDA benchmark phase III. Similarly to phase I, the figure shows Singles, Doubles and Triples. Values for ONMS are again based on the average of 10 independent runs, and the simulated measurement time for each run was either assumed to be either long enough to generate more than 20 million neutron events per run for the larger samples or 1000 s. The figure also shows experimental values, which are typically lower than the point model values, and

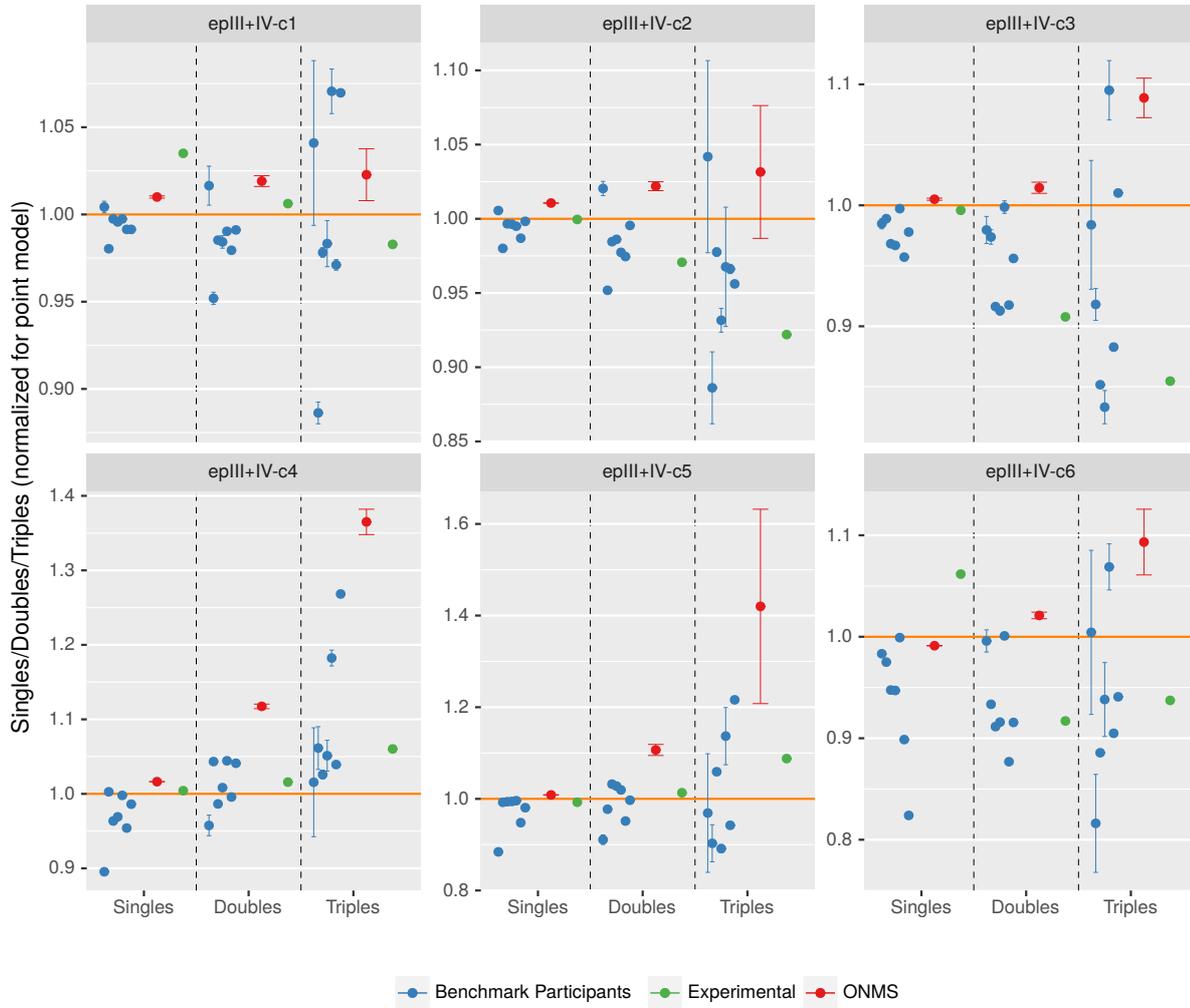


Figure 6.10.: Results of ONMS simulations for ESARDA Benchmark phase III in comparison with benchmark participants' results. The error bars for the ONMS values are show the standard deviation of ten independent calculations for each sample. Results are normalized with regard to the point model values given in [Pee+09] (orange line).

also most of the simulations. This is due to inherent physical dead time effects of the measurement systems, which were not accounted for in the simulations.

Very good results with regard to all three rates are achieved by ONMS for the two ^{252}Cf samples (epIII+IV-c1, epIII+IV-c2), as well as from most of the benchmark participants. For the four plutonium sample, the single rates are slightly overestimated, as well as the Doubles. The absolute overestimation is smaller than the general spread of the results by different benchmark participants. As the Doubles are directly proportional to Singles, the overestimation in the latter also increases the values of the first. For the Triples, results are inside the range of benchmark participants for sample epIII+IV-c3, and very close for epIII+IV-c6. For epIII+IV-c4 and epIII+IV-c5, results for Triples are overestimated more.

An important factor that might contribute to the differences between point model, benchmark participants and the ONMS results is the uncertainty of the four sample's isotopic composition. [Pee+09] only specified isotopic composition at the time of sample characterization, not at the time when the actual measurements took place. The age of the samples used for the ONMS calculations was received in personal communication with the author. It is not clear, however, what ages were assumed by the benchmark participants. Other possible contributing factors are similar to those discussed previously - detector design, sample geometry, and also differences in used nuclear data.

Overall, the results are good, although some of the validation simulations show values slightly off the point model. Hence, ONMS should be considered a viable tool to carry out neutron multiplicity simulations.



7 Conclusion and Outlook

Currently and in the future, effective nuclear arms control agreements require technical verification mechanisms. If new agreements will be negotiated, as for example a ban of nuclear weapons, such needs for verification technology will increase as well. Scientists and engineers play important roles carrying out developments to meet these necessities. Constantly, new ideas are proposed and studied, and their implementation is discussed and tested.

As a technical contribution to the field of nuclear arms control, an open source code to carry out simulations of neutron multiplicity measurements was developed in this thesis. Such measurements can be employed to estimate plutonium masses of samples, for example in the context of nuclear safeguards or for nuclear disarmament verification. The code, “Open Neutron Multiplicity Simulation” (ONMS) is able to carry out all required tasks for simulations of such measurements. These tasks include

- Monte Carlo particle transport of neutrons through complex detector and sample geometries, including the treatment of thermal neutron scattering in moderators as polyethylene;
- adequate source definitions, including neutrons from spontaneous fission reactions and (α, n) reactions;
- subsequent analysis of the neutron pulsetrains that are produced in the simulation by neutron detection in specific detector elements.

Several codes exist that can carry out all tasks, developed over years by groups of researchers in research centers around the world. Some more applications are available that can carry the last task only. Nearly all existing comprehensive codes for the simulation of neutron multiplicity measurements rely on MCNP⁴¹, an export-controlled and proprietary application, for the Monte Carlo transport of neutrons. ONMS uses a different approach that has not been used before - it is based on the Geant4 framework, a freely available set of routines for Monte Carlo particle transport.

Many functions for the three described tasks were implemented from scratch for ONMS, Geant4 mainly supplied the basic Monte Carlo transport routines. In the ONMS source routines, special functions for the treatment of (α, n) reactions were included, which do not exist in other codes. They allow for an automatic calculation of (α, n) source strength and neutron energy spectrum based on material specifications given by users. The routine for the analysis of the detected neutron pulsetrain was implemented based on commonly used shift register approaches. This routine was implemented to be usable

⁴¹ Before they were merged, this includes MCNP and MCNPX alike.

with the application developed in the Geant4 framework, but could also be used as a library for other programs. During the development, it was also made sure that all the required nuclear data sets are available without restrictions.

The functionality of ONMS was validated in three ways. First, simulated results were compared to experimental data generated by Malte Göttsche, based on his measurements of eight different plutonium samples in metallic and oxide form. Second, the pulsetrain analysis module was validated using two sets of pulsetrains that had been generated for a large benchmark exercise by the Non-Destructive Assay Working Group of ESARDA. The results of the analysis could be compared to results produced by other code systems and theoretical values. Lastly, full Monte Carlo simulations have been carried out based on the described benchmark, again comparing results to other codes and theoretical values. While showing slightly larger discrepancies for large samples, overall the validation showed that the code works well and could be a versatile tool to carry out simulations of neutron multiplicity measurements.

Based on these results of this thesis, future research opportunities and needs can be identified with regard to the development of ONMS. It might be useful to further analyze the differences shown in the results of the discussed validation efforts. That could involve a renewed effort to check accuracy of detector and sample geometries and materials, and also make use of different nuclear cross section evaluations. While a small set of different nuclear data sets has been already tested for some samples, no full study of the influence of various parameters has been carried out. In the longer run, it is also possible to continue pursuing the approach for (α, n) treatment that actually tracked the transport of α particles through the geometry. This approach has not been followed up in this thesis due to high computational needs. An increase in computing power might make that approach feasible. While simulations can be carried out without such an approach, it would be a valuable addition for the simulation of complex sample geometries.

In the current version of ONMS, the pulsetrain analysis and subsequent calculation of plutonium mass follows the most common approach used, which is based on several assumptions as listed in chapter 4. Over time, several additions to this approach have been proposed, for example the use of sample specific correction factors that improve the results for high multiplication [Göt15], or the use of different methods to generate the gates for the extraction of the measured multiplicity distributions [CHH12; Hen+12]. It would be feasible to improve ONMS by implementing some of these approaches, depending on intended further use cases.

While running many calculations using the Geant4 module NeutronHP / ParticleHP, it became also clear that there would also be room for performance improvements. Currently, Geant4 carries out Doppler broadening of neutron interaction cross sections on-the-fly, i.e. whenever such a reaction could take place. Other Monte Carlo codes allow for the use of specially prepared cross sections for specific temperatures, which include already broadened peaks. It would be possible to implement a similar routine

for Geant4, and prepare temperature-specific cross sections. This would involve changing parts of the Geant4 code - to add such a functionality permanently to Geant4 would require consent by the Geant4 Collaboration.

Beyond the development of ONMS as a technical tool, it has a significance for an important broader context, which was also identified and studied in this thesis. Verification technologies have an inherent problem which goes beyond common research that focuses on technical aspects or direct effects of a technologies implementation. The issue can be subdivided in at least three parts: First, technology not only need to produce valid results, but all actors involved, states and others need trust the results. It is crucial to establish confidence in the complex technical systems required for verification. Second, there is a need for transparency of the means and assumptions that a technology is based on, to allow everyone to understand and follow the conclusions drawn. And lastly, it is important that the tools are available to at least every actor involved, but also to the society at large, to ensure broad participation and democratic mechanisms in the technical arms control process. All these points are relevant because the results of verification technologies are often the basis for decision making by and among states, nevertheless they are currently not addressed in research related to nuclear arms control.

The problems have been discussed in detail in the beginning of the thesis, focusing on the role of software. After identifying and describing the problem, a new solution has been proposed. Software as part of verification technologies should adhere to three open source criteria, defined as follows:

1. No restrictions for access to the program.
2. Distribution of the program must include the full source code.
3. Modifications of the program are allowed to anybody.

These criteria have been derived from definitions of open source software and free software, as they are common in software development in general. Open source software and the connected open source movement have become increasingly popular in the last two decades. By offering a different way to create and maintain software, the open source approach has contributed to significant changes in the world of software engineering. To implement the criteria for nuclear arms control, either new software would have to be developed or the distribution methods of existing software would need to be changed significantly. With the development of ONMS, it can be shown that it is possible to write such an application as an open source tool and publish it on a common source distribution platform⁴². While both, the technical work and the study of the role of open source software could have been done in separate works, it was very beneficial as a combined project - an approach that could also be repeated in the future.

Applying open source criteria to software in nuclear arms control has a number of benefits. Software that fulfills the three criteria could enable more trusted and transparent

⁴² <http://github.com/nuclearfreesoftware/onms>

verification exercises. Source code availability allows experts and a broader audience to put tools under extensive scrutiny, looking for attempts of malicious and also unintentional cheating and misbehavior. Easily accessible source code also makes underlying assumptions transparent, at least to the extent of a third parties' code reading abilities. Allowing everyone to access software would allow for broad participation in the usage for arms control purposes, but also for the development of the tools. This could lead to better software, and to increased capacity building.

Looking at the process of other open source projects, some experiences would have to be taken towards the field of nuclear arms control. While it is possible for many to participate in a project, it is not guaranteed to happen. Important applications implemented as open source software still need the structural backing of a state or an international organization. Similarly important are clear decision making structures - the process to allow developers to implement changes to critical software should include several steps of adequate quality assessment. Often, free availability of software is perceived as a risk due to dual-use aspects, for example using the software for weapons development instead to help disarm weapons. This should be addressed by trying to develop software with reduced dual-use characteristics. At the same it should be carefully checked that access limitations do not harm possible verification uses for more than needed. Overall, the benefits most likely outweigh the risks.

From early on, ideas of similar notion have been discussed with regard to nuclear arms control. As early as in 1946 in the Acheson Lilienthal Plan proposed a global community of knowledge as a necessary precondition for successful international cooperation for the control of nuclear weapons. Clearly, the open source approach can help with the creation of such a community, and would also help to make nuclear arms control more democratic.

With regard to future steps, a concrete possibility based on the open source nature of ONMS could be a new research opportunity for detector development. Instead of relying on conventional approaches, one could also host an open challenge, where groups would submit designs for such detectors, focusing their optimization for example on very high efficiency or the reduction of ^3He needs. That should not indicate that current detectors are badly designed, but by allowing participants from different backgrounds to participate, probably new innovative ideas could be generated. Such a challenge benefits from the open source characteristics, but also requires a good documentation and specification of the task. If carried out on a broad scale, it would also be a good way to spread knowledge and increase the number of experts on neutron multiplicity counting and simulation.

In relation to the open source criteria, the most important next step would be the implementation of the criteria for further software tools. In several presentations given as part of this project, such an outreach was attempted and often well received. At the same time, the discussions on this work showed the need for a set of instructions, on how to convert software to open source software, explaining the exact meaning of the term and the necessary steps to make software open source. A generic "handbook"

containing these information might be worth considering, and could help the transition process.

The introduction of more open source tools should be accompanied by a broader discussion of open source aspects in the arms control community. It is important to know what reasons there are that limit different actors from using open source software and from making software available, and what would be required to overcome this reasons.

A different pathway forward are outreaching attempts to different communities. It is one of the benefits of the open source approach that it does not limit participation to arms control specialists. For example, it might be useful to address the group of so-called hackers. The word hacker is often misunderstood for people that have bad intentions. Here, it is not used in that way, but to describe talented and innovative software developers. If involved, they could help creating more trusted systems, and also support the task of spotting potential malfunctions in important code parts.

Besides software, the approach could be extended to hardware, to all technologies involved in verification. Open source criteria for hardware are more difficult to review successfully. It is possible to specify design criteria, electric circuits and physical properties, but to check if an actual part of an equipment adheres to all these criteria is harder to achieve. While software can be exactly copied to allow others to study it, hardware always consists of unique physical objects.

Beyond making the tools openly available, an even broader aspect should be taken into consideration: It would not help to make software and hardware openly available if there is a lack of open physics knowledge, which might exist due to classification and sensitivity issues that states apply. Over time, solutions should be found to ensure that all relevant physics necessary for verification technologies is accessible. Without such a step, the differences in knowledge could again create mistrust and transparency, even with fully open software and hardware.

Probably, such secrecy will not be necessary anymore in a world without nuclear weapons. On the way towards such a world, open source software in general and the tool developed as part of this work hopefully can contribute to foster sharing and participation, trust and transparency.



Part III.

Appendix



A JENDL/AN-2005 Cross Sections

The data set JENDL/AN-2005 is an evaluation of cross sections for inelastic interactions of α -particles, with a particular focus on (α , n) reactions. It has been put together by members of the Charged Particle Nuclear Data Working Group, which was established under the Japanese Nuclear Data Committee at the Japan Atomic Energy Research Institute and is described in detail in [MMS06]. It is an updated version of JENDL/AN-2003 ([MS02; MY02] evaluation. The evaluation includes several low-Z isotopes, listed in table A.1. Comprehensive description of measurement data (and references) are given in the report [MMS06]. Typically, the authors used available data from a number of measurements, and the mEXIFON code to produce the cross sections.

Table A.1.: Isotopes included in JENDL/AN-2005, including a list of reactions available for each isotope. In the reaction column, “np” stands for total neutron production cross section, which combines the respective neutron producing cross sections of the isotope.

Isotope	Z AID	Filename	MT	Reactions
⁶ Li	3006	Li006.dat	4, 22, 28, 50-53/91, 201	(a,n), (a,a'n), (a,pn), (a,n0-3/c), np
⁷ Li	3007	Li007.dat	4, 22, 50-54/91, 201	(a,n), (a,a'n), (a,n0-4/c), np
⁹ Be	4009	Be009.dat	4, 22, 50-52/91, 201	(a,n), (a,a'n), (a,n0-2/c), np
¹⁰ B	5010	B010.dat	4, 22, 28, 50-54/91, 201	(a,n), (a,a'n), (a,pn), (a,n0-4/c), np
¹¹ B	5011	B011.dat	4, 16, 28, 50-54/91, 201	(a,n), (a,2n), (a,pn), (a,n0-4/c), np
¹² C	6012	C012.dat	4, 50, 201	(a,n), (a,n0) - both equal, np
¹³ C	6013	C013.dat	4, 22, 28, 50-54/91, 201	(a,n), (a,a'n), (a,pn), (a,n0-4/c), np
¹⁴ N	7014	N014.dat	4, 22, 28, 50-54/91, 201	(a,n), (a,a'n), (a,pn), (a,n0-4/c), np
¹⁵ N	7015	N015.dat	4, 22, 50-54/91, 201	(a,n), (a,a'n), (a,n0-4/c), np
¹⁷ O	8017	O017.dat	4, 22, 50-53/91, 201	(a,n), (a,a'n), (a,n0-3/c), np
¹⁸ O	8018	O018.dat	4, 16, 22, 50-54/91, 201	(a,n), (a, 2n), (a,a'n), (a,n0-4/c), np
¹⁹ F	9019	F019.dat	4, 22, 28, 50-77/9, 201	(a,n), (a,a'n), (a,pn), (a,n0-27/c), np
²³ Na	11023	Na023.dat	4, 28, 50-78/9, 201	(a,n), (a,pn), (a,n0-28/c), np
²⁷ Al	13027	Al027.dat	201	np
²⁸ Si	14028	Si028.dat	201	np
²⁹ Si	14029	Si029.dat	201	np
³⁰ Si	14030	Si030.dat	201	np

According to the report, for every cross section thick-target neutron yields were calculated based on [Zie77] stopping power data. After that, the authors claim to have adjusted the cross sections to closely match yield data from [BC79] and [WS82], how-

ever no information on uncertainties is given in the report. The cross sections data files are available online for download⁴³.

A set of small scripts has been written to process these cross sections, and turn them into a format that is similar to the G4NDL file format for low energy neutron cross sections in Geant4. The scripts are all part of the Github Repository for ONMS, located under the subfolder tools/JENDL-conversion

The script `download.sh` downloads and extracts the cross sections into a new folder `JENDL-AN-2005`. As the files have Windows newline (CRLF), an additional folder is created and files are converted to UNIX style newlines. The original files have minor ENDF errors with regard to the used notation. As this does lead to readout errors depending on the used readout routine, the errors have been fixed using patch files. The first MAT, 1,451/ blank, blank, MF1, MT1, NC1, MOD1]CONT entry, which holds the number of lines for every MF/MT entry lists one line less than necessary for the MF=1, MT=451 entry. This error is occurring in B010.dat, B011.dat, C012.dat, C013.dat, F019.dat, Li006.dat, Li007.dat, N014.dat, N015.dat, Na023.dat, O017.dat, O018.dat.

In addition, all evaluation files have the following headline:

```
JENDL/AN-2005                                0          0
```

but should have

```
JENDL/AN-2005                                0 0 0 0
```

Two patch files, `451-lines.patch` and `headline.patch` were created that contain all necessary changes to correct the missing lines. The patches can be executed using the script `patch.sh`. Conversion to G4NDL format is significantly more complex. While the ENDF file format is well documented [HT09], only parts of the G4NDL structure are documented in [Gea15b]. A short script `convert.py` was written to do the conversion. It relies on the three Python modules, that were specifically developed for this task. `newmf.py` uses routines from PyNE to create a complete data structure in Python that holds the data of the ENDF files. `geantdata.py` can produce G4NDL formatted files based on this data structure. `endfgeant4converter.py` combines the other two modules and takes care of the management of input and outputfiles.

Important for the routine for (α, n) reactions in `NMSMaterialDecaySource` are not only the production cross sections, but also the partial cross sections (MT=50-91). These cross sections describe reactions that leave the residual nucleus in an excited state and are used for the calculation of the neutron energy spectrum. The files produced by the conversion script are stored in the folder called `output`. To be able to use them with ONMS, the environmental variable `NMSALPHALEDATA` has to be set to include that path.

The scripts also read MF=6 data from the ENDF files. The MF=6 section contains energy-angle distributions. These distributions are also converted to G4NDL files. They

⁴³ <http://wwwndc.jaea.go.jp/ftpnd/ftp/JENDL/jendlan2005.tar.gz>

are not used for typical ONMS reactions, but have been used in the development process to test the implementation of new (α , n) cross sections for α -particle transport in Geant4.



B Example GDML file

The following listing shows a GDML file for the model of the sample “PM1”, as described in section 6.1.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2 <gdml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../schema/gdml.xsd"
3   ">
4   <define>
5     <position name="NMSGDML-SamplePosition" unit="cm" x="0" y="0" z="-6.45" />
6   </define>
7
8   <materials>
9     <isotope N="58" Z="28" name="Ni58">
10      <atom unit="g/mole" value="57.935342" />
11    </isotope>
12    <isotope N="60" Z="28" name="Ni60">
13      <atom unit="g/mole" value="59.930786" />
14    </isotope>
15    <isotope N="61" Z="28" name="Ni61">
16      <atom unit="g/mole" value="60.931056" />
17    </isotope>
18    <isotope N="62" Z="28" name="Ni62">
19      <atom unit="g/mole" value="61.928345" />
20    </isotope>
21    <isotope N="64" Z="28" name="Ni64">
22      <atom unit="g/mole" value="63.927967" />
23    </isotope>
24    <isotope N="63" Z="29" name="Cu63">
25      <atom unit="g/mole" value="62.929598" />
26    </isotope>
27    <isotope N="65" Z="29" name="Cu65">
28      <atom unit="g/mole" value="64.927790" />
29    </isotope>
30    <isotope N="238" Z="94" name="Pu238">
31      <atom unit="g/mole" value="238.049560" />
32    </isotope>
33    <isotope N="239" Z="94" name="Pu239">
34      <atom unit="g/mole" value="239.052164" />
35    </isotope>
36    <isotope N="240" Z="94" name="Pu240">
37      <atom unit="g/mole" value="240.053814" />
38    </isotope>
39    <isotope N="241" Z="94" name="Pu241">
40      <atom unit="g/mole" value="241.056852" />
41    </isotope>
42    <isotope N="242" Z="94" name="Pu242">
43      <atom unit="g/mole" value="242.058743" />
44    </isotope>
45    <isotope N="241" Z="95" name="Am241">
46      <atom unit="g/mole" value="241.056829" />
47    </isotope>
48    <element name="El-Ni_of_NMSSourceMaterial">
49      <fraction n="0.680770" ref="Ni58" />
50      <fraction n="0.262230" ref="Ni60" />
51      <fraction n="0.011399" ref="Ni61" />
52      <fraction n="0.036346" ref="Ni62" />
53      <fraction n="0.009255" ref="Ni64" />
54    </element>
55    <element name="El-Cu_of_NMSSourceMaterial">
56      <fraction n="0.691500" ref="Cu63" />
57      <fraction n="0.308500" ref="Cu65" />
58    </element>
59    <element name="El-Pu_of_NMSSourceMaterial">
60      <fraction n="0.000040" ref="Pu238" />
61      <fraction n="0.954385" ref="Pu239" />
```

```

62 <fraction n="0.045110" ref="Pu240" />
63 <fraction n="0.000317" ref="Pu241" />
64 <fraction n="0.000148" ref="Pu242" />
65 </element>
66 <element name="El-Am_of_NMSSourceMaterial">
67 <fraction n="1.000000" ref="Am241" />
68 </element>
69 <material name="NMSSourceMaterial" state="solid">
70 <D unit="g/cm3" value="20.059704" />
71 <fraction n="0.055000" ref="El-Ni_of_NMSSourceMaterial" />
72 <fraction n="0.035000" ref="El-Cu_of_NMSSourceMaterial" />
73 <fraction n="0.907770" ref="El-Pu_of_NMSSourceMaterial" />
74 <fraction n="0.002229" ref="El-Am_of_NMSSourceMaterial" />
75 </material>
76
77 <material Z="13" name="G4_A1" state="solid">
78 <MEE unit="eV" value="166"/>
79 <D unit="g/cm3" value="2.699"/>
80 <atom unit="g/mole" value="26.9815"/>
81 </material>
82 <isotope N="54" Z="26" name="Fe54">
83 <atom unit="g/mole" value="53.9396"/>
84 </isotope>
85 <isotope N="56" Z="26" name="Fe56">
86 <atom unit="g/mole" value="55.9349"/>
87 </isotope>
88 <isotope N="57" Z="26" name="Fe57">
89 <atom unit="g/mole" value="56.9354"/>
90 </isotope>
91 <isotope N="58" Z="26" name="Fe58">
92 <atom unit="g/mole" value="57.9333"/>
93 </isotope>
94 <element name="Fe">
95 <fraction n="0.05845" ref="Fe54"/>
96 <fraction n="0.91754" ref="Fe56"/>
97 <fraction n="0.02119" ref="Fe57"/>
98 <fraction n="0.00282" ref="Fe58"/>
99 </element>
100 <isotope N="50" Z="24" name="Cr50">
101 <atom unit="g/mole" value="49.946"/>
102 </isotope>
103 <isotope N="52" Z="24" name="Cr52">
104 <atom unit="g/mole" value="51.9405"/>
105 </isotope>
106 <isotope N="53" Z="24" name="Cr53">
107 <atom unit="g/mole" value="52.9407"/>
108 </isotope>
109 <isotope N="54" Z="24" name="Cr54">
110 <atom unit="g/mole" value="53.9389"/>
111 </isotope>
112 <element name="Cr">
113 <fraction n="0.04345" ref="Cr50"/>
114 <fraction n="0.83789" ref="Cr52"/>
115 <fraction n="0.09501" ref="Cr53"/>
116 <fraction n="0.02365" ref="Cr54"/>
117 </element>
118 <isotope N="58" Z="28" name="Ni58">
119 <atom unit="g/mole" value="57.9353"/>
120 </isotope>
121 <isotope N="60" Z="28" name="Ni60">
122 <atom unit="g/mole" value="59.9308"/>
123 </isotope>
124 <isotope N="61" Z="28" name="Ni61">
125 <atom unit="g/mole" value="60.9311"/>
126 </isotope>
127 <isotope N="62" Z="28" name="Ni62">
128 <atom unit="g/mole" value="61.9283"/>
129 </isotope>
130 <isotope N="64" Z="28" name="Ni64">
131 <atom unit="g/mole" value="63.928"/>
132 </isotope>
133 <element name="Ni">
134 <fraction n="0.680769" ref="Ni58"/>
135 <fraction n="0.262231" ref="Ni60"/>
136 <fraction n="0.011399" ref="Ni61"/>
137 <fraction n="0.036345" ref="Ni62"/>

```

```

138     <fraction n="0.009256" ref="Ni64"/>
139 </element>
140 <material name="Aisi304Steel" state="solid">
141   <T unit="K" value="293.15"/>
142   <MEE unit="eV" value="282.839371081725"/>
143   <D unit="g/cm3" value="7.92"/>
144   <fraction n="0.705" ref="Fe"/>
145   <fraction n="0.19" ref="Cr"/>
146   <fraction n="0.105" ref="Ni"/>
147 </material>
148
149 <isotope N="12" Z="6" name="C12">
150   <atom unit="g/mole" value="12"/>
151 </isotope>
152 <isotope N="13" Z="6" name="C13">
153   <atom unit="g/mole" value="13.0034"/>
154 </isotope>
155 <element name="C">
156   <fraction n="0.9893" ref="C12"/>
157   <fraction n="0.0107" ref="C13"/>
158 </element>
159 <isotope N="14" Z="7" name="N14">
160   <atom unit="g/mole" value="14.0031"/>
161 </isotope>
162 <isotope N="15" Z="7" name="N15">
163   <atom unit="g/mole" value="15.0001"/>
164 </isotope>
165 <element name="N">
166   <fraction n="0.99632" ref="N14"/>
167   <fraction n="0.00368" ref="N15"/>
168 </element>
169 <isotope N="16" Z="8" name="O16">
170   <atom unit="g/mole" value="15.9949"/>
171 </isotope>
172 <isotope N="17" Z="8" name="O17">
173   <atom unit="g/mole" value="16.9991"/>
174 </isotope>
175 <isotope N="18" Z="8" name="O18">
176   <atom unit="g/mole" value="17.9992"/>
177 </isotope>
178 <element name="O">
179   <fraction n="0.99757" ref="O16"/>
180   <fraction n="0.00038" ref="O17"/>
181   <fraction n="0.00205" ref="O18"/>
182 </element>
183 <isotope N="36" Z="18" name="Ar36">
184   <atom unit="g/mole" value="35.9675"/>
185 </isotope>
186 <isotope N="38" Z="18" name="Ar38">
187   <atom unit="g/mole" value="37.9627"/>
188 </isotope>
189 <isotope N="40" Z="18" name="Ar40">
190   <atom unit="g/mole" value="39.9624"/>
191 </isotope>
192 <element name="Ar">
193   <fraction n="0.003365" ref="Ar36"/>
194   <fraction n="0.000632" ref="Ar38"/>
195   <fraction n="0.996003" ref="Ar40"/>
196 </element>
197 <material name="G4_AIR" state="gas">
198   <MEE unit="eV" value="85.7"/>
199   <D unit="g/cm3" value="0.00120479"/>
200   <fraction n="0.000124000124000124" ref="C"/>
201   <fraction n="0.755267755267755" ref="N"/>
202   <fraction n="0.231781231781232" ref="O"/>
203   <fraction n="0.0128270128270128" ref="Ar"/>
204 </material>
205 </materials>
206
207 <solids>
208   <tube aunit="deg" deltaphi="360" lunit="mm" name="SolidPu" rmax="3.75" rmin="0" startphi="0" z="15.5"/>
209   <tube aunit="deg" deltaphi="360" lunit="mm" name="totalAlCase" rmax="7" rmin="0" startphi="0" z="46"/>
210   <tube aunit="deg" deltaphi="360" lunit="mm" name="innerAlCase" rmax="4" rmin="0" startphi="0" z="31"/>
211   <subtraction name="AlCaseSolid">
212     <first ref="totalAlCase"/>
213     <second ref="innerAlCase"/>

```

```

214     <position name="AlCaseSolid_pos" unit="mm" x="0" y="0" z="-2.5"/>
215 </subtraction>
216 <tube aunit="deg" deltaphi="360" lunit="mm" name="PerlaCavity" rmax="11.5" rmin="0" startphi="0" z="69.5"
    />
217 <tube aunit="deg" deltaphi="360" lunit="mm" name="totalPerla" rmax="18" rmin="0" startphi="0" z="90"/>
218 <tube aunit="deg" deltaphi="360" lunit="mm" name="outerPerlaNon" rmax="18.1" rmin="12.5" startphi="0" z="
    65"/>
219 <subtraction name="PerlaSolid">
220   <first ref="totalPerla"/>
221   <second ref="outerPerlaNon"/>
222   <position name="PerlaSolid_pos" unit="mm" x="0" y="0" z="-12.6"/>
223 </subtraction>
224 </solids>
225
226 <structure>
227   <volume name="PM1">
228     <materialref ref="NMSSourceMaterial"/>
229     <solidref ref="SolidPu"/>
230   </volume>
231   <volume name="AlCase">
232     <materialref ref="G4_Al"/>
233     <solidref ref="AlCaseSolid"/>
234   </volume>
235   <volume name="PerlaCavity">
236     <materialref ref="Aisi304Steel"/>
237     <solidref ref="PerlaCavity"/>
238     <physvol name="NMSSourceVolume">
239       <volumeref ref="PM1"/>
240       <position name="PM1_pos" unit="mm" x="0" y="0" z="-22.0"/>
241     </physvol>
242     <physvol name="AlCase">
243       <volumeref ref="AlCase"/>
244       <position name="AlCase_pos" unit="mm" x="0" y="0" z="-11.75"/>
245     </physvol>
246   </volume>
247   <volume name="NMSLVName-SampleMother">
248     <materialref ref="G4_AIR"/>
249     <solidref ref="PerlaSolid"/>
250     <physvol name="PerlaCavity">
251       <volumeref ref="PerlaCavity"/>
252       <position name="PerlaCavity_pos" unit="mm" x="0" y="0" z="-7.25"/>
253     </physvol>
254   </volume>
255 </structure>
256
257 <setup name="Default" version="1.0">
258   <world ref="NMSLVName-SampleMother"/>
259 </setup>
260
261 </gdml>

```

C Usage of Open Neutron Multiplicity Simulation

C.1 Command Line Options

A small number of command line options has been defined for ONMS, which can be used when ONMS is started. Most settings specified using command line options cannot be changed at runtime.

Table C.1.: Command line options for ONMS.

-h / -help	Show help
-D <detector>	GDML File with Detector specification (Default: gdml/detector/PSMC.gdml)
-S <sample>	GDML File with Sample specification (Default: gdml/sample/PM1.gdml)
-n	Do not use PhysicsList with thermal neutron scattering (it is used as default)
-m <macrofilename>	Load <macrofilename> (Default: vis.mac)
-b <macrofilename>	Run <macrofilename> in batch mode
-l <logfile>	Write all output to log file <logfile>
-p <name>	Name for all output files (log, result etc.).
-r <no1> <no2>	Create file random_seeds.txt with <no2> lines, each line containing <no1> random seeds

C.2 Messenger Options

Geant4 allows application developers to make use of specific messenger classes to define user interface commands. These commands can be called at runtime and allow users to change settings, but also to control the program flow. All commands listed in table C.2 have been specifically programmed for ONMS. They can be used to work with the program interactively, e.g. to load and analyze pulsetrain data files using different settings.

Table C.2.: Messenger Commands

Command	Description
Control event runs	
/ONMS/run/runtime	Set measurement time to be simulated
/ONMS/run/printmodulo	Every time the event number modulo the given value is zero, status information is printed
/ONMS/run/beamOnRuntime	Start calculation, the number of events is calculated based on source activity
Parameters for analysis and result output	
/ONMS/analysis/detectorvolume	Set volume name of the volume where neutron absorption is added to pulsetrain
/ONMS/analysis/lostvolume	Set volume name of the volume where neutron absorption is counted as lost neutron
/ONMS/analysis/showresults	Show results of current run or loaded pulsetrain data
/ONMS/analysis/ writesourceneutronenergies	Write list of source neutron energies to file
/ONMS/analysis/eventoffset	Set an event-id offset for pulsetrain (e.g. for multiple parallel runs)
/ONMS/analysis/writeresults	Write results to files (detector statistics, multiplicity results and settings)
/ONMS/analysis/writeresultsafterrun	Automatically export results after each run
/ONMS/analysis/writeincludepulsetrain	Turn pulsetrain file export on/off for normal result write
/ONMS/analysis/readpulsetrain	Load a pulsetrain from a pulsetrain file. “.pulsetrain” is automatically added to filename
/ONMS/analysis/readshakespulsetrain	Load a pulsetrain from a file that contains a list of events with eventtimes in shakes
/ONMS/analysis/multiplicitylength	Set maximal multiplicity for R+A / A distributions
/ONMS/analysis/registerlength	Set length of shift register (in positions!)
/ONMS/analysis/registerperiod	Set register period of shift register
/ONMS/analysis/setpredelay	Predelay for pulsetrain analysis
/ONMS/analysis/setlongdelay	Long delay for pulsetrain analysis
/ONMS/analysis/setgate	Gate length for pulsetrain analysis
/ONMS/analysis/setderandomizeperiod	Set period for derandomize-quantization step
/ONMS/analysis/derandomizedo	Turn on/off derandomize quantization step
/ONMS/analysis/registerquantizationdo	Turn on/off register quantization step

Table continued on next page

Table C.2 – continued from previous page

Command	Description
/ONMS/analysis/efficiency	Set detector efficiency for mass calculations
/ONMS/analysis/dieawaymethod	Set method to calculate dieaway (0 = fixed dieaway time)
/ONMS/analysis/dieaway	Set neutron dieaway time for detector
/ONMS/analysis/infinitegate	Set length for 'infinite' gate to be used for gate fraction calculations
/ONMS/analysis/gatefractionmethod	Choose method for calculation of gate fractions
/ONMS/analysis/gatefractions	Set fixed gate fractions
/ONMS/analysis/predeadttime	Dead time for algorithm carried out pre quantization
/ONMS/analysis/predeadttimeupdating	Set dead time to be/not to be updating for algorithm carried out PRE quantization
/ONMS/analysis/postdeadttime	Dead time for algorithm carried out post quantization
/ONMS/analysis/postdeadttimeupdating	Set dead time to be/not to be updating for algorithm carried out POST quantization
/ONMS/analysis/pulsetrainanalysismode	Set pulsetrain analysis method (forward/backward)
General source settings	
/ONMS/source/verbose	Set verbose level for MaterialDecaySource
/ONMS/source/material	Select material for ONMSMaterialDecaySource
/ONMS/source/sourcefromvolume	Specify a physical volume name that should be used as source
/ONMS/source/cf252sfoptions	Set options for multiplicity and energy distribution of Cf252 spontaneous fission.
/ONMS/source/activevolume	Set active volume of source (density as in material specification)
/ONMS/source/activity	Set activity of source
/ONMS/source/activityfixed	Set to true to override automatic activity calculation.
/ONMS/source/sfn	Enable/disable spontaneous fission (neutron emission) as a decay mode for source
/ONMS/source/sfg	Enable/disable spontaneous fission (gamma emission) as a decay mode for source
/ONMS/source/alpha	Enable/disable α -decay for source
/ONMS/source/neutronAlphaN	Enable/disable neutrons from (α ,n) reactions for source

Table continued on next page

Table C.2 – continued from previous page

Command	Description
/ONMS/source/dumpstatus	Output current source status
Special settings for (α , n) reaction source	
/ONMS/source/alphan/directionSampling	Set method for sampling the position
/ONMS/source/alphan/energySampling	Set method for sampling the energy
/ONMS/source/alphan/positionSampling	Set method for sampling the position
/ONMS/source/alphan/energy	Set energy of neutrons emitted by (α , n) reactions
/ONMS/source/alphan/filename	Set file for input (α , n) reaction data
/ONMS/source/alphan/energyFilename	Set file with neutron energy spectrum for alpha,n reaction
/ONMS/source/alphan/writeEnergyFile	Output (α , n) reaction spectrum to file
/ONMS/source/alphan/energySpectrumIncludeMT91	Turn on/off inclusion of MT91 cross section (and Q-value) for spectrum calculation
/ONMS/source/alphan/activityCalculation	Set method for calculation of activity for
/ONMS/source/alphan/activity	Set activity of (α , n) neutron source
Source geometry settings	
/ONMS/source/geometry/type	Set ONMSMaterialDecaySource distribution type
/ONMS/source/geometry/shape	Set ONMSMaterialDecaySource shape for Surface or Volume source
/ONMS/source/geometry/volume	Confine source to volume (NULL to unset)
/ONMS/source/geometry/radius	Set radius.
/ONMS/source/geometry/inner_radius	Set inner radius when required.
/ONMS/source/geometry/halfx	Set inner radius when required.
/ONMS/source/geometry/halfy	Set inner radius when required.
/ONMS/source/geometry/halfz	Set z half length.
/ONMS/source/geometry/centre	Set centre coordinates of ONMSMaterialDecaySource
Random seeds	
/ONMS/randomSeedList	Creates a list of pairs of random seeds

It is also possible to combine commands in a macro file, which can be used to automatically run calculations. An example macro file is shown in the next section.

C.3 Example Macro file

The following macro file was used to calculate sample “PuO₂-20”. It was used as a split run, where many runs with a measurement length of 5 s were simulated using different random seeds. It uses neutron from spontaneous fission and (α , n) reactions as source particles. In

the end, results including statistics of the run (started / absorbed neutrons) as well as source settings and the produced pulsetrain are written to different files.

Listing C.1: Macro file for calculation of PuO₂-20 sample.

```
1 # Source Volume & Material
2 /NMS/source/material NMSSourceMaterial
3 /NMS/source/sourcefromvolume NMSSourceVolume
4
5 # Detector Settings
6 /NMS/analysis/detectorvolume HE1-He
7 /NMS/analysis/lostvolume PolyethyleneArea
8
9
10
11 # Source Type and settings
12 /NMS/source/sfg false
13 /NMS/source/sfn true
14 /NMS/source/alpha false
15 /NMS/source/neutronAlphaN true
16 # /NMS/source/activity - not given
17
18 # AlphaN Settings
19 /NMS/source/alphan/directionSampling 0
20 /NMS/source/alphan/energySampling 2
21 /NMS/source/alphan/positionSampling 0
22 # /NMS/source/alphan/energy - not given
23 # /NMS/source/alphan/filename - not given
24 /NMS/source/alphan/activityCalculation 2
25 # /NMS/source/alphan/activity - not given
26
27 # Runtime settings
28 /random/setSeeds 28607207 36633217
29 /NMS/run/runtime 5
30 /NMS/run/printmodulo 1000
31 /NMS/analysis/eventoffset 0
32
33 # Run
34 /NMS/run/beamOnRuntime 1
35
36 # Create some output
37 /NMS/source/dumpstatus
38 /NMS/analysis/writesourceneutronenergies puo2-20
39 /NMS/source/alphan/writeEnergyFile puo2-20
40 /NMS/analysis/showresults
41 /NMS/analysis/writeincludepulsetrain true
42 /NMS/analysis/writeresults puo2-20
```

C.4 Formats of result files produced by ONMS

After a pulsetrain analysis has been carried out, a results file similar to file in listing C.2 is produced. In the beginning, it includes information on the source particles generated (primaryalpha, primarygamma, primaryneutron, neutronpertime, primaryneutronenergy). The next set of information is related to the processes in the detector (secondaryneutron, totalneutron, secondaryneutronenergy, neutron-absorbeddetector, neutronabsorbedlostvolume). Often of interest are the values `efficiencysource`, giving the number of neutrons detected divided by the number of neutrons started in a source; `efficiencytotal` giving the number of neutrons detected divided by the number produced in any neutron producing reaction; and `neutronlifetime`, which gives the average lifetime of a neutron before leakage or absorption. All these values only depend on source definition and Monte Carlo transport.

All other values depend on the settings that were made for the pulsetrain analysis. The moments of derived from the pulsetrain are given as totals (singlestotals, etc.), per time (singlespertime). Also, values are calculated based on splitting the pulsetrain into ten smaller timesteps (avgsinglespertime). From this treatment, it is also possible to calculate statistical uncertainties (the standard deviation of the distribution, uncsinglespertime). The remaining values are the multiplication, spontaneous fission rate and α ratio as well as plutonium effective mass. The total plutonium mass is calculated based on the isotopic composition of the specified source material.

Listing C.2: Result file of a calculation of PuO₂-20 sample.

```
1 runtime,5,s
2 lastevent,4.99483,s
3 primaryalpha,0,1
4 primarygamma,0,1
5 primaryneutron,5989,1
6 secondaryneutron,345,1
7 totalneutron,6334,1
8 sfubar,0,n/SF
9 neutronpertime,1197.8,1/s
10 primaryneutronenergy,2.03991,MeV
11 secondaryneutronenergy,1.52004,MeV
12 neutronabsorbeddetector,3315,1
13 detectorvolname,HE1-He,-
14 neutronabsorbedlostvolume,1741,1
15 lostvolumename,PolyethyleneArea,-
16 efficiencysource,0.553515,1
17 efficiencytotal,0.523366,1
18 neutronlifetime,52.1742,ms
19 singlestotal,3315,1
20 doublestotal,625,1
21 triplestotal,104.17,1
22 singlespertime,663,1/s
23 doublespertime,125,1/s
24 triplespertime,20.8341,1/s
25 avgsinglespertime,663,1/s
26 avgdoublespertime,125,1/s
27 avgtriplespertime,20.8716,1/s
28 uncsinglespertime,52.7276,1/s
29 uncdoublespertime,27.47,1/s
30 unctriplespertime,11.4518,1/s
31 multiplication,1.00082346352,1
32 alpha,0.682605114316,1
33 fissionrate,334.761405542,1/s
34 pu240effmass,0.698876,g
35 pumass,5.03887,g
```

D Additional results for dead time calculations

The following figures show additional results for ESARDA Benchmark, phase II, including different dead time corrections. All results are normalized with regard to the point model results given in [PS06] (orange line). ONMS calculations are shown in red, Benchmark participants in blue. Results of ONMS calculation in comparison with benchmark participant values.

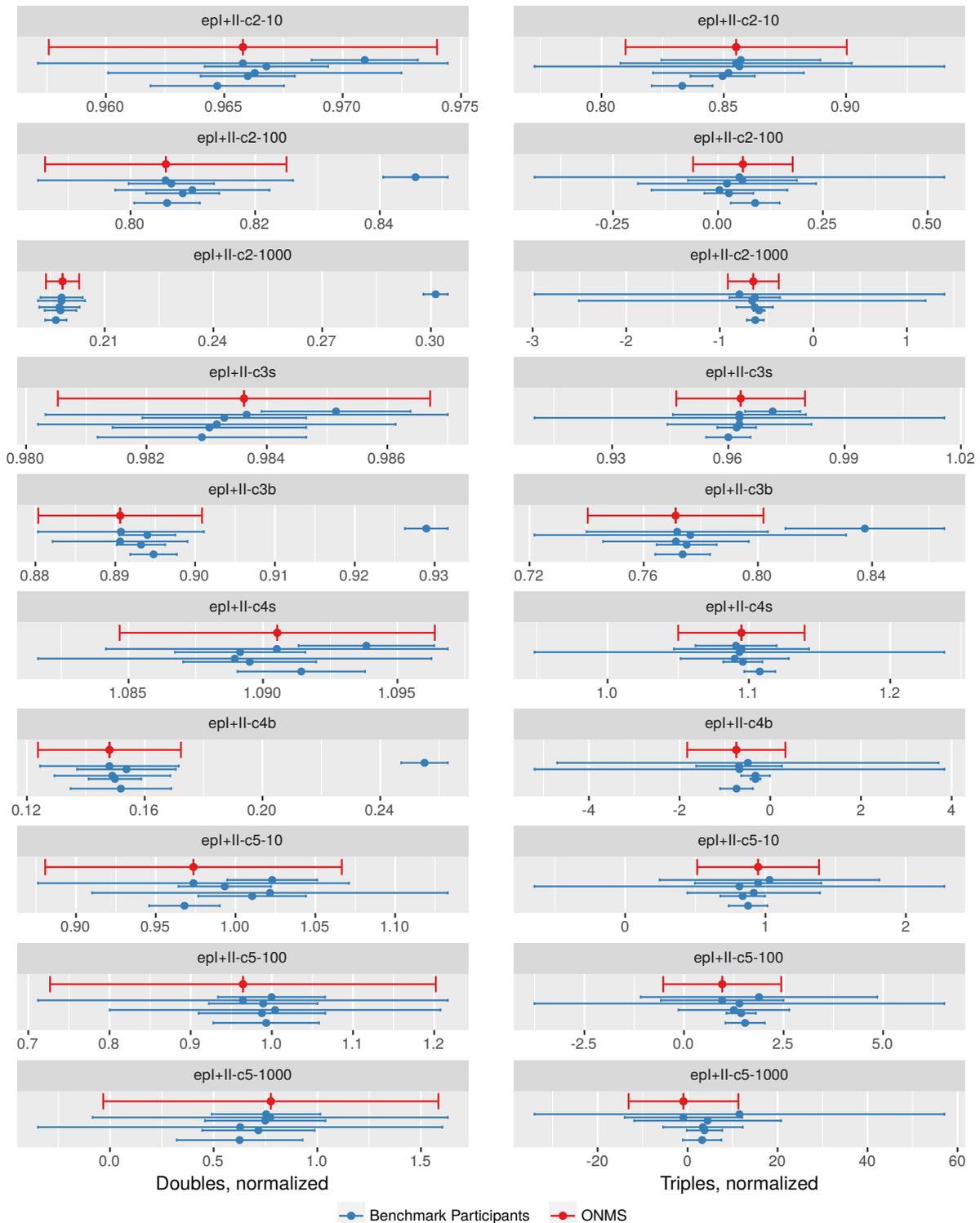


Figure D.1.: Pulsetrain analysis results, ESARDA benchmark phase II, assuming a dead time of $0.5 \mu\text{s}$, non-updating.

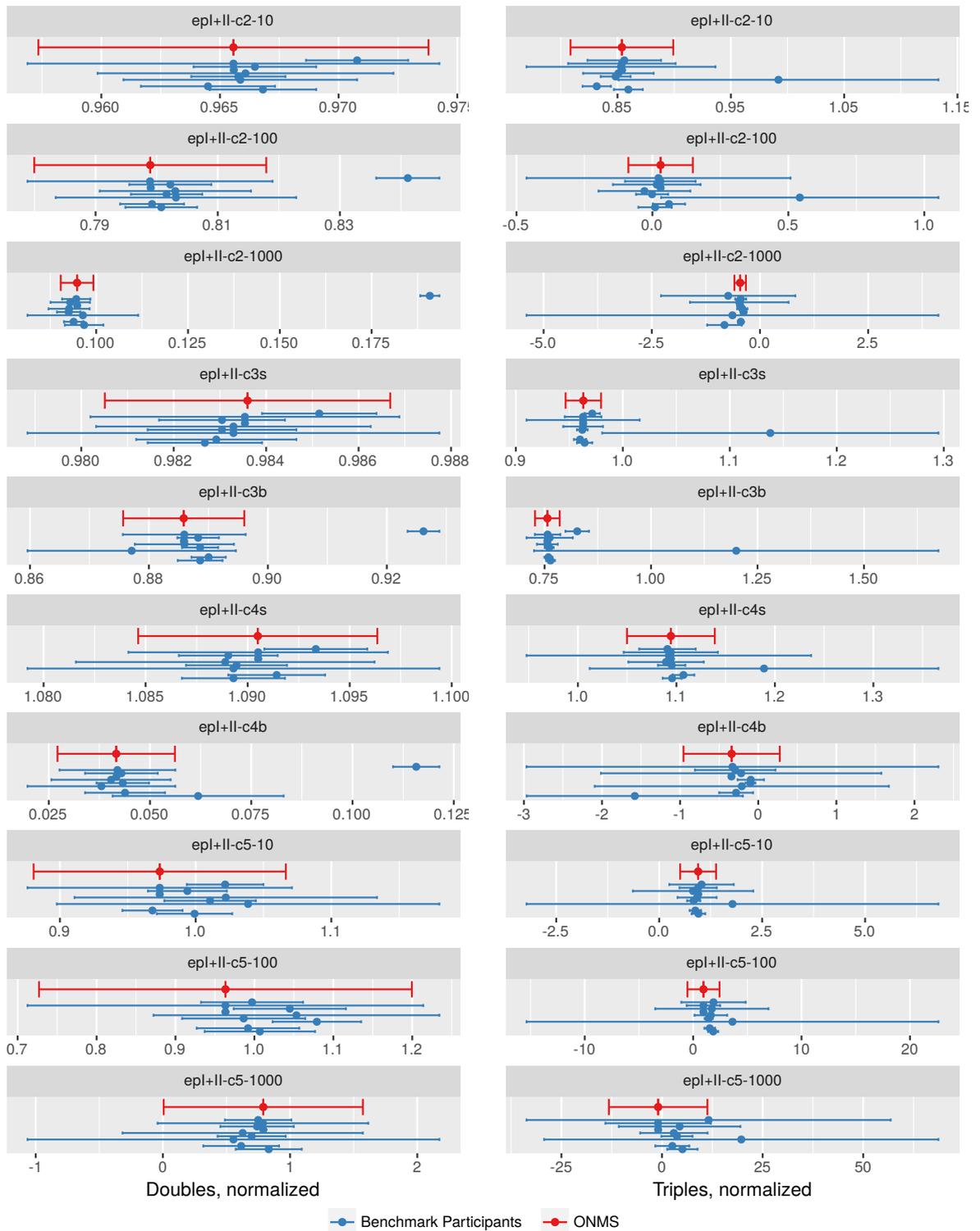


Figure D.2.: Pulsetrain analysis results, ESARDA benchmark phase II, assuming a dead time of $0.5 \mu\text{s}$, updating.

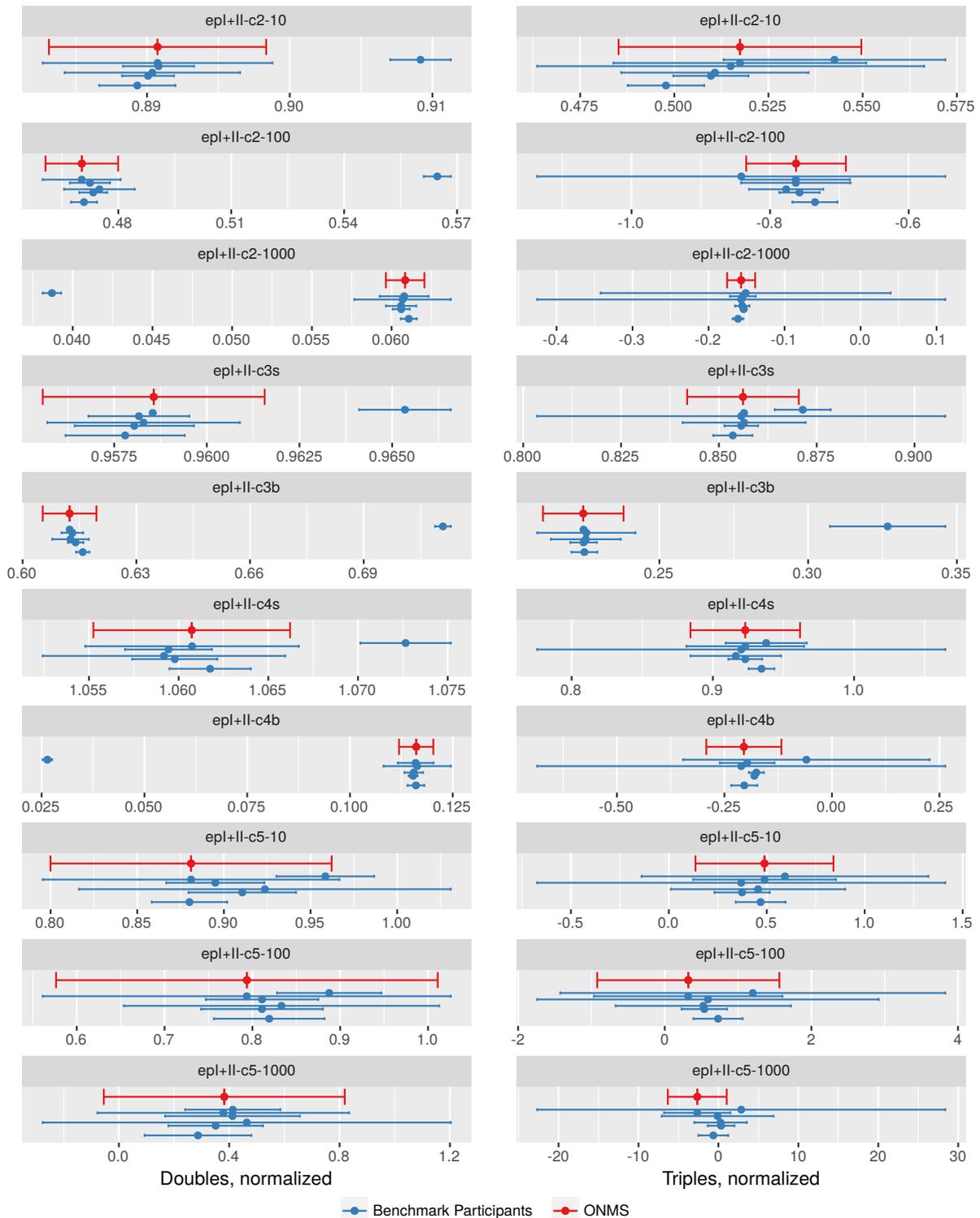


Figure D.3.: Pulsetrain analysis results, ESARDA benchmark phase II, assuming a dead time of $2\mu s$, non-updating.

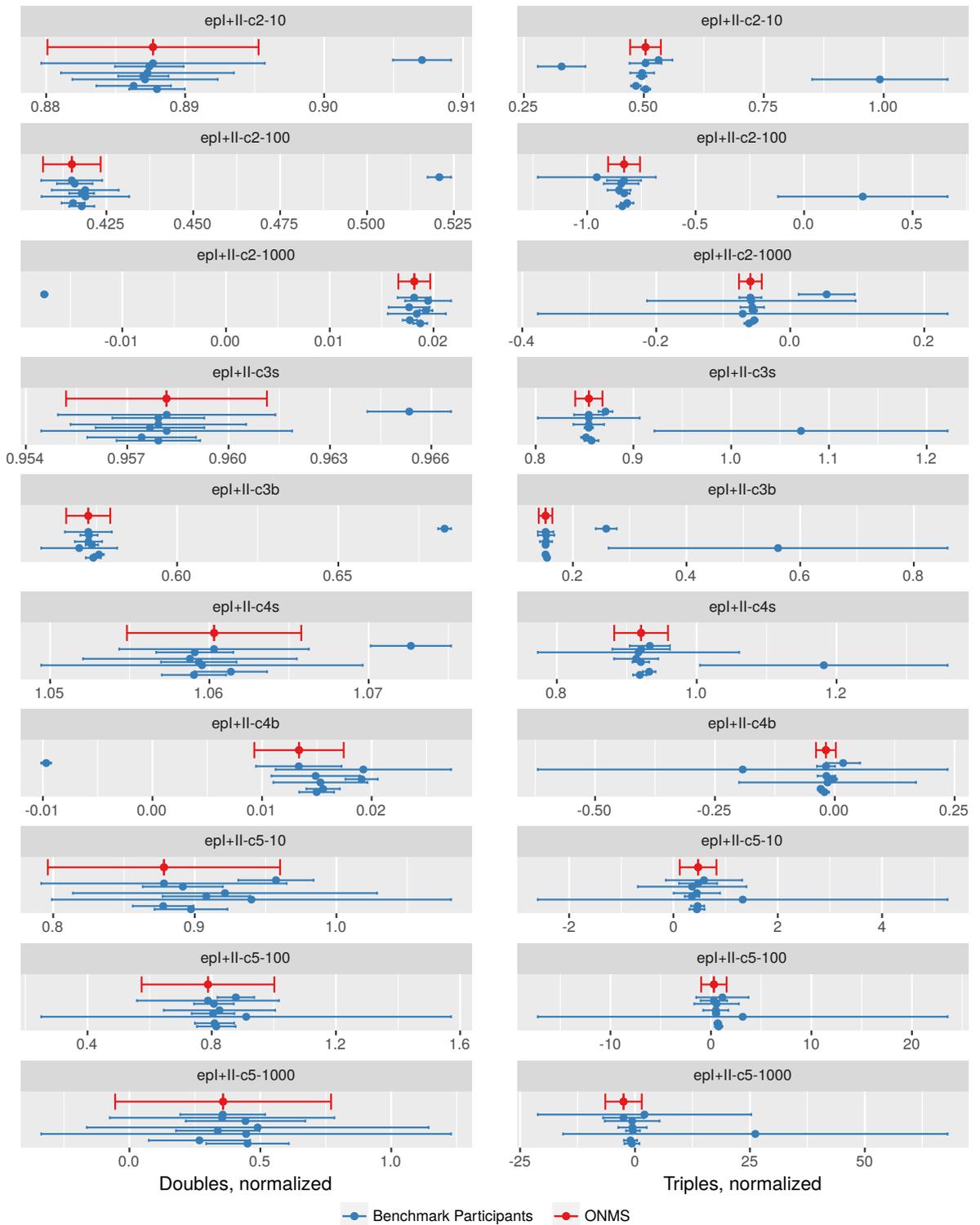


Figure D.4: Pulsetrain analysis results, ESARDA benchmark phase II, assuming a dead time of $2\mu\text{s}$, updating.



Bibliography

- [AG15] A. Ahmad and A. Glaser. “A Conversion Proposal for Iran’s IR-40 Reactor with Reduced Plutonium Production”. *Science & Global Security* 23.1 (Jan. 2015), pp. 3–19. DOI: 10.1080/08929882.2015.996074.
- [Ago+03] S. Agostinelli et al. “Geant4 - a simulation toolkit”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (3 2003), pp. 250–303. DOI: 10.1016/S0168-9002(03)01368-8.
- [Ako02] Y. A. Akovali. “Nuclear Data Sheets for A = 242”. *Nuclear Data Sheets* 96.1 (2002), pp. 177–239. DOI: 10.1006/ndsh.2002.0011.
- [Ako03] Y. A. Akovali. “Nuclear Data Sheets for A = 244”. *Nuclear Data Sheets* 99.1 (2003), pp. 197–273. DOI: 10.1006/ndsh.2003.0008.
- [All+06] J. Allison et al. “Geant4 developments and applications”. *IEEE Trans. Nucl. Sci.* 53.1 (Feb. 2006), pp. 270–278. DOI: 10.1109/tns.2006.869826.
- [All+13] K. Allen et al. “UK-Norway Initiative (UKNI) approach for the development of a Gamma Ray Attribute Measurement System with an integrated Information Barrier”. In: *35th ESARDA Symposium proceedings, Bruges, 27-30 May 2013*. Ed. by F. Sevini. 2013.
- [Arg+16] Argentina, Brazil, Costa Rica, Ecuador, Guatemala, Indonesia, Malaysia, Mexico, and Zambia. *Addressing nuclear disarmament: Recommendations from the perspective of nuclear-weapon-free zones*. Working Paper A/AC.286/WP.34. United Nations General Assembly - Open-ended Working Group taking forward multilateral nuclear disarmament negotiation, 2016.
- [Aud+12] G. Audi, M. Wang, A. Wapstra, F. Kondev, M. MacCormick, X. Xu, and B. Pfeiffer. “The AME2012 atomic mass evaluation”. *Chinese physics C* 36.12 (2012), p. 1287. URL: <https://www-nds.iaea.org/amdc/>.
- [Bas06] M. Basunia. “Nuclear Data Sheets for A = 237”. *Nuclear Data Sheets* 107.8 (2006), pp. 2323–2422. DOI: 10.1016/j.nds.2006.07.001.
- [BB73] J. K. Bair and H. M. Butler. “Neutron yield from a small high purity 238 PuO₂ source”. *Nuclear Technology* 19.3 (1973), pp. 202–203.
- [BC79] J. K. Bair and J. G. del Campo. “Neutron Yields from Alpha-Particle Bombardment”. *Nuclear Science and Engineering* 71 (1979), pp. 18–28.
- [Ber+05] M. J. Berger, J. S. Coursey, M. A. Zucker, and J. Chang. *Stopping-Power and Range Tables for Electrons, Protons, and Helium Ions*. 2005. URL: <http://www.nist.gov/pml/data/star/index.cfm>.

-
- [Boe85] K. Boehnel. “The effect of multiplication on the quantitative determination of spontaneously fissioning isotopes by neutron correlation analysis”. *Nucl. Sci. Eng* 90 (1985), pp. 75–82.
- [Bou+01] S. C. Bourret, W. C. Harker, P. A. Hypes, D. C. Langner, D. R. Mayo, S. D. Salazar, R. Siebelist, M. K. Smith, and M. R. Sweet. “Fissile Material Transparency Technology Demonstration Neutron Multiplicity Counting System”. In: *43th Annual Meeting of the Institute of Nuclear Materials Management (INMM), Indian Wells, CA*. 2001.
- [BR03] A. Bonaccorsi and C. Rossi. “Why Open Source software can succeed”. *Research Policy* 32.7 (July 2003), pp. 1243–1258. DOI: 10.1016/S0048-7333(03)00051-9.
- [Bra13] D. C. Brabham. *Crowdsourcing*. Cambridge, Massachusetts: MIT Press, 2013.
- [Bro75] F. P. Brooks. *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley, 1975.
- [BT06] E. Browne and J. Tuli. “Nuclear Data Sheets for A = 236”. *Nuclear Data Sheets* 107.10 (2006), pp. 2649–2714. DOI: 10.1016/j.nds.2006.09.002.
- [BT07] E. Browne and J. Tuli. “Nuclear Data Sheets for A = 234”. *Nuclear Data Sheets* 108.3 (2007), pp. 681–772. DOI: 10.1016/j.nds.2007.02.003.
- [BT14a] E. Browne and J. Tuli. “Nuclear Data Sheets for A = 235”. *Nuclear Data Sheets* 122 (2014), pp. 205–292. DOI: 10.1016/j.nds.2014.11.002.
- [BT14b] E. Browne and J. Tuli. “Nuclear Data Sheets for A = 239”. *Nuclear Data Sheets* 122 (2014), pp. 293–376. DOI: 10.1016/j.nds.2014.11.003.
- [BT15] E. Browne and J. Tuli. “Nuclear Data Sheets for A = 238”. *Nuclear Data Sheets* 127 (2015), pp. 191–332. DOI: 10.1016/j.nds.2015.07.003.
- [Cha+09] P. M. J. Chard, S. Croft, M. Looman, P. Peerani, H. Tagziria, M. Brugge-
man, and A. Laure-Weber. “A Good Practice Guide for the use of Modelling
Codes in Non Destructive Assay of Nuclear Materials”. *ESARDA Bulletin* 42
(2009), pp. 26–69.
- [Cha+10] D. M. Chambers et al. “UK-Norway Initiative: Research into Information
Barriers to allow warhead attribute verification without release of sensi-
tive or proliferative information.” In: *51st Annual Meeting of the Institute
of Nuclear Materials Management (INMM), Baltimore, MD*. Institute of Nu-
clear Materials Management, 2010.
- [Cha+11] M. Chadwick et al. “ENDF/B-VII.1 Nuclear Data for Science and Technol-
ogy: Cross Sections, Covariances, Fission Product Yields and Decay Data”.
Nuclear Data Sheets 112.12 (2011). Special Issue on ENDF/B-VII.1 Library,
pp. 2887–2996. DOI: 10.1016/j.nds.2011.11.002.

-
- [CHH12] S. Croft, D. Henzlova, and D. Hauck. “Extraction of correlated count rates using various gate generation techniques: Part I theory”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 691 (Nov. 2012), pp. 152–158. DOI: 10.1016/j.nima.2012.06.011.
- [Chy+06] R. Chytrcek, J. McCormick, W. Pokorski, and G. Santin. “Geometry description markup language for physics simulation and analysis applications”. *Nuclear Science, IEEE Transactions on* 53.5 (2006), pp. 2892–2896.
- [CM13] S. Croft and R. D. McElroy. *The thick-target integrated-over-angle (α , n) yield curve for U3O8 over the energy range from 1.5 to 10 MeV and associated specific (α , n) yields of the uranium isotopes*. Tech. rep. Oak Ridge National Laboratory (ORNL), 2013.
- [Com10] Comprehensive Test Ban Treaty Organisation. *CTBTO’s data processing capabilities increase with overhaul of computer systems and migration to Linux*. 2010. URL: <https://www.ctbto.org/press-centre/highlights/2010/ctbtos-data-processing-capabilitiesincrease-with-overhaul-of-computersystems-and-migration-to-linux/> (visited on 11/25/2015).
- [Cou16] Council of the European Union. *Joint Comprehensive Plan of Action and restrictive measures*. 2016. URL: <http://www.consilium.europa.eu/en/policies/sanctions/iran/jcpoa-restrictive-measures/> (visited on 06/05/2016).
- [Cox14] M. J. Cox. *Heartbleed Timeline*. 2014. URL: <https://plus.google.com/+MarkJCox/posts/TmCbp3BhJma> (visited on 06/05/2016).
- [Cro+12] S. Croft, L. Evans, A. Favalli, D. Hauck, D. Henzlova, P. Santi, and M. Swinhoe. “A new alternative singles rate dead time correction algorithm for use in neutron multiplicity analysis”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 681 (July 2012), pp. 16–20. DOI: 10.1016/j.nima.2012.04.028.
- [Dah13] O. Dahlman. “How Can Science Support a Process Towards a World Free of Nuclear Weapons?” *Science & Global Security* 21.2 (2013), pp. 95–105. DOI: 10.1080/08929882.2013.798980.
- [Ein46] A. Einstein. *Letter from the Emergency Committee of Atomic Scientists, December 11*. 1946. URL: <http://www.einsteinsworld.com/images/Einstein-TSL-1946-Dec-11-ECAS-Appeal-Rubber-Stamp/Einstein-TSL-1946-ECAS-Appeal-Rubber-Stamp-1024px.jpg> (visited on 05/17/2016).
- [Eng09] M. Englert. “Neutronenphysikalische Simulationsrechnungen zur Proliferationsresistenz nuklearer Technologien”. PhD thesis. Technische Universität Darmstadt, 2009.

-
- [Ens+07a] N. Ensslin, W. H. Geist, M. S. Krick, and M. M. Pickrell. 7. *Active Neutron Multiplicity Counting*. LA-UR-07-1403. Los Alamos National Labs, 2007.
- [Ens+07b] N. Ensslin, M. S. Krick, D. G. Langner, M. M. Pickrell, T. D. Reilly, and J. E. Stewart. 6. *Passive Neutron Multiplicity Counting*. LA-UR-07-1402. Los Alamos National Laboratories, 2007.
- [Ens+98] N. Ensslin, W. C. Harker, M. S. Krick, D. G. Langner, M. M. Pickrell, and J. E. Stewart. *Application Guide to Neutron Multiplicity Counting*. LA-13422-M. Los Alamos National Laboratories, 1998.
- [Fei+14] H. A. Feiveson, A. Glaser, Z. Mian, and F. N. von Hippel. *Unmaking the Bomb: A Fissile Material Approach to Nuclear Disarmament and Nonproliferation*. Cambridge: MIT Press, 2014.
- [Fel+05] J. Feller, B. Fitzgerald, S. A. Hissam, and K. R. Lakhani, eds. *Perspectives on free and open source software*. Cambridge: MIT Press, 2005.
- [Fet+90] S. Fetter, V. A. Frolov, M. Miller, R. Mozley, O. F. Prilutsky, S. N. Rodionov, and R. Z. Sagdeev. "Detecting nuclear warheads". *Science & Global Security* 1.3-4 (1990), pp. 225–253. DOI: 10.1080/08929889008426333.
- [FF01] J. Feller and B. Fitzgerald. *Understanding Open Source Software Development*. Addison-Wesley Professional, 2001.
- [FGL02] M. G. Fry, E. Goldstein, and R. Langhorne, eds. *Guide to International Relations and Diplomacy*. Continuum, 2002.
- [Fod71] A. H. Foderaro. *The elements of neutron interaction theory*. Cambridge: MIT Press, 1971.
- [FOK68] Y. Feige, B. G. Oltman, and J. Kastner. "Production rates of neutrons in soils due to natural radioactivity". *Journal of Geophysical Research* 73.10 (1968), pp. 3135–3142. DOI: 10.1029/JB073i010p03135.
- [For15] Forbes. *The World's Biggest Public Companies*. 2015. URL: <http://www.forbes.com/global2000/list/#industry:Software%20&%20Programming> (visited on 05/21/2016).
- [FP13] F. Fernandez-Alonso and D. L. Price, eds. *Neutron Scattering - Fundamentals*. Vol. 44. Experimental Methods in the Physical Sciences. Academic Press, 2013.
- [Fra+45] J. Franck, D. Hughes, L. Szilard, T. Hogness, E. Rabinowitch, G. Seaborg, and C. Nickson. *The "Franck Report" - A Report To The Secretary Of War, June 1945*. 1945. URL: <http://www.fas.org/sgp/eprint/franck.html> (visited on 05/19/2016).
- [Ful99] J. Fuller. *The functional Requirements and Design Basis for Information Barriers*. PNNL-13285. Pacific Northwestern National Laboratory, 1999.
- [Gay86] N. Gayler. "Verification, Compliance and the Intelligence Process". In: *Arms Control Verification - The Technologies That Make It Possible*. Ed. by K. Tsipis, D. W. Hafemeister, and P. Janeway. New York: Pergamon Press, 1986.

-
- [GC13] B. Goddard and S. Croft. “High-fidelity passive neutron multiplicity measurements and simulations of uranium oxide”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 712 (June 2013), pp. 147–156. DOI: 10.1016/j.nima.2013.02.007.
- [GDML] *Geometry Description Markup Language (GDML)*. 2016. URL: <http://gdml.web.cern.ch/GDML/> (visited on 02/20/2016).
- [Gea15a] Geant4 Collaboration. *Geant4 User’s Guide for Application Developers - Version: geant4 10.2 (4 December 2015)*. 2015. URL: <http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/fo/BookForAppliDev.pdf> (visited on 05/27/2016).
- [Gea15b] Geant4 Collaboration. *Physics Reference Manual - Version: geant4 10.2 (4 December 2015)*. 2015. URL: <http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/PhysicsReferenceManual/fo/PhysicsReferenceManual.pdf> (visited on 05/27/2016).
- [GK14] M. Göttsche and G. Kirchner. “Measurement Techniques for Warhead Authentication with Attributes: Advantages and Limitations”. *Science & Global Security* 22.2 (2014). DOI: 10.1080/08929882.2014.918805.
- [GK15] M. Göttsche and G. Kirchner. “Improving neutron multiplicity counting for the spatial dependence of multiplication: Results for spherical plutonium samples”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 798 (Oct. 2015), pp. 99–106. DOI: 10.1016/j.nima.2015.07.007.
- [Goo+13] J. T. Goorley et al. *Initial MCNP Release Overview - MCNP 6 Version 1.0*. Tech. rep. LA-UR-13-22934. Los Alamos National Lab, 2013.
- [Göt15] M. Göttsche. “Reducing Neutron Multiplicity Counting Bias for Plutonium Warhead Authentication”. PhD thesis. Universität Hamburg, 2015.
- [Gua93] *ESARDA International Workshop On Passive Neutron Coincidence Counting*. CL-NA-15102-EN-C. European Communities - Commission, 1993.
- [Hay+10] T. Hayakawa, N. Kikuzawa, R. Hajima, T. Shizuma, N. Nishimori, M. Fujiwara, and M. Seya. “Nondestructive assay of plutonium and minor actinide in spent fuel using nuclear resonance fluorescence with laser Compton scattering”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 621.1–3 (2010), pp. 695–700. DOI: 10.1016/j.nima.2010.06.096.
- [HE98] M. A. Heller and R. S. Eisenberg. “Can patents deter innovation? The anticommons in biomedical research”. *Science* 280.5364 (1998), pp. 698–701.

-
- [Hea+89] R. Heaton, H. Lee, P. Skensved, and B. Robertson. “Neutron production from thick-target (α , n) reactions”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 276.3 (1989), pp. 529–538. DOI: 10.1016/0168-9002(89)90579-2.
- [Hen+12] D. Henzlova, S. Croft, H. Menlove, and M. Swinhoe. “Extraction of correlated count rates using various gate generation techniques Part II Experiment”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 691 (Nov. 2012), pp. 152–158. DOI: 10.1016/j.nima.2012.04.091.
- [HH12] C. Hinderstein and K. Hartigan. “The Opportunities and Limits of Societal Verification”. In: *54th Annual Meeting of the Institute of Nuclear Materials Management*. Institute of Nuclear Materials Management. Palm Desert, July 18, 2012.
- [HK03] E. von Hippel and G. von Krogh. “Open-Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science”. *Organization Science* 14.2 (März 2003), pp. 208–223.
- [Hři+11] I. Hřivnáčová, O. Datskova, A. Gheata, A. Morsch, E. Sicking, and the ALICE Collaboration. “The ALICE Geant4 Simulation”. *Journal of Physics: Conference Series* 331.3 (2011), p. 032016.
- [HT09] M. Herman and A. Trkov. *ENDF-6 Formats Manual - Data Formats and Procedures for the Evaluated Nuclear Data File ENDF/B-VI and ENDF/B-VII*. Report BNL-90365-2009. Brookhaven National Laboratory, 2009.
- [HWG11] K. D. Huff, P. P. Wilson, and M. j. Gidden. “Open Architecture and Modular Paradigm of Cyclus”. In: *American Nuclear Society Annual Conference*. 2011.
- [ICR93] ICRU. *Stopping Powers and Ranges for Protons and Alpha Particles*. ICRU Report 49. International Commission on Radiation Units and Measurements, 1993.
- [III07] International Physicians for the Prevention of Nuclear War, International Association of Lawyers Against Nuclear Arms, and International Network of Engineers and Scientists Against Proliferation. *Securing Our Survival - The Case for a Nuclear Weapons Convention*. 2007. URL: <http://icanw.org/files/sos.pdf> (visited on 05/20/2012).
- [Int97] International Atomic Energy Agency. *Press Statement on the Trilateral Initiative*. 1997. URL: <https://www.iaea.org/newscenter/pressreleases/press-statement-trilateral-initiative> (visited on 05/10/2016).
- [Jon+13] G. Jonkmans, V. Anghel, C. Jewett, and M. Thompson. “Nuclear waste imaging and spent fuel verification by muon tomography”. *Annals of Nuclear Energy* 53 (2013), pp. 267–273. DOI: 10.1016/j.anucene.2012.09.011.

-
- [KBM09] M. Kellett, O. Bersillon, and R. Mills. *The JEFF-3.1/-3.1.1 radioactive decay data and fission yields sub-libraries*. JEFF Report 20. OECD, 2009.
- [Ker83] A. Kerckhoffs. “La cryptographie militaire”. *Journal des sciences militaires* IX (1883), pp. 5–38, 161–191.
- [KGE14] M. Kütt, A. Glaser, and M. Englert. “Open Source meets Nuclear Arms Control”. In: *55th Annual Meeting of the Institute for Nuclear Materials Management, Atlanta, GA*. 2014.
- [KGG16] M. Kütt, M. Götttsche, and A. Glaser. “Disarmament Hacking 2.0: Toward a Trusted, Open-Hardware Computing Platform for Nuclear Warhead Verification”. In: *57th Annual Meeting of the Institute for Nuclear Materials Management, Atlanta, GA*. 2016.
- [KN16] H. M. Kristensen and R. S. Norris. *Status of World Nuclear Forces*. 2016. URL: <http://fas.org/issues/nuclear-weapons/status-world-nuclear-forces/> (visited on 05/03/2016).
- [KPG15] M. Kütt, S. Philippe, and A. Glaser. “Leveraging the Wisdom of the Crowd: Hardware and Software Challenges for Nuclear Disarmament Verification”. In: *56th Annual Meeting of the Institute for Nuclear Materials Management, Indian Wells, CA*. 2015.
- [KS84] M. Krick and J. Swansen. “Neutron multiplicity and multiplication measurements”. *Nuclear Instruments and Methods in Physics Research* 219.2 (1984), pp. 384–393. DOI: 10.1016/0167-5087(84)90349-1.
- [Küt11] M. Kütt. “Neutronic Calculations: Proliferation risks of Fast Reactors”. Master Thesis. Technische Universität Darmstadt, 2011.
- [Lan14] I. Landwerth. *.NET Core is Open Source*. 2014. URL: <https://blogs.msdn.microsoft.com/dotnet/2014/11/12/net-core-is-open-source/> (visited on 01/30/2016).
- [Lee+08] T.-H. Lee, H. O. Menlove, S.-Y. Lee, and H.-D. Kim. “Development of the ACP safeguards neutron counter for PWR spent fuel rods”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 589.1 (Apr. 2008), pp. 57–65. DOI: 10.1016/j.nima.2008.02.054.
- [Lev75] S. Levy. *Hackers: Heroes of the Computer Revolution*. O’Reilly Media, 1975.
- [LFL97] D. Langner, J. Franco, and R. Larsen. *Assay of impure plutonium oxide with the large neutron multiplicity counter for IAEA verification of excess weapons material at the Rocky Flats Environmental Technology Site*. LA-UR-97-2650. Los Alamos National Laboratory, 1997.
- [Lil+46] D. E. Lilienthal, C. I. Barnard, C. A. Thomas, J. R. Oppenheimer, and H. A. Winne. *The Acheson-Lilienthal Report: Report on the International Control of Atomic Energy*. 1946.

-
- [Lin14] Linux Foundation. *Amazon Web Services, Cisco, Dell, Facebook, Fujitsu, Google, IBM, Intel, Microsoft, NetApp, Rackspace, VMware and The Linux Foundation Form New Initiative to Support Critical Open Source Projects*. Apr. 24, 2014. URL: <http://www.linuxfoundation.org/news-media/announcements/2014/04/amazon-web-services-cisco-dell-facebook-fujitsu-google-ibm-intel> (visited on 05/28/2016).
- [LKE90] D. Langner, M. Krick, and N. Ensslin. “Pyrochemical multiplicity counter design”. In: *Annual Meeting of the Institute for Nuclear Materials Management (INMM)*. 1990, pp. 411–415.
- [LLNLFIS] Lawrence Livermore National Laboratory. *Physics Simulation Packages - Fission*. URL: http://nuclear.llnl.gov/simulation/fission_v1.9.1/fission_v1.9.1.tar.gz (visited on 02/20/2016).
- [LP77] H. Liskien and A. Paulsen. “Neutron yields of light elements under α -bombardment”. *Atomkernenergie* 30 (1977), pp. 59–61.
- [LPT09] M. Looman, P. Peerani, and H. Tagziria. “Monte Carlo simulation of neutron counters for safeguards applications”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 598.2 (Jan. 2009), pp. 542–550. DOI: 10.1016/j.nima.2008.09.040.
- [MBH11] M. McDaniel, D. Bleakly, and K. Horak. “Harvesting Geospatial Safeguards Information With Open Source Tools”. In: *51nd Annual Meeting of the Institute of Nuclear Materials Management (INMM)*. Institute of Nuclear Materials Management, Palm Desert, 2011, pp. 2179–2185.
- [Men+12] E. Mendoza, D. Cano-Ott, C. Guerrero, and R. Capote. *IAEA technical report INDC(NDS)-0612, Vienna*. 2012. URL: <http://www-nds.iaea.org/geant4>. (visited on 02/20/2016).
- [Men+14] E. Mendoza, D. Cano-Ott, T. Koi, and C. Guerrero. “New Standard Evaluated Neutron Cross Section Libraries for the GEANT4 Code and First Verification”. *IEEE Trans. Nucl. Sci.* 61.4 (Aug. 2014), pp. 2357–2364. DOI: 10.1109/tns.2014.2335538.
- [Men+93] H. O. Menlove, J. Baca, M. Krick, K. Kroncke, and D. Langner. *Plutonium scrap multiplicity counter operation manual*. LA-12479-M. Los Alamos National Laboratory, 1993.
- [Mil+11] E. Miller, B. Dennis, S. Clarke, S. Pozzi, and J. Mattingly. “Simulation of polyethylene-moderated plutonium neutron multiplicity measurements”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 652.1 (2011). Symposium on Radiation Measurements and Applications (SORMA) {XII} 2010, pp. 540–543. DOI: 10.1016/j.nima.2011.01.042.
- [Mil12] E. Miller. “Characterization of Fissionable Material using a Time-Correlated Pulse-Height Technique for Liquid Scintillators”. PhD thesis. University of Michigan, 2012.

-
- [MM94] H. O. Menlove and N. Miura. *The Use of Curium Neutrons to verify Plutonium in Spent Fuel and Reprocessing Wastes*. LA-12774-MS. Los Alamos National Laboratory, 1994.
- [MMS06] T. Murata, H. Matsunobu, and K. Shibata. *Evaluation of the (α , xn) Reaction Data for JENDL/AN-2005*. JAEA-Research 2006-052. Japan Atomic Energy Agency, 2006.
- [MS02] T. Murata and K. Shibata. “Evaluation of the (α , n) Reaction Nuclear Data for Light Nuclei”. *Journal of Nuclear Science and Technology, Supplement 2* (2002).
- [MS10] S. E. Miller and S. D. Sagan. “Alternative nuclear futures”. *Daedalus* (Winter 2010).
- [MY02] H. Matsunobu and N. Yamamuro. “Evaluation of the (α , n) Reaction for F, Na, Al, Cr, Fe, Ni, and Cu”. *Journal of Nuclear Science and Technology, Supplement 2* (2002).
- [Nak+10] H. Nakamura, D. Beddingfield, H. Nakamichia, Y. Mukaia, and K. Yoshimoto. *Feasibility Study of Neutron Multiplicity Assay for a Heterogeneous Sludge Sample containing Na, Pu and other Impurities*. IAEA-CN-184. International Atomic Energy Agency, 2010.
- [Nak+14] S. Nakajima et al. “Performance test results for the Advanced Fuel Assembly Assay System (AFAS) on the active length verification of LWR MOX fuel assembly by neutron detectors”. In: *55th Annual Meeting of the Institute for Nuclear Materials Management, Atlanta, GA*. 2014.
- [NCP13] S. Naeem, S. Clarke, and S. Pozzi. “Validation of Geant4 and MCNPX-PoliMi simulations of fast neutron detection with the EJ-309 liquid scintillator”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 714 (June 2013), pp. 98–104. DOI: 10.1016/j.nima.2013.02.017.
- [Nes15] C. Nesaraja. “Nuclear Data Sheets for A = 241”. *Nuclear Data Sheets* 130 (2015), pp. 183–252. DOI: 10.1016/j.nds.2015.11.004.
- [Nic05] N. Nica. “Nuclear Data Sheets for A = 252”. *Nuclear Data Sheets* 106.2 (2005), pp. 813–834. DOI: 10.1016/j.nds.2005.11.003.
- [NK11] R. S. Norris and H. M. Kristensen. “US tactical nuclear weapons in Europe, 2011”. *Bulletin of the Atomic Scientists* 67.1 (Jan. 2011), pp. 64–73. DOI: 10.1177/0096340210393931.
- [NNDC] National Nuclear Data Center. *Evaluated Nuclear Structure Data File (ENSDF) Search and Retrieval*. Brookhaven National Laboratory. URL: <http://www.nndc.bnl.gov/ensdf/> (visited on 03/16/2016).
- [NNDCa] National Nuclear Data Center. *ENDF B-VII.1 - Decay Reaction Sublibrary*. Oak Ridge National Laboratory. URL: <http://www.nndc.bnl.gov/ensdf/b7.1/zips/ENDF-B-VII.1-decay.zip>.

-
- [OSI] Open Source Initiative. *The Open Source Initiative*. URL: <http://opensource.org/> (visited on 05/30/2016).
- [Pag16] J. Pagliery. *Your Internet security relies on a few volunteers*. Apr. 18, 2016. URL: <http://money.cnn.com/2014/04/18/technology/security/heartbleed-volunteers/> (visited on 05/30/2016).
- [Pee+09] P. Peerani, M. Swinhoe, A. L. Weber, and L. Evans. "ESARDA Multiplicity Benchmark Exercise - Phases III and IV". *ESARDA Bulletin* 42 (2009), pp. 2–25.
- [Pis06] C. Pistner. "Neutronenphysikalische Untersuchungen zu uranfreien Brennstoffen". PhD thesis. Technische Universität Darmstadt, 2006.
- [Poz+12] S. Pozzi et al. "MCNPX-PoliMi for nuclear nonproliferation applications". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 694 (2012), pp. 119–125. DOI: 10.1016/j.nima.2012.07.040.
- [PPM03] S. A. Pozzi, E. Padovani, and M. Marseguerra. "MCNP-PoliMi: a Monte-Carlo code for correlation measurements". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 513.3 (2003), pp. 550–558.
- [PS06] P. Peerani and M. Swinhoe. "ESARDA Multiplicity Benchmark Exercise - Final Report". *ESARDA Bulletin* 34 (2006), pp. 2–32.
- [PVE07] M. M. Pickrell, K. Veal, and N. Ensslin. *8. Fast and Epithermal Neutron Multiplicity Counters*. LA-UR-07-1602. Los Alamos National Labs, 2007.
- [PW12] P. Peerani and A.-L. Weber. "Analysis of uncertainties affecting the Monte Carlo simulation of a neutron multiplicity counter". *Radiation Measurements* 47.7 (2012), pp. 475–480. DOI: 10.1016/j.radmeas.2012.05.010.
- [PW81] R. T. Perry and W. B. Wilson. *Neutron Production from (α,n) Reactions and Spontaneous Fission in ThO₂, UO₂, and (U,Pu)O₂ Fuels*. Los Alamos National Laboratory, 1981.
- [Ray00] E. S. Raymond. *The Cathedral and the Bazaar*. 2000. URL: <http://www.catb.org/esr/writings/homesteading/cathedral-bazaar/cathedral-bazaar.ps> (visited on 09/26/2013).
- [Raz+10] S. Razinkov, M. Bulatov, S. Kondratov, A. Livke, M. Leplyavkina, D. MacArthur, D. Sivachev, J. Thron, S. Tsybryaev, and A. Vayushin. "AVNG System Objectives and Concept". In: *51st Annual Meeting of the Institute of Nuclear Materials Management (INMM)*, Baltimore, MD. 2010, pp. 11–15.
- [RE08] D. Reilly and N. Ensslin. "Nuclear Material Measurement Technologies". In: J. Doyle. *Nuclear Safeguards, Security and Nonproliferation: Achieving Security with Technology and Policy*. Ed. by J. E. Doyle. Elsevier, 2008, pp. 31–62.
- [Rei+91] D. Reilly, N. Ensslin, H. S. Jr., and S. Kreiner. *Passive Nondestructive Assay of Nuclear Materials*. US Nuclear Regulatory Commission, 1991.

-
- [Rei07] D. Reilly, ed. *Passive Nondestructive Assay of Nuclear Materials - 2007 Addendum*. 2007.
- [RF13] P. K. Romano and B. Forget. “The OpenMC Monte Carlo particle transport code”. *Annals of Nuclear Energy* 51 (Jan. 2013), pp. 274–281. DOI: 10.1016/j.anucene.2012.06.040.
- [Rim+04] A. Rimoldi, A. Dell’Acqua, M. Gallas, A. Nairz, J. Boudreau, V. Tsulaia, and D. Costanzo. “The simulation for the ATLAS experiment. Present status and outlook”. In: *Nuclear Science Symposium Conference Record, 2004 IEEE*. Vol. 3. Oct. 2004, pp. 1886–18903. DOI: 10.1109/NSSMIC.2004.1462613.
- [Rot+93] J. Rotblat, J. Steinberger, B. Udgaonkar, E. *, and U. Bhalchandra. *A Nuclear-weapon-free World: Desirable? Feasible? (Pugwash Monograph)*. Westview Press, 1993.
- [Rus60] B. Russell. “The Social Responsibilities of Scientists”. *Science* 131.3398 (Feb. 1960), pp. 391–392. DOI: 10.1126/science.131.3398.391.
- [San+09] A. Santamarina et al. *The JEFF-3.1.1 Nuclear Data Library*. JEFF Report 22. OECD, 2009.
- [SB08] B. Singh and E. Browne. “Nuclear Data Sheets for A=240”. *Nuclear Data Sheets* 109 (2008), pp. 2439–2499. DOI: 10.1016/j.nds.2008.09.002.
- [Sch14] B. Schneier. *The Insecurity of Secret IT Systems*. 2014. URL: https://www.schneier.com/blog/archives/2014/02/the_insecurity_2.html (visited on 06/05/2016).
- [Sch99] B. Schneier. *Schneier on Security - Crypto-Gram Newsletter*. 1999. URL: <https://www.schneier.com/crypto-gram-9909.html> (visited on 05/27/2014).
- [Sco+12] A. Scopatz, P. K. Romano, P. P. Wilson, and K. D. Huff. “PyNE: Python for Nuclear Engineering”. In: *Am. Nuc. Soc. Winter Meeting 2012*. Vol. 107. San Diego, CA, USA, Nov. 2012.
- [SF02] J. K. Shultis and R. E. Faw. *Fundamentals of Nuclear Science and Engineering*. CRC Press, 2002.
- [She+05] J. Shergur, D. Bracken, L. Carrillo, T. Elmont, Frame, Hirsch, P. Hypes, J. Kuzminski, R. Landry, D. MacArthur, et al. “An Overview of the Design of a Next Generation Attribute Measurement System”. In: *46th Annual Meeting of the Institute of Nuclear Materials Management (INMM), Phoenix, AZ*. 2005.
- [SLG02] R. M. Stallman, L. Lessig, and J. Gay. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Free Software Foundation, 2002.
- [SM08] P. Santi and M. Miller. “Reevaluation of Prompt Neutron Emission Multiplicity Distributions for Spontaneous Fission”. *Nuclear science and engineering* 160.2 (2008), pp. 190–199.
- [Söd08] J. Söderberg. *Hacking capitalism: The FOSS software movement*. New York: Routledge, 2008.

-
- [Sta10] R. Stallman. *Free Software Free Society, 2nd Edition*. Free Software Foundation, Inc., 2010.
- [Str13] B. Stroustrup. *The C++ Programming Language*. Addison Wesley, May 9, 2013.
- [Sur05] J. Surowiecki. *The Wisdom of Crowds*. London: Abacus, 2005.
- [Tam+10] T. Tamura, M. Komatsu, T. Iwamoto, M. Swinhoe, C. Real, J. Marlow, and H. Menlove. “A Study of an Epithermal Neutron Multiplicity Counter (ENMC) for Nuclear Material Accountancy”. In: *51st Annual Meeting of the Institute of Nuclear Materials Management (INMM), Baltimore, MD*. 2010.
- [TL12] H. Tagziria and M. Looman. “The ideal neutron energy spectrum of $^{241}\text{AmLi}(\alpha, n)^{10}\text{B}$ sources”. *Applied Radiation and Isotopes* 70.10 (2012), pp. 2395–2402. DOI: 10.1016/j.apradiso.2012.07.008.
- [UKN15] UK-Norway Initiative. *Information Barrier*. 2015. URL: <http://ukni.info/project/information-barrier/> (visited on 11/25/2015).
- [UNG15] United Nations General Assembly. *Humanitarian pledge for the prohibition and elimination of nuclear weapons*. Resolution A/RES/70/48. Dec. 7, 2015.
- [UNG46] United Nations General Assembly. *Establishment of a Commission to Deal with the Problem Raised by the Discovery of Atomic Energy*. Resolution A/RES/1 (I). 1946.
- [Uni11] United States General Accounting Office. *Neutron detectors - Alternatives to using helium-3*. Technical Assessment GAO-11-753. United States General Accounting Office, 2011.
- [UNO12] United Nations Office for Disarmament Affairs. *Status of the Treaty on the Non-Proliferation of Nuclear Weapons (NPT)*. 2012. URL: <http://disarmament.un.org/treaties/t/npt> (visited on 09/05/2012).
- [UNO15] United Nations Office for Disarmament Affairs. *2015 Review Conference of the Parties to the Treaty on the Non-Proliferation of Nuclear Weapons (NPT)*. 2015. URL: <http://www.un.org/en/conf/npt/2015/> (visited on 12/10/2015).
- [VHW10] J. M. Verbeke, C. Hagmann, and D. Wright. *Simulation of Neutron and Gamma Ray Emission from Fission and Photofission*. UCRL-AR-228518. Lawrence Livermore National Laboratory, 2010. URL: <http://nuclear.llnl.gov/simulation/main.html> (visited on 02/20/2016).
- [VW07] O. Ventä and B. Wahlström. “Investigating the case of Open Source applications within nuclear power”. In: *Enlarged Halden Programme Group meeting, Storefjell, Norway*. 2007, pp. 12–15.
- [Wan+12] M. Wang, G. Audi, A. Wapstra, F. Kondev, M. MacCormick, X. Xu, and B. Pfeiffer. “The AME2012 atomic mass evaluation”. *Chinese Phys. C* 36.12 (Dec. 2012), pp. 1603–2014. DOI: 10.1088/1674-1137/36/12/003. URL: <https://www-nds.iaea.org/amdc/>.

-
- [War+12] G. A. Warren, D. E. Archer, M. Cunningham, S. McConchie, and J. Thron. “Concepts for the Measurements Subsystems of the Third Generation Attributes Measurement System”. In: *53rd Annual Meeting of the Institute for Nuclear Materials Management (INMM)*, Orlando, FL, 2012.
- [Web04] S. Weber. *The Success of Open Source*. Harvard University Press, 2004.
- [Wes79] D. West. “The calculation of neutron yields in mixtures and compounds from the thick target (α,n) yields in the separate constituents”. *Annals of Nuclear Energy* 6.11 (1979), pp. 549–552. DOI: 10.1016/0306-4549(79)90003-3.
- [WFK12] T. M. Willig, C. Futsaether, and H. Kippe. “Converting the Iranian Heavy Water Reactor IR-40 to a More Proliferation-Resistant Reactor”. *Science & Global Security* 20.2-3 (May 2012), pp. 97–116. DOI: 10.1080/08929882.2012.713767.
- [Whi01] G. White. “Increasing Inspectability of Hardware and Software for Arms Control and Nonproliferation Regimes”. In: *43th Annual Meeting of the Institute of Nuclear Materials Management (INMM)*, Indian Wells, CA. 2001.
- [Whi12] G. White. “Review of Prior U.S. Attribute Measurement Systems”. In: *53rd Annual Meeting of the Institute of Nuclear Materials Management*. 2012.
- [Wil+02] W. B. Wilson et al. *SOURCES 4C: a code for calculating (α,n), spontaneous fission, and delayed neutron sources and spectra*. LA-UR-02-1839. Los Alamos National Laboratory, Jan. 2002.
- [Wit09] L. S. Wittner. *Confronting the Bomb: A Short History of the World Nuclear Disarmament Movement*. Palo Alto: Stanford University Press, 2009.
- [Wol+01] J. J. Wolford, B. Geelhood, V. Hamilton, J. Ingraham, D. MacArthur, D. Mitchell, J. Mullens, P. Vanier, G. White, and R. Whiteson. “Software Authentication”. In: *43th Annual Meeting of the Institute of Nuclear Materials Management (INMM)*, Indian Wells, CA. 2001.
- [WS82] D. West and A. Sherwood. “Measurements of thick-target (α, n) yields from light elements”. *Annals of Nuclear Energy* 9.11–12 (1982), pp. 551–577. DOI: 10.1016/0306-4549(82)90001-9.
- [WVV07] B. Wahlström, O. Ventä, and J. Valkonen. “Proving absence of CCFs; a case for Open Source”. In: *Technical Meeting on Common-Cause Failures in Digital Instrumentation and Control Systems of Nuclear Power Plants*, Bethesda, Maryland, USA. 2007.
- [ZBZ08] J. F. Ziegler, J. P. Biersack, and M. D. Ziegler. *SRIM, the Stopping and Range of Ions in Matter*. Chester, Maryland: SRIM, 2008.
- [Zie77] J. F. Ziegler. *Helium: stopping powers and ranges in all elemental matter*. Vol. 4. New York: Pergamon Press, 1977.



Lebenslauf

Persönliche Daten

Geburtsdatum 23. März 1985
Geburtsort Frankfurt / Main
Eltern Gerlinde Jilg-Kütt und Hermann Kütt

Bildungsweg

2011 - 2016 Promotionsstudium an der Technischen Universität Darmstadt, Interdisziplinäre Arbeitsgruppe Naturwissenschaft, Technik und Sicherheit (IANUS)
10/2015 - 06/2016 Visiting Research Student Collaborator, Program on
04/2014 - 07/2014 Science and Global Security, Princeton University
2004 - 2012 Bachelor Studium Politikwissenschaft, TU Darmstadt
2008 - 2011 Master Studium Physik, TU Darmstadt
08/2009 - 02/2010 ERASMUS-Studium, Ecole Polytechnique Fédéral de Lausanne, Schweiz
2004 - 2008 Bachelor Studium Physik, TU Darmstadt
Juni 2004 Abitur
2001 - 2004 Gewerblich-Technische Schulen, Offenbach/Main
1995 - 2001 Adolf-Reichwein Gymnasium, Heusenstamm
1993 - 1995 Grundschule, Obertshausen
1991 - 1993 Grundschule, Hergershausen



Erklärung zur Dissertation

Hiermit versichere ich, die vorliegende Dissertation ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 04. August 2016

(M. Kütt)