ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská
Katedra dozimetrie a aplikace ionizujícího záření

DIPLOMOVÁ PRÁCE

Development and validation of new features for the G4SEE radiation effect simulation toolkit

Autor:	Bc. Eva Fialová
Vedoucí:	MSc. Dávid Lucsányi
Akademický rok:	2023/2024

CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Fialová Eva** Personal ID number: **486350**

Faculty / Institute: **Faculty of Nuclear Sciences and Physical Engineering**

Department / Institute: **Department of Dosimetry and Application of Ionizing Radiation**

Study program: **Nuclear Engineering**

Specialisation: **Applied Physics of Ionizing Radiation**

II. Master’s thesis details

Master's thesis title in English:

Development and validation of new features for the G4SEE radiation effect simulation toolkit

Master's thesis title in Czech:

Vývoj a validace nových funkcí pro simulační nástroj radiačních efektů G4SEE

Guidelines:

1. Introduce the Single Event Effects in electronics, Monte Carlo simulation of SEEs, and the G4SEE toolkit.
2. Implement new event-by-event scoring features in the G4SEE toolkit.
3. Validate the simulated data and the newly implemented features against measured data from experimental test campaigns.
4. Support G4SEE users on how to use the new features (documentation, examples, user interfaces).
5. Summarize the work done and describe future work.

Bibliography / sources:

[1] LUCSÁNYI, Dávid, 2023, G4SEE. Available from: https://g4see.web.cern.ch/

[2] Radiation to Electronics project at CERN, 2023. R2E. Available from: https://r2e.web.cern.ch/

[3] LUCSANYI, David, Ruben Garcia ALIA, Kacper BILKO, Matteo CECCHETTO, Salvatore FIORE a Elisa PIROVANO. G4SEE: A Geant4-Based Single Event Effect Simulation Toolkit and Its Validation Through Monoenergetic Neutron Measurements. IEEE Transactions on Nuclear Science. 2022, 69(3), 273-281. ISSN 0018-9499. DOI: 10.1109/TNS.2022.3149989.

[4] Open-source GitLab repositories of the G4SEE toolkit, 2023. Available from: https://gitlab.cern.ch/g4see/

[5] FLORACZAK, Josua. TA04-57: Sensitivity analysis of COTS SMOS sensors for radiation detection. RADNEXT Transnational Access Summary Report, Zenodo. Available from: https://doi.org/10.5281/zenodo.7463656

[6] NGO, Khai. TA02-14: Neutron response of EJ-276 and 7CLYC scintillators for characterization of the new NILE facility. RADNEXT Transnational Access Summary Report, Zenodo. Available from: https://doi.org/10.5281/zenodo.6594913

Name and workplace of master’s thesis supervisor:

MSc. Dávid Lucsányi CERN, Systems Department; Sources, Targets, Interactions Group; Radiation to Electronics (R2E) project

Name and workplace of second master’s thesis supervisor or consultant:

RNDr. Jan Smolík, Ph.D. Department of Dosimetry and Application of Ionizing Radiation, FNSPE CTU in Prague

Date of master’s thesis assignment: **28.02.2023** Deadline for master's thesis submission: **08.01.2024**

Assignment valid until: ______________

MSc. Dávid Lucsányi
Supervisor’s signature

Head of department’s signature

doc. Ing. Václav Čuba, Ph.D.
Dean’s signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

6.3. 2023
Date of assignment receipt

Fialová
Student's signature

Prohlášení

Prohlašuji, že jsem svoji diplomovou práci vypracovala samostatně a použila jsem pouze podklady uvedené v přiloženém seznamu.

Fialová

V Praze 9.5.2024 ....

Bc. Eva Fialová

Acknowledgements

First of all, I would like to thank Msc. Dávid Lucsányi, my supervisor, for his professional guidance and willingness, that he gave me during the preparation of my master thesis. He always helped me solve any problem and provided valuable advice to help me advance my research.

Dále bych chtěla poděkovat RNDr. Janu Smolíkovi, Ph.D., který mě podpořil v rozhodnutí absolvavot stáž v CERNu a vždy mi byl nápomocen s jakýmkoliv problémem.

I would also like to acknowledge the team from ETH Zurich, in particular Corinna Martinella, for the discussions about SiC power MOSFETs, and Carlo Cazzaniga and Michela Paoletti, for the experimental data for the diamond detector.

I would also like to thank my colleagues from the SY-STI-BMI section, who always managed to turn a bad lunch into enjoyable time. I would especially like to thank Auriane, my office mate and climbing buddy.

Special thanks goes to Daniel. Thank you for always supporting me and making my stay at CERN an amazing experience, it would not have been the same without you.

Chtěla bych také srdečně poděkovat mým rodičům. Děkuji, že mě podporujete v mých snech a stojíte vždy při mě. Doufám, že vám to budu schopná jednoho dne oplatit.

Bc. Eva Fialová

Název práce: **Development and validation of new features for the G4SEE radiation effect simulation toolkit**

Autor: Bc. Eva Fialová

Obor: Dozimetrie a aplikace ionizujícího záření

Druh práce: Diplomová práce

Vedoucí práce: MSc. Dávid Lucsányi, CERN, SY-STI-BMI

Konzultant: RNDr. Jan Smolík, Ph.D., FJFI ČVUT v Praze, Katedra dozimetrie a aplikace ionizujícího záření

Abstrakt: Single Event Efekty představují riziko pro elektronická zařízení nejen v prostředí s vysokým radiačním zatížením (jako je například vesmír nebo vysokoenergetické urychlovače), ale také v prostředí, kde je radiační úroveň nízká. Monte Carlo simulační nástroje těchto efektů jsou zásadní součástí pro jejich pochopení. Takovým nástrojem je G4SEE, Geant4-based Single Event Effects simulation toolkit, vyvíjený v CERNu pro komunitu zabývající se radiačními efekty. Tento simulační nástroj je neustále vyvíjen a validován.

V této práci je nejprve uveden obecný přehled Monte Carlo simulací, Single Event Efektů a jejich simulací. Poté následuje podrobný popis simulačního nástroje G4SEE a jeho schopností. Dále jsou představeny nově implementované funkce, konkrétně skórování LET, TID a deponované energie pro jednotlivé typy interakcí. Nakonec jsou představeny validace G4SEE s neutrony a protony.

Klíčová slova: G4SEE, Monte Carlo Simulace, Single Event Efekty, Lineární přenos energie, Deponovaná energie

Title: **Development and validation of new features for the G4SEE radiation effect simulation toolkit**

Author: Bc. Eva Fialová

Abstract: Single Event Effects pose a risk for electronic devices not only in high-level radiation environments (such as space, or high-energy accelerators) but also in low radiation environment at ground-level. The Monte Carlo simulations of such effects are a crucial part of understanding them. Such tool is G4SEE, a Geant4-based Single Event Effects simulation toolkit, developed at CERN for the radiation effects community. It is constantly being developed and validated.

In this work firstly a general overview of Monte Carlo simulations, Single Event Effects and their simulations is given. This follows with detailed description of G4SEE and its capabilities. Afterwards newly implemented features, namely LET, TID and energy deposition per interaction scoring, are described. Lastly the validations of the toolkit with neutrons and protons are described.

Key words: G4SEE, Monte Carlo Simulations, Single Event Effects, Linear Energy Transfer, Energy Deposition

Contents

Chapter 1

Introduction

This thesis aims to develop new features and validate the Geant4-based Single Event Effect simulation toolkit (G4SEE) for simulating radiation effects in electronics. This work has been carried out in the context of the Radiation to Electronics (R2E) project at the European Organisation for Nuclear Research (CERN). In the following, a brief description of each chapter is provided.

Single Event Effects (SEE) pose a risk for electronic devices and systems. They can be critical not only in the high-level radiation environments (such as space, or high-energy particle accelerators), but also in low radiation environment at ground-level. Since the commercial off-the-shelf (COTS) devices are not specifically designed or tested for high radiation levels, it is important to focus on the radiation hardening and SEEs testing specifically for space applications or in case of CERN for high-energy accelerator applications. To understand the SEEs, not only experimental irradiation testing, but also Monte Carlo (MC) simulations are necessary. In the Chapter 2 the MC simulations of radiation transport, the Geant4 MC toolkit and SEEs are described.

Not only general-purpose MC radiation transport simulation tools exist, but also there are tools developed specifically for simulations of SEEs in electronics. One of such tools is G4SEE toolkit (based on Geant4), which compared to others is a freely available, open-source code, and is being developed as general as possible based on the needs of radiation effect community. It is being developed in the scope of CERN R2E project. In Chapter 3, detailed description of the toolkit structure and capabilities are given.

The development of the toolkit is constantly ongoing. In Chapter 4 the newly implemented features (compared to version 0.5.1) are described, namely Dose scoring, Linear Energy Transfer (LET) scoring, and improved Energy Deposition per Interaction scoring. The implementations of these are described and the example simulation results are shown. At the end of the chapter, the newly added and improved features, and also other minor changes implemented in G4SEE are summarized.

Complementing the development, also validation of the toolkit is continuously ongoing. In Chapter 5 of this thesis, validation results with a diamond detector irradiated by 14 MeV and 2.5 MeV neutrons are presented. Furthermore, simulation results for a silicon carbide (SiC) power MOSFET irradiated by 200 MeV and

3 MeV protons are also presented. Lastly, simulations for CMOS image sensors (CIS) irradiated by high-energy proton beam are shown. These simulations were preformed in preparation for a planned experimental irradiation test campaign, briefly described at the end of Chapter 5.

Chapter 2

Monte Carlo Simulations of Single Event Effects

2.1 Monte Carlo Radiation Transport Simulation

Monte Carlo (MC) method is a class of mathematical methods involving the use of random numbers in calculations based on stochastic processes. [1] It was first used in the 1940s by scientists working on the Manhattan project in Los Alamos, namely Nicholas Metropolis, Stanislav Ulam, Richard von Neumann, Richard Feynman, and others.[2] MC methods can be used to solve many different problems in various fields, such as solving partial differential equations and modeling economics. Additionally, they are commonly used for radiation transport simulations. This thesis will focus on the MC simulations of radiation transport.

The aim of MC radiation transport simulation is to numerically simulate a large number of random particle trajectories undergoing defined interaction mechanisms and to combine contributions of particle tracks to the statistical estimator of observable. By radiation is meant a collection of particles (for example γ, e$^{\pm}$, $\mu^{\pm}$, ν_i, n, p, $\pi^{\pm}$, ions, ...) propagating in matter and undergoing series of different radiation-matter interactions. These interactions include angular deflections, energy losses, generation of secondary particles, and the production of residual nuclei, which can possibly be radioactive. [3]

In the conventional MC simulation of radiation transport, the particle trajectory is going through homogeneous media, where particle is randomly scattered (such as gases, liquids, and amorphous solids). In such simulations, the material of the media is not affected by previous particles, and the secondary particles do not interact with each other. [4]

2.1.1 Basics of Monte Carlo method

Particle density

One of the basic quantities used in radiation transport simulations is particle density $n_i(r, E, \Omega, t)$, where r represents the particle's position in media, E its energy, Ω the direction along which the particle travels, and t time. This quantity represents the number of particle species i per unit volume, unit energy, unit solid angle, and at a given time. [3]

Interaction cross section

For particle-atom/molecule interaction, various interaction mechanisms are possible, each characterised by differential cross section (DXS). This cross section depends on the particle state variables that undergo a change during the interaction process. Following equation describes the DXS:

$$\frac{d^2\sigma}{d\Omega dW} \equiv \frac{\dot{N}_{count}}{|J_{inc}|d\Omega dW} \tag{2.1}$$

Here, σ represents the microscopic cross section, J_{inc} is the current density of the incident beam, W is a certain energy loss of a particle, $d\Omega$ represents a small angle under which the detector is placed, and $\dot{N}_{count}$ is the number of counts per unit time. The dimension of DXS is [area/(solid angle × energy)]. Essentially, the DXS represents the likelihood of a particle being scattered into a direction Ω with energy loss W. [4] An illustration of the above mentioned quantities is shown in the figure 2.1.

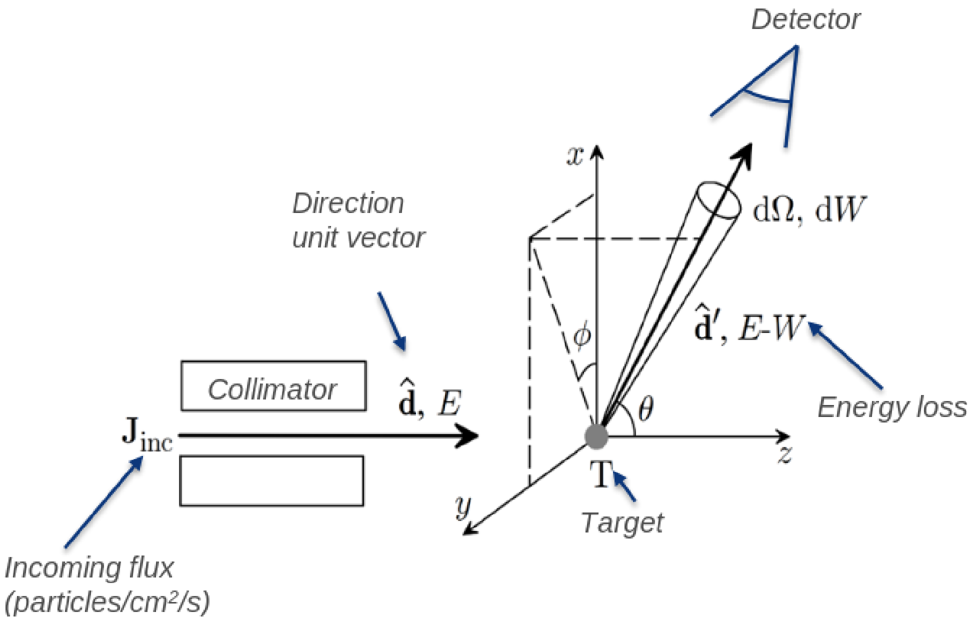
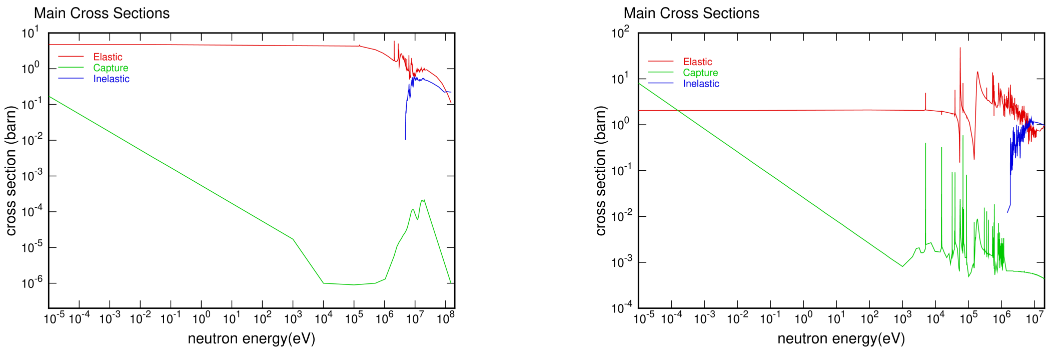


Figure 2.1: Schematic diagram showing an experiment to measure the DXS. Incident particles move in the direction of the z axis, θ is the polar and ϕ is the azimuth scattering angle. [4]

Total microscopic cross section can be computed from DXS as:

$$\sigma_{TOT} \equiv \int_0^E (\int \frac{d^2\sigma}{d\Omega dW} d\Omega) dW \tag{2.2}$$

The dimension of σ_{TOT} is area, with the typical unit being 1 barn (equivalent to 10^{24} cm^2). Geometrically, the total microscopic cross section can be imagined as the area of a plane surface perpendicular to the incident beam, crossed by a number of projectiles undergoing various angular deflections and energy losses. [4] Figure 2.2 shows an example of cross section data for neutrons used by Geant4.



(a)

(b)

Figure 2.2: Dependence of σ_{TOT} total microscopic cross section on incident energy in case of neutrons interacting with a) ${}^{12}_{6}$C nuclei, and with b) ${}^{28}_{14}$Si nuclei. This data are used by Geant4 [5, 6].

The transport equation

Solving the Boltzmann equation of particle transport is one of the basic ingredients of MC simulations. This equation describes the change of the radiation field[1] at another position in later time within a small volume V, with surface S. It is expressed as follows [7]:

$$\begin{aligned}\int_V d\mathbf{r}\,\frac{dn_i(\mathbf{r},E,\Omega,t)}{dtdEd\Omega} &= -\oint_S dA\, j((\mathbf{r},E,\Omega,t)\cdot\hat{\mathbf{a}}\\ &-N\int_V d\mathbf{r}\, n_i(\mathbf{r},E,\Omega,t)v(E)\sigma(E)\\ &+N\int_V d\mathbf{r}\int dE'\int d\Omega'\, n_i(\mathbf{r},E',\Omega',t)v(E')\frac{d\sigma}{d\Omega'' dW''}\\ &+N\int_V d\mathbf{r}\int dE'\int d\Omega' \sum_j n_j(\mathbf{r}E,',\Omega',t)v(E')\frac{d\sigma_{\text{sec},i}}{d\Omega'' dW''}\\ &+\int_V d\mathbf{r}\, Q_{\text{source}}(\mathbf{r},E,\Omega,t)\end{aligned} \tag{2.3}$$

In simpler terms, equation 2.3 determines the time evolution of particle density n_i in a small volume for a list of particle species and the cross sections for respective

[1] Radiation field is an ensemble of particles (it may consist of different particle species), at a position r, energy E, moving along the direction of Ω

interaction mechanisms. The Boltzmann equation consists of several contributions (terms). The first term of right side describes the particle transit, i.e. particles passing through the volume without scattering. The second term represents particles that were originally in the volume but were then scattered out of it, and third term represents particles that were originally outside the volume were then scattered in it. The out-scattered particles can lead to production of one or more secondary particles, as added by the fourth term. The last term represents the source of particles inside the volume, which can be, for example, radioactive nuclei. N in the equation represents number of target atoms per unit volume. [7]

All of the above mentioned represent a statement of the balance of incoming, produced, and outgoing particles, which is represented by an integro-differential equation. Analytical solutions to this equation are only possible for simplified scenarios such as infinite medium, few interaction mechanisms, or few particle species. Therefore, this problem needs to be solved numerically. One method that can be used to solve equation 2.3 is numerical quadrature integration, which can handle general cases, but is inefficient for high-dimensional integrals. The best to use is the MC method, which is well-suited for this purpose. The MC method solves the equation 2.3 for arbitrary source variations, geometries, and an arbitrary number of particle species. [7]

Mean free path

Another important quantity for MC radiation transport simulations is particle mean free path. It is determined by the probability density function (PDF) of the particle's path length from its current position until the next interaction with an atom or molecule of the medium, which has N atoms or molecules per unit volume. The mean free path is defined as:

$$\lambda \equiv \frac{1}{N\sigma} \tag{2.4}$$

This represents the average path length between interactions. [4]

Particle tracks

In the simulation process, generated particle tracks (trajectories), start off at a given position with a given initial direction and energy. These are the source characteristics. The state of a particle just after an interaction is defined by position coordinates, energy, and the direction cosines of the flight direction. Each particle track consists of a series of these particle states. [4]

Markov process

The preceding definitions are based on the theory that particle transport can be modelled as a Markov process, which means that the future of a particle does not depend on its past history and particles interact with an individual atom or nucleus. Thanks to this the particle history generation can be stopped at any state of the particle and the simulation can be continued from that point without any bias in the results. This makes the MC simulations much simpler and faster to implement. [4]

2.1.2 Variance reduction methods

Analogue[2] simulations can take a lot of computational time. To reduce variance (uncertainties), which directly reduces computational time, variance-reduction (biasing) techniques are used. They reduce the statistical uncertainty of a simulated quantity without increasing computational time. [4]

Any MC simulation is equivalent to the calculation of an integral for the expectation value, equation 2.5, where Q is the quantity of interest (such as deposited energy in certain volume, or particle fluence), $q(x)$ represents the contribution of an individual event, x is a set of random variables characterizing each shower, and $p(x)$ is a PDF of that specific shower.

$$\overline{Q} \simeq \int q(x)p(x)dx. \tag{2.5}$$

Statistical uncertainties are evaluated from the variance of the quantity, as follows:

$$var_Q \simeq \int [q(x) - \overline{Q}]^2 p(x)dx = \int q^2(x)p(x)dx - \overline{Q}^2. \tag{2.6}$$

To optimize the efficiency of the simulation, the weight w is applied to each particle, usually initialized equally for all primary particles, and then appropriately adjusted as the particle history unfolds. Applying the weight is shown in equation 2.7 and the variance is computed in equation 2.8. [4]

$$\overline{Q} = \int wq(x)\frac{p(x)}{w}dx = \int q'(x)p'(x)dx \tag{2.7}$$

$$var'_Q = \int q'^2(x)p'(x)dx - \overline{Q}^2 = \int wq^2(x)p(x)dx - \overline{Q}^2 \tag{2.8}$$

By choosing a suitable weights such that the $var'_Q < var_Q$, the efficiency of the simulated quantity of interest is improved. This implies that the desired level of accuracy can be achieved in less computational time. It also means that for simulations of very rare events, a significantly higher number of events or particles can be obtained, improving the statistics for rare events.

Types of variance reduction methods

Interaction forcing (cross section biasing): In this method the probability of an interaction process of interest is artificially increased. Essentially, this means that the likelihood of interactions that typically occur with low probability is artificially increased, resulting in high variance results. This is done by shortening the mean free path of a particle. To ensure the simulation remains unbiased, the applied weights must be corrected. This method should only be employed for particles above certain energy threshold. [8]

Splitting and Russian roulette: These two methods are commonly used together, especially in cases where the focus lies on a localized spatial region, often

[2] "analogue" simulation refers to simulation that does not apply any variance-reduction method

referred to as a region of interest (ROI). The idea behind these methods is to promote the flux of radiation towards the ROI, while inhibiting the radiation that moves away from it. This approach saves numerical work by avoiding tracking particles that are unlikely to significantly contribute to the scores in the ROI. As in the interaction forcing, the variance reduction is achieved by assigning modified weights to particles. Secondary particles inherit the initial weight of their creator primary particle. When splitting technique is applied, particle with weight w_0 is transformed into multiple copies with weight $w = w_0/S$, where $S > 1$ is the number of copies made in the same state. This technique is applied when particle enters the ROI. Russian roulette, on the other hand, is the reverse process of splitting. Instead of creating particle copies, particles are "killed" when they move away from ROI with certain probability. "Killing" a particle means that it will no longer be tracked and therefore will not contribute to the score from that point onwards. [8]

Woodcock method: This method, also known as the **delta scattering method**, is applicable to photons. It is based on the fact, that photons have high penetration capability through materials with complex geometries, so their tracking can be simplified. The method operates on the assumption that photons undergo not only physical interactions but also delta interactions, which are interactions that do not modify the state variable of the particle. During photon transport across the geometry, an extended inverse mean free path is used. At the end of each free flight, a real interaction or delta interaction (which has no effect on photon) can occur. In this method the probability of real interaction per unit path length remains unchanged. The Woodcock method eliminates the requirement for computing intersections of particle rays with interfaces, albeit at the cost of having to determine which material is at the end of each free flight. This method improves the efficiency for geometries, where classical tracking is slower than finding a location of a particle. [8]

Other methods: Another way to reduce variation is to simply avoid unnecessary calculations. For general-purpose simulation codes, this can involve leveraging geometric symmetries of the studied problem. Additionally, another method involves rejecting or killing particles with energy lower than chosen threshold. These low-energy particles can be absorbed when they cannot leave the ROI. [8]

2.1.3 General-purpose Monte Carlo tools

The Capabilities of general-purpose MC codes are following [3]:

- transport of e^-, e^+, γ as well as muons, hadrons and ions
- source of radiation can be particle beam, particle-particle collision or radioactive isotope
- cover a broad energy range (up to TeV for accelerator physics, down to 1 keV for e-, e+, γ, and down to meV for neutrons)
- offer up-to-date and benchmarked physics models
- offer a flexible and efficient geometry settings

- capability to score physical observable of interest, for example energy deposition, particle kinetic energy, fluence, etc.
- capability to run in parallel or distributed

Examples of the general-purpose MC tools are Geant4 [9, 10, 11], FLUKA.CERN [12, 13, 14], MCNP [15]. Geant4 will be described in more details in the section 2.2, as the G4SEE toolkit is based on it.

Every MC simulation code of radiation transport has to include definition of geometry and radiation source, radiation-matter interactions, and random sampling techniques. These are described in more detail in sections below.

Radiation source and geometry definition

The initial state of the simulation must be characterised, including sampling schemes for selecting the initial values of all independent variables from a source PDF. This includes characteristics such as production per unit volume, per unit direction, per unit energy, and per unit time. The radiation source can be a particle beam, radioactive isotopes or particle-particle collisions. [1]

Another important ingredient of MC simulation is geometry definition. Geometries are composed of basic objects (bodies) such as spheres, cylinders, and planes. These basic objects can be combined via boolean operations (union, intersection, subtraction) to form complex regions. Additionally, MC codes have the ability to clone or replicate regions, allowing for the efficient modeling of complex geometries.

Radiation-matter interactions

The radiation-matter mechanisms or physics models are crucial ingredients of every MC code. While they all implement the same basic interaction mechanisms, each MC code utilizes different models, which may perform better or worse depending on the simulated situations. Therefore, it is important for users to cross-check if all relevant physics processes for the simulation are included in the code. [3]

In the basic interactions of **photons** include **Rayleigh scattering**, **photoelectric absorption**, **Compton scattering** and **e^+-e^- pair production**. For the photoelectric absorption and Compton scattering processes, the emission of fluorescence photons and Auger electrons is also taken into account. For the photon energy above a few MeV, photonuclear reactions may also be observed. Except for the Rayleigh scattering, all these interactions result in at least two particles in the final state, leading to a rapid increase in particle numbers, and producing electromagnetic particle showers. [3]

Similarly, interactions involving **electrons** and **positrons** are also fundamental in radiation transport simulations. These particles can interact **elastically**, which is a very frequent type of interaction. To make this efficient, multiple scattering algorithms are employed. **Inelastic scattering** is another common interaction type, and effective approach to reducing computational time is aggregating energy losses along particle steps via stopping power calculations. Additionally, two other significant interactions are **Bremsstrahlung emission** and the **annihilation**. [3]

The aforementioned Electromagnetic (EM) interactions are driven by the Coulomb potential and can be processed analytically. However, **hadronic interactions** pose greater challenges in modeling due to their reliance on parameterised reaction cross section, which are adjusted to experimental data. Accurate description of these interactions requires the combination of multiple physics models. Hadronic interactions undergo two distinct stages: a fast stage lasting approximately 10^{-22} s, during which neutrons, protons, and other fragments are emitted, followed by a slow stage lasting approximately 10^{-16} s. In the slow stage additional nuclear fragments may be emitted, γ rays may be produced from deexcitacion, or heavier nuclei may undergo fission, leaving behind residual nuclei (which can be radioactive). This interaction result in production of several tens or even more secondary particles (hadronic particle shower). [3]

Lastly, **low-energy neutron** interactions are also included in the group of basic interactions. Neutrons lose energy rapidly while interacting with material, primarily through elastic scattering, or the production of low-energy neutron secondaries. Neutron capture process have a high cross section for low-energy neutrons, leading to the creation of ∼MeV photons and the production of residual nuclei. This aspect is critical for radiation protection and shielding aspects. Modeling of low-energy neutron interactions is complicated due to the rich resonant structure of neutron cross sections, as depicted in figure 2.3. No single model effectively reproduce all resonant details. Therefore, two approaches for the simulations can be chosen: using less precise **group-wise** neutron interaction model, or more precise **point-wise** neutron interaction model. Both models rely on evaluated nuclear data libraries such as Evaluated Nuclear Data Files (ENDF) [16], Joint Evaluated Fission and Fusion (JEFF) [17] and others, covering elastic scattering, capture, fission and explicit inelastic reaction channels. [3]

Random sampling techniques

Every MC calculation relies on a sequence of random numbers that are uniformly distributed over the open interval $[0, 1)$. Since computers are not able to generate true random numbers, pseudo-random numbers are used instead. A pseudo-random number generator is a deterministic algorithm that, based on the previous state in the sequence, calculates the next number. It requires a seed to start a sequence and consistently produces the same sequence of numbers when initialized with the same seed. This enables the reproduction of the same result with the same code running on different computers. [19]

In Geant4, HEPRandom [20] module of CLHEP library (Computing Library for High Energy Physics) is used as a random number generator. This module consists of classes implementing various random engines and different random distributions associated with an engine, representing a random generator. The distribution class collects different algorithms and calling sequences for each method to define distribution parameters or range intervals. The basic algorithm for generating pseudo-random numbers is implemented by the engine. In G4, the static generator is used, where

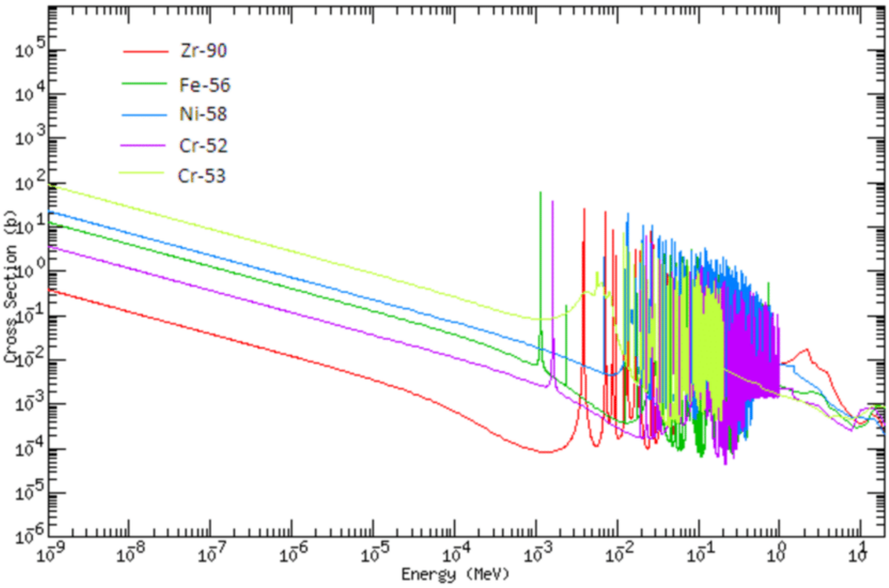


Figure 2.3: Neutron cross sections plotted as a function of incident neutron energy for the highest abundance isotopes in zirconium, iron, nickel and chromium. [18]

random values are generated using static methods defined for each distribution class. [21]

In radiation transport simulations, quantities such as step lengths, energy losses, and deflections must be sampled. Since typically these values are not uniformly distributed, sampling techniques such as inverse sampling, rejection sampling, and composition methods are employed. These techniques allow for the generation of random values according to given distributions (for example exponential PDF for particle step), ensuring accurate simulation results. [19]

Steps of a MC radiation transport simulation

In every simulation of radiation transport, the first step involve defining the radiation source and material geometry. Subsequently, a sufficiently large number of primary particles, denoted as N, should be started, as the statistical uncertainty decreases as $1/\sqrt{N}$, as implied by the central limit theorem[3]. Due to the Markov process the simulation can progresses through following steps [3]:

1. New primary event occurs, adding one (or more) particles to the stack.
2. One particle from the stack is selected and advances in the material medium, with the contribution of this step to scoring observable.
3. The mean free path $\lambda(E)$ for this particle is evaluated.
4. A random step length to the next interaction is sampled.

[3] Central limit theorem says that if the estimator receives a sufficiently large number of contributions N, than the distribution of the mean value tends to a Gaussian distribution centered around the true expectation value, with standard deviation $\frac{\sigma}{\sqrt{N}}$

5. The contribution of sampled step to the statistical estimators is accumulated.
6. The type of interaction that will occur is determined.
7. The final state of the particle is sampled, possibly adding secondaries to the stack if any produced.
8. The contribution of the interaction is accumulated to the statistical estimators.
9. If the particle survives, its energy and direction is updated, and the process repeats from the step 3. Otherwise, the process returns to the step 2., or to the first step, if there are no more particles in stack.

2.2 Geant4 Simulation Toolkit

Geant4 (G4) is a freely distributed, open-source MC particle transport software toolkit used to accurately simulate the passage of particles through matter and their interactions. It encompasses all aspects of the simulation process and supports stand-alone applications based on G4. [21]

Figure 2.4: Geant4 logo [21]

Written in C++, the toolkit follows an object-oriented design. When developing a specific applications based on G4, application developers and users have a range of options for implementing different features and can incorporate their code into user action classes provided by the toolkit. Geant4 evolved from its predecessor, Geant3, which was a FORTRAN based MC code. The first release of G4 was in December 1998, nowadays one public release is made every year. [21]

G4 includes a wide range of physics models, covering electromagnetic processes, hadronic and nuclear processes, optical photon processes, and more. With the toolkit any type of geometry, including solid geometry, complex boolean geometry, dynamic geometries can be constructed. Further discussions on geometry and physics models will be provided in chapter 3. [22]

On the diagram 2.5, the key functionalities (class categories) of G4 are displayed. The categories highlighted in red are handled by G4 kernel, which provides frameworks for geometrical representation and physics processes. The G4 working group maintain the interfaces between categories (e.g. top level design). The categories shown at the bottom of the diagram serve as the foundation for the toolkit and are used by all the categories displayed above them. Here is a more detailed description of each category [21]:

- Global: This category covers the system of units, constants, numeric and random number handling.
- Materials and particles: These implement the necessary infrastructure to describe the physical properties of particles and materials for simulating particle-matter interactions.
- Geometry: This category enables the description of geometrical structures, allowing particles to be effectively propagated through them.
- Track and processes: The track category contains classes for tracks and steps. Features of particle track are than used in the processes category, where models of physical interactions are implemented.
- Tracking: This category manages the invocation of all processes and their contribution to the evolution of track's state. It also provides information for hits and digitization in sensitive volume.
- Event: This category is in charge of managing events in terms of their tracks.
- Run: Events that share a common beam and detector implementation are collected in this category.
- Readout: This category allows the handling of pile-up.
- Visualization, Persistency and user interface capabilities: Finally, all of these three categories use all of above mentioned categories, and are able to connect them with facilities outside the toolkit through an abstract interface.

Objects of G4 user action classes are explained in more detail below. Namely these are track, step, event, and run. Their hierarchy is displayed in figure 2.6. User action classes are used either to setup or modify the simulation, or to collect information about the run.

Track

In G4, a Track represents a snapshot of a particle's state at a specific point in its trajectory. It involves physical quantities such as position, energy, and momentum, providing information about particle's current state, the past states are not stored in the simulation history. A track object is terminated when the particle leaves the World volume, disappears in an interaction, has zero kinetic energy, or is "killed" below user-defined threshold cuts. This implies that at the end of an event no more track exists. Tracks are managed by the `G4TrackingManager` class, and are represented by the `G4Track` class. [23]

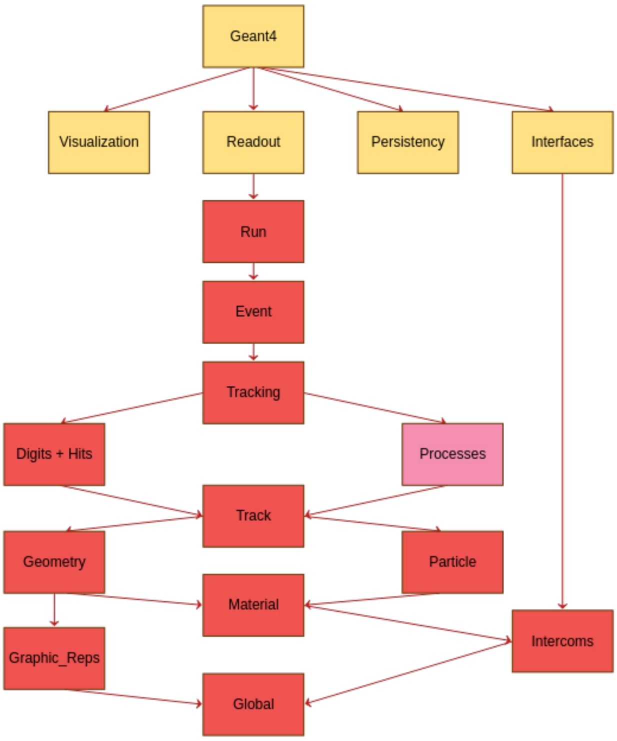


Figure 2.5: Geant4 categories diagram. G4 kernel consists of red colored categories. [21]

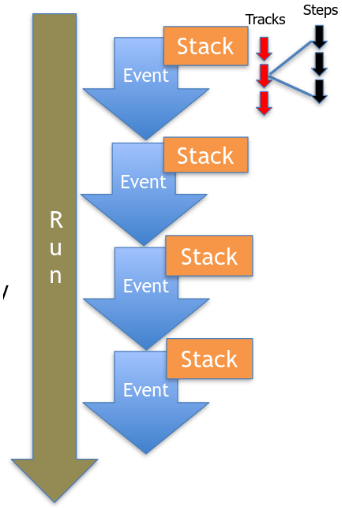


Figure 2.6: Diagram of hierarchy of the G4 user actions [23]

Step

In Geant4 (G4), particle move through medium by taking discrete steps along their trajectory. Each step represent the most recent information about particle state, meaning its current position, energy, momentum etc. Step is defined by pre-step and post-step points. If step is limited by volume boundary than the Post-step point physically stands on the boundary and belongs to the next volume. The kinematic information stored in Post-step point is also stored in `G4Track` at recent time, because the track represents the current state of a particle. `G4Step` class corresponds to a Step object. [23]

Event

An Event in G4 is fundamental unit of simulation in terms of Tracks. At the beginning of an event, all primary tracks are generated and stacked. During event processing, tracks are picked one at a time from stack and it are tracked through the geometry. Once all tracks in the event have been processed, the event is considered complete. The G4Event class represents an Event object, containing information about primary vertices, particles, hits and trajectories collections. [23]

Run

The `G4Run` object is composed of a simulation configuration and a set of Events. Before initializing a Run, the detector setup, source characteristics, and physics processes settings must be defined. It is represented by `G4Run` object.

The hierarchy of above mentioned objects is shown in figure 2.6. In G4 a particle is represented by G4Track, this includes information about its position, geometrical information etc. Further it is defined by `G4DynamicParticle`, this class describe particle dynamical properties, such as momentum, energy, and polarization. Lastly `G4ParticleDefinition` class represents the static properties of a particle, such as charge, mass, life time, and decay. [23]

2.3 Single Event Effects

The trend in the semiconductor integrated circuits (ICs) manufacturing is to continually reduce their sizes, leading to higher density and nanometer feature size. This development is fueled by the demand for reduced power consumption and increased operational speeds of circuits. Combination of smaller feature size, lower supply voltage and higher packing density increases the probability of single ionizing particle inducing Single Event Effects (SEEs) due to the lower critical charge required. [24]

SEE first gain prominence in space-borne electronics systems when they were reported in the operation of Hughes satellite system in 1975, resulting in communication failures. Follow-up observations of SEEs (bit upsets) in random access memory circuits, attributed to heavy ion cosmic rays, further highlighted the impact of SEEs on electronic systems. These incidents initialized the need for radiation hardening and SEE tolerance in semiconductor ICs, leading to the development of various mitigation strategies. [25]

SEE pose a challenge not only in space missions but also in other environments with high radiation, such as high-energy accelerators. Nowadays problems of SEEs occurring are not only restricted for the high-level radiation environments, but also occurring in low radiation environments. They pose a risk for example for self driving cars, medical implants, and data centers. [25]

The natural space radiation environment, consists of particles trapped by planetary magnetosphere (Van Allen belts), galactic cosmic rays and solar particles events (such as coronal mass ejection and flares), including protons, electrons and heavier ions. Terrestrial atmospheric and ground-level radiation environment, made out of showers of secondary particles produced in interactions of cosmic rays with atmosphere are the source of SEEs on Earth. Notably, neutrons play a significant role in causing SEEs, due to the inelastic interactions with nuclei of atoms present in ICs, leading to either ejection of nuclear fragments or to spallation reaction products. Another key source of SEEs are alpha particles (in the energy range of 4 MeV to 6 MeV) emitted in radioactive decay of uranium and thorium impurities contained in the packaging materials. Lastly, low-energy or thermal neutrons are important source of SEEs as well. They pose a danger when there is ^{10}B present in the packaging, which isotope has high capture cross section for thermal neutrons. After neutrons are absorbed, ionizing fragments are produced, which can induce soft errors in the sensitive volume of electronics. [24]

The basic mechanics of SEE involve energetic particle passing through semiconductor or insulating materials used in the ICs, see in figure 2.7. This excites electrons from the valence band, leaving behind free electrons and holes. When passing near or through a reverse-biased semiconductor P-N junction, the high electric field can effectively separate these carriers, leading to current generation at the terminals of a semiconductor device. This charge collection occurs within tens of picoseconds in advanced events. Another way of charge collection is through diffusion of charge carriers from semiconductor's bulk or substrate near the depletion region. There they can be collected by a circuit node, leading to higher total charge collected. This process happens over nanoseconds time scale. The junction electric field can be locally extended by the charge generated along particle's track, leading to creation of a field funnel region. Extended junction field can interfere with the substrate creating an increase in charge collection at the struck node, charges which are deposited far away from the junction can be collected due to the drift.

In Complementary Metal-Oxide-Semiconductor (CMOS) devices, the created electron-hole pairs are restricted to the well region, where transistors are located. Here, the parasitic bipolar effect[4] can enhance charge collection and potentially lead to logic state changes in adjacent transistors. [24]

SEEs in electronic devices are characterized by threshold Linear Energy Transfer

[4] To every core transistor in an IC comes a parasitic bipolar transistor under the transistor channel. If one of these parasitic transistors ever turns on during IC operation than the logic state from one of the transistors can leak through the parasitic transistor to the adjacent transistor and cause its logic state to change. [26]

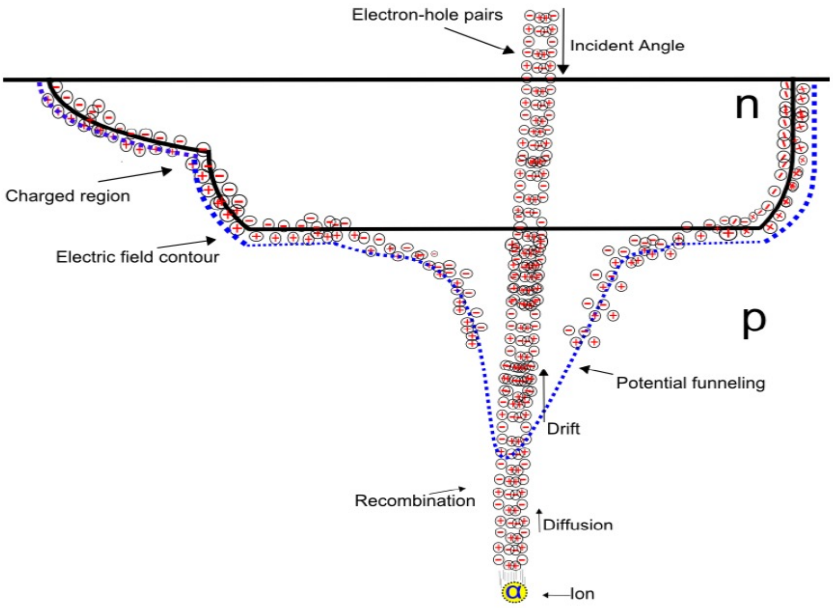


Figure 2.7: Basic mechanics of SEE [27]

(LET), which is the lowest LET at which SEE occurs, and by SEE cross section, expressing the likelihood of an SEE occurring. This is calculated as the ratio of single event error count and the fluence of particles potentially causing an error.

SEEs can be classified either as non-destructive ("soft") or destructive ("hard") errors, as shown in figure 2.8. Non-destructive errors are transient and can be corrected, while destructive errors result in permanent damage of the device. The non-destructive error rate, typically expressed as a unit of number of failures-in-time (FIT), estimates the frequency of non-destructive errors over a specified operation period, where 1 FIT equals to one soft error per billion hours of operation of the device. [24]

Additionally, there is another large group of radiation effects called cumulative effects. In these effects the Total ionizing dose (TID)[5] impacts the conductive properties of the material. The non-ionizing energy loss contribute to the structural damage (e.g. atomic rearrangement) on the crystal lattice of material, this is called Displacement Damage (DD). [28]

2.3.1 Types of SEEs

On diagram 2.8, the division of different types of stochastic SEEs in non-destructive and destructive groups is depicted. Below more detailed description of these effects is given.

Non-destructive Single Event Effects [27]:

- **Single Event Upset** (SEU): It is a change of a latched logic cell from state 1 to

[5] TID is the total amount of energy deposited by particle that results in e-h pair production (ionization) in a unit mass, unit: Gray (Gy).

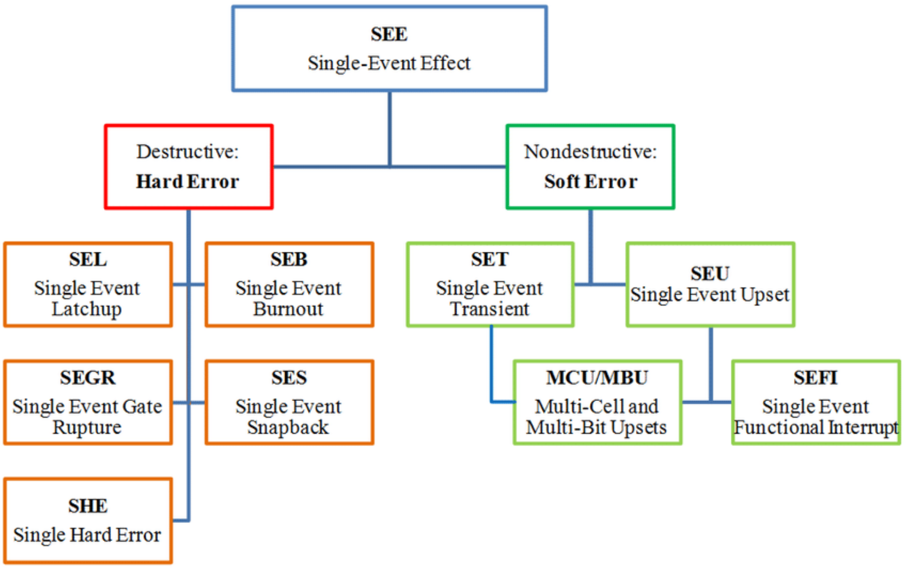


Figure 2.8: Types and classification of SEEs [29]

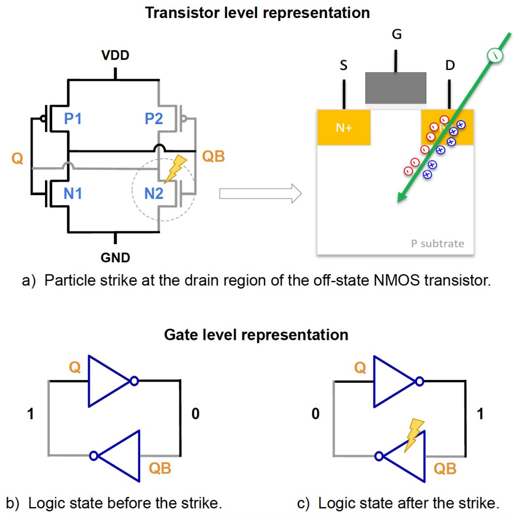
0 or vice-versa. These logic elements, commonly found in Static Random Access Memory (SRAM) cells, can be rewritten or reset. [30] Figure 2.9 illustrates this phenomenon.

Not every particle striking a memory cell and depositing energy will cause a SEU. The energy has to be deposited within a range sufficiently close to the sensitive node of the device. This is why the sensitive volume (SV) is defined as the volume where the collected charge actively contributes to the SEU. The deposited energy in the SV must also exceed the critical charge of the SRAM cell to upset it. [31]

Both heavy ions and hadrons can cause SEU. In space heavy ions have enough stopping power to deposit high enough energy in the small SV of semiconductor, and thereby cause SEU. On the other hand, hadrons usually cannot deposit enough energy through direct ionization, however they can still cause SEU indirectly by interacting with the nuclei of the memory cell or its close surrounding. The recoiled nuclei from these interactions have high enough stopping power to cause a SEU. This indirect ionization is significant for high-energy accelerators (like Large Hadron Collider (LHC)) at CERN, where the radiation environment consists mainly of neutrons and charged hadrons. [31]

There is a wide variety of mitigation techniques against SEUs, including hardware, software, and device tolerant solutions. One such method is placing the the thin layer of silicon on insulator during the chip manufacture, silicon-on-insulator technology process. The thin Si layer protects the bulk from charged particles, reducing the SEUs.[32]

- **Multiple Cell Upset** (MCU) and **Multiple Bit Upset** (MBU): MCU occurs when two or more logic cells, usually placed next to each other, change their current state. MBU is a special case of MCU where the corrupted cells lie within a single logical word. It can not be corrected by simple error correction code. Both of these effects typically represent only small fraction of all SEEs. [30]



a) Particle strike at the drain region of the off-state NMOS transistor.

b) Logic state before the strike.

c) Logic state after the strike.

Figure 2.9: Illustration of SEU in NMOS transistor, where in a) is illustrated the region where particle strikes causing an SEU and in b) and c) the logic states are shown before and after the particle strike. [27]

- **Single Event Transient** (SET): This effect appears as a temporary voltage deviation at a node in logic or linear IC. It propagates through the circuit and may be stored as incorrect data, causing disruption of the circuit operation. [30] A schematic of SET is shown in figure 2.10. [30]

- **Single Event Latch-up** (SEL): This effect can be potentially destructive. Semiconductor regions in CMOS transistors can contain parasitic bipolar - PNPN thyristor structures, as shown in figure 2.11. In this type of structure, an increase of a PNP collector current leads to an increase in the NPN base current, which in turn increases the collector current of the NPN, leading to an increase of the PNP base current. If the overall gain of the PNPN structure is high enough, an ionization particle strike can trigger latch-up by turning on one of the bipolar junction transistor structures. [31, 24]

  For recovery of the device a power cycle is needed. If the power supply is promptly cut after the latch-up is initiated, the effect is non-destructive. However, if not, it can result in catastrophic damage to metallization and junctions, potentially leading to a destructive error. To reduce SEL sensitivity of the device, the best approach is to decrease the gain of the parasitic PNPN structure. This can be achieved either by using epitaxial substrates, or trench isolation instead of junction isolation, or by layout modifications, such as increasing the distance between the complementary devices, using guard rings, or employing numerous substrate and well contacts. [24, 31]

- **Single Event Functional Interrupt** (SEFI): It is a soft error that leads to component lockup or malfunction, typically occurring in critical registers. This affects the device's functionality. Two different types are distinguished based on how the operational state is restored, which can be done by software reset or by power cycling (hard reset), and may cause data loss. [30]

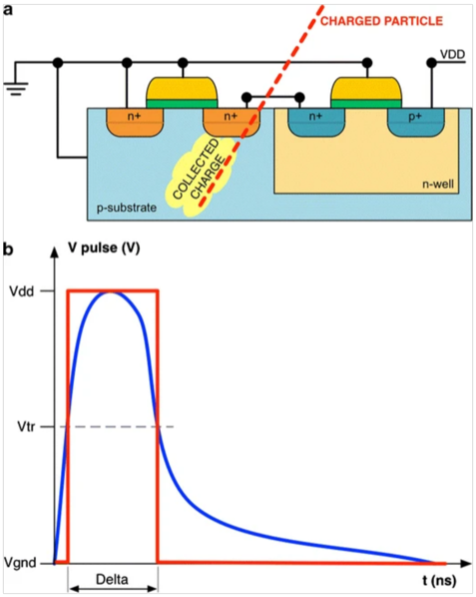


Figure 2.10: Schematic of SET (a) interaction of charged particle with the transistor (b) output of the affected cell, showing the voltage spike plotted as function of time. [33]

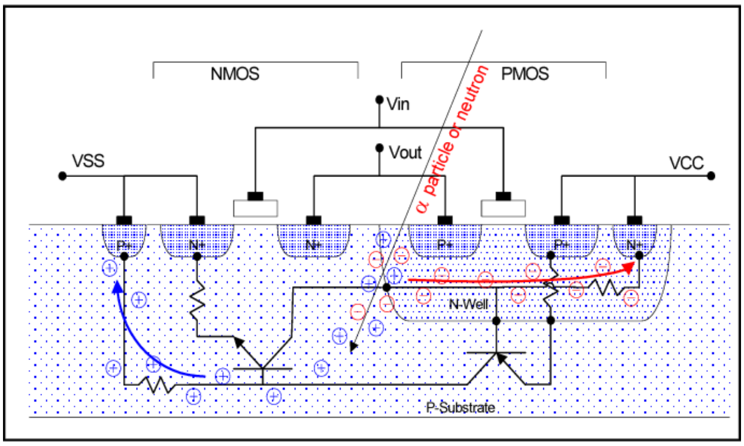


Figure 2.11: SEL caused by energetic particle in CMOS structure [34]

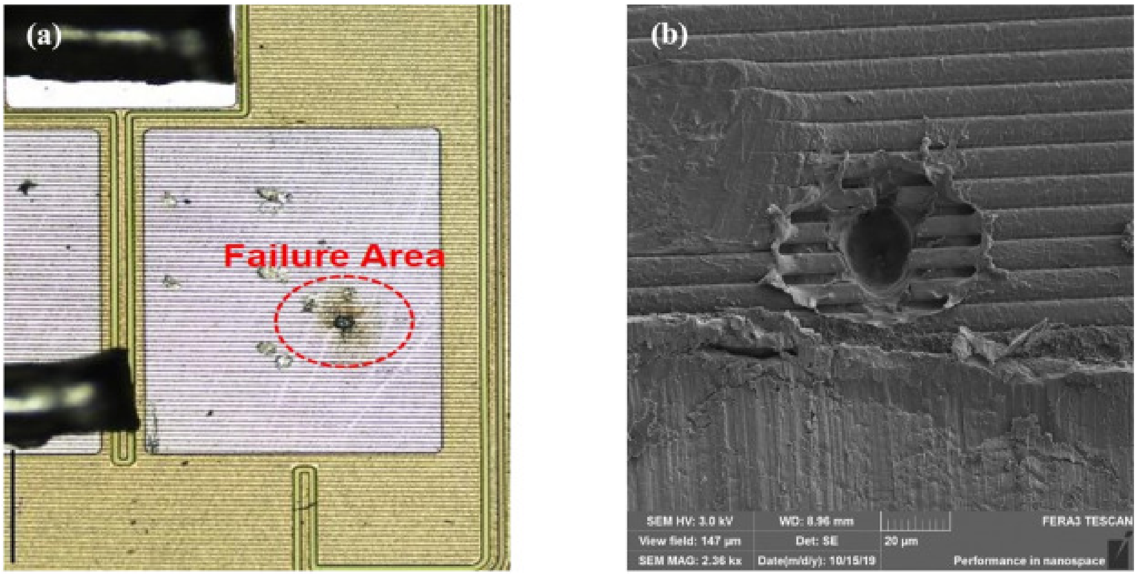


Figure 2.12: Two different images of the failure on the die surface of a SiC crystal [37]

The second group are destructive SEEs, as shown in figure 2.12, where visible destruction in the device is evident. Beside destructive SEL, this group includes [35]:

- **Single Event Burnout** (SEB): This effect occurs in power transistors, where it triggers a parasitic bipolar structure together with regenerative feed-back, leading to high current condition and avalanche. This can result in a destructive burnout if not suitably protected. [30]

  SEBs pose a danger for power MOSFETs. It is induced when heavy charged particle penetrates the device with enough energy to generate electron-hole pairs along its track. When such event occur the created charge occurs, the created charge carriers are multiplied in depletion region, leading to parasitic bipolar transistor activation. If there is no interruption, second breakdown occurs and the regenerative current increase can destroy the device.[36]

- **Single Event Gate Rupture** (SEGR) and **Single Event Dielectric Rupture** (SEDR): These effects lead to a destructive rupture of a gate oxide or any dielectric layer, resulting in a leakage current through gate oxide or dielectric under bias voltage. They can be observed in power MOSFETs, linear ICs, or as stuck bits in digital components. [24]

It is very important to select radiation-tolerant or even radiation-hard devices and systems critical for operation, or make them radiation-hardened, especially if they are vulnerable to radiation effects. There are several radiation hardening and mitigation techniques developed already to get higher reliability and better performance for devices in radiation environments.

Radiation Hardness Assurance (RHA) consists of deploying and applying all activities needed to ensure that electronic components, devices of a system, regardless of containing potentially radiation-sensitive or radiation-tolerant parts, meet their design specifications and operational requirements. At CERN there is Radiation to

Electronics (R2E) project, with the main high-level purpose to ensure the successful and uninterrupted operation of CERN particle accelerators regarding radiation effects on electronics, which therefore has RHA related tasks as well. [28] Further discussion on activities of R2E project are provided below in section 2.3.2.

2.3.2 CERN Radiation to Electronics project

Main responsibility of the R2E group at CERN is to prevent beam dumps due to electronics failures caused by radiation. The need to start this project arose in 2008, when the CERN Neutrino to Gran Sasso target area was suffering from many SEEs on microcontroller in the ventilation system causing the beam dump. From the beginning of the LHC operation the R2E project assists with the estimation of radiation-induced failures of the electronics system. The effect of improvements made since 2011 can be seen of figure 2.13. Mitigation techniques involve placement of shielding, equipment repositioning, and also radiation hardness assurance procedures. [38]

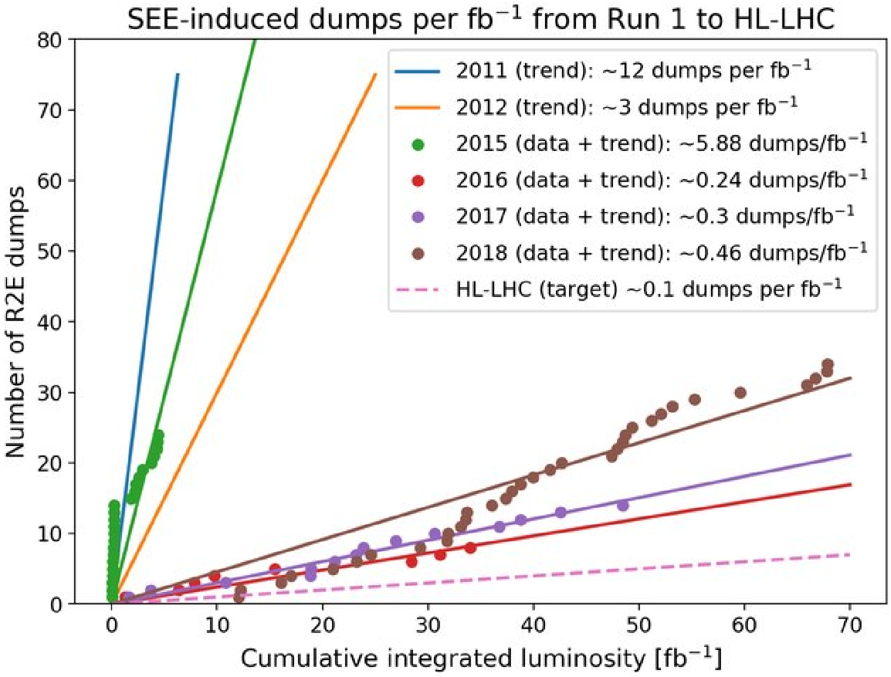


Figure 2.13: Number of LHC beam dumps induced by electronic failures due to radiation as a function of the cumulative integrated luminosity in 2011-2012 (Run 1, trend only), 2015-2018 (Run 2, data and trend) and HL-LHC (target). [39]

High-energy hadron accelerators require a significant number of electronic components being placed close to the beam-line and to the interaction points. The LHC can contain up to the thousands of commercial off-the-shelf (COTS) units, such as power converters (providing suitable voltage and current levels for the magnets), quenching protection system (protecting superconducting equipment from incident caused by excessive heat), cryogenics and many others. [28]

The key elements for developing radiation tolerant device are: precise definition of the radiation environment, radiation testing of the critical electronic components and systems, and the validation of the radiation tolerance of the equipment at system-level. The assessment of radiation damage to electronics is a complex process and requires also a detailed description of the radiation source, as well as characterization of the calculated quantities. For the monitoring and calculation to measure and predict radiation levels at the LHC, Beam Loss Monitors (ionisation chambers) and RadMon devices (COTS-based radiation environment and effect monitoring system) are placed in the LHC tunnel, and general-purpose FLUKA.CERN MC simulation code is used in combination. [38]

2.4 SEE simulation tools

As mentioned before, there are some SEE simulation tools exist already. One of such tools is MRED, which is briefly presented in this section.

Monte Carlo Radiative Energy deposition (MRED), a Geant4-based radiation transport code, is developed at Vanderbilt University. It is a Python application that simulates radiation transport integrating selected physics processes of the G4 toolkit, Fortran codes of PENELOPE[6] 2008 [4] and CEM03[7] [40] and LAQGSM[8]. [41]

It serves as the core simulation engine (back-end) for CREME-MC[9] [42]. It computes deposited energy in semiconductor device size structures by assuming the well known relationship between deposited energy and average rate of generated charge in silicon (3.6 eV per electron-hole pair), the excess charge produced by radiation in devices is approximated. Other capabilities are for example determination of energy required to dislocate atoms from a crystal lattice (displacement damage) and electronic excitation (charge generation). Energy deposition simulation is the principal function, the aim is to compute it as accurately and precisely as current knowledge enables. [41] In figure 2.14 are shown observed LET data from a space satellite compared with CREME-MC data, which were used to determine the SEU rates.

In [44], authors were examining the influence of tungsten placed near the sensitive volume. They used MRED to identify the worst case energy for SEE susceptibility applied to certain bulk-Si and silicon-on-insulator (SOI) technologies.

In [45], the authors examined the upsets caused in 65 nm bulk CMOS SRAM by low energy protons. The experimental data were compared with MRED simulations and used for its calibration. Thanks to the good agreement the code can be used for general rate prediction in arbitrary space environments.

The MRED is not and open-source code and its features, publicly available only via CREME-MC, are very limited, for example simulations of neutron induced SEEs are not possible. Other existing codes are even more inaccessible or restricted. This is the reason why G4SEE is being developed and used instead of using the MRED.

[6] PENELOPE is electron and photon transport MC code

[7] CEM03 is a MC code calculating nuclear reactions in the framework of the improved cascade-exciton model

[8] LAQSM is a MC code simulating spallation events for any incident particle

[9] CREME-MC is a online web interface for modeling energy deposition in materials due to heavy ion radiation

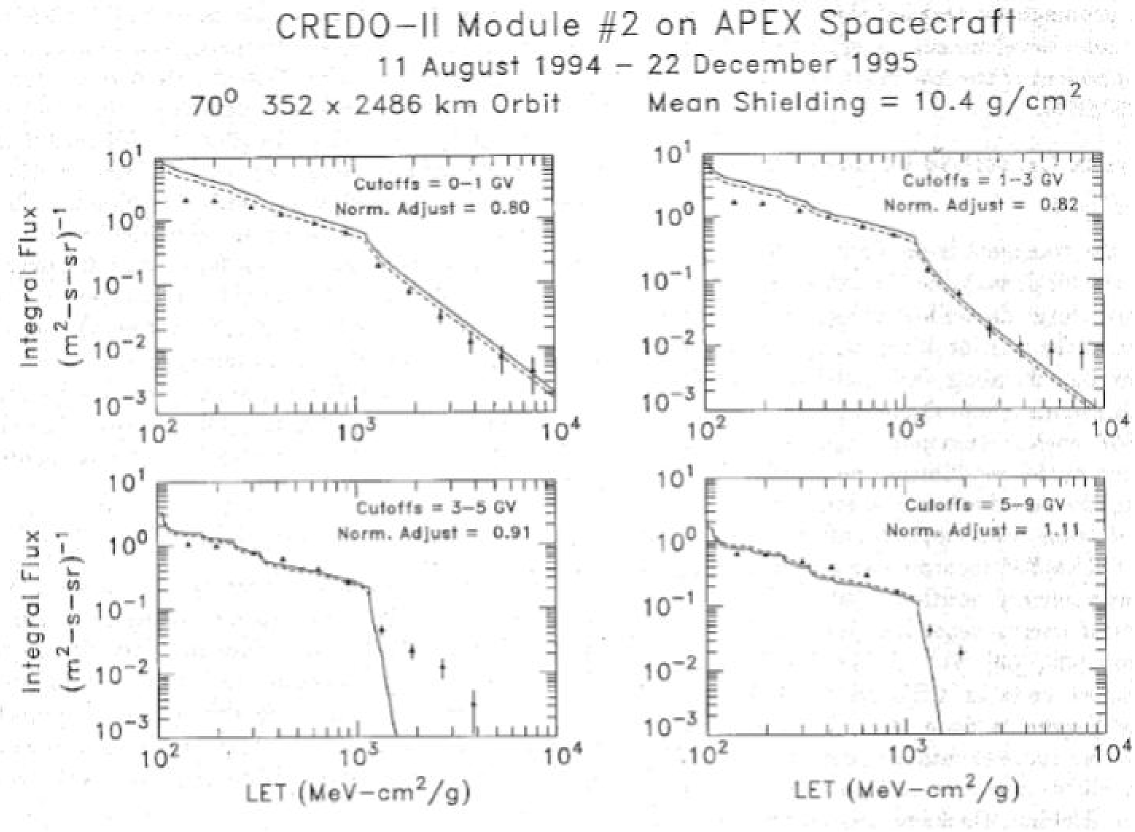


Figure 2.14: Observed LET spectra from a space satellite (filled triangles with statistical error bars), compared to absolute CREME96 predictions (solid lines) and normalization-adjusted (dashed lines) CREME96 predictions. [43]

Chapter 3

The G4SEE toolkit

Geant4-based Single Event Effect simulation toolkit (G4SEE) [46, 47, 48] is being developed at CERN within the Radiation to Electronics project [38] for the whole radiation effects community for a wide range of practical applications. G4SEE is freely available open-source code. [49] It has been validated experimentally by measurements of inelastic energy deposition events of mono-energetic neutrons in the 1.2 MeV - 17 MeV energy range, measured by a silicon diode detector. [46] Main goal of G4SEE is to focus on energy deposition scoring in microelectronics structures.

In the sections below is given more detailed description of G4SEE capabilities, user support and example use cases.

Figure 3.1: G4SEE logo

3.1 G4SEE capabilities

The architecture, inputs and outputs of the G4SEE toolkit version 0.5.1 is shown in the figure 3.2, which will be described in detail in the following sections together with its main simulation features.

3.1.1 User input

An input file is a G4-style ASCII macro file. All user-defined settings and parameters for a single G4SEE simulation run are set by G4SEE commands and some general G4 commands. If user wants to run a simulation with variable parameters, for example with different sizes of SV, a parametric YAML file is used. There, the

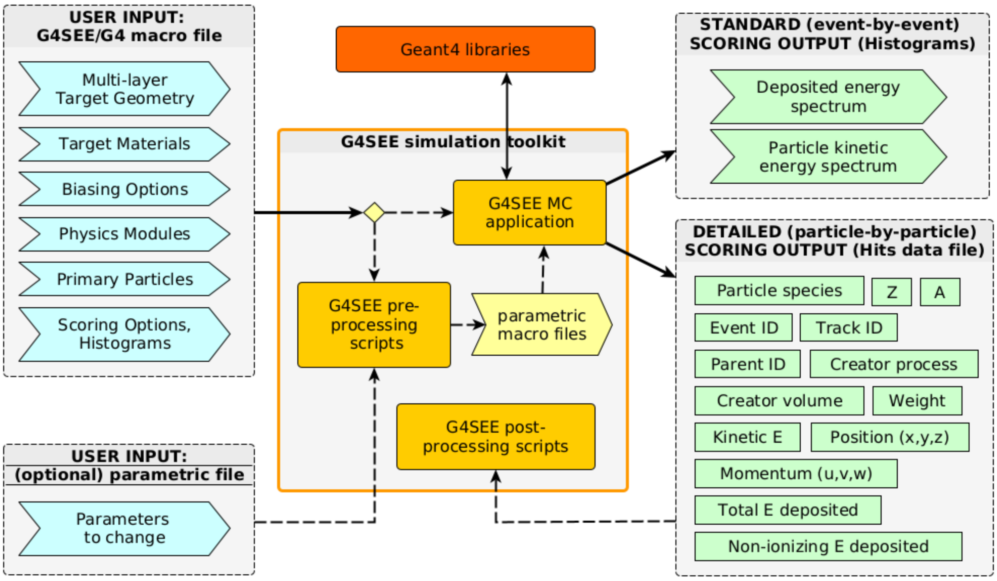


Figure 3.2: G4SEE architecture of version 0.5.1 [46]

different parameter values are defined, and based on this different input macro files are generated. Set of multiple parameters can be changed simultaneously as well. First the geometry, physics, biasing, and particle source can be defined in the macro file in the pre-initialization state of the simulation. They are described in more detail below. [48]

Geometry and materials

The user can define all the necessary materials for a simulation. Arbitrary number of Elements (defined according to the isotope nucleon number and isotope abundance), Materials (any element of the periodic table can be used, density can be set) and also Mixtures can be implemented. All the predefined G4 elements or materials in the G4 database [50] can be used. [48]

The target geometry is placed within a "World" volume and always contains a "Bulk" volume inside which the "Sensitive Volume" ("SV"), where all the quantities of interest are being scored, is placed. For now only one SV can be defined. Additionally an arbitrary number of Back End Of Line (BEOL) layer volumes can be placed on top of the Bulk. Each newly defined BEOL is added on top of the previous one. The top center point of the Bulk volume is always placed at the origin (0,0,0) and position of SV is defined relative to it. By default cuboids are used for every geometry layer, but also cylindrical volumes can be used. In figure 3.3 can be seen an example of G4SEE geometry (side view). [48]

Physics

Users can define an arbitrary physics list, which are build from different predefined G4 physics modules [51] and lists for various particle processes and interactions. The

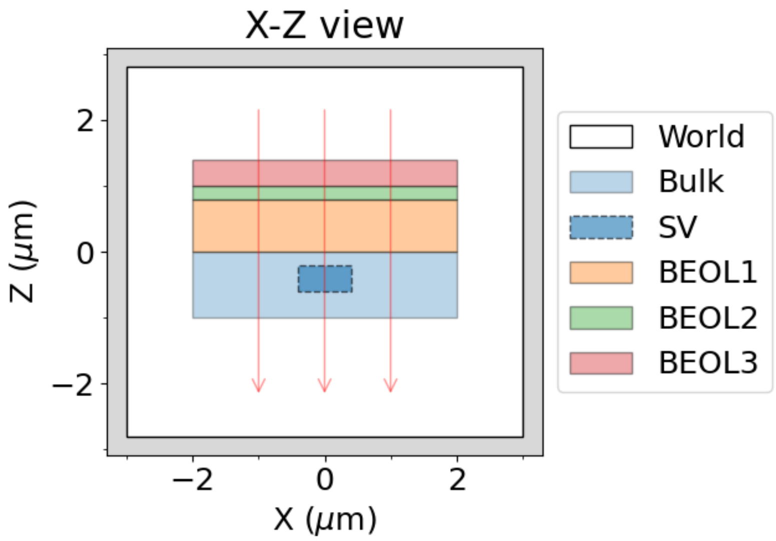


Figure 3.3: Example of G4SEE geometry [46]

running time depends on the chosen physics modules. All particles of the Standard Model are defined and can be used with G4SEE. For electrons, positrons, γ particles and all hadrons, the production range cut can be set. [48]

Users can build their own physics list from various electromagnetic, hadron elastic and inelastic, and ion physics modules, as follows:

- **Electromagnetic physics**: In every simulation run EM module is mandatory, but only one can be used at a time. Default and recommended is `G4EmStandardPhysics_option4`, it includes the most accurate EM physics models. Other EM modules can be also used, but they are not as accurate as the default, or they can be used only in special cases, like `G4EmPenelopePhysics` module, which uses PENELOPE [4] physics. It is developed for low-energy γ, e^- and e^+ from a few hundred eV to about 1 GeV.

- **Hadronic physics**: this module can be further divided into following submodules:

  - **Elastic**: Module for the elastic interactions of hadrons. For example, `G4HadronEasticPhysicsHP` is strongly recommended for neutrons with energy below 20 MeV, and `G4IonElasticPhysics` is recommended for heavy ions.

  - **Inelastic**: Several different models are combined in this module, applicable to different energy ranges. In the most important models belong: Quark-Gluon string model for >15 GeV, Fritiof Parton model for >5 GeV, Binary Cascade model <10 GeV, Bertini Intranuclear Cascade model for <10 GeV, Liège Intranuclear Cascade model for <10 GeV, Precompound/de-excitation model and High Precision (HP) neutron model for neutrons below 20 MeV. For a simulation it is always recommended to try different modules for comparison. They are usually very similar, but in some cases, there could be a discrepancy in the higher energy range of deposited energy distribution.

  - **Ion Inelastic**: These modules are for inelastic ion-ion processes for deuteron, triton, ^{3}He, α and GenericIon projectiles, for example `G4IonPhysics`.

- **Decay physics**: Contains decay processes of unstable particles and nucleons. The default and mandatory module is `G4DecayPhysics`, and the other optional module is `G4RadioactiveDecayPhysics`.

Biasing

In G4SEE a microscopic cross section (XS) biasing of particle interaction is implemented. This type of biasing was described (as interaction forcing) in section 2.1.2. If used, it artificially multiplies the occurrence (microscopic cross section) of a specific interaction by a user-defined factor. This biasing is automatically corrected by particle weights during scoring. It should always be used with caution since the simulation becomes a non-analogue MC simulation and if the biasing would not be used correctly the results could get over-biased, leading to wrong results (not corresponding to reality). In the input macro the user defines for which particle and particle processes the XS biasing should be applied to, and the XS bias weight factor. By default only primary particles are biased, but optionally secondary particles can be biased as well. Biasing can be selectively enabled or disabled in any volume of the target geometry. For example users can bias all the inelastic processes of primary neutrons with factor 1000, but only in SV. [48]

After the definition of above mentioned user settings and parameters, the simulation has to be initialized, which is done by `/run/initialize` command. Then the particle source and scoring of the quantities can be defined in macro file.

General Particle Source

G4 General Particle Source (GPS) [52] macro commands can be used to define primary particles, their attributes and distributions. User can define for example ion properties, particle momentum direction, source location, energy distribution, beam shape and many others. [48]

3.1.2 Scoring

Standard scoring

In version 0.5.1, it is possible to score the total deposited energy E_{dep} event-by-event in SV using the "Standard scoring". Another quantity possible to score is the kinetic energy E_{kin} of a particle. This is implemented to be scored particle-by-particle for SV and it is mandatory to define the particle species in the input macro (any particle species even isotopes can be defined). The particle kinetic energy is scored once for particle entering or produced in the SV. Both deposited energy and kinetic energy, are scored in one dimensional user-defined histograms with logarithmic or linear binning, then these are printed to ASCII histogram files. From an E_{dep} histogram, using the related post-processing script the reverse cumulative sum of deposited energy can be calculated in order to retrieve the SEE cross section (for SEUs in SRAMs, desribed in 2.3.1) as a function of the critical charge. [48]

Detailed scoring

Another type of scoring implemented in G4SEE is the "Detailed scoring". In this feature, each CPU thread (when multi-threading is used) saves all individual particle hits in SV particle-by-particle to a `Hits_t#.csv` output file, providing the user with detailed information about individual particles entering or produced in SV. For e^-, e^+, and γ particles created in one event, their deposited energy is summed in order to reduce verbosity and size of detailed scoring output file. The user can also set kinetic energy threshold, below which these particles are grouped, and above it they are scored individually. Default threshold value for e$^-$ and e$^+$ is 10 keV, and for γ it is 100 keV. Additionally, e$^-$ and e$^+$ grouping by particle ID of their closest non-electron ancestor within events is possible. This makes it possible to categorize the main secondary products or nuclear reactions that occurred in an event. Users can save many useful quantities to the CSV file, for example deposited energy by a particle, kinetic energy, atomic number, particle's position, momentum and more. All the user-selected quantities for all individual particles are printed in one line. Since a large amount of information is printed in the output CSV file, it can quickly grow very large in file size. It is important to keep this in mind before running the simulation, otherwise, the disk space may be exceeded. Table 3.1 shows an example of how the data in an output file looks like, and figure 3.4 [46] demonstrates a final analysis result a user can get by post-processing and plotting detailed scoring output file with deposited energy per particle. [48]

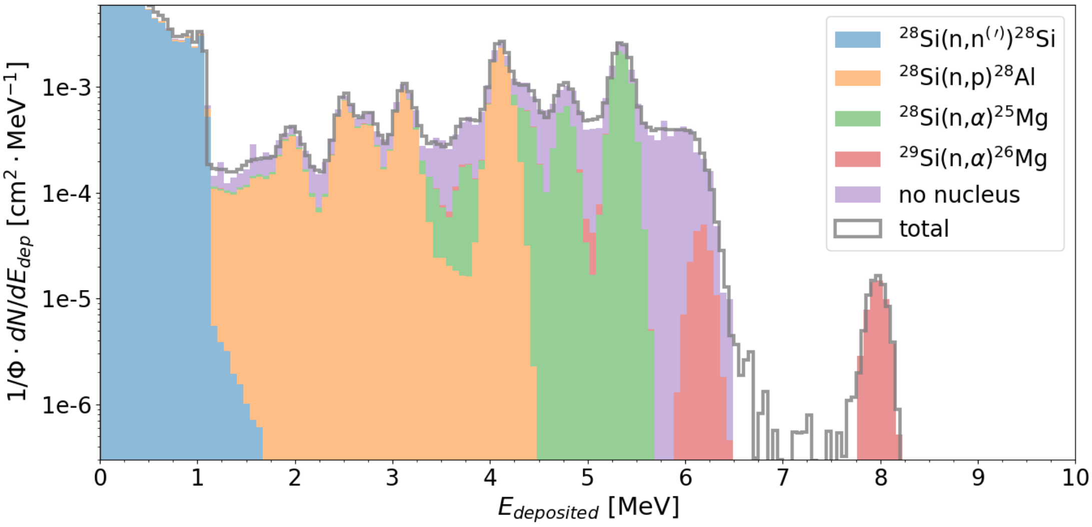


Figure 3.4: G4SEE final analysis result after post-processing and plotting detailed scoring output file with E_{dep} per particle data. Simulation was run for Si diode irradiated by 8 MeV neutrons [46]

Table 3.1: Example of output `Hits_t0.csv` file created for 14.04 MeV neutron induced elastic and inelastic interactions in diamond (carbon atoms). In this example only some columns (quantities) and chosen events (rows) are shown.

event	particle	Z	A	track	parent	E_kin	pos_x	mom_x	process	volume	E_dep
1	neutron	0	1	1	0	1.3972e+01	-2.11e+00	-1.61e+00	primary	Air2	0
2	neutron	0	1	1	0	1.4006e+01	-1.05e+00	-8.04e-01	primary	Air2	0
3	neutron	0	1	1	0	1.3822e+01	4.49e-01	3.40e-01	primary	Air2	0
1336	neutron	0	1	2	1	9.0059e+00	8.18e-02	1.07e+02	b(neutronInelastic)	Sensitive	0
1336	C13	6	13	3	1	1.0649e+00	8.18e-02	-1.10e+02	b(neutronInelastic)	Sensitive	1.0649e+00
1336	gamma	0	0	4	1	1.6930e-01	8.18e-02	-1.55e-01	b(neutronInelastic)	Sensitive	0
1336	gamma	0	0	5	1	3.6845e+00	8.18e-02	2.67e+00	b(neutronInelastic)	Sensitive	0
1339	neutron	0	1	1	0	1.4011e+01	-1.47e+00	-1.13e+00	primary	Air2	0
2144	neutron	0	1	1	0	1.4265e+01	-4.94e-01	-3.80e-01	primary	Air2	0
2144	^{12}C	6	12	2	1	2.8354e-02	-4.94e-01	1.25e+01	hadElastic	Sensitive	2.8354e-02
2493	neutron	0	1	1	0	1.3827e+01	2.10e+00	1.60e+00	primary	Air2	0
2493	neutron	0	1	2	1	6.9765e+00	2.11e+00	9.00e+01	b(neutronInelastic)	Sensitive	0
2493	^{12}C	6	12	3	1	2.3469e+00	2.11e+00	-8.72e+01	b(neutronInelastic)	Sensitive	2.3469e+00
2493	gamma	0	0	4	1	4.4390e+00	2.11e+00	-1.30e+00	b(neutronInelastic)	Sensitive	0
3306	neutron	0	1	1	0	1.4086e+01	8.75e-01	6.70e-01	primary	Air2	0
3306	neutron	0	1	2	1	1.3967e+00	8.75e-01	5.93e+00	b(neutronInelastic)	Sensitive	0
3306	alpha	2	4	3	1	3.4478e+00	8.75e-01	1.06e+02	b(neutronInelastic)	Sensitive	3.4478e+00
3306	alpha	2	4	4	1	1.0584e+00	8.75e-01	-6.88e+01	b(neutronInelastic)	Sensitive	1.0584e+00
3306	alpha	2	4	5	1	9.0927e-01	8.75e-01	-4.29e+01	b(neutronInelastic)	Sensitive	9.0927e-01
5108	neutron	0	1	1	0	1.4176e+01	2.82e-01	2.17e-01	primary	Air2	0
5108	alpha	2	4	2	1	6.9718e+00	2.82e-01	1.44e+02	b(neutronInelastic)	Sensitive	6.9718e+00
5108	Be9	4	9	3	1	1.4938e+00	2.82e-01	-1.43e+02	b(neutronInelastic)	Sensitive	1.4938e+00
5109	neutron	0	1	1	0	1.3992e+01	-2.24e+00	-1.71e+00	primary	Air2	0
99971	neutron	0	1	1	0	1.4259e+01	5.71e-01	4.40e-01	primary	Air2	0
99971	alpha	2	4	2	1	3.4524e+00	5.71e-01	1.58e+02	b(neutronInelastic)	Sensitive	3.4524e+00
99971	^{12}C	6	12	6	3	1.4578e-01	1.19e-01	-2.82e+01	hadElastic	Sensitive	1.4578e-01
99971	neutron	0	1	3	1	1.5794e+00	5.71e-01	-4.58e+01	b(neutronInelastic)	Sensitive	0
99971	alpha	2	4	4	1	6.1543e-01	5.71e-01	-4.56e+01	b(neutronInelastic)	Sensitive	6.1543e-01
99971	alpha	2	4	5	1	1.3376e+00	5.71e-01	-6.65e+01	b(neutronInelastic)	Sensitive	1.3376e+00

After all of the above mentioned items defined in the input file, the run starts with last command `run/beamOn 1000`. By this command 1000 primary particles will be generated. Number of primary particles should be used by users to normalize the output data. The higher the number of primary particles, the better the statistics and lower variance (std. deviation) of each histogram bin is obtained (as described in section 2.1.2). The above described input parameters are shown in appendix A, where example G4SEE input macro of diamond detector example is shown.

3.1.3 Post-processing

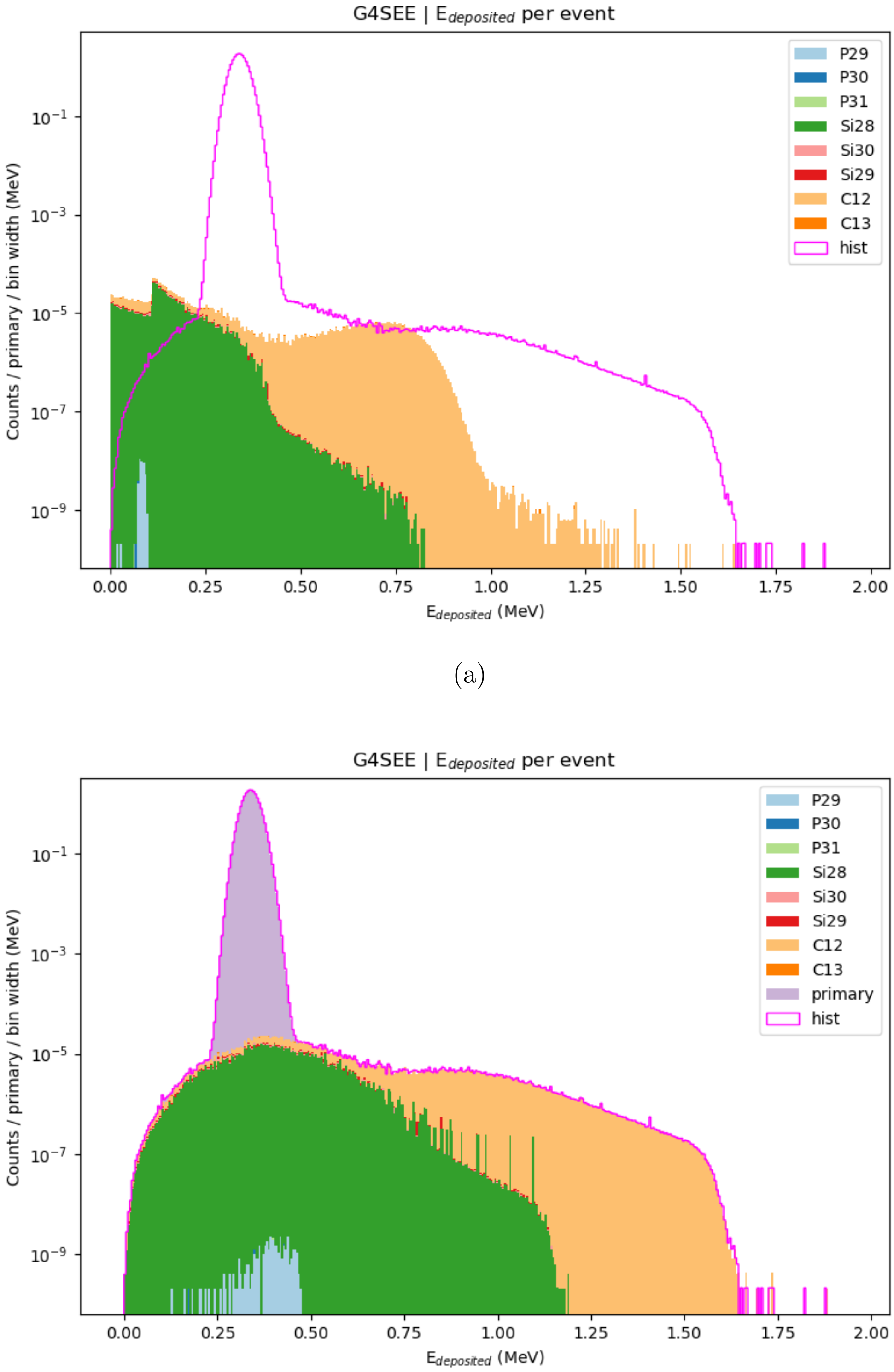
G4SEE toolkit includes pre-processing and post-processing Python 3 scripts. The G4-based application is capable to run on any number of CPU threads at the same time (multi-threaded), as well as run in multi-processing mode by submitting any number of parallel jobs to independent computer cluster nodes and monitor or delete these jobs. Output histogram files can be merged recursively into a single histogram file by a post-processing script. With a post-processing script, users can also visualize geometry, plot histograms and also the SEEs cross section can be computed and plotted.

For the processing of the CSV output file created by the detailed scoring, a post-processing script is also available. This script is capable to separate the information from the output CSV file into separate CSV files containing information only about secondary particles, interaction types, creator volume of secondary particles, or only about the deposited energy. Also with this script is possible to plot the histograms of deposited energy by each particle species, depositing energy in SV. Up to the version 0.5.1, this script was used only for neutron-induced interactions, and has not yet been shared as part of the toolkit. These data corresponded well with the standard E_{dep} scoring and measured data, see in paper [46]. While processing data for proton-induced interactions, the processed data in histograms were significantly different from the standard scoring histogram. They were shifted to lower energies and also some of the deposited energy was missing from the data, see figure 3.5a. These particular results will be discussed more in chapter 5.2. The problem was that the energy deposited directly by primary particles was not taken into account. While this approach is in the case of neutrons correct, since neutrons are ionizing only indirectly, for directly ionizing charged particles it was a problem to be fixed.

To get correct results I made following changes in the script. Firstly, I included the energy deposited by primary particle (direct ionization of the material) to the total energy deposited in event, and made histogram. While processing the data the heavy ions (with $Z>2$) are being differentiated. If there are secondaries with $Z\leq2$, these are all included in the "None" group. In the script, the deposited energy is summed over the event, when secondary particles are created. This means, that the energy deposited by all secondaries is summed and assigned to the main secondary product of the event (for example ^{28}Si), and then added to the total deposited energy histogram of that product. The post-processing script in the previous version did not include the energy deposited by primary particle in the summing per event, which is not a correct approach for charged particles. Based on this I changed the script such as the energy of primary particle is also added to the sum of energy per event. The result after the mentioned changes can be seen in figure 3.5b.

3.1.4 Visualization

In the version 0.5.1 the G4 visualization feature was not enabled, tested and used. Therefore, for the upcoming version, I have enabled the G4 visualization of target geometry and particle tracks, and also wrote a documentation with explanation of the G4 user macro commands for visualization. In figure 3.6, an example can be seen, how the visualization looks for G4SEE SRAM example simulation. Using the commands explained in the documentation, users can customize the visualization according to their needs.



(a)

(b)

Figure 3.5: Detailed scoring E_{dep} histograms for 3 MeV protons interacting with SiC: a) wrong histograms before any change in the post-processing script, b) correct histograms after added direct E_{dep} of primary particles to total E_{dep} per event. Each histogram represent deposited energy by different interaction, the legend shows the main secondary particles produced in these interactions. "Hist" is the histogram of deposited energy obtained by standard scoring.

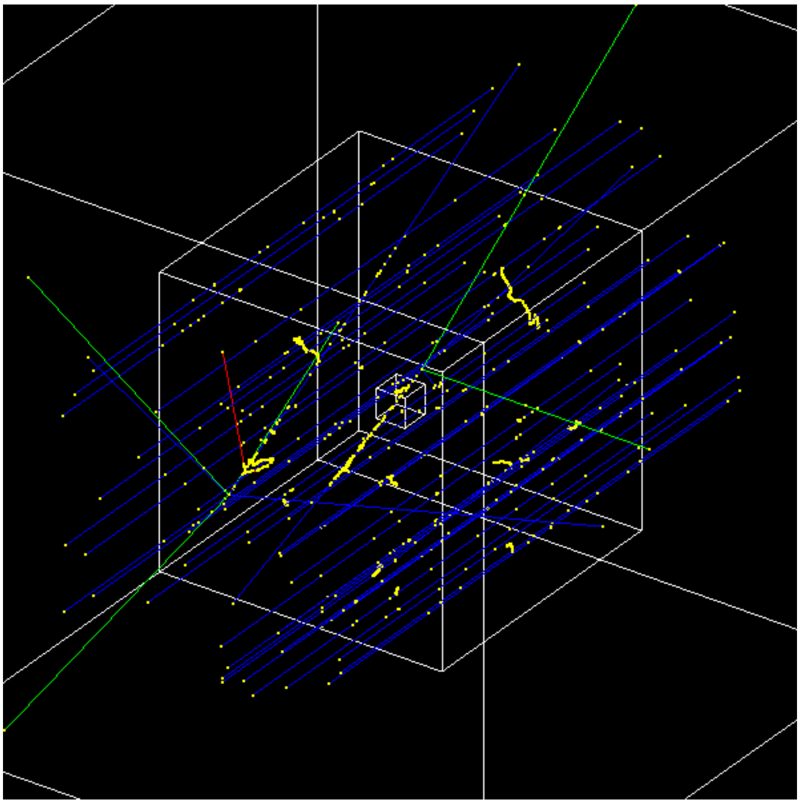


Figure 3.6: Visualized G4SEE example of 600 nm-thick SRAM memory, placed in 1 μm-thick Bulk, irradiated with 10 MeV protons. On top of Bulk 1.6 μm-thick SiO_2, Cu and Si_3N_4 mixture layer and 0.8 μm-thick SiO_2 layer are placed. The proton beam (blue tracks) is centered on the Bulk center and is 10 um away from it in the z-axis direction. Yellow dots represent step points.

3.1.5 Application structure

Since G4SEE is based on Geant4, the core application is an object-oriented C++ application build from several G4SEE classes, which include and use classes of the G4 C++ library implementing the essential, smaller components of the MC simulation. Main G4SEE classes will be described in more details in the following paragraphs.

The geometry construction is handled by the `DeviceConstruction` class. User commands for defining the dimensions and materials of the volumes are provided by the `DeviceMessenger` class. Similarly, the available physics processes and related user commands are defined in `PhysicsList` and `PhysicsListMessenger` classes.

In the `BiasingXS` and `BiasingMultiParticlesXS` classes, the cross section (XS) biasing is implemented, and related user commands are also implemented in its messenger class. The `BiasingMultiParticlesXS` class is responsible for applying biasing to several particle species. In the source, first the collection of processes under biasing starts. Subsequently, the required biasing is created, and needed operations are associated with the processes. The adjustment of XS fundamentally involves change of process occurrence, the objects are selected in the method

```
BiasingXS::ProposeOccurenceBiasingOperation(const G4Track* track,
 ↪  const G4BiasingProcessInterface* callingProcess)
```

of the `BiasingXS` operator.

The primary particles can be any particle species, impacting the defined geometry at user-defined angles and distances. This is defined in `PrimaryGeneratorAction` and the `G4GeneralParticleSource` class.

All user action classes (`PrimaryGeneratorAction`, `RunAction`, `EventAction`, `TrackingAction` and `SteppingAction`) are instantiated and registered to G4 kernell by the `ActionInitialization` class. In single-threaded mode, the action classes are initialized once via the invocation of the `ActionInitialization::Build()` method. In multi-threaded mode, the same method is invoked for each thread worker. All user action classes are defined thread-local. A run consists of set of events (defined in `RunAction`/`EventAction` class).

The `ParticleHit` class records step-by-step all the necessary information for simulating and analyzing quantities scored in the SV. These quantities include eventID, TrackID, ParentID, particle name, E_{dep}, N_{dep}, position in volume, particles momentum, E$_{kin}$, creator physics process, weight, Z, A and the volume where the particle was created.

The deposited energy and particle kinetic energy values are recorded via the `SensitiveVolume` class. Physics quantities are stored in `ParticleHit` through `SensitiveVolume` object associated with SV in the `ConstructSDandField()` method of `DeviceConstruction`.

The scored physics quantities saved in histogram files are created in `Scoring` class, and can be defined by user input commands also defined in related messenger class.

3.2 User support

A significant advantage of G4SEE is the fact that it is open-source and freely available for everyone. On the G4SEE website [47] introduction to G4SEE and all the useful links can be found. Among these links, one leads to the G4SEE online documentation [48], providing comprehensive information necessary for utilizing G4SEE. This documentation covers everything from download and installation to how to define geometry, scoring, and others.

The documentation includes a step-by-step tutorial on downloading G4SEE for local installation. Alternatively, users can opt to run G4SEE through Docker. The G4SEE Docker image is a Debian-based Docker image, with G4 pre-installed and G4SEE pre-compiled inside, eliminating the need for users to install and manage dependencies locally, which can consume significant amount of time and disk space.

In the freely accessible G4SEE GitLab repository [49], users can find the source code, and illustrative G4SEE simulation examples and tutorials as well, which can serve as templates for beginner users' simulations.

As many open-source projects, G4SEE maintains a dedicated User Forum [53], where users can seek assistance and guidance. This forum is primarily managed by the toolkit developers. Additionally, as part of this work, support was provided for two students from INSA Toulouse (supervised by Gabriel Duran (ALTER) and Marine Ruffenach (CNES)). The students have engaged in G4SEE simulations and authored tutorials on the installation process of G4SEE.

3.3 G4SEE applications

G4SEE is developed according to the needs of its user community. Collaborative efforts with various individuals and organizations are described in chapter 5, demonstrating the adaptability and relevance of the toolkit.

Members of R2E project at CERN actively use G4SEE in their research. In some of these applications, G4SEE is used for simulating neutron or proton-induced SEUs in SRAM memories (like ISSI SRAM), calculating their SEU cross sections and comparing simulation results with experimental data [54, 55, 56].

The focus remains on implementing and validating new features relevant for SEEs simulations for electronics. Implementing new features was one of the main tasks of this work, these are described in the following chapter. In the near future, progressive integration of G4SEE features into the FLUKA v5 will be made. FLUKA v5 general-purpose MC simulation tool is currently under development and beta testing by FLUKA.CERN Collaboration. [12, 14, 13] It is going to be a G4-based tool, but with existing and widely used FLUKA v4 physics models and features, such as geometry definition, scoring, biasing, etc. So all the G4SEE features (including the ones which were implemented in this work) will be integrated into FLUKA v5 and used by many FLUKA users worldwide in the future.

Chapter 4

G4SEE toolkit development

Three new features have been implemented as a part of this work: Dose scoring, Linear Energy Transfer scoring, and Energy Deposition per Interaction scoring. These enhancements will be described in details in following sections. The extended architecture of G4SEE including these new scoring features is shown in figure 4.1.

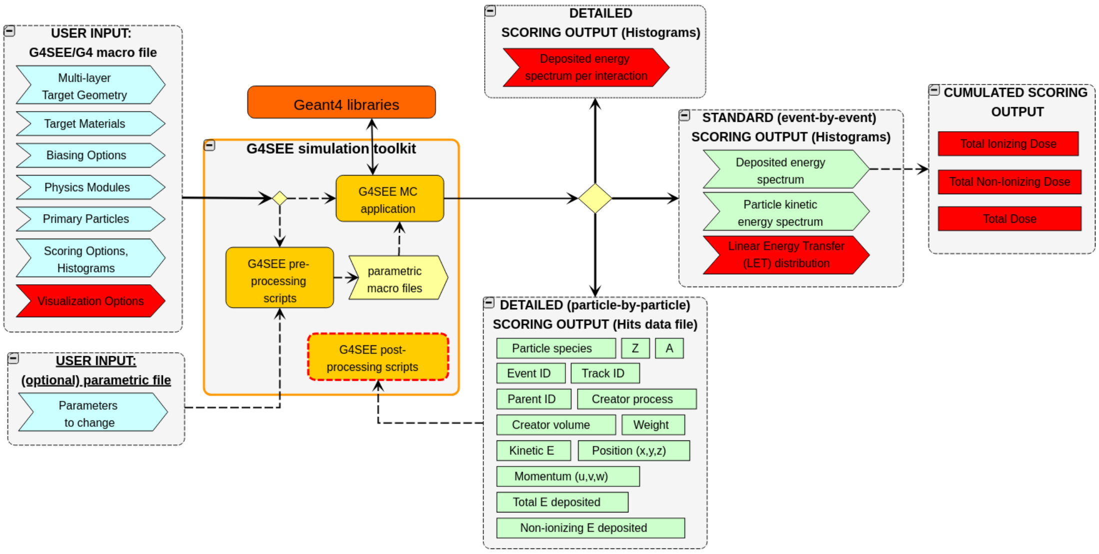


Figure 4.1: Extended architecture of G4SEE with newly implemented scoring features highlighted in red

4.1 Dose scoring

This thesis mainly focus on SEEs and their impact on electronic devices, in which the deposited energy and LET are the main quantities of interest. But also cumulative effects pose a risk in device operation in radiation environments, for which the TID and TNID doses are the relevant physics quantity of interest. Therefore, I have implemented dose scoring as a new scoring feature in G4SEE.

Since the dose represents the fraction of deposited energy in a volume and the volume's mass, the idea was to accumulate the deposited energy over the simulation run and divide it by the SV mass. The deposited energy in G4SEE is collected step-by-step in SV and accumulated event-by-event, both in `SensitiveVolume` class. These values are then summed at the end of simulation run in the `RunAction` class. The total deposited dose is computed at `RunAction::EndOfRunAction()`, where the SV mass is also calculated using the `GetMass()` function of the `G4LogicalVolume` class [57].

Not only the total dose, but also the Total Ionizing Dose (TID) and Total Non-Ionizing Dose (TNID) can be computed similarly. To get the total deposited energy, the `GetTotalEnergyDeposit()` function is used, it gives the deposited energy value along one step of particle track. These values are then summed at the end of each event (E_{dep} value). Similarly there is `GetNonIonizingEnergyDeposit()` function. Its sum at the end of each event is the N_{dep} value. Both of these functions are from `G4Step` class [58]. By subtracting these values, the energy deposited via ionization only is obtained.

To enable the dose scoring in the input macro file, the command `/SEE/run/doseScoring True` needs to be added by the user. This generates a `dep_dose_t#.out` file, where the three scored dose values (total, ionizing, and non-ionizing) are printed, see such output below. These values are not normalized, but computed for all the number of primary particles (or events) of the simulation run. The main parts of implementation of dose scoring in the G4SEE source code can be seen in Appendix B. Below an example dose scoring output file can be seen after a simulation with 10^5 3 MeV primary protons to score dose values in a 11 μm-thick SiC volume:

```
Total deposited dose: 4676.62 Gy
Total non-ionizing deposited dose: 467.873 milliGy
Total ionizing deposited dose: 4676.16 Gy
Note: The above calculated doses correspond to 100000 primary particles
```

4.2 Linear Energy Transfer scoring

Another new feature implemented in this work is the LET scoring. Compared to dose scoring, its implementation was more complex. LET can be defined as a ratio of energy lost by the charged particle due to electronic interactions in material when passing along a short trajectory, and this trajectory. Units of LET are typically (keV/μm), or, when normalized by material density, (MeV·cm^2/mg). [59] The challenge lay in determining how to compute the two quantities for LET, the energy deposition and the related length, leading to several possible approaches. Ultimately, I have implemented three different LET scoring options, which will be further described and compared.

The **first scoring option** (named after `G4EmCalculator`) computes LET using the G4 `ComputeDEDX()` function from the `G4EmCalculator` class [60]. This function requires as input values the particle species, for which the LET is computed, particle kinetic energy, the interaction process, the material involved in the interaction, and also the particle energy cut can be set. These values are obtained in `SensitiveVolume::ProcessHits()` function for every particle species step. Based on these inputs, the function then returns the corresponding LET value from G4 physics tables.

The **second option** (named after `G4Event`) computes LET by dividing the total deposited energy per event in SV (E_{dep}) by the track length of primary particle in SV. The track length is computed as the distance between the point where the particle entered the SV and the point where it stopped in the volume or left it. The entry point is obtained when the particle's pre-step point crosses the SV boundary with the `GetPosition()` function from `G4Track` class [61]. The end point is also obtained with the `GetPosition()` function but from the particle track rather than a step point. At the end of an event in the `SensitiveVolume::EndOfEvent()` function the total deposited energy per event is divided by the total track length of primary particle to compute the LET. This approach allows for comparison with real measurement data, as detectors can measure the deposited energy by primary particle along with the secondaries produced and estimate the particle track. In this option it is possible to compute the stochastic LET for the whole event (practically from experimental point of view for the primary particle) event-by-event.

In the **third option** (named after `G4Step`), LET is also calculated by dividing the deposited energy in SV by the particle track length in SV. However, unlike the second option, the deposited energy and corresponding track length are obtained per each particle species step. LET is computed in the `SensitiveVolume::ProcessHits()` function, where the particle's deposited energy per step is divided by the step length. The LET value is then added to the hits and based on the particle species defined in the input macro, the LET value for given particle is added to the histogram at the end of an event. The main difference here, compared to the second option, is that LET is computed per each particle in an event separately, whereas in second option the LET is computed for the event. While LET scored using this approach cannot be measured experimentally, it provides a more detailed insight into what is happening in the SV, thanks to the information about secondary particles' LET.

The difference between the second and third options is illustrated in figure 4.2 and the implemented source code is shown in Appendix C.

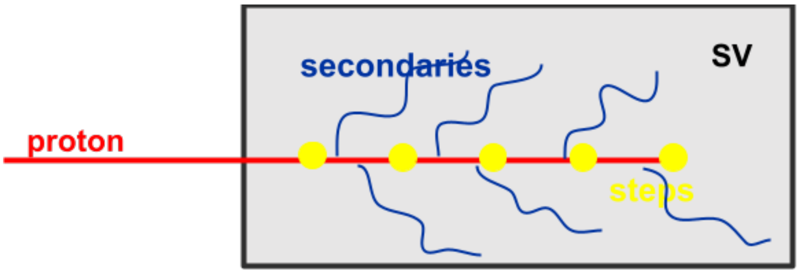


Figure 4.2: Simplified illustration of proton interacting in SV. The LET computed per event takes into account the energy deposited by primary proton and also its secondaries, on the contrary the LET computed per step computes it only for the proton for each step, not taking into account the secondaries.

The comparison of the different LET scoring is shown in figure 4.3. The distributions differ, since the methods of LET determination vary. The difference between LET computed per G4Event and per G4Step distributions is caused by the fact that LET per G4Step the is computed only for protons interacting in SV, while per G4Event, the LET is computed for every event when proton interacted in SV. In this case, the energy deposition can be done only via ionization or also secondary particles can be created whose deposited energy is also included in the calculation.

Similarly as the E_{dep} or E_{kin} standard scoring, the LET scoring can be defined in the input file as follows:

```
/SEE/scoring/addScoring    hist_ID  LET_type  particle_species
/SEE/scoring/setHistogram  scale    min_LET   max_LET   bins
```

In the input macro it can look like this:

```
/SEE/scoring/addScoring    1      LET1        proton
/SEE/scoring/setHistogram  lin    0 keV/um    100 keV/um   100
/SEE/scoring/addScoring    2      LET2
/SEE/scoring/setHistogram  lin    0 keV/um    100 keV/um   100
/SEE/scoring/addSoring     3      LET3        proton
/SEE/scoring/setHistogram  lin    0 keV/um    100 keV/um   100
```

The histograms `.out` files for selected particles are created.

4.3 Energy Deposition per Interaction scoring

In version 0.5.1, both the event-by-event standard scoring and the particle-by-particle detailed scoring features are available and used by many users to obtain energy deposition distributions (see section 3.1.2). The detailed scoring feature is very beneficial when users seeking a deeper understanding of the simulated process, for example to find out which particle interactions (nuclear reactions) contributes to the total energy deposition distribution. This is already possible with post-processing of detailed scoring output as shown in section 3.1.2. However, the fast growth of the output CSV file in size implies limitations on the number of events (primary particles) simulated, leading to lower event statistics and higher variances. Also the need of post-processing (see section 3.1.3) of such large data files represents a disadvantage that can be critical in some use cases.

These limitations became a problem when I was running the simulation of diamond detector (described more in chapter 5.1). Higher event statistics was needed in this case to see rare events, but the CSV file was too large (orders of GB), making it difficult to achieve good enough statistics for rare events. This is why I have implemented new type of deposited energy per interaction scoring, which scores the deposited energy particle-by-particle (same way as detailed scoring), allowing users to see types of secondary products created in the SV, while output files are only event-by-event histogram files (same as standard scoring histograms). Histogram files require very small disk space, and their file names contain the list of secondary

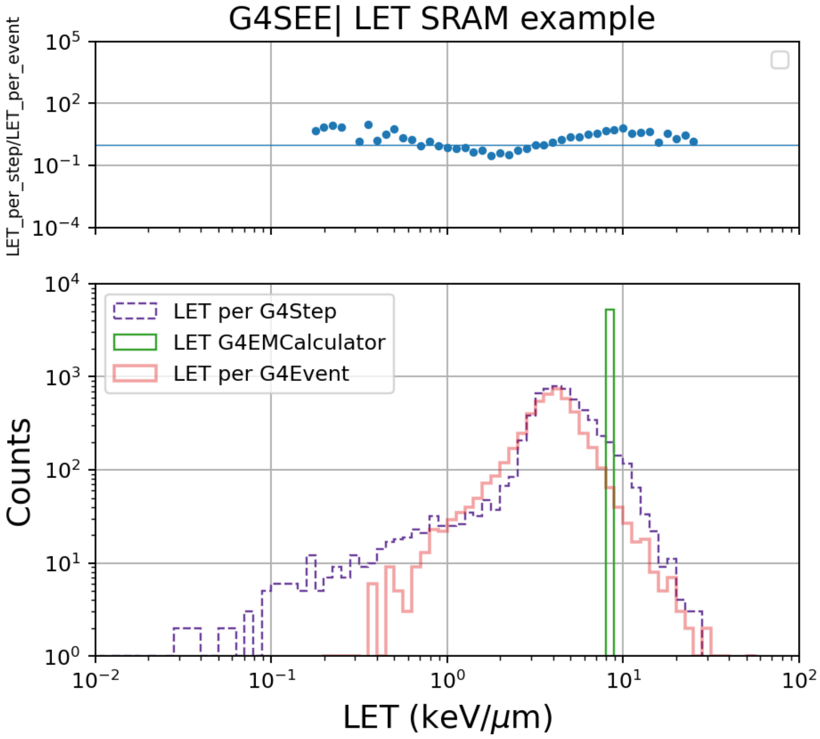
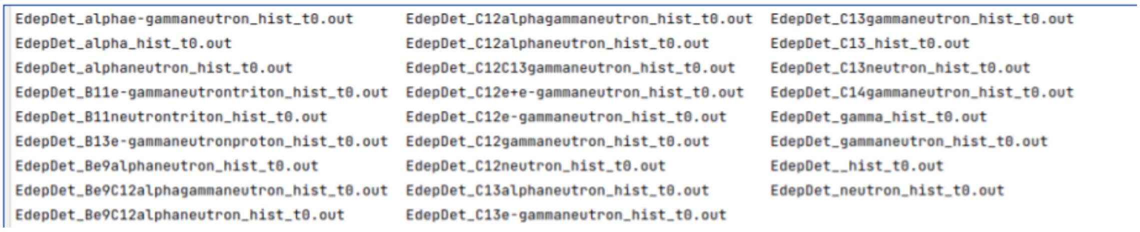


Figure 4.3: Comparison of different types of LET scoring implemented. This example is made for 0.6 μm-thick SRAM memory irradiated by 10 MeV proton beam.

particles (interaction products). Example the output files is shown in the figure 4.4. Compared to the existing detailed scoring, this new scoring method provides less, but compared to the standard E$_{dep}$ scoring it provides more information about the interaction processes and their contribution to total deposited energy.



```
EdepDet_alphae-gammaneutron_hist_t0.out        EdepDet_C12alphagammaneutron_hist_t0.out  EdepDet_C13gammaneutron_hist_t0.out
EdepDet_alpha_hist_t0.out                      EdepDet_C12alphaneutron_hist_t0.out       EdepDet_C13_hist_t0.out
EdepDet_alphaneutron_hist_t0.out               EdepDet_C12C13gammaneutron_hist_t0.out    EdepDet_C13neutron_hist_t0.out
EdepDet_B11e-gammaneutrontriton_hist_t0.out    EdepDet_C12e+e-gammaneutron_hist_t0.out   EdepDet_C14gammaneutron_hist_t0.out
EdepDet_B11neutrontriton_hist_t0.out           EdepDet_C12e-gammaneutron_hist_t0.out     EdepDet_gamma_hist_t0.out
EdepDet_B13e-gammaneutronproton_hist_t0.out    EdepDet_C12gammaneutron_hist_t0.out       EdepDet_gammaneutron_hist_t0.out
EdepDet_Be9alphaneutron_hist_t0.out            EdepDet_C12neutron_hist_t0.out            EdepDet__hist_t0.out
EdepDet_Be9C12alphagammaneutron_hist_t0.out    EdepDet_C13alphaneutron_hist_t0.out       EdepDet_neutron_hist_t0.out
EdepDet_Be9C12alphaneutron_hist_t0.out         EdepDet_C13e-gammaneutron_hist_t0.out
```

Figure 4.4: List of created output files for energy deposition per interaction scoring

This scoring is implemented in the `SensitiveVolume` class. For each track, a particle name is obtained and the names of the particles created in the event are accumulated at the end of the event. This creates a specific combination of particle names. The deposited energy is obtained per particle step and accumulated at the end of event (same as for the before mentioned deposited energy computations). Each particle combination created is assigned a corresponding deposited energy per event. This deposited energy is then added to the histogram.

After the simulation run, histograms are written to histogram `.out` files, each containing E_{dep} data for different inelastic particle interactions. Some of the interaction product list differ only in the presence or absence of gamma or electron/positron particles. In this case the particle set strings are different and multiple histogram files are created, but from the interaction point of view they belong to the same. As a temporary fix, I have created a Python 3 post-processing script that merges these files into one. In the future, this merging will be included in the G4SEE source code and performed automatically during scoring.

Comparison of the output histograms of the new scoring and standard energy deposition scoring is shown in figure 4.5. The data correspond well to each other, indicating that this new type of scoring is implemented correctly and can be used.

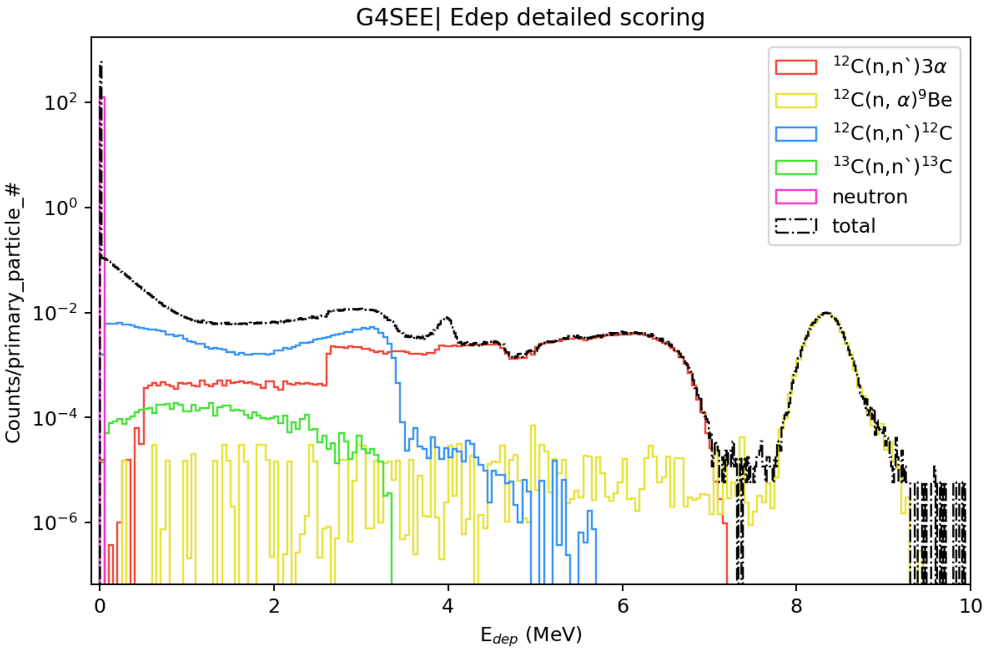


Figure 4.5: Output histograms of the newly implemented E_{dep} per interaction scoring. This example shows deposited energy by products created by 14.04 MeV neutron induced interactions in diamond.The total histogram shows deposited energy obtained with standard scoring.

4.4 Contribution summary

To summarize my contributions to G4SEE development. I enabled the visualization of the simulation run and geometry and wrote documentation for it. The post-processing script for the detailed scoring was improved, namely to include also deposited energy by directly ionizing primary particles in histograms. Another minor change made was making the biasing parameter optional in volumes, so that it is not always necessary to specify that the biasing is disabled (set to false).

I implemented three new scoring features. The first one is scoring of the total dose, which is divided into total ionizing (TID) and total non-ionizing (TNID) doses. Second feature is LET scoring. With this scoring users can score three different LET values, which are implemented differently and have their advantages and disadvantages. Last implemented scoring feature is energy deposition per interaction scoring with standard histogram output. Compared to the detailed scoring already implemented, it is possible to get event-by-event deposited energy per different nuclear reactions, but the disc space and also computational time are saved, and statistics of rare events are improved.

Chapter 5

G4SEE validation

A necessary part of a toolkit development is its validation. As it was mentioned, G4SEE was validated by measurements of inelastic energy deposition events of mono-energetic neutrons in the 1.2 MeV - 17 MeV energy range by a silicon diode [46], but further validations are constantly ongoing. In this chapter, new validation studies of G4SEE performed by me are described.

5.1 Diamond detector simulations

In collaboration with Carlo Cazzaniga (STFC UKRI), Michela Paoletti (University of Milano-Bicocca) and Davide Rigamonti (CNR ISTP), who performed diamond detector characterization test with 2.5 MeV and 14.04 MeV neutrons, a MC simulation study for the neutron test campaign have been performed. In following paragraphs a short overview about diamond detectors and the experiment is given, and then simulation results are presented and compared with experimental data.

5.1.1 Diamond detector

Earlier, natural diamond crystals have been used as detectors, but they were facing big disadvantages, such as large variations of impurities and defects in diamond matrix, making them impractical to use. That is why the production of artificial diamond detectors started. There are many different manufacturing techniques, for example chemical vapor deposition (CVD). With this technique the detector thickness, volume and electronic grade quality can be well controlled. [62]

The diamond detector has advantages like ability to differentiate neutrons from γ rays, radiation and temperature hardness, sensitivity to fast neutrons, high electron-hole mobility and low leakage current. They are a first choice detectors for measurements with 5.7 MeV or higher energetic neutrons. This is due to the production of α particles and ^{9}Be ions, creating an easily recognizable sharp peak in the measured spectrum, see figure 5.1. [62]

Diamond detectors are used in many different applications, for example boron-neutron capture therapy, microdosimetry, beam monitoring, and high-resolution

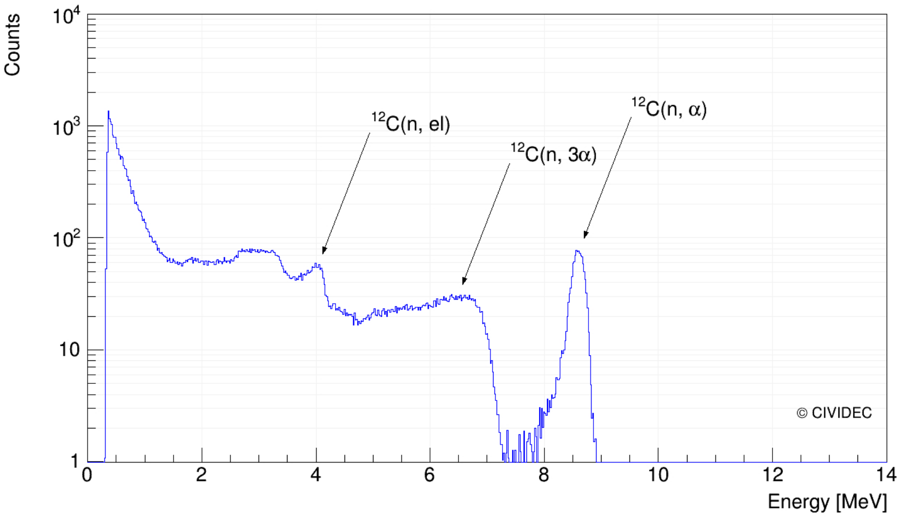


Figure 5.1: Example of 14.3 MeV neutron spectra for sCVD diamond detector. The ^{12}C(n,α) peak correspond to the ^{9}Be. [63]

neutron spectroscopy. [62] The advancements in crystal quality manufacture in recent years enabled their successful utilization in neutron spectroscopy measurements, both at spallation sources and fusion experimental devices. [64] In figure 5.2 is shown an image of diamond detector used in the experimental measurements described in following chapter.

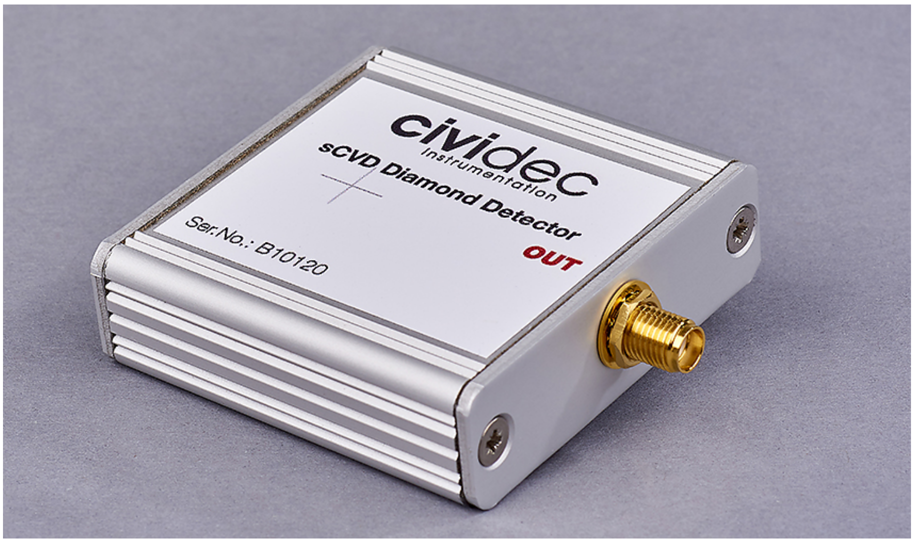


Figure 5.2: sCVD diamond detector manufactured by CIVIDEC [65]

5.1.2 Experimental measurement

The experimental test campaign was performed at Physikalisch-Technische Bundesanstalt (PTB) national metrology institute in Germany by Carlo Cazzaniga, Michela Paoletti and Davide Rigamonti. During the test campaign measurements with multiple different detectors and neutron energies were carried out. Purpose of the test was to calibrate detectors according to the PTB standard, which will be/are installed at the Neutron Irradiation Laboratory for Electronics (NILE) facility to characterise the radiation field and for dosimetry. [66]

In this chapter it will be focused only on the diamond detector measurements (other detectors tested were for example 7CLYC ands EJ-276 scintillators), since those are the ones the G4SEE simulations were performed for. At NILE facility two compact generators of 14 MeV and 2.5 MeV neutrons are installed. The neutrons are produced by deuterium-deuterium and deuterium-tritium fusion reactions. The role of diamond detector is to measure with precision the neutron energy as a function of the emission angle and the spectral broadening of the neutron peak - high-resolution neutron spectroscopy. [66]

The 2.5 MeV neutrons are detected by the elastic scattering on ^{12}C nuclei, which creates an edge at the maximum energy allowed. The position and the broadening of this edge is analyzed by spectroscopy. For the 14 MeV neutrons the peak created in (n,α) interactions is used for the spectroscopy. Both of these spectra are shown in result section 5.1.4. [64]

The experimental test were performed with diamond detector manufactured by CIVIDEC instrumentation [65]. The crystal dimensions were 4.5 mm×4.5 mm×0.5 mm, above the crystal the gold metallization layer was placed, with assumed thickness of 250 nm. This all was placed in the aluminium box of 55 mm×55 mm×16 mm. The device under test (DUT) was placed 20 cm from the target. For the 14.04 MeV neutrons with FWHM=435 keV the DUT was placed perpendicular to the beam direction. This is shown in figure 5.3 for better illustration. For the 2.5 MeV neutrons with FWHM=103 keV the DUT was placed in the beam direction. For the back scattering corrections the shadow cones were used.

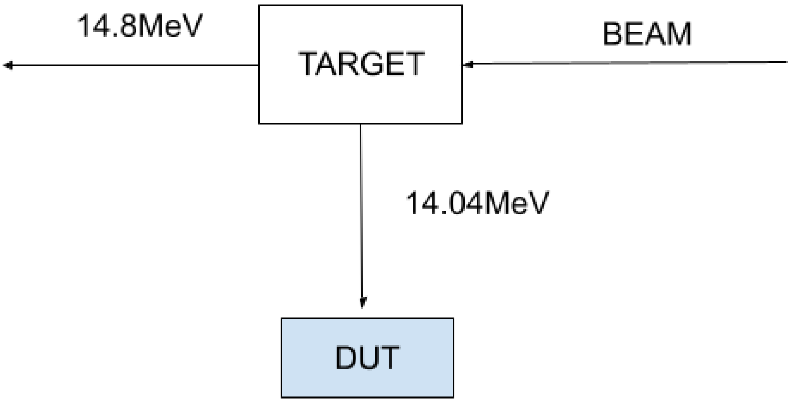


Figure 5.3: Placement of the DUT for the 14.04 MeV neutrons source

5.1.3 Simulations

The geometry was defined, according to the description in the previous section, and is shown in figure 5.4. The diamond material density was set to $\rho = 3.52\ g/cm^3$. Following characteristics of the source were implemented: Gaussian beam energy distribution, point source distribution type, and the beam dimensions were set to cover the aluminium box. For the 14.04 MeV case, the neutron inelastic interaction biasing was applied. When the biasing was applied on the elastic interactions, the data became over-biased and did not correspond with the experimental results. For the 2.5 MeV case, no biasing was applied.

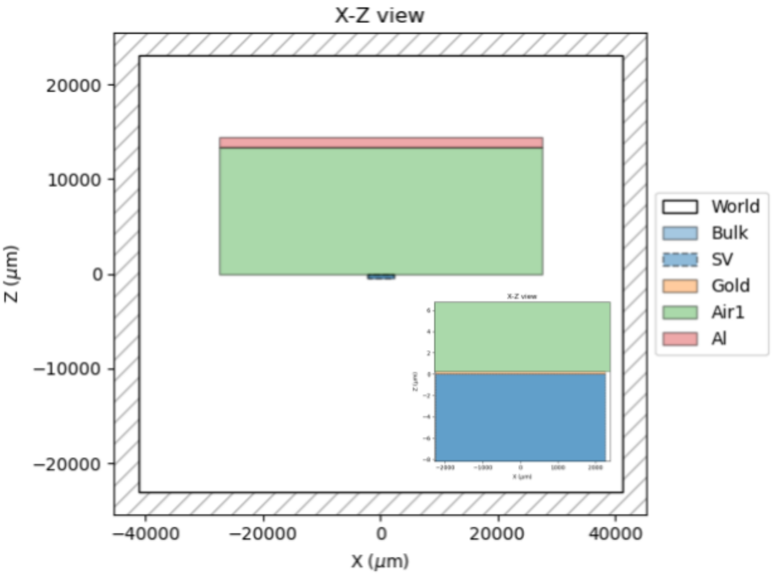


Figure 5.4: G4SEE diamond detector simulation geometry (side view)

For both simulations (2.5 MeV and 14.04 MeV), the G4 HP model was used. In the case of 14.04 MeV neutrons, NRESP71 [67] model (developed at PTB with study of organic scintillators for fast neutrons in energy range 0.2 MeV - 20 MeV) had to be enabled for the complete description of neutron induced α interactions on carbon nuclei. This is not enabled by default in the HP model. It has to be done by additional command in the input macro file:

```
/process/had/particle_hp/use_NRESP71_model true
```

The problem, why only the standard HP model is not sufficient is the description of ^{12}C(n,α) interaction (threshold energy for this interaction is 8.18 MeV), is that the HP model is missing complete description of breakup reactions proceeding in multiple steps, shown on 5.1 and 5.2. In the HP model these interactions are considered only to the first step, leading to the decay of excited intermediate nucleus without particle emission. Therefore two different interaction mechanisms has to be considered. They both end up with two α decay, but they differ in initial and intermediate state. For the first reaction the Q energy value of intermediate nucleus is -8.13 MeV and the threshold neutron energy is 8.81 MeV. For the second interaction the Q value differs according to the excited state of ^{12}C, these values are in the range of -7.65 MeV to -17.0 MeV, the threshold neutron energy is 8.29 MeV. [67]

$$n +^{12}\mathrm{C} \rightarrow \alpha +^{9}\mathrm{Be}^{*} \quad | \quad {}^{9}\mathrm{Be}^{*} \rightarrow n' +^{8}\mathrm{Be} \quad | \quad {}^{8}\mathrm{Be} \rightarrow 2\alpha \tag{5.1}$$
$$n +^{12}\mathrm{C} \rightarrow n' +^{12}\mathrm{C}^{*} \quad | \quad {}^{12}\mathrm{C}^{*} \rightarrow \alpha +^{8}\mathrm{Be} \quad | \quad {}^{8}\mathrm{Be} \rightarrow 2\alpha \tag{5.2}$$

The difference between standard G4 HP model and the NRESP71 model is the library from which they take the nuclear data. NRESP71 takes data for neutron-carbon interactions from ENDF/B-VI [68] library, where the cross section data are retrieved from experiments performed at PTB. [67]

5.1.4 Results

In the figures 5.5 and 5.6 the comparison of simulated and experimental data for 2.5 MeV and 14.04 MeV neutrons can be seen.

In figure 5.7, the different hadron-elastic models are compared for 2.5 MeV neutrons, it can be seen that the `G4HadronDElasticPhysics` model is in the best agreement with the measured data. In the figure 5.8 the comparison of results for 14.04 MeV with and without the NRESP71 model is shown. As can be seen without the NRESP71 model the G4 HP model was not able to differentiate between the two peaks resulting in either 3α particles or ^{9}Be nuclei.

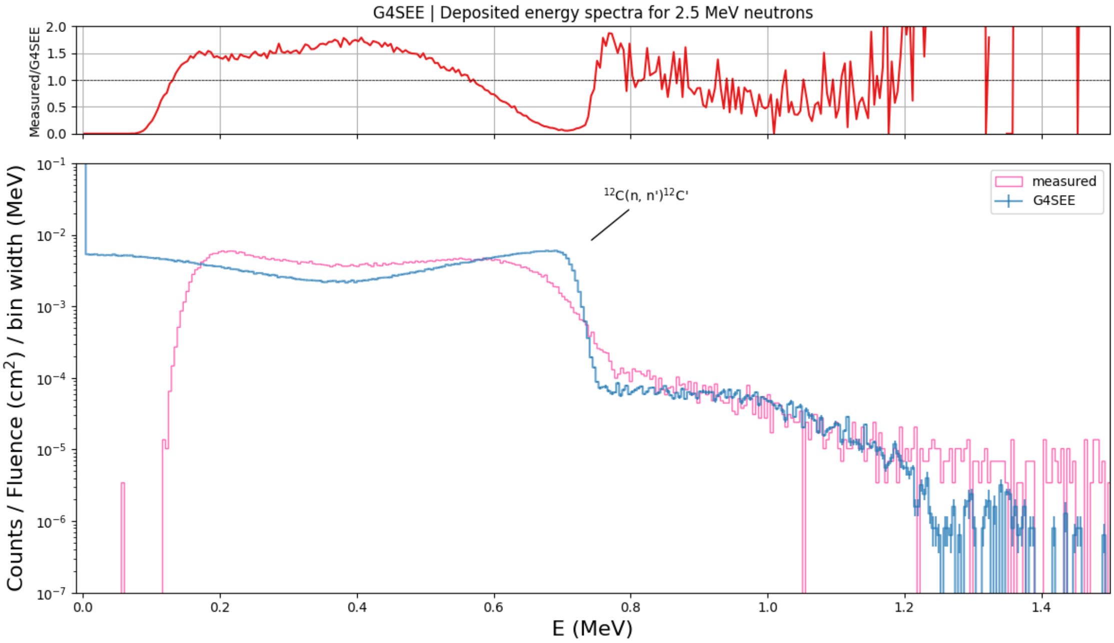


Figure 5.5: Results for 2.5 MeV neutrons. The top plot shows the ratio of measured and simulated data, the bottom plot shows the comparison of measured and simulated spectra.

Since the development of G4SEE falls under the R2E project which belongs to the CERN group where FLUKA.CERN code is developed, it was interesting to compare the G4SEE simulation results with FLUKA results for the 14.04 MeV neutrons, to see if in FLUKA the two peaks will be differentiated. This comparison can be seen in figure 5.9. As can be seen in the figure the simulated data from G4SEE and FLUKA correspond well with each other. The FLUKA simulations of diamond detector have been performed by Mario Sacristan Barbero (CERN/SY-STI-BMI).

From the figures showing the results for 14.04 MeV there can be seen large discrepancy between experimental and simulated data above 9 MeV. To see what

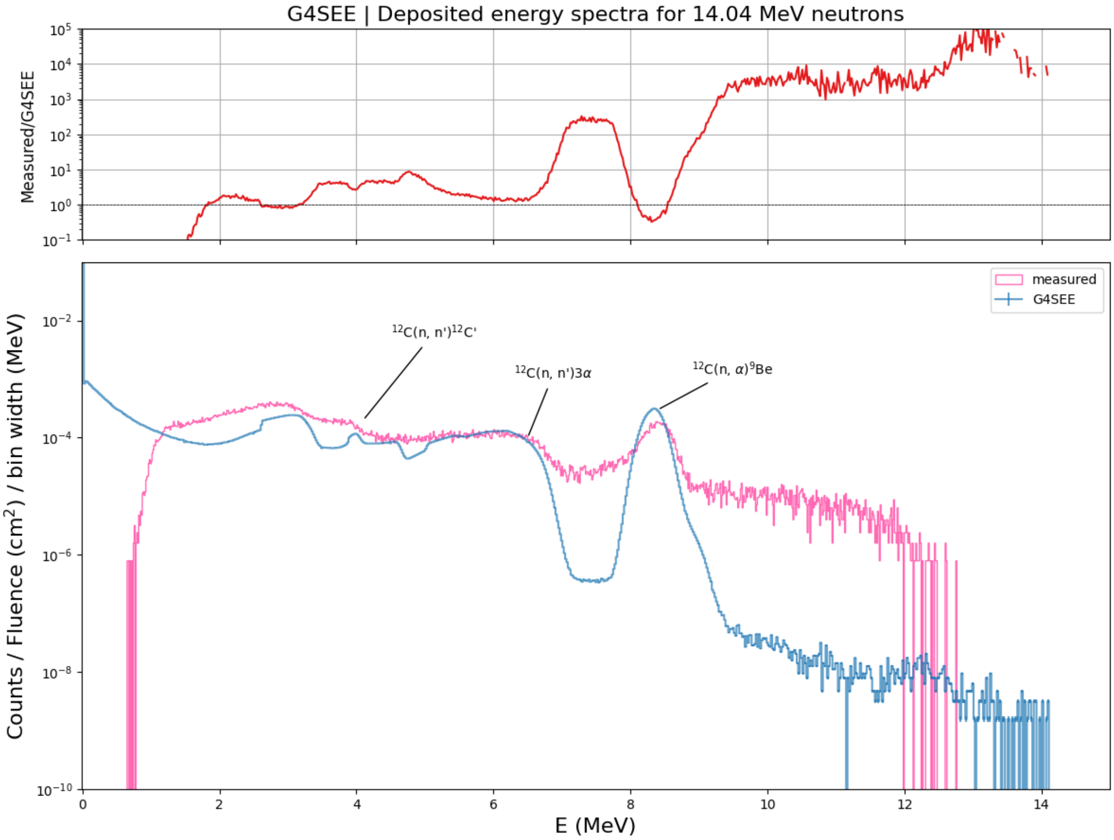


Figure 5.6: Results for 14.04 MeV neutrons. The top plot shows the ratio of measured and simulated data, the bottom plot shows the comparison of measured and simulated spectra.

types of secondary particles are produced in that region, detailed scoring simulation was performed. To be able to see the rare events above 9 MeV, high enough event statistic was necessary. The results of the energy deposition per interaction scoring for 1e+8 primary particles is shown in figure 5.10. As can be seen this was not enough to see the rare events, but increasing the number of primaries for this scoring would create output files too large (orders of hundreds GB). That is why the new energy deposition per interaction scoring was implemented and used. Comparison of these results with standard scoring result can be seen in figure 5.11, in this case the number of primaries was 1e+9 for the detailed scoring. Based on the results of energy deposition scoring per interaction, it can be concluded that the rare high E_{dep} is caused by secondary protons.

The simulation data for 2.5 MeV correspond well with the measured data, the small discrepancies are caused by the detector's energy resolution - that is why we don't see the sharp cutoff in case of experimental data. For the 14.04 MeV data there is discrepancy around 7 MeV, that is caused by the detector's energy resolution as well. The discrepancy above 9 MeV cannot be caused by the detector's resolution. The exact reason for this is not quiet clear, it can be caused by multiple reasons, like incomplete MC simulation of the experimental setup (upstream materials, neutron source, etc.), missing interactions in G4 and FLUKA physics models, or incorrect

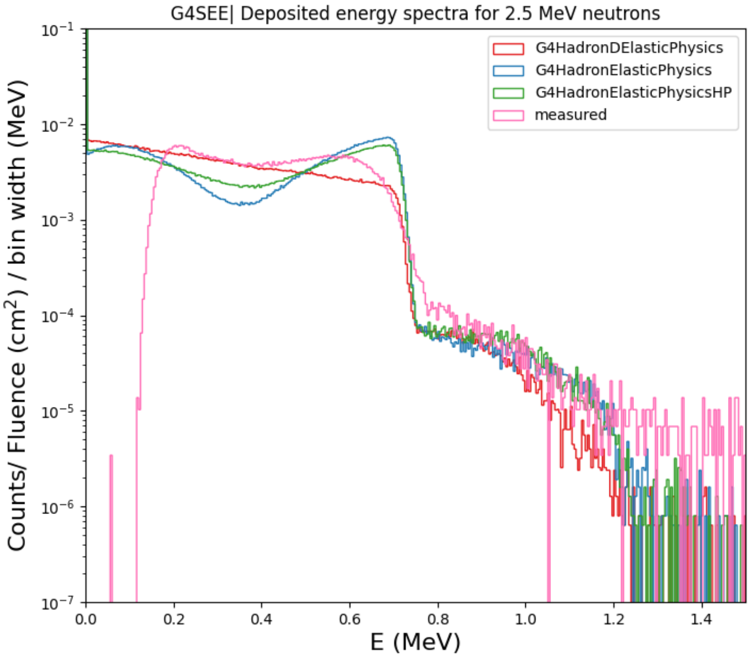


Figure 5.7: Comparison of E_{dep} distributions using different G4 elastic physics models for 2.5 MeV neutron simulation

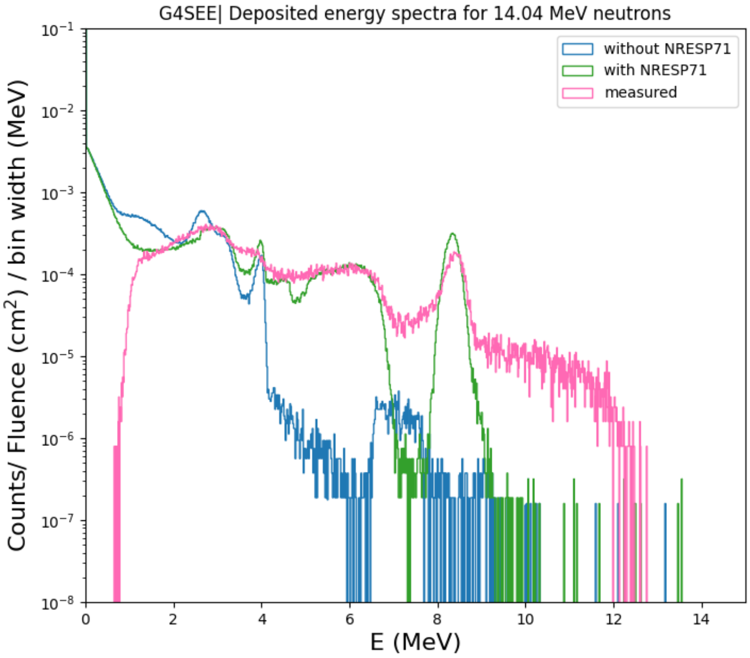


Figure 5.8: Comparison of E_{dep} distributions with and without NRESP71 model enabled for 14.04 MeV neutron simulation

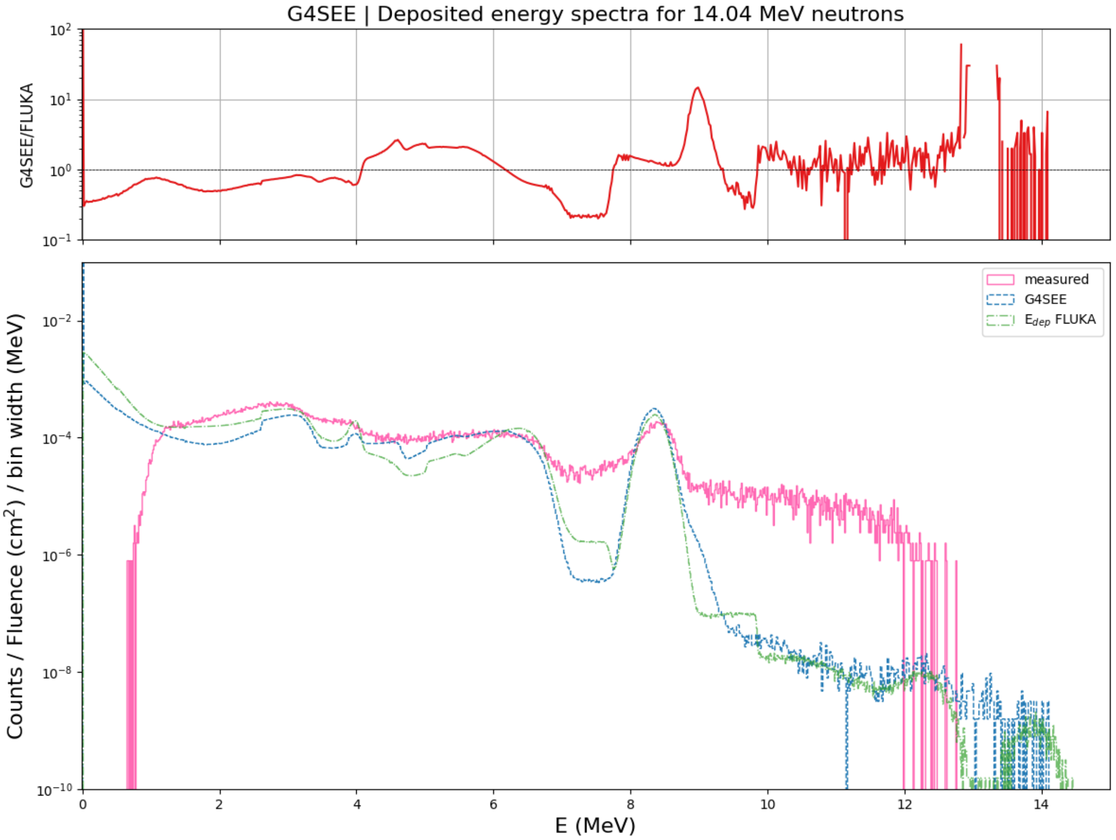


Figure 5.9: On the top plot the ratio of G4SEE and FLUKA simulated data is shown. Comparison of E_{dep} for experimental data with G4SEE and also FLUKA simulation results for 14.04 MeV neutrons is shown on the bottom plot.

neutron cross section data used by G4 and FLUKA, or due to electronic noise. Results of detailed scoring show, that above 9.5 MeV only secondary protons contribute to the deposited energy spectra.

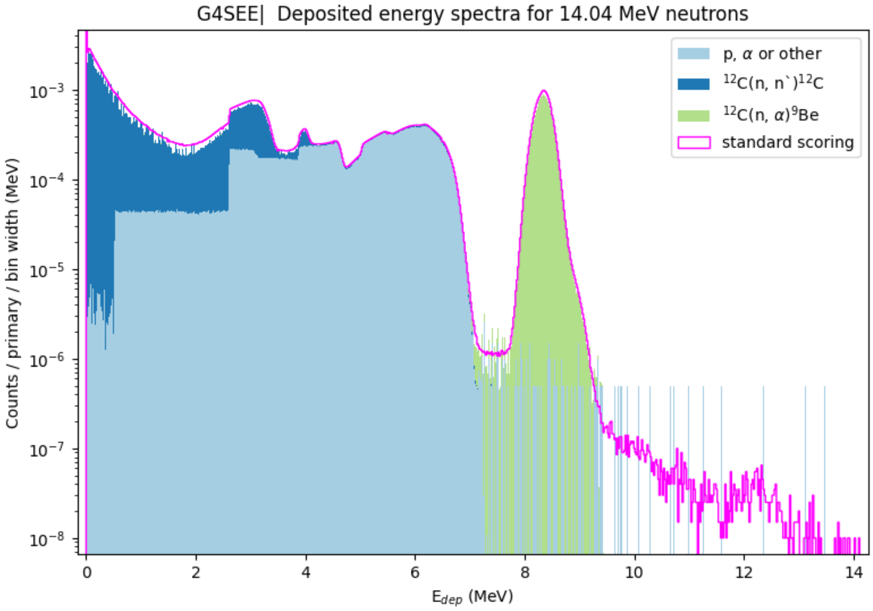


Figure 5.10: Results of detailed scoring for 14.04 MeV neutrons with 1e+8 primary particles. The dark blue and green histograms show contribution of a specific nuclear interaction on carbon nuclei and the light blue histogram shows the contribution of interactions when light secondary ions were created. Histograms are stacked on top of each other.

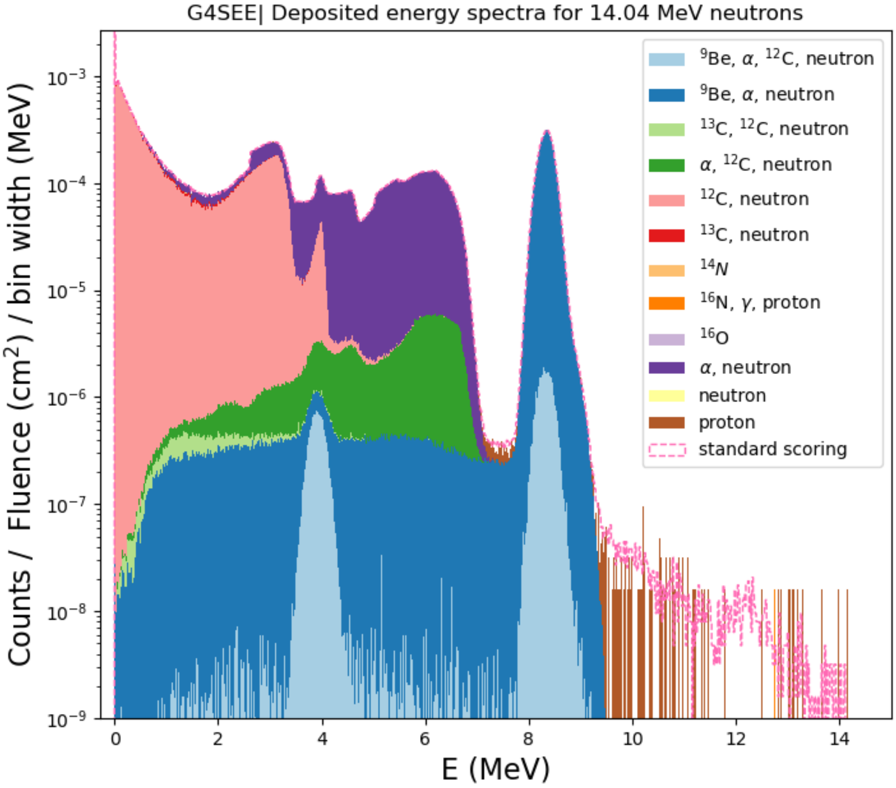


Figure 5.11: Results of newly implemented energy deposition per interaction scoring for 14.04 MeV neutrons with 1e+9 primary particles. Each histograms represent deposited energy by the secondary particles listed, histograms are stacked on top of each other.

5.2 SiC power MOSFET simulations

The constantly increasing demand for higher power, higher frequency and higher temperature resistant devices for variety of applications requires more than Si-based components can deliver. The Si-based power devices are approaching their material limits, so the need to find a new material that can fulfill the future needs arose. One of these materials is silicon carbide (SiC), which is one of the best alternatives to silicon. [69]

Many research groups are investigating the SiC properties and also its radiation resistance. One of these research groups is at Department of Information Technology and Electrical Engineering at ETH Zurich, where they have dedicated Advanced Power Semiconductor (APS) Laboratory. Their research focus is on power semiconductors and their modules, in improving the devices necessary for power electronics. Related to the SiC, they are developing new devices and establishing a SiC MOSFET manufacturing process. [70]

In collaboration with Dr. Corinna Martinella (ETH Zurich, APS) and her colleagues, we are providing the simulations of SiC power MOSFET for their study of SEEs. In the following sections the SiC power devices will be described in more details, and afterwards the SiC simulations results will be presented.

5.2.1 SiC power MOSFETs

The SiC is a wide-bandgap (WBG) semiconductor material, consisting of 50% silicon and 50% carbon. It can exist in around 250 different crystalline forms, mostly synthetically grown. It belongs to one of the hardest known materials, and can also maintain high hardness and elasticity at high temperatures. Thanks to low intrinsic carrier density in room and also high (700-800 K) temperatures, the SiC devices can be operated with low leakage current (compared to Si, it is orders of magnitude lower). Another advantage compared to Si is that SiC can withstand electric fields over eight times greater without undergoing avalanche breakdown, which makes it suitable for high-voltage and high-power devices, such as power transistors, diodes and thyristors. In figure 5.12 can be seen the comparison of SiC material properties to Si and also GaN materials. [37]

Thanks to the high electric field strength the SiC devices can be manufactured with much thinner drift layers in respect to Si. Since the resistance is determined by the thickness of the drift layers, the resistance per unit area can be reduced for same blocking voltage capability[1]. This reduces the conduction losses, thus makes it suitable for high-voltage operations. The adaption of SiC in various applications has been relatively slow until 2017. It was difficult especially for replacement of Si MOSFET and Si insulated gate bipolar transistor devices. Nowadays it has applications for example in electric vehicles, space and avionics and also for high-energy accelerators at CERN. [37]

SiC power MOSFET has a metal gate electrode, which is insulated with an oxide

[1] blocking voltage capability is the ability to withstand a high reverse voltage without breakdown or conduction

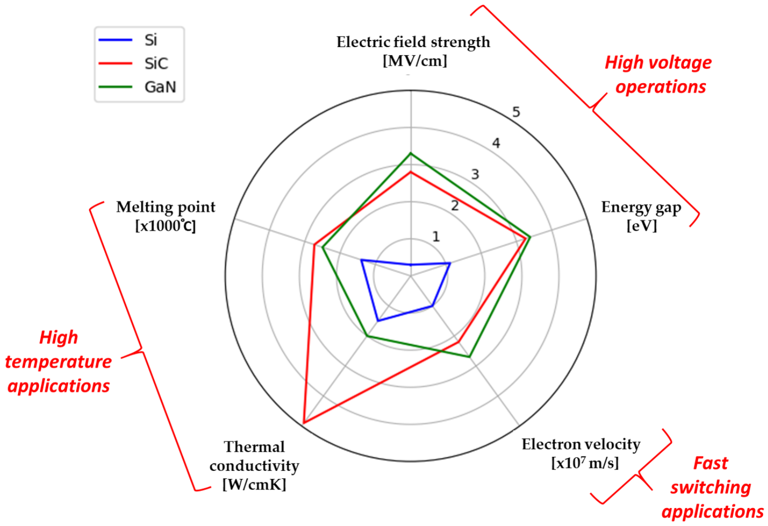
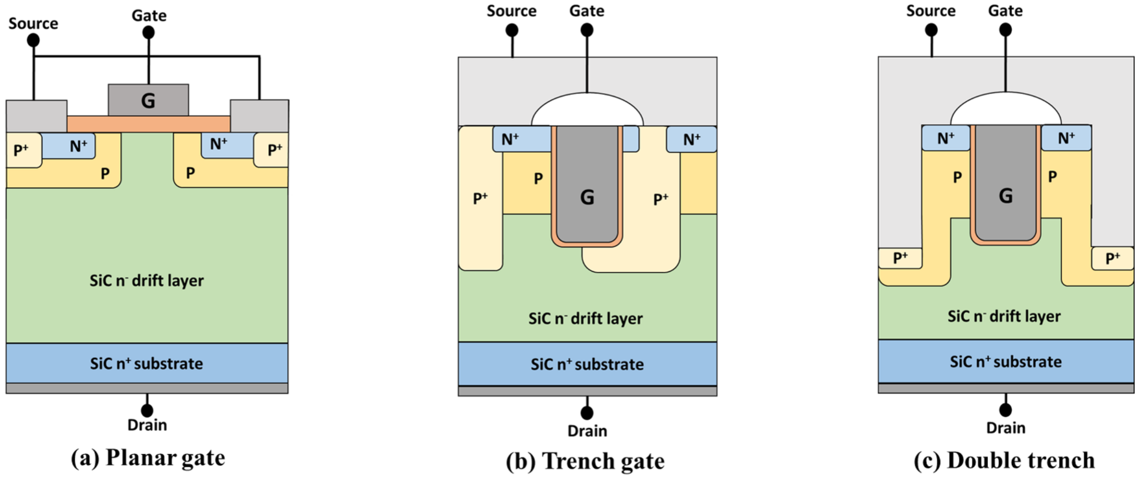


Figure 5.12: Radar chart of Si, SiC and GaN relevant material properties [37]

layer from the semiconductor channel. They are used in wide range of applications, such as power supplies, DC-to-DC converters, low voltage motor controllers and others. Commercially two types of SiC power MOSFETs are available, namely planar gate devices and trench devices, these can be further divided into single- or double-trench design. Single-trench device has only a trench gate, whereas double-trench device has trench source and also a trench gate. These types of architectures are shown in figure 5.13. The gate source voltage determines the conductivity of the MOSFET independently from the architecture. From this it follows that the current flowing through the main channel between the drain and source is dependent on the gate voltage. Changing the conductivity with the gate voltage leads to the switching or amplifying the signals. During normal operation no current flows in the gate, because gate oxide isolates gate electrode from the rest of the device structure. Most power MOSFETs are being manufactured in a single-chip structure with a large number of closely stacked identical cells. Purpose of this is to minimize resistance between the drain and the source when device is in the on-state. With trench structures it is possible to achieve higher cell densities compared to planar structures, but they are more difficult to manufacture. [37]

SEEs in SiC power MOSFETs

The SEB and SEGR (explained in section 2.3.1) are observed for low LET particles (high-energy neutrons or protons), they are the source of failures of SiC power MOSFETs in space applications. The TID plays main role in cumulative effects, it changes the I-V and C-V characteristics. For these effects certain natural resistance exists. When the device is irradiated by neutrons the recoiled nuclei from the lattice atoms can cause a functional failure of the device. Permanent damage of the device can be also caused by proton and heavy ion irradiation. High LET values will lead to the increase in the gate and source leakage of the device. After irradiation the safe working voltage significantly reduces, which affects the device reliability index. This implies that the SEE assessment of voltage range for safe operation is needed. [72]



(a) Planar gate (b) Trench gate (c) Double trench

Figure 5.13: Schematics of architecture types of SiC power MOSFETs [71]

5.2.2 Simulations

In the collaboration with APS group of ETH Zurich, simulations of SiC power MOSFET with 200 MeV and 3 MeV protons were performed. Following characteristics were set for both 200 MeV and 3 MeV monoenergetic proton beam: Gaussian beam energy distribution, source position 5 cm from the DUT in z-axis direction, rectangular shape of beam covering the DUT, plane source distribution type. In figure 5.14 the geometry implemented in their 1-D Technology Computer-Aided Design (TCAD) simulations can be seen (the information about the dimensions of individual volumes are not described here, since they are confidential). The examined scoring volumes are SV1 and SV2 shown in the figure. In the figure only one cell is show, in reality (and also this was implemented in the simulations) three of such cells are stacked next to each other. The whole geometry (all three cells) implemented in G4SEE can be seen in figures 5.15 and 5.16. Because in G4SEE the BEOL can be only stacked on top of each other, the epoxy was placed on top of the aluminium layer and the SiO_2 and poly-silicon layers were placed only above the SV2, so above SV1 is vacuum up to the aluminium layer. As G4SEE allows only one SV, for each SV in each cell different simulation run had to be done.

The deposited energy (charge) is an important input for the estimation of the SEE rate. Therefore for each SV E_{dep} was scored. Another interesting and useful results for better understanding of SEEs, are the results from the detailed scoring. Thanks to these results the interaction types and their products can be known, also position (volume) of creation of the products and energy deposited by them can be obtained. All these information are relevant for the SEE estimations and can serve as an input for future TCAD simulations. The simulation results are presented in the next sections.

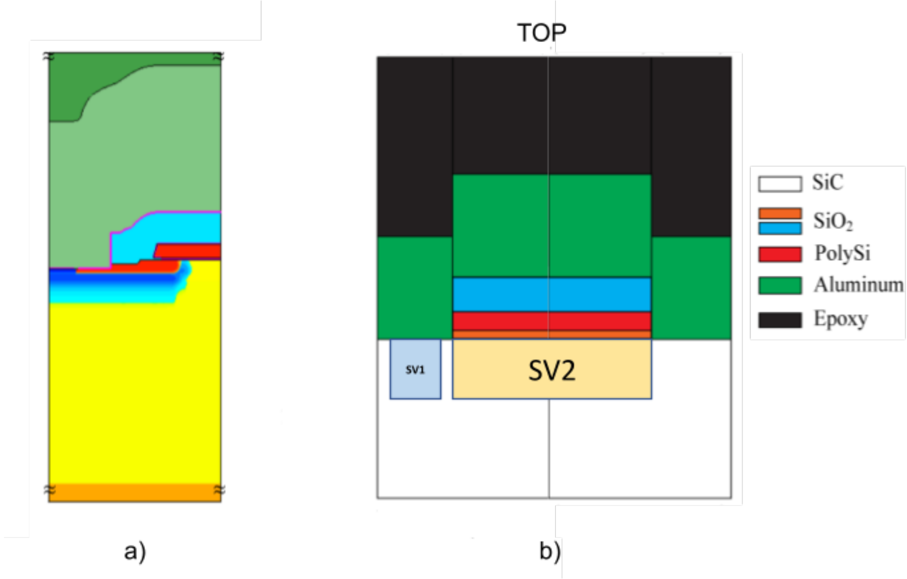


Figure 5.14: Geometry of the SiC power MOSFET provided by our collaborators from ETH Zurich, a) TCAD electric field of the device (only half of one cell), b) simplified 2D geometry, which was than implemented in G4SEE (one cell)

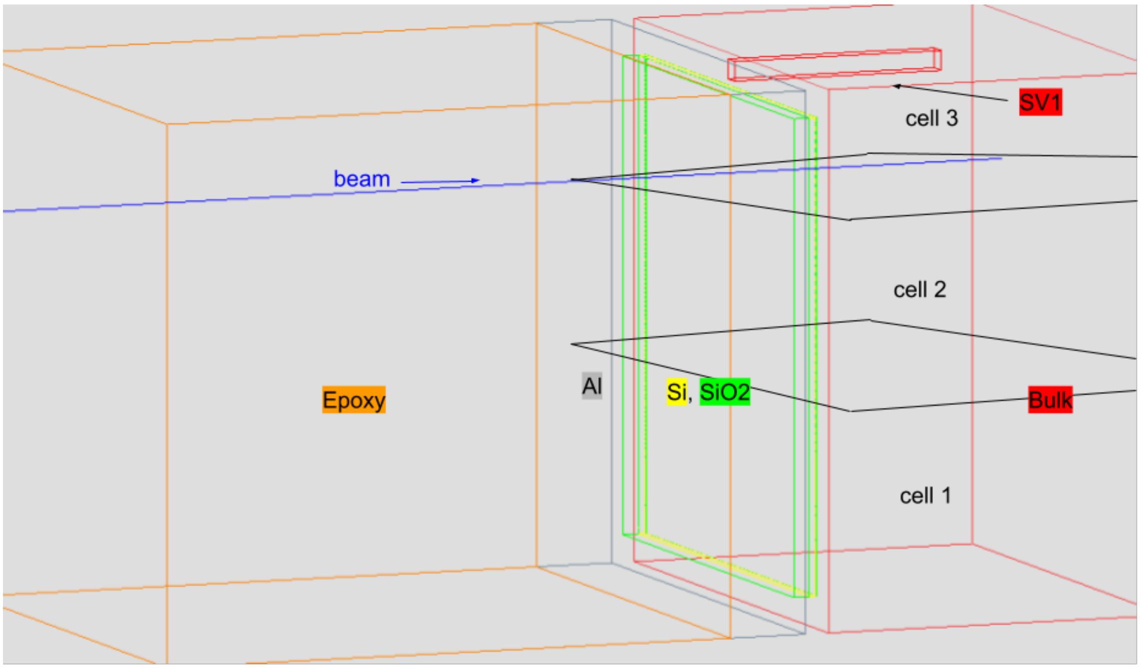


Figure 5.15: Visualisation of implemented geometry in G4SEE, showing SV1 placed in cell 3

5.2.3 Results

Simulations of 200 MeV protons

For the 200 MeV protons the whole geometry described in previous section was implemented in the simulation. The deposited energy spectra for SV1 and SV2 in

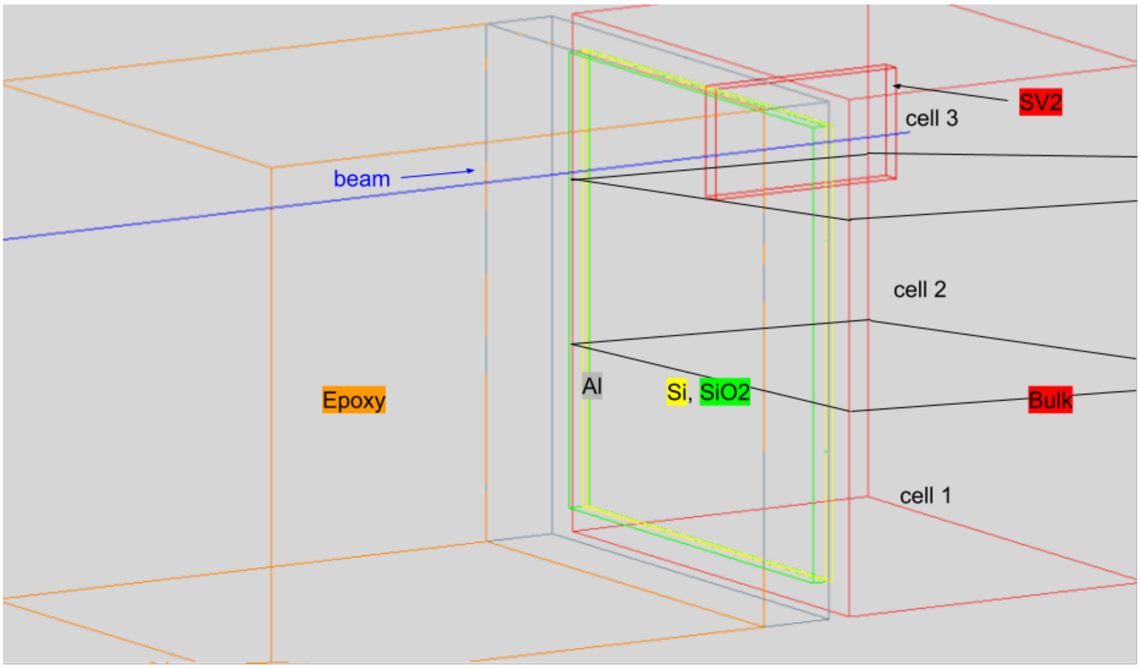


Figure 5.16: Visualisation of implemented geometry in G4SEE, showing SV2 placed in cell 3

cells 1-3 are shown in figures 5.17 and 5.18.

Comparison of deposited energy for both SVs in each cell is shown in figure 5.19. Reason, why there is bigger discrepancy for SVs in cell 2 compared to cell 1 and cell 3 is because in this case the geometry is different. I had to make the SiO_2 and poly-silicon layers smaller, for them not to interfere with the area above the SV1. In this figure, there is also histogram of deposited energy for SV only. This is result of simulation, where the geometry consisted only of SiC SV, rest was filled with vacuum. I ran this simulation because the results of the detailed scoring were not unambiguous, since the defined materials can contain different isotopes of an element (for example ^{28}Si and ^{29}Si) and also some products can be created in BEOLs and than deposited their energy in SV. With results of this simulation I was able to determine correctly in which types of processes where the secondary particles created, because it contains products created only in the SV.

The results of detailed scoring of energy deposition can be seen for both SV1 and SV2 in cell 3 in figures 5.20 and 5.21 (for cell 1 and cell 2 the results were almost identical, that is why results for only one cell are shown). For both of these plots only the most represented secondary products are shown (for example for SV2 79 different secondary products were created), that is why in case of results for SV2 the histogram of deposited energy retrieved from standard scoring differs from the detailed scoring.

In table 5.1 volumes in which the secondary particles are created are shown, and in table 5.2 the types of processes that create the secondary particles are shown. The results are shown only for the SV2, for the SV1 the results are very similar.

Results of standard scoring, displayed in figures 5.17 and 5.18, show that in SV2

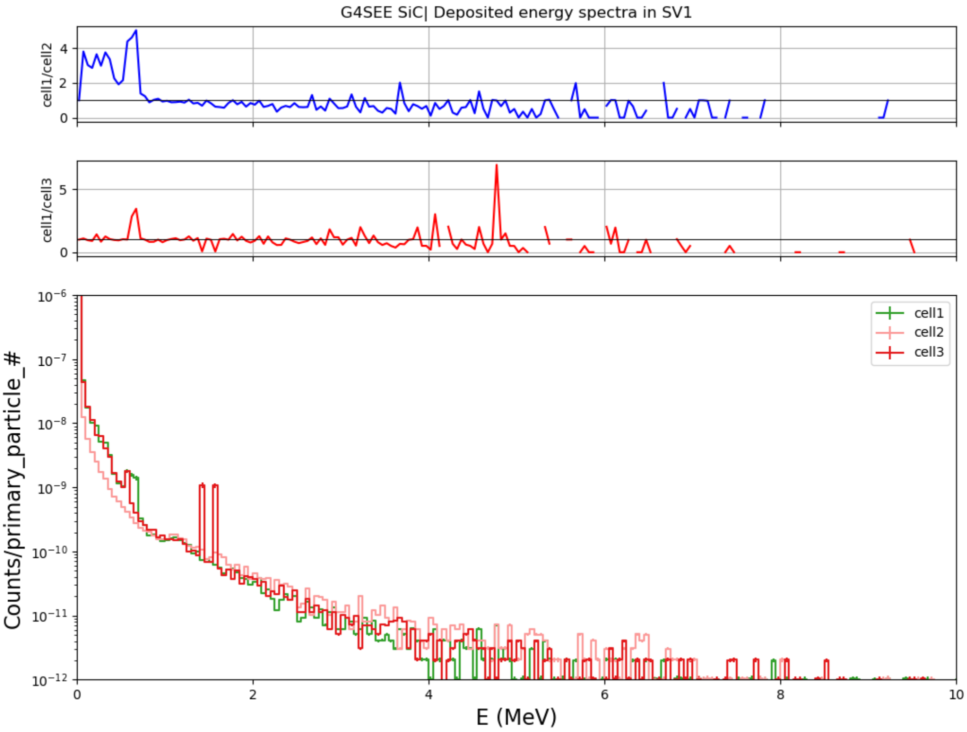


Figure 5.17: On the top two plots the ratios of deposited energy in SV1 in cell 1 to cell 2 and cell 3 can be seen. On the bottom plot is shown the comparison of E_{dep} in SV1 in each cell for 200 MeV protons.

Table 5.1: Creator volumes for 200 MeV simulation with 10^8 primary particles for SV2 in cell 1

	reaction products ($Z > 2$ nuclei)																
volumes	None	^{11}C	^{12}C	^{25}Mg	^{28}Si	^{10}B	^{15}N	^{20}Ne	^{3}He	^{8}Be	^{27}Al	^{26}Al	^{11}B	^{26}Mg	^{23}Na	^{24}Mg	^{27}Si
Bulk	536080	408	1119	421	716	524	29	200	1644	1653	684	233	860	522	330	598	250
SV	11998	801	3048	1297	5869	578	37	507	142	1163	2273	788	1288	1446	847	1914	928
Al	13191	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Epoxy	8861	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SiO_2 second	3610	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Si	1989	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SiO_2 first	224	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.2: Creator processes for 200 MeV simulation with 10^8 primary particles for SV2 in cell 1

	reaction products ($Z > 2$ nuclei)																
processes	None	^{11}C	^{12}C	^{25}Mg	^{28}Si	^{10}B	^{15}N	^{20}Ne	^{3}He	^{8}Be	^{27}Al	^{26}Al	^{11}B	^{26}Mg	^{23}Na	^{24}Mg	^{27}Si
proton-inelastic	478271	1199	738	1694	990	1092	66	694	1783	2798	2907	1008	2128	1940	1158	2473	1167
ionization	95625	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hadron elastic	3	0	3374	0	5509	0	0	0	0	0	0	0	0	0	0	0	0
proton-inelastic, ionization	2492	11	6	29	8	12	0	15	7	29	58	18	30	35	23	44	15
electron ionization	177	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hadron-elastic, ionization	1	0	54	0	93	0	0	0	0	0	0	0	0	0	0	0	0
Bremmstrahlung	78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

the deposited energy reaches higher values compared to the SV1. This is important

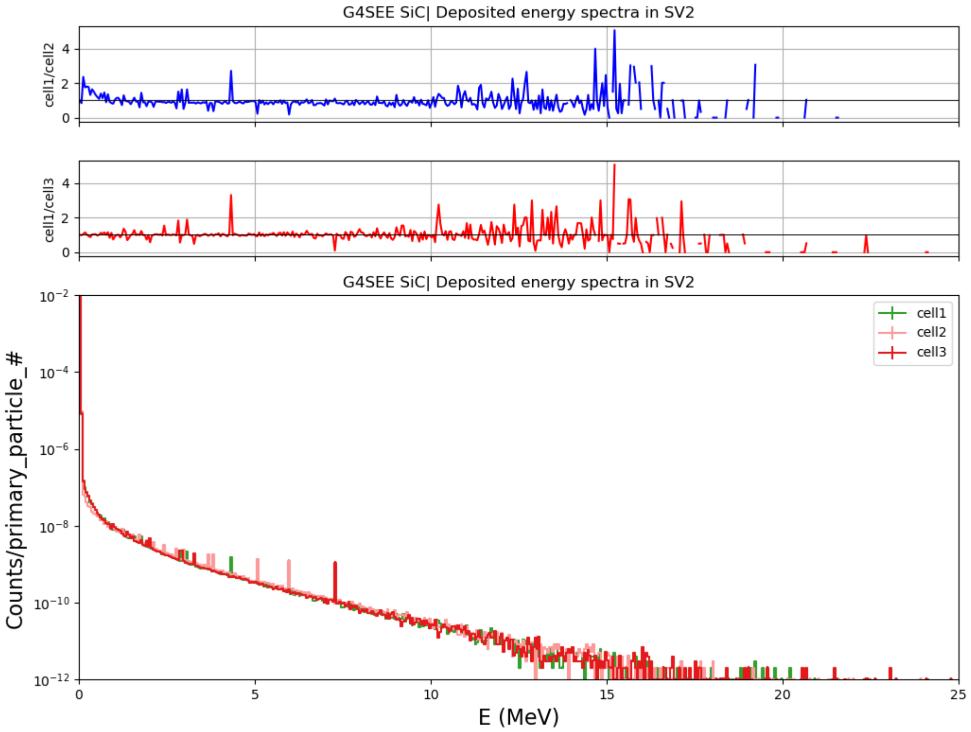


Figure 5.18: On the top two plots the ratios of deposited energy in SV2 in cell 1 to cell 2 and cell 3 can be seen. On the bottom plot is shown the comparison of E_{dep} in SV2 in different cells for 200 MeV protons.

for SEEs, since for higher energies the probability of SEE increases. This means that the SV2 (under gate) is in bigger danger of some SEE. The results of detailed scoring show that interactions where primary proton deposit energy only via ionization and inelastic interactions with creation of light secondary particles (such as protons, neutrons, and α particles) prevail. These secondaries are mostly created in SV itself, but also in the BEOL layers. Heavy ions interacting in SV are produced mostly in proton inelastic interactions in the SiC bulk. Heavy ions produced in BEOL layers don't reach the SV. Although heavy ions production is not high, due to their high ionisation ability (high particle LET) they are the main source of SEEs.

Particle LET plays a big role in the triggering of SEEs. In case of SiC power MOSFET the critical particle LET on inducing the SEB is studied. For example in paper [73], authors are showing some results of critical LET values for different particle species for SEB probability based on particle LET. In the figure 5.22 comparison of LET distributions for the 200 MeV protons is shown. In case of the LET scoring per particle species, it could be in the future also interesting to see the LET values for the heavy ions created in the inelastic interactions.

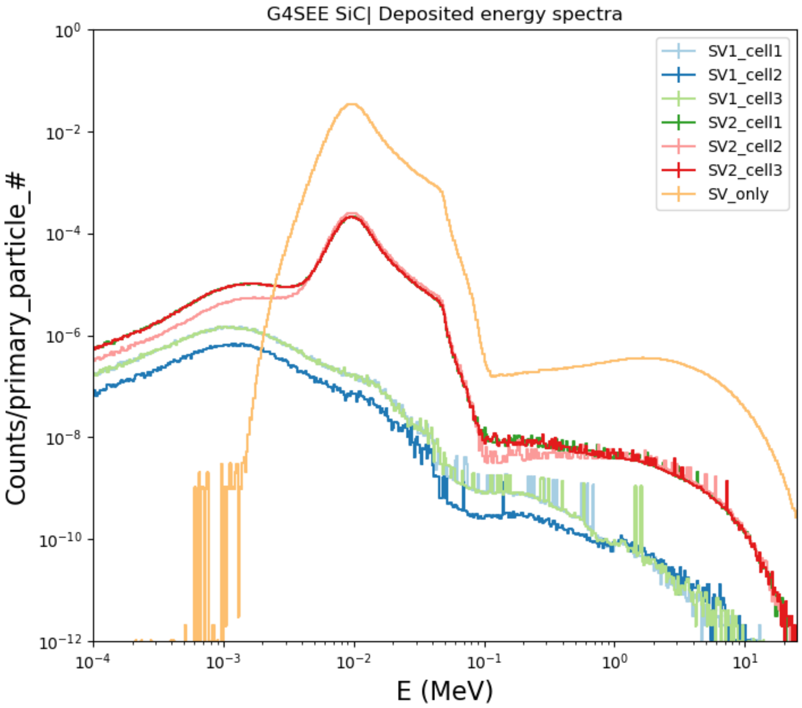


Figure 5.19: Comparison of E_{dep} in SV1 and SV2 in different cells and also for SV only (no BEOL were simulated) for 200 MeV protons

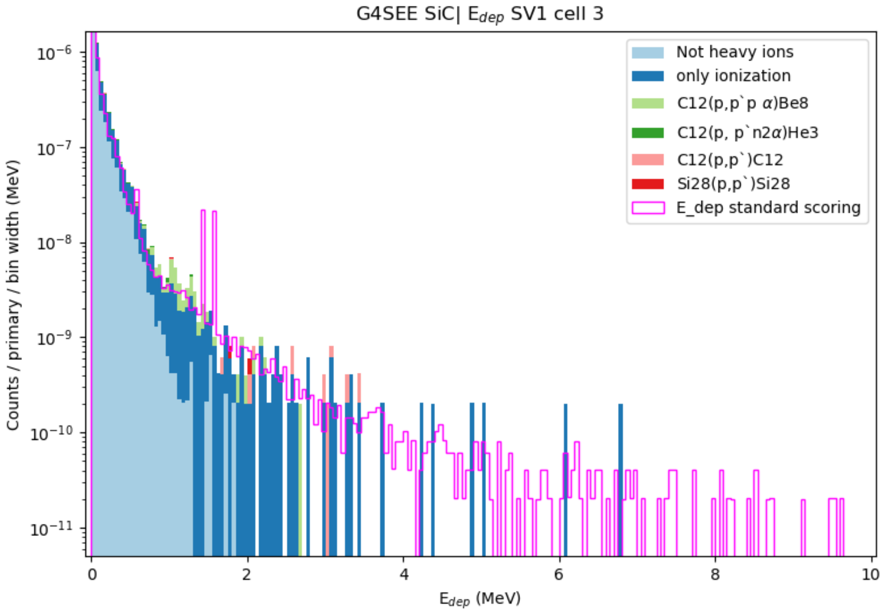


Figure 5.20: Results of detailed scoring for SV1 for 200 MeV protons, only the most frequent nuclear reactions (based on secondary products) are shown. The histograms are stacked on top of each other.

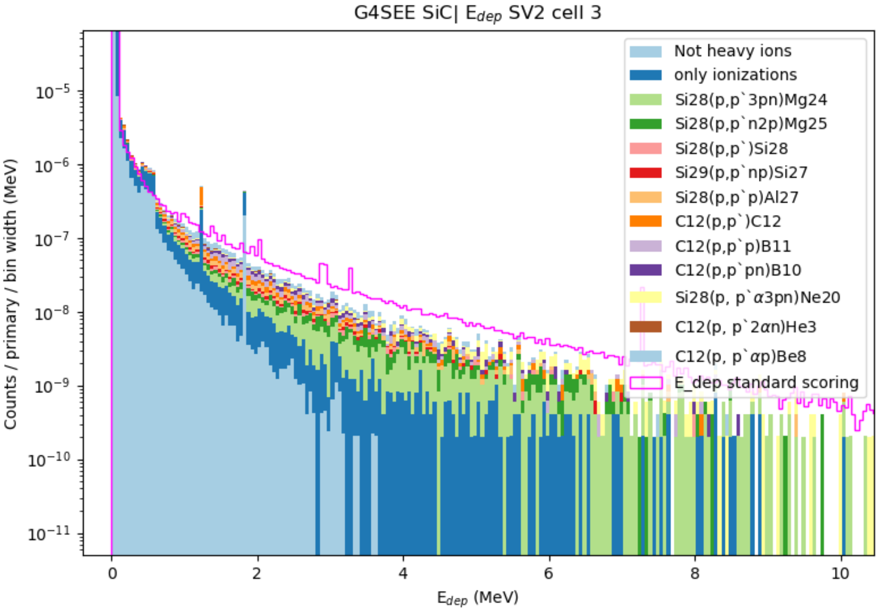


Figure 5.21: Results of detailed scoring for SV2 for 200 MeV protons, only the most frequent nuclear reactions (based on secondary products) are shown. The histograms are stacked on top of each other.

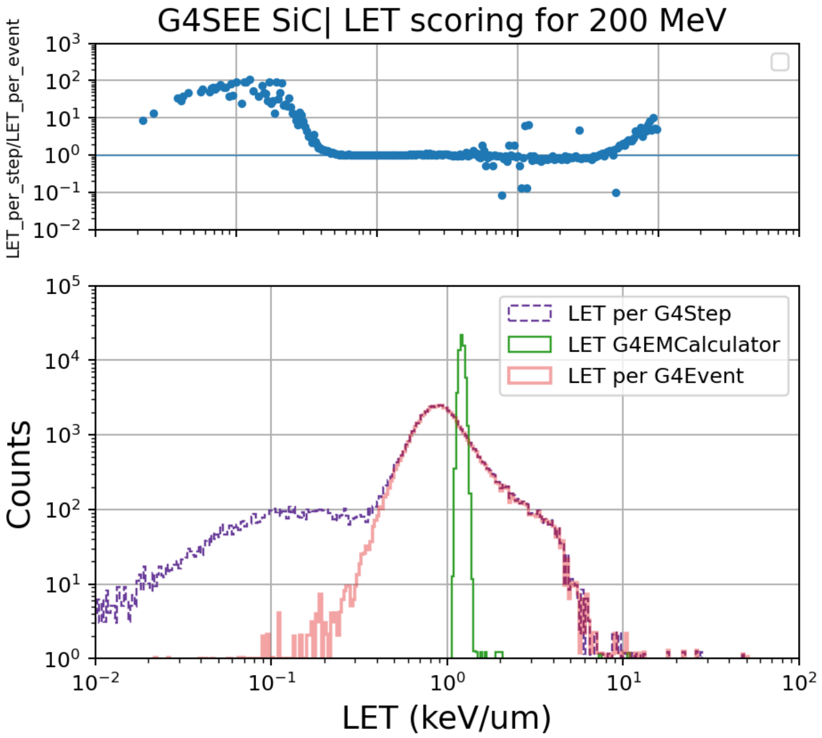


Figure 5.22: On the top plot the ratio of LET computed per particle step and per one event is shown. On the bottom plot comparison of all three implemented LET scoring for 200 MeV protons is shown.

Simulations of 3 MeV protons

For the 3 MeV proton simulations the geometry had to be changed compared to the 200 MeV proton simulations. Specifically the epoxy layer had to be removed, otherwise all the protons would stop in it, this can be seen in figure 5.23, the epoxy layer was also removed in experimental measurements done by APS group at ETH Zurich. The deposited energy spectra for both SVs in cells 1-3 can be seen in figures 5.24 and 5.25.

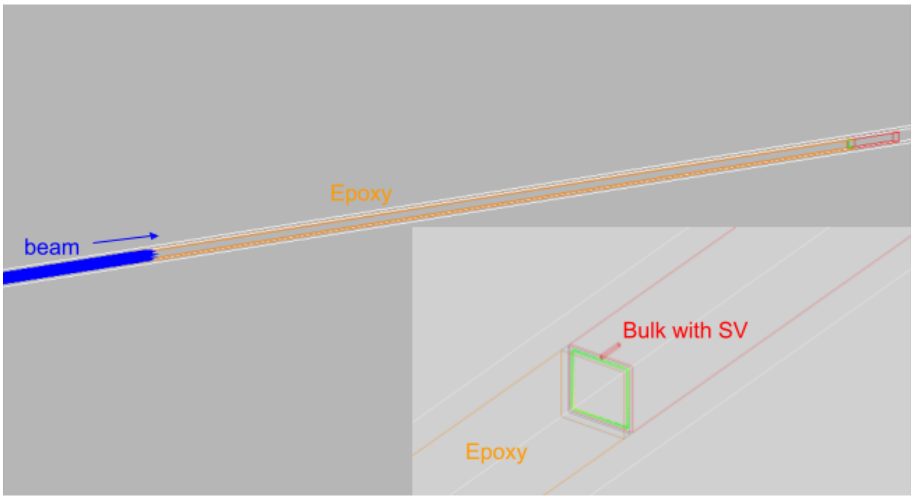


Figure 5.23: Visualisation of simulation run for 3 MeV protons with geometry including epoxy layer (orange), which had to be removed.

Same as for the 200 MeV proton simulations a simulation with SV only was performed, comparison of E_{dep} can be seen in figure 5.26.

The result of post-processed detailed scoring for SV1 and SV2 in cell 3 can be seen in figures 5.27 and 5.28 (same as for the 200 MeV simulations the results for cell 1 and cell 2 are almost identical, therefore not shown). For the SV2 the deposited energy retrieved from detailed scoring (figure 5.28) exceeds above 0.5 MeV the results of standard scoring. This is due to the primary protons. The reason here could be due to wrong biasing weight correction in the post-processing script, further investigations of this discrepancy has to be made.

In table 5.3 volumes, in which the secondary particles are created, are shown and in table 5.4 are shown the types of processes that create secondary particles.

In case of 3 MeV simulations again, the energy deposited is higher in SV2, which means that this volume is under higher risk of damage due to SEE. From the detailed scoring results it can be seen that in the case of 3 MeV protons most of the secondary particles were created due to the hadron elastic interactions and were mostly created in bulk.

Similarly as for the 200 MeV protons the LET simulations for 3 MeV protons were also performed and are shown in figure 5.29.

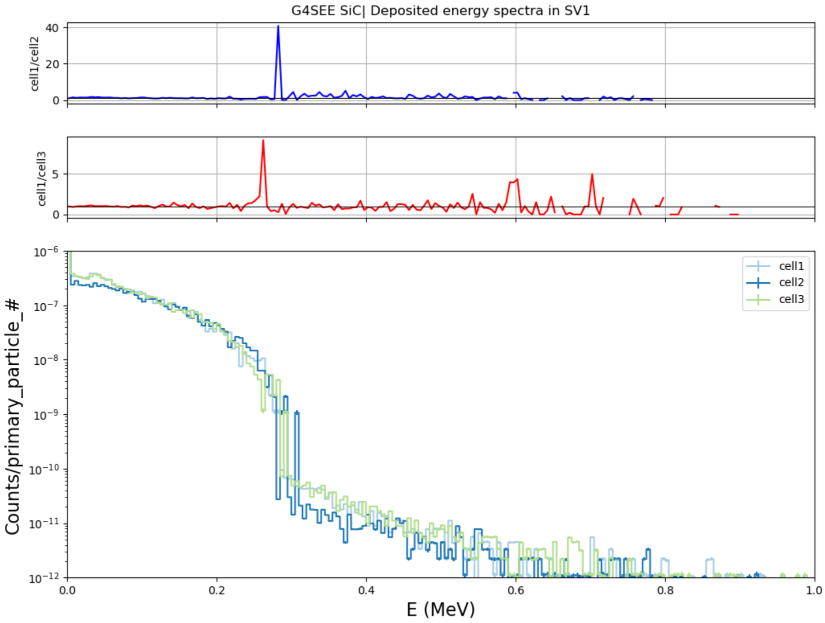


Figure 5.24: On the top two plots the ratios of deposited energy in SV1 of different cells can be seen. On the bottom plot is shown the comparison of E_{dep} in SV1 in different cells for 3 MeV protons.

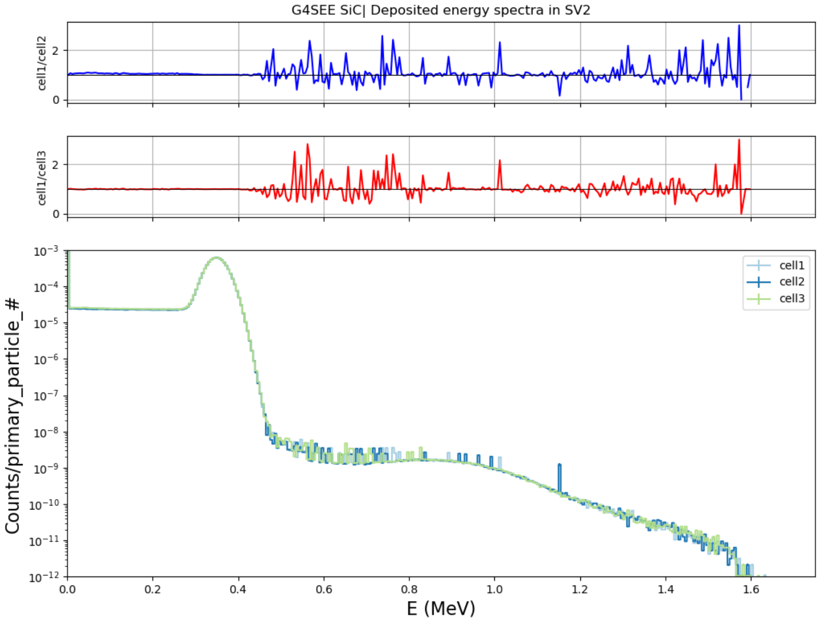


Figure 5.25: On the top two plots the ratios of deposited energy in SV2 of different cells can be seen. On the bottom plot is shown the comparison of E_{dep} in SV2 in different cells for 3 MeV protons.

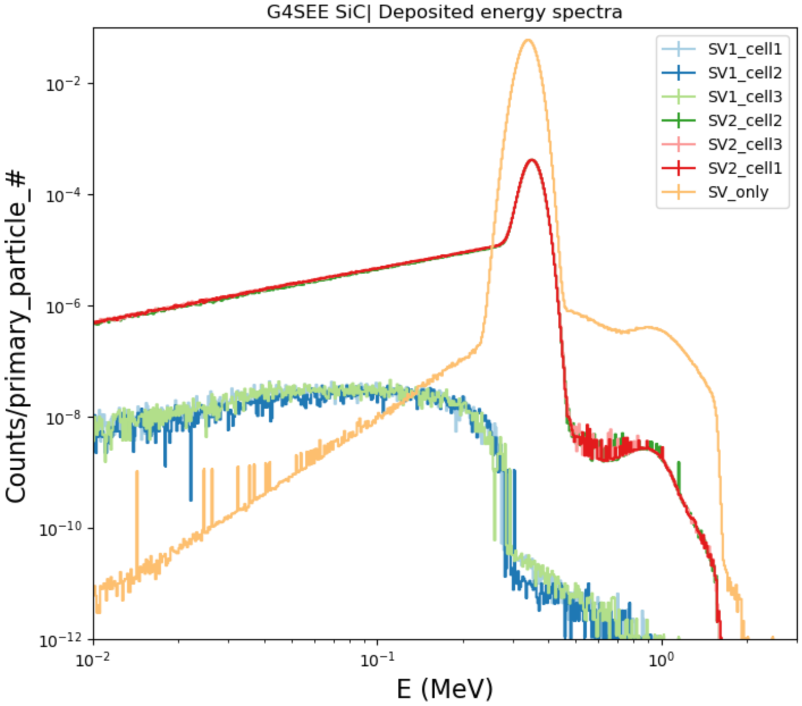


Figure 5.26: Comparison of E_{dep} in SV1 and SV2 in each cell and also for SV only (no BEOLs were simulated) for 3 MeV protons

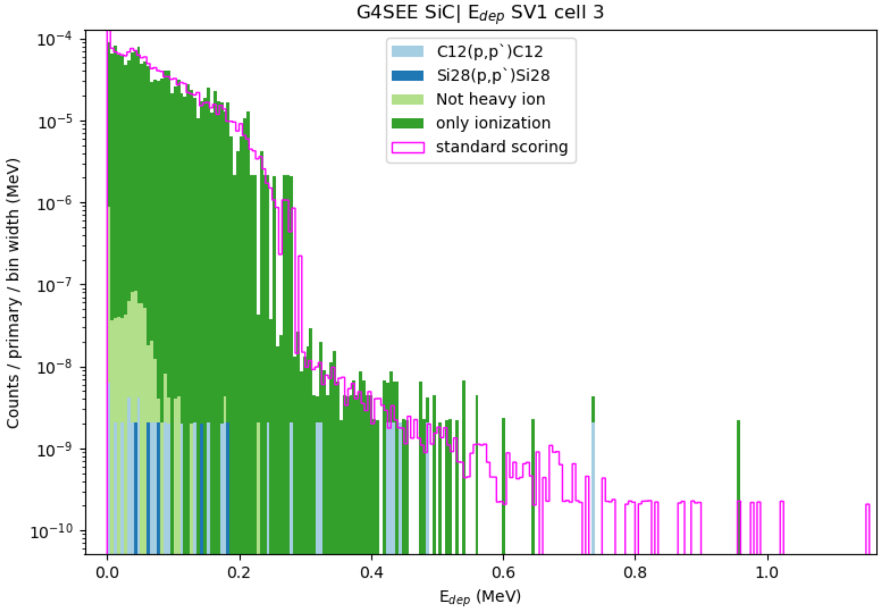


Figure 5.27: Results of detailed scoring for SV1 for 3 MeV protons, only the most frequent interactions are shown. The histograms are stacked on top of each other.

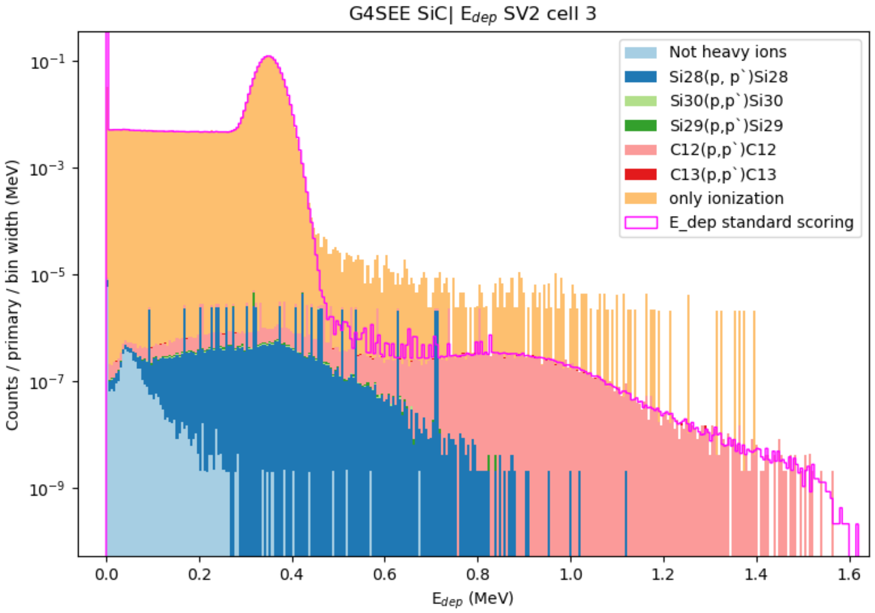


Figure 5.28: Results of detailed scoring for SV2 for 3 MeV protons, only the most frequent interactions are shown. The histograms are stacked on top of each other.

Table 5.3: Creator volumes for 3 MeV simulation with 10^8 primary particles for SV2 in cell 1

	reaction products ($Z > 2$ nuclei)						
volumes	None	^{28}Si	^{12}C	^{29}Si	^{30}Si	^{13}C	^{29}P
SV	0	14704	25396	727	574	295	6
Bulk	3800	809	3811	44	25	41	0
Al	1	0	0	0	0	0	0
Si	0	1	0	0	0	0	0

Table 5.4: Creator processes for 3 MeV simulation with 10^8 primary particles for SV2 in cell 1

	reaction products ($Z > 2$ nuclei)						
processes	None	^{28}Si	^{12}C	^{29}Si	^{30}Si	^{13}C	^{29}P
hadron elastic	0	15255	29063	754	592	335	0
proton inelastic	3801	233	134	16	7	1	6
coulomb scattering	0	26	10	1	0	0	0

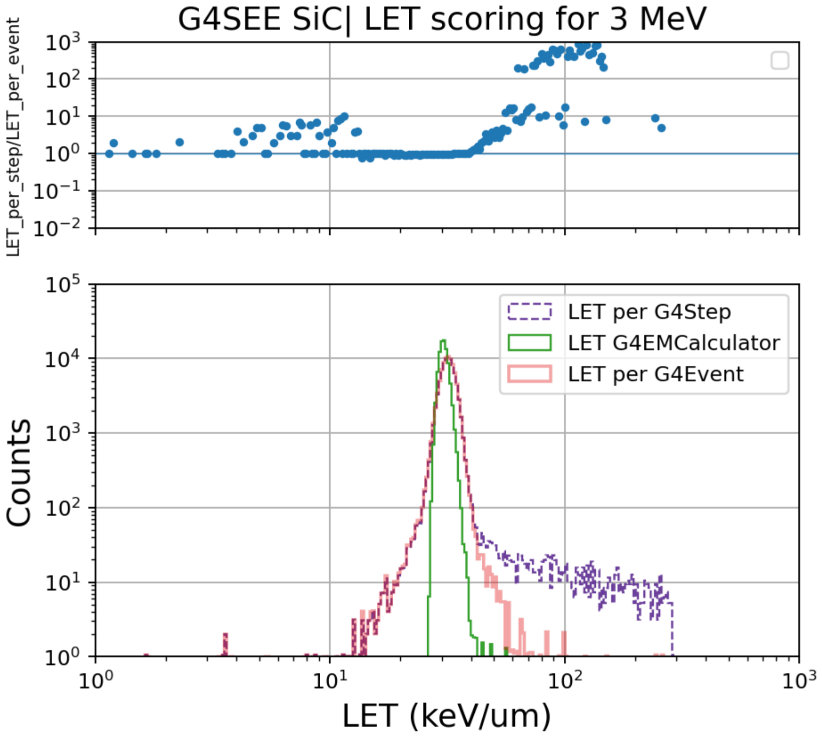


Figure 5.29: On the top plot the ratio of LET computed per particle step and per one event is shown. On the bottom plot comparison of all three implemented LET scoring for 3 MeV protons is shown.

5.3 LET simulations

For the experimental validation of the LET scoring, the data obtained during proton test campaign will be used. This test campaign has been postponed several times unfortunately. Currently it is planned to be performed at Paul Scherrer Institut (PSI) at Proton Irradiation Facility (PIF). The test campaign at this facility was postponed due to a problem with the delivery of one of the devices to be tested. Beam time is funded by the RADNEXT Transnational Access. This test campaign was originally planned to happen in summer 2023 at PARTREC facility, unfortunately this was cancelled due to technical problems with accelerator. Both of these facilities and preparations for the test campaign will be described in following sections. After that the results of simulations made in preparation for the test will be shown.

5.3.1 Proton test campaign

During the test campaign the CMOS Active Pixel Sensors of a COTS camera and a custom-made CIS-based instrument will be irradiated by protons. The purpose of the test is to characterize and calibrate these image sensors, and also to validate newly implemented scoring features of G4SEE. The characterized sensors are planned to be used as radiation environment monitors both at CERN and in space.

The beam requirements for the test campaign were following: a proton beam with a wide energy (and LET) range between 30 MeV – 300 MeV, very low proton beam flux: ideally 10–100 p/cm^2/s (but maximum 1000 p/cm^2/s), in general as low as possible due to the high sensitivity of the sensors, the total fluence over the whole test should be at least 10^6–10^7 p/cm^2, small beam size and as homogeneous and mono-directional beam as possible.

PARTREC facility

The RADNEXT proposal for the test campaign was approved already in 2023 and the Particle Therapy Research Center (PARTREC) facility in the Netherlands was assigned to the test campaign. At this facility they operate a large superconducting cyclotron for research purposes. The test campaign was postponed two times by the facility, and in the end cancelled by the facility due to technical issues with their accelerator cryogenic system.

In the figure 5.30 an illustration of the beam line set-up is shown, which was implemented as the simulation geometry as accurately as possible. The beam energy at the DUT position reaches 184 MeV. With usage of various thicknesses of aluminium degraders the beam energy can be decreased. [74]

I have run multiple simulations in preparation for this test campaign to check the expected E_{dep} and LET distributions to be measured experimentally. Some of the results are presented in section 5.3.3.

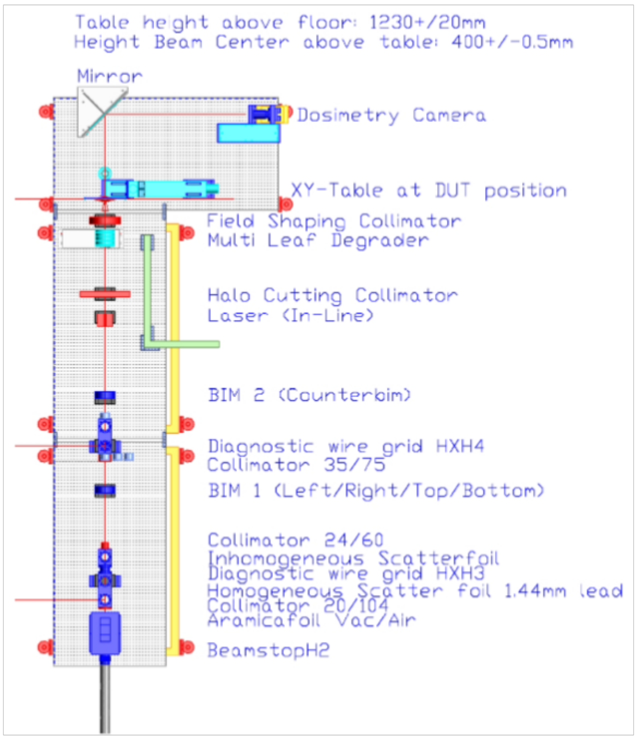


Figure 5.30: Image of beam line of facility for irradiation of materials beam line set-up, the beam enters from the bottom. [75]

PSI PIF facility

Since the test campaign at PARTREC was cancelled, another facility the PIF at PSI was assigned by the RADNEXT TA coordinators. This facility is designed for testing components for space applications. They can provide proton environment comparable to the one in space and mono-energetic beams for radiation hardness assurance tests. The proton beam (250 MeV) is delivered by the PROSCAN accelerator (superconducting cyclotron, also used for proton therapy). The primary energy degrader allows to have proton beam energy ranging from 74 MeV to 230 MeV right after the protons exit the accelerator. At the DUT position, the minimum beam energy is approximately 30 MeV, due to the second degrader placement in the beam. Figure 5.31 shows the hall with the accelerator and also the PIF area. The test campaign was supposed to happen in spring 2024, but unfortunately it had to be postponed due to international delivery problems of a DUT instrument on the way of shipping from Hungary to France.

5.3.2 Device Under Test

One of the DUTs will be the Basler camera, which could be used as a radiation environment monitor. Detailed description of Basler camera is given in following section 5.3.2, since for this device the G4SEE simulations (which results are shown in section 5.3.3), were performed.

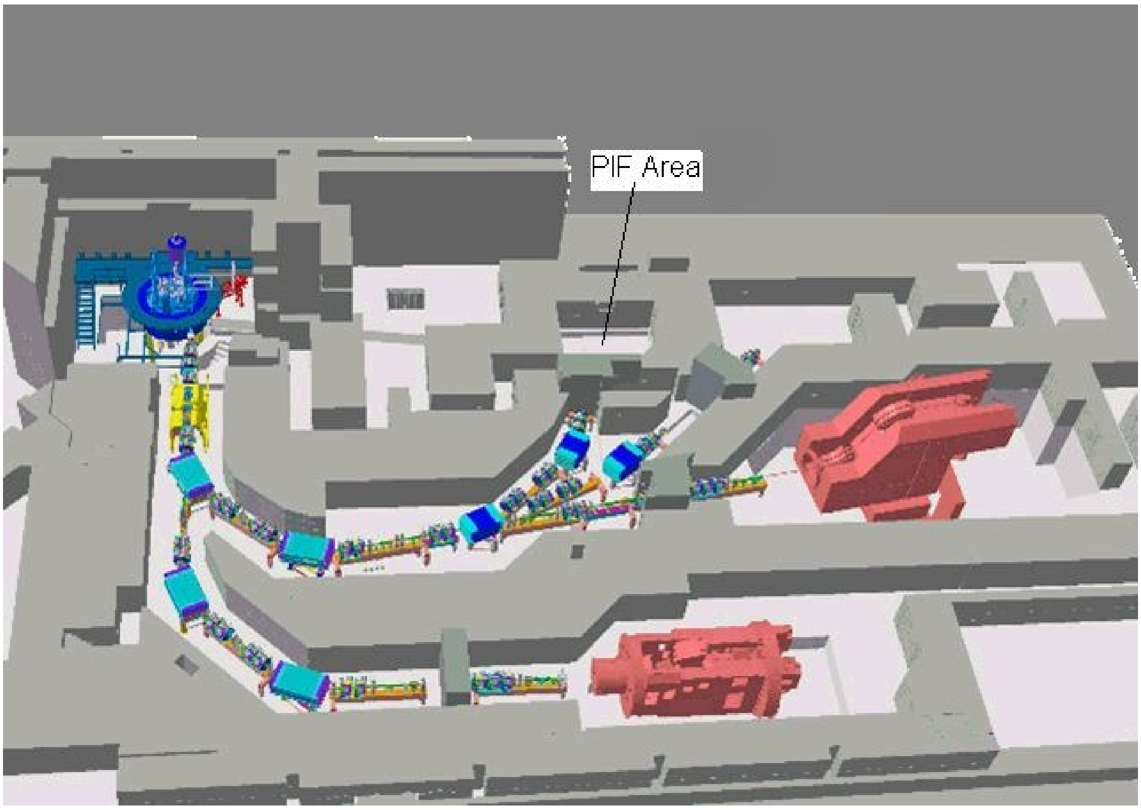


Figure 5.31: PIF-PROSCAN Hall with the PIF Experimental Area at PSI [76]

CMOS image sensors

Complementary Metal-Oxide Semiconductor (CMOS) image sensor (CIS) is a silicon pixel detector capable to capture digital images. Another such device is Charged-Coupled Device (CCD) that represents main competitor of CMOS sensors. The principle of CIS operation is conversion of optical signal into electric signal. Firstly, the incident photon is converted into electron-hole pairs, and secondly, the charge created by these pairs is converted into voltage. In CIS, the charge is converted into voltage in each pixel. Both CCD and CIS are based on MOS technology, MOSFET amplifiers are the main building blocks of a CIS. [77]

CMOS sensors started to be used in the early 1990s. Typically a photodiode is used to generate the charges at each pixel. Supporting circuitry can be located either in the same chip or next to each photodiode. This makes it cheaper to manufacture compared to CCD sensors. Also in comparison with CCDs, lower voltage is needed, which results in a lower power consumption. Due to the sensor design, where each pixel includes an amplifier, the noise is higher compared to CCDs. This is the reason why in the low noise applications rather CCDs are used instead of CIS. The architecture and the image of a CMOS sensor can be seen in figure 5.32. [77]

Basler camera

During the test campaign specifically the Basler CMOS a2A3840-13gmBAS [78] image sensor will be used. Basler [79] is a global producer of digital cameras for

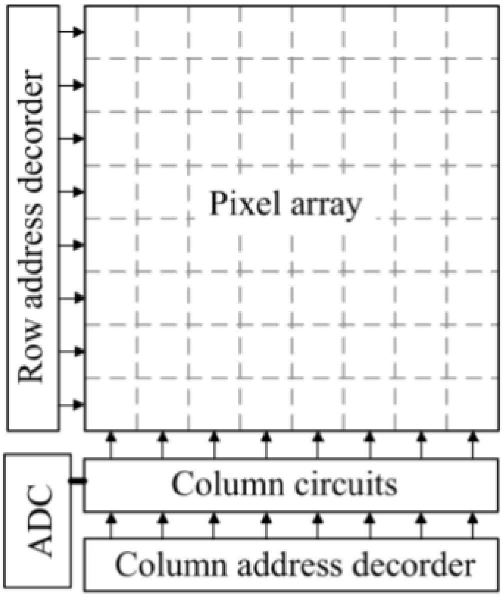


Figure 5.32: Architecture of CMOS image sensor (CIS) [77]

industrial applications. I was running simulations for this device and developed a data acquisition Python 3 script for it. Image of this specific camera is shown in figure 5.33.

The camera is equipped with a Sony IMX334 CMOS sensor, with size 7.7 mm×4.3 mm. The resolution is 8.3 MP (3860×2178 pixels), size of each pixel is 2 μm×2 μm. The frame rate is 12.4 fps. The sensor is capable to see the visible spectrum. The quantum efficiency is typically 72.2%, the dark noise is 1.96e^{-}, saturation capacity is 7.2 ke^{-}, dynamic range is 69.2 dB, and signal-to-noise ratio is 38.6 dB. The housing of the sensor is a 55.5×29×29 mm box, which weights around 100 g. The operating temperature ranges from -10 to 60 degrees Celsius. The Basler camera which will be used during the test campaign will be shielded from light with an aluminium foil placed in front of its Sony CMOS sensor.

The camera is accessible and can be controlled via the Basler pylon Camera Software Suite [80], or with the use of the pypylon Python package, which is an official open-source python wrapper for the Basler pylon Software Suite. With the use of this Python package, I developed a Python script for the data acquisition of the camera, where all the necessary camera and acquisition parameters can be set, and it also real-time displays captured images, monitors camera's temperature and plots the pixel value histogram of last frame. The displayed image and plots running the script can be seen in figure 5.34.

5.3.3 Simulations and Results

In the preparations for the experimental test campaign I performed many simulations. Their purpose was to get an idea of what we can expect from the experimental measurements. I ran and present here the simulations only for the PARTREC facility. For the PSI PIF, we have not yet received enough information to be able to perform

Figure 5.33: Photos of a Basler a2A3840-13gmBAS COTS industrial camera [78]

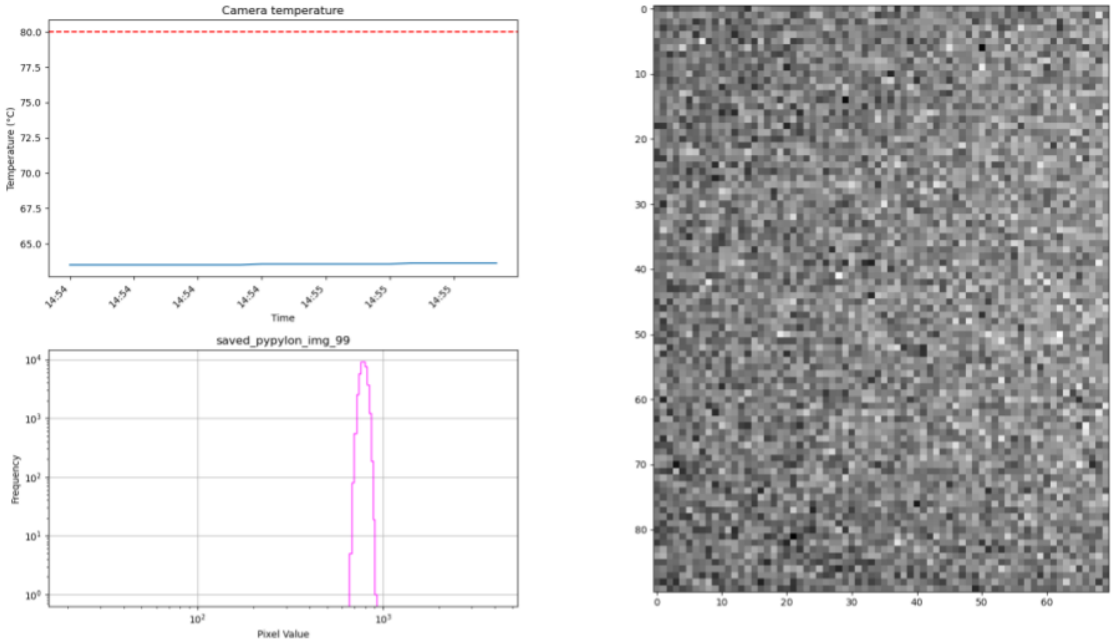


Figure 5.34: Screenshot of the visualized data read out from the camera during data acquisition (without any irradiation). In the top left plot, the temperature of the camera over time is displayed, and in the bottom left plot the histogram of pixel values is shown. On the right, the last image frame taken by the camera is displayed.

detailed simulation study which result would give us meaningful information, but it is expected to have similar results for PIF as for the PARTREC facility.

In the simulation geometry, I implemented as many components as accurately as possible based on the information received from the facility. I performed simulations for 3 μm, and 10 μm-thick SV. The primary beam energy was set to 189.9 MeV. To have proton beams with lower energy and higher LET during experiment, an aluminium degrader would be used, therefore I performed simulations with different aluminium degrader thicknesses. Results can be seen in figure 5.35.

During the test campaign the CMOS sensor's structure will be characterized, similar to the work described in [81]. This will be done by the rotation of the DUT with respect to the beam, so the effective thickness of the sensor's sensitive volume, from where deposited charge is read out changes in function of the incident angle. Simulations for different incident angles were therefore performed. In figure 5.36 results of these simulations are displayed.

One of the purposes of this test campaign is also to validate the newly implemented LET scoring. The LET values will be obtained similarly as is described in [81]. I run simulations for the different types of LET scoring for both different incident angles and different degrader thicknesses, which resulted in different proton energy and LET distributions at DUT position. In the figure 5.37 it can be seen a comparison of different LET scoring for 10° beam incident angle. The LET distributions scored per event or per step are almost identical in this case, which is caused by the fact that the proton made only one step in the SV, therefore the step length and track length in this case are almost identical.

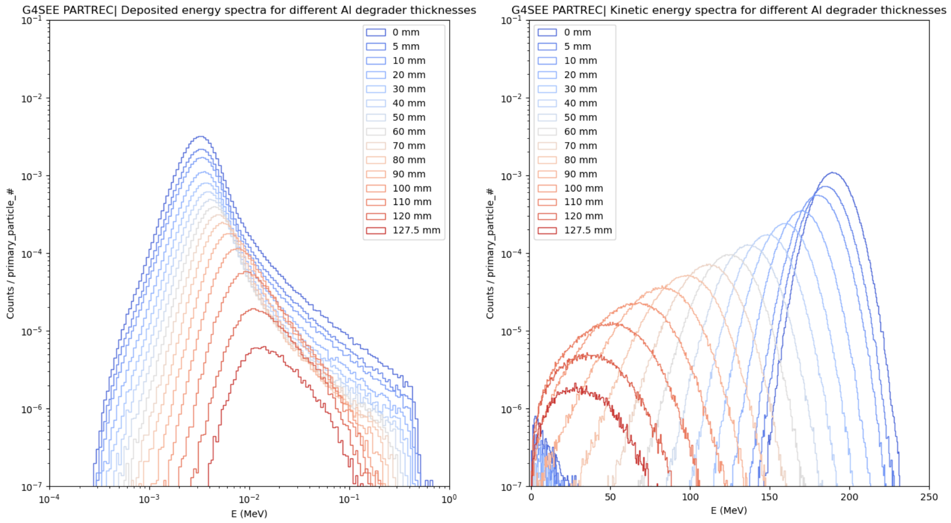


Figure 5.35: Comparison of deposited (left) and kinetic (right) energy by protons in 10 μm-thick SV for different aluminium thicknesses. Results shown are for 189.9 MeV beam initial energy, they were simulated for the test at PARTREC.

Once the test campaign will be performed all the simulation results will be compared with the experimental data to validate the new LET scoring of the G4SEE toolkit.

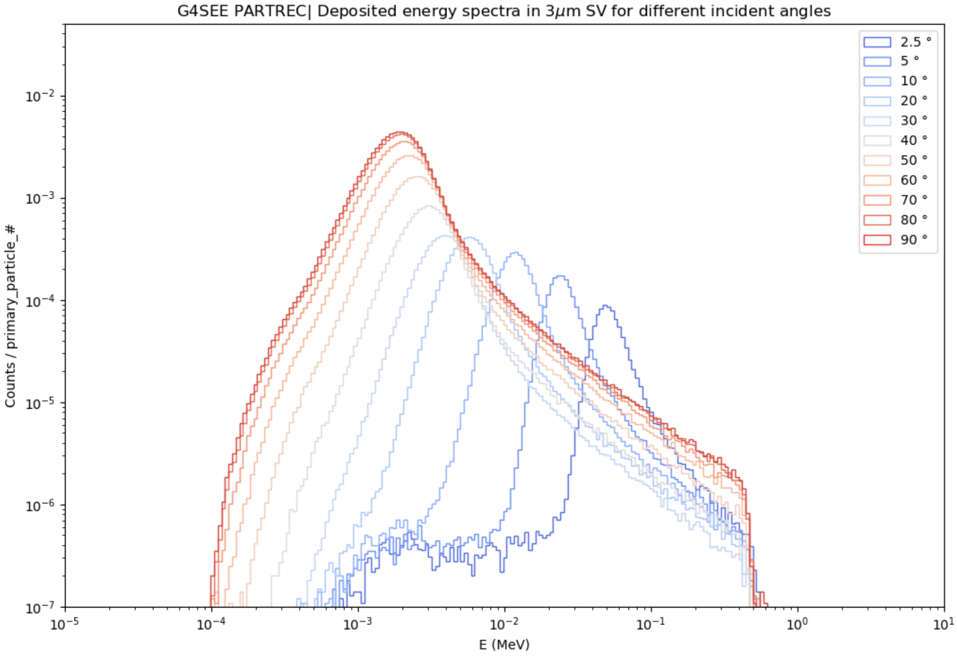


Figure 5.36: Comparison of deposited energy for different beam incident angles for 3 μm-thick SV. Results shown are for 189.9 MeV beam energy, they were simulated for the test at PARTREC.

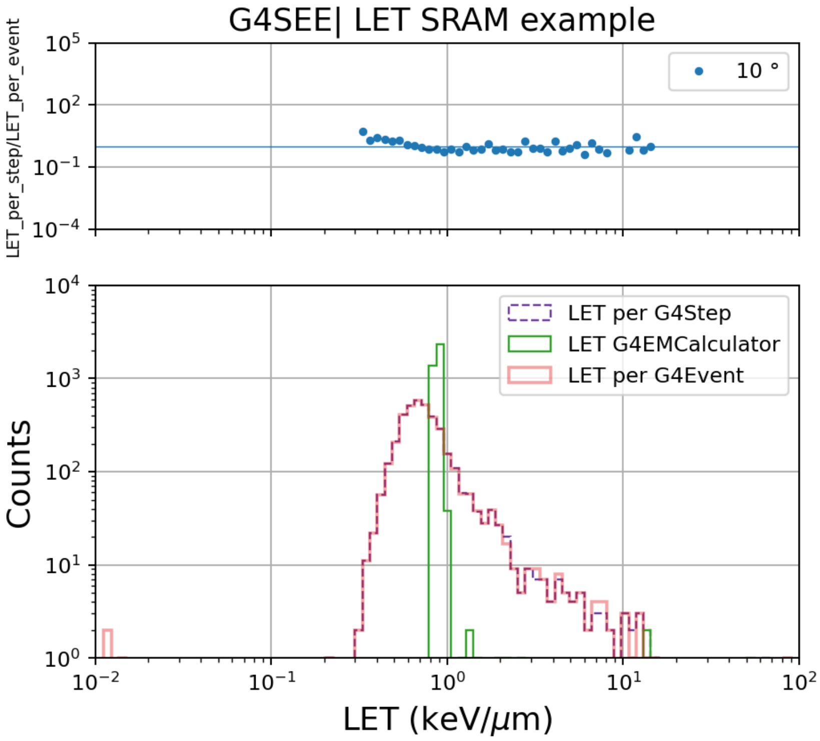


Figure 5.37: On the top plot is shown the ratio of LET computed per particle step to LET computed per one event. On the bottom plot is show a comparison of all three possible LET scoring for protons. Results shown are for 189.9 MeV beam energy and 10° incident angle, they were simulated for the test at PARTREC.

Chapter 6

Conclusions

The main purposes of this work were to implement new features in Geant4-based Single Event Effect simulation toolkit (G4SEE) and to validate these features. The minor changes and features in G4SEE compared to v0.5.1 are following. I have enabled the visualization of the geometry and simulation run and written documentation for it. Improvement of the post-processing script for the detailed scoring has been made, to include also the energy deposited by primary particles via direct ionization.

I have implemented three major features for G4SEE. The first one is Dose scoring. With this scoring users can get information not only about the total dose deposited in the Sensitive Volume, but also the total ionizing and non-ionizing doses are computed for the SV. Second implemented feature is LET scoring, for which three different options of computing the LET values were defined. One of these options computes the LET from the Geant4 EM tabulated data. The other two options compute LET by dividing the deposited energy of particle by its trajectory length in the SV. Two different approaches have been implemented: computing it event-by-event, and computing it particle-by-particle (per particle step). Lastly, energy deposition per interaction scoring was implemented as well. The improvement compared to the already implemented detailed scoring and post-processing is that instead of creating and post-processing large data files, G4SEE already creates histograms of deposited energy during simulation run, for all the different groups of secondary particles created together in an interaction and then scored in the SV. It saves a significant amount of disk space, time and made the use of post-processing script obsolete for this type of scoring.

As a validation study, simulations of diamond detector irradiated by 2.5 MeV and 14.04 MeV neutrons were performed. The simulation results were compared with experimental data taken during test campaign at PTB by collaborators. For the 2.5 MeV simulations the data correspond well with the experimental measurement. Small discrepancies are most probably caused by the energy resolution of the detector. For the 14.04 MeV simulation the NRESP71 model had to be enabled to get correct results for the α induced interactions on carbon nuclei. Two large discrepancies between measured and simulated data was found, one around 7 MeV also probably caused by the detector response, and the other can be caused by the missing interactions in physics models, or by electronic noise. The cause of this discrepancy should be further investigated. Apart from this discrepancy, the simulated data are in a good agreement with measured data.

Another simulation study for a SiC power MOSFET with 200 MeV and 3 MeV protons were carried out in collaboration with ETH Zurich APS laboratory. For this purpose, the simulations of deposited energy, detailed scoring simulations of interaction types and creator volumes, as well as LET simulations were performed. Two SV were examined. From the results it is clear that the SV below the devices' gate is in more danger to get damaged due to SEE, since more energy is deposited there and also more secondary heavy ions are created in this volume. For the 200 MeV proton beam the majority of interactions are inelastic interactions, whereas for 3 MeV protons the majority of interactions are hadron elastic interactions. Since LET is an important quantity for understanding the SEEs, simulations using LET scoring were also performed. All of these results are being shared and consulted with people at ETH Zurich, as part of the still ongoing collaboration project.

Furthermore, simulations have been made and camera image acquisition script has been developed in preparation for the upcoming experimental proton test campaign, which will be performed at PSI PIF facility. Purposes of the test campaign are to characterise CMOS image sensors under test, and to validate the new LET scoring feature of G4SEE by comparing simulation results with proton LET distributions measured using the camera. Unfortunately this test campaign was postponed for the date later than the submission of this thesis, so only the results of simulations in preparation for the experimental test campaign are presented.

Finally, in the near feature, changes and features implemented during this work will be merged into upcoming major release of the open-source G4SEE toolkit for the benefits of the whole radiation effects to electronics community.

List of Abbreviations

BEOL Back End Of Layers. 36, 65, 67, 69, 70, 74

CERN European Organisation for Nuclear Research. 8, 9, 11, 28, 31, 32, 35, 45, 77

CIS CMOS image sensor. 12, 77, 79, 80

CMOS Complementary Metal-Oxide Semiconductor. 12, 26, 29, 30, 33, 77, 79, 80, 82, 86

COTS commercial off-the-shelf. 11, 32, 33, 77

DUT Device Under Test. 10, 55, 65, 77, 78, 82

DXS Differential Cross Section. 14

EM Electromagnetic. 20, 37, 85

FIT Failure In Time. 27

G4 Geant4. 8, 11, 15, 20, 22–25, 33, 35, 36, 38, 41, 44, 45, 47, 48, 56–58, 85

G4SEE Geant4-based Single Event Effect simulation toolkit. 7–11, 33, 35–40, 43–47, 51, 53, 55–57, 60, 65–67, 77, 78, 85, 96, 99, 104

HP High Precision model. 37, 56, 57, 96

IC Integrated Circuit. 25, 26, 29, 31

LET Linear Energy Transfer. 8, 10, 11, 26, 27, 33, 34, 46–51, 64, 69, 71, 72, 76, 77, 82, 84–86, 104

LHC Large Hadron Collider. 28, 32

MBU Multiple Bit Upset. 28

MC Monte Carlo. 8, 9, 11, 13, 15–22, 33, 38, 45

MCU Multiple Cell Upset. 28

MOSFET metal-oxide-semiconductor field-effect transistor. 10, 11, 31, 63–66, 69, 79

MRED Monte Carlo Radiative Energy Deposition. 33

NILE Neutron Irradiation Laboratory for Electronics. 55

PARTREC Particle Therapy Research Center. 77, 81

PDF probability density function. 16, 17, 19, 21

PIF Proton Irradiation Facility. 77, 78, 80, 81, 86

PSI Paul Scherrer Institut. 77, 78, 80, 86

PTB Physikalisch-Technische Bundesanstalt. 54–57, 85

R2E Radiation to Electronics. 11, 32, 45, 57

RHA Radiation Hardness Assurance. 31, 32

ROI Region Of Interest. 18

SEB Single Event Burn-out. 31, 64, 69

SEDR Single Event Dielectric Rupture. 31

SEE Single Event Effect. 8, 9, 11, 25–28, 31–33, 39, 40, 45, 46, 63–65, 69, 72, 86

SEFI Single Event Functional Interrupt. 29

SEGR Single Event Gate Rupture. 31, 64

SEL Single Event Latch-up. 29–31

SET Single Event Transient. 29, 30

SEU Single Event Upset. 27–29, 33, 45

SRAM Static Random Access Memory. 28, 33, 43, 45, 50

SV Sensitive Volume. 28, 35, 36, 38, 39, 41, 44, 47–49, 65–75, 81–83, 85, 86, 96, 99

TID Total Ionizing Dose. 8, 27, 46, 47, 64, 99

TNID Total Non-Ionizing Dose. 46, 47, 99

WBG wide bandgap. 63

XS cross section. 38, 44

Bibliography

[1] M. H. Kalos and P. A. Whitlock. *Monte Carlo Methods*. Weinheim: Wiley-VCH, 2008.

[2] Nicholas Metropolis and S. Ulam. “The Monte Carlo Method”. In: *Journal of the American Statistical Association* 44.247 (1949). PMID: 18139350, pp. 335–341. DOI: 10.1080/01621459.1949.10483310. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1949.10483310. URL: ttps://www.tandfonline.com/doi/abs/10.1080/01621459.1949.10483310.

[3] Francesc Salvat Pujol. *An introduction to the Monte Carlo simulation of radiation transport*. Presentation at CERN Academic Training. September 12, 2022.

[4] OECD. *PENELOPE 2018: A code system for Monte Carlo simulation of electron and photon transport: Workshop Proceedings, Barcelona, Spain, 28 January – 1 February 2019*. en. Paris: Organisation for Economic Co-operation and Development, 2019. URL: https://www.oecd-ilibrary.org/nuclear-energy/penelope-2018-a-code-system-for-monte-carlo-simulation-of-electron-and-photon-transport_32da5043-en (visited on 02/27/2024).

[5] E. Mendoza et al. “IEEE Trans. Nucl. Science 61 (2014) 2357”. In: *IEEE Transactions on Nuclear Science* 61 (2014), p. 2357.

[6] Emilio Mendoza et al. “New Standard Evaluated Neutron Cross Section Libraries for the GEANT4 Code and First Verification”. In: *IEEE Transactions on Nuclear Science* 61.4 (Aug. 2014), pp. 2357–2364. ISSN: 0018-9499, 1558-1578. DOI: 10.1109/TNS.2014.2335538. URL: http://ieeexplore.ieee.org/document/6869047/ (visited on 03/23/2024).

[7] *Introduction to the Monte Carlo simulation of radiation transport*. Presentation at th Beginner course - ULB. May, 2022.

[8] Salvador García-Pareja, Antonio M. Lallena, and Francesc Salvat. “Variance-Reduction Methods for Monte Carlo Simulation of Radiation Transport”. In: *Frontiers in Physics* 9 (Oct. 2021), p. 718873. ISSN: 2296-424X. DOI: 10.3389/fphy.2021.718873. URL: https://www.frontiersin.org/articles/10.3389/fphy.2021.718873/full (visited on 03/10/2024).

[9] S. Agostinelli et al. “Geant4—a simulation toolkit”. en. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (July 2003), pp. 250–303. ISSN: 01689002. DOI: 10.1016/S0168-9002(03)01368-8. URL: https://linkinghub.elsevier.com/retrieve/pii/S0168900203013688 (visited on 03/16/2024).

[10] J. Allison et al. "Geant4 developments and applications". In: *IEEE Transactions on Nuclear Science* 53.1 (Feb. 2006), pp. 270–278. ISSN: 0018-9499. DOI: 10.1109/TNS.2006.869826. URL: http://ieeexplore.ieee.org/document/1610988/ (visited on 03/16/2024).

[11] J. Allison et al. "Recent developments in Geant4". en. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 835 (Nov. 2016), pp. 186–225. ISSN: 01689002. DOI: 10.1016/j.nima.2016.06.125. URL: https://linkinghub.elsevier.com/retrieve/pii/S0168900216306957 (visited on 03/16/2024).

[12] C. Ahdida et al. "New Capabilities of the FLUKA Multi-Purpose Code". In: *Frontiers in Physics* 9 (Jan. 2022), p. 788253. ISSN: 2296-424X. DOI: 10.3389/fphy.2021.788253. URL: https://www.frontiersin.org/articles/10.3389/fphy.2021.788253/full (visited on 03/16/2024).

[13] *Home | The official CERN FLUKA website.* URL: https://fluka.cern/ (visited on 03/16/2024).

[14] Giuseppe Battistoni et al. "Overview of the FLUKA code". en. In: *Annals of Nuclear Energy* 82 (Aug. 2015), pp. 10–18. ISSN: 03064549. DOI: 10.1016/j.anucene.2014.11.007. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306454914005878 (visited on 03/16/2024).

[15] Michael Evan Rising et al. *MCNP® Code Version 6.3.0 Release Notes*. Tech. rep. LA-UR-22-33103, Rev. 1. Los Alamos, NM, USA: Los Alamos National Laboratory, Jan. 2023. DOI: 10.2172/1909545. URL: https://www.osti.gov/biblio/1909545.

[16] *ENDF: Evaluated Nuclear Data File.* URL: https://www-nds.iaea.org/exfor/endf.htm (visited on 03/24/2024).

[17] *The JEFF Nuclear Data Library.* en. URL: https://www.oecd-nea.org/jcms/pl_27365/the-jeff-nuclear-data-library (visited on 03/24/2024).

[18] Chary Rangacharyulu et al. "An Integrated Approach to Selecting Materials for Fuel Cladding in Advanced High-Temperature Reactors". In: Aug. 2014.

[19] Dr. Lucas Norberto Burigo. *Introduction to Monte Carlo Particle Transport*. Presentation. https://www.dkfz.de/en/medizinische-physik-radioonkologie/Teaching/03_Introduction_MC_particle_transport.pdf. Heidelberg, Germany, May 2020.

[20] *Global Usage Classes — Book For Application Developers 11.2 documentation.* URL: https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/Fundamentals/global.html (visited on 03/24/2024).

[21] Geant4 Collaboration. *Geant4 Documentation.* 2024. URL: https://geant4.web.cern.ch/docs/.

[22] Marc Verderi. *The Geant4 Particle Transport Simulation Toolkit.* Presentation at CERN Academic Training. September 13, 2022.

[23] John Apostolakis. *What is Geant4, Introduction.* Presentation at Geant4 Training at CERN. May 25, 2021.

[24] Balaji Narasimham. "Challenges and Opportunities for Radiation Hardening in Advanced Technologies". In: *Proceedings of the 2021 IEEE Nuclear and Space Radiation Effects Conference (NSREC)*. IEEE. Virtual Conference, 2021.

[25] E. Petersen. *Single Event Effects in Aerospace*. New York, NY: Wiley - IEEE Press, 2011.

[26] Howard Tigelaar. *How Transistor Area Shrank by 1 Million Fold*. en. Cham: Springer International Publishing, 2020. ISBN: 9783030400200 9783030400217. DOI: 10.1007/978-3-030-40021-7. URL: http://link.springer.com/10.1007/978-3-030-40021-7 (visited on 03/12/2024).

[27] Ygor Aguiar. *Radiation Effects I: Single-Event Effects*. Presentation at Basic Introduction to Radiation Effects on Electronics School. June 8, 2023.

[28] Rubén García Alía. *Introduction 'I&' Radiation Environment*. Presentation at Basic Introduction to Radiation Effects on Electronics School. June 7, 2023. 2023.

[29] Daniel Gomez Toro. "Temporal Filtering with Soft Error Detection and Correction Technique for Radiation Hardening Based on a C-element and BICS". PhD thesis. Dec. 2014.

[30] European Space Agency. *SINGLE EVENT EFFECTS TEST METHOD AND GUIDELINES ESCC Basic Specification No. 25100*. Tech. rep. 2014.

[31] Federico Faccio. *Radiation Effects in the Electronics for CMS*. 2001.

[32] Fernanda Gusmão de Lima, Exame de Qualificação, and Ricardo Augusto da Luz Reis. "Single event upset mitigation techniques for programmable devices". In: *Porto Alegre* 14 (2000).

[33] Enrico Costenaro et al. "A Practical Approach to Single Event Transient Analysis for Highly Complex Design". en. In: *Journal of Electronic Testing* 29.3 (June 2013), pp. 301–315. ISSN: 0923-8174, 1573-0727. DOI: 10.1007/s10836-013-5385-9. URL: http://link.springer.com/10.1007/s10836-013-5385-9 (visited on 05/08/2024).

[34] H. Puchner et al. "Elimination of Single Event Latchup in 90nm SRAM Technologies". In: *2006 IEEE International Reliability Physics Symposium Proceedings*. ISSN: 1938-1891. Mar. 2006, pp. 721–722. DOI: 10.1109/RELPHY.2006.251342. URL: https://ieeexplore.ieee.org/document/4017283 (visited on 03/17/2024).

[35] Mario Sacristan Barbero. *Radiation Effects I: Single-Event Effects – Destructive events*. Presentation at Basic Introduction to Radiation Effects on Electronics School. June 8, 2023. 2023.

[36] Saulo G. Alberton et al. "Charge deposition analysis of heavy-ion-induced single-event burnout in low-voltage power VDMOSFET". In: *Microelectronics Reliability* 137 (2022), p. 114784. ISSN: 0026-2714. DOI: https://doi.org/10.1016/j.microrel.2022.114784. URL: https://www.sciencedirect.com/science/article/pii/S0026271422003080.

[37] Corinna Martinella. "Single-Event Radiation Effects in Silicon Carbide Power MOSFETs". PhD thesis. University of Jyväskylä, 2021. ISBN: 978-951-39-8726-8. URL: `https://jyx.jyu.fi/handle/123456789/76476?locale-attribute=en`.

[38] *R2E: Radiation to Electronics at CERN*. `https://r2e.web.cern.ch/`. Accessed: May 10, 2024.

[39] Ygor Aguiar et al. "Radiation to Electronics Impact on CERN LHC Operation: Run 2 Overview and HL-LHC Outlook". en. In: *Proceedings of the 12th International Particle Accelerator Conference* IPAC2021 (2021), 4 pages, 0.576 MB. ISSN: 2673-5490. DOI: `10.18429/JACOW-IPAC2021-MOPAB013`. URL: `https://jacow.org/ipac2021/doi/JACoW-IPAC2021-MOPAB013.html` (visited on 03/18/2024).

[40] S G Mashnik et al. "CEM03 and LAQGSM03—new modeling tools for nuclear applications". In: *Journal of Physics: Conference Series* 41 (May 2006), pp. 340–351. ISSN: 1742-6588, 1742-6596. DOI: `10.1088/1742-6596/41/1/037`. URL: `https://iopscience.iop.org/article/10.1088/1742-6596/41/1/037` (visited on 05/09/2024).

[41] Robert A. Reed et al. "Physical Processes and Applications of the Monte Carlo Radiative Energy Deposition (MRED) Code". In: *IEEE Transactions on Nuclear Science* 62.4 (2015), pp. 1441–1461. DOI: `10.1109/TNS.2015.2454446`.

[42] *Welcome to the CRÈME site — CREME-MC site*. URL: `https://creme.isde.vanderbilt.edu/` (visited on 05/08/2024).

[43] A.J. Tylka et al. "CREME96: A Revision of the Cosmic Ray Effects on Micro-Electronics Code". In: *IEEE Transactions on Nuclear Science* 44.6 (Dec. 1997), pp. 2150–2160. ISSN: 0018-9499, 1558-1578. DOI: `10.1109/23.659030`. URL: `https://ieeexplore.ieee.org/document/659030/` (visited on 03/23/2024).

[44] N. A. Dodds et al. "Charge Generation by Secondary Particles From Nuclear Reactions in BEOL Materials". In: *IEEE Transactions on Nuclear Science* 56.6 (Dec. 2009), pp. 3172–3179. ISSN: 0018-9499, 1558-1578. DOI: `10.1109/TNS.2009.2034160`. URL: `http://ieeexplore.ieee.org/document/5341385/` (visited on 03/05/2024).

[45] Brian D. Sierawski et al. "Impact of Low-Energy Proton Induced Upsets on Test Methods and Rate Predictions". In: *IEEE Transactions on Nuclear Science* 56.6 (Dec. 2009), pp. 3085–3092. ISSN: 0018-9499, 1558-1578. DOI: `10.1109/TNS.2009.2032545`. URL: `http://ieeexplore.ieee.org/document/5341357/` (visited on 03/08/2024).

[46] D. Lucsányi et al. "G4SEE: a Geant4-based Single Event Effect simulation toolkit and its validation through monoenergetic neutron measurements". In: *IEEE Transactions on Nuclear Science* 69.3 (Mar. 2022), pp. 273–281. DOI: `10.1109/TNS.2022.3149989`.

[47] *G4SEE: Toolkit for simulating radiation effects in electronics*. `https://cern.ch/g4see`. Accessed: May 10, 2024.

[48] *G4SEE Documentation*. URL: `https://g4see-docs.web.cern.ch/v0.5.1/index.html#g4see-documentation` (visited on 05/08/2024).

[49] *G4SEE toolkit's main repository in CERN GitLab.* URL: https://gitlab.cern.ch/g4see/g4see (visited on 03/21/2024).

[50] *Geant4 Material Database — Book For Application Developers 11.2 documentation.* URL: https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/Appendix/materialNames.html (visited on 03/28/2024).

[51] *Physics Reference Manual — Physics Reference Manual 11.2 documentation.* URL: https://geant4-userdoc.web.cern.ch/UsersGuides/PhysicsReferenceManual/html/index.html (visited on 05/09/2024).

[52] *Geant4: G4GeneralParticleSource.hh Source File.* URL: https://apc.u-paris.fr/~franco/g4doxy/html/G4GeneralParticleSource_8hh-source.html (visited on 05/08/2024).

[53] *G4SEE Community Forum.* URL: https://g4see-forum.web.cern.ch/ (visited on 03/21/2024).

[54] Matteo Cecchetto et al. "0.1–10 MeV Neutron Soft Error Rate in Accelerator and Atmospheric Environments". In: *IEEE Transactions on Nuclear Science* 68.5 (May 2021), pp. 873–883. ISSN: 0018-9499, 1558-1578. DOI: 10.1109/TNS.2021.3064666. URL: https://ieeexplore.ieee.org/document/9373367/ (visited on 03/21/2024).

[55] Matteo Cecchetto. "Experimental and Simulation Study of Neutron-Induced Single Event Effects in Accelerator Environment and Implications on Qualification Approach". PhD thesis. CERN, 2021. URL: https://cds.cern.ch/record/2809556 (visited on 03/21/2024).

[56] Andrea Coronetti. "Relevance and guidelines of radiation effect testing beyond the standards for electronic devices and systems used in space and at accelerators". eng. In: *JYU dissertations* (2021). URL: https://jyx.jyu.fi/handle/123456789/78496 (visited on 03/21/2024).

[57] *Geant4: G4LogicalVolume Class Reference.* URL: https://apc.u-paris.fr/~franco/g4doxy/html/classG4LogicalVolume.html (visited on 04/16/2024).

[58] *Geant4: G4Step Class Reference.* URL: https://apc.u-paris.fr/~franco/g4doxy/html/classG4Step.html (visited on 04/15/2024).

[59] *Linear energy transfer - ICRPaedia.* URL: http://icrpaedia.org/Linear_energy_transfer (visited on 03/11/2024).

[60] *Geant4: G4EmCalculator Class Reference.* URL: https://apc.u-paris.fr/~franco/g4doxy/html/classG4EmCalculator.html (visited on 04/11/2024).

[61] *Geant4: G4Track Class Reference.* URL: https://apc.u-paris.fr/~franco/g4doxy/html/classG4Track.html (visited on 04/16/2024).

[62] Maurizio Angelone and Claudio Verona. "Properties of Diamond-Based Neutron Detectors Operated in Harsh Environments". In: *Journal of Nuclear Engineering* 2.4 (2021), pp. 422–470. DOI: 10.3390/jne2040032. URL: https://www.mdpi.com/2673-4362/2/4/32.

[63] Christina Weiss et al. “Selective data analysis for diamond detectors in neutron fields”. In: *EPJ Web of Conferences* 146 (2017). Ed. by A. Plompen et al., p. 03004. ISSN: 2100-014X. DOI: 10.1051/epjconf/201714603004. URL: http://www.epj-conferences.org/10.1051/epjconf/201714603004 (visited on 04/17/2024).

[64] Carlo Cazzaniga et al. “Characterization Measurements of Compact Neutron Generators of the New NILE Facility”. In: *IEEE Transactions on Nuclear Science* 70.8 (Aug. 2023), pp. 1616–1624. ISSN: 0018-9499, 1558-1578. DOI: 10.1109/TNS.2023.3293797. URL: https://ieeexplore.ieee.org/document/10180224/ (visited on 03/03/2024).

[65] *CIVIDEC Instrumentation - CVD Diamond Technology applications.* URL: https://cividec.at/detectors-B1.html (visited on 03/29/2024).

[66] Khai Ngo. “TA02-14: Neutron response of EJ-276 and 7CLYC scintillators for characterization of the new NILE facility”. en. In: (May 2022). DOI: 10.5281/ZENODO.6594912. URL: https://zenodo.org/record/6594912 (visited on 03/30/2024).

[67] A.R. Garcia et al. “New physics model in GEANT4 for the simulation of neutron interactions with organic scintillation detectors”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 867 (2017), pp. 148–155. DOI: 10.1016/j.nima.2017.06.021.

[68] *ENDF/B-VI.8 Evaluated Nuclear Data Library.* URL: https://www.nndc.bnl.gov/endf-b6.8/ (visited on 03/30/2024).

[69] Luciano F. S. Alves et al. “SIC power devices in power electronics: An overview”. In: *2017 Brazilian Power Electronics Conference (COBEP)*. 2017, pp. 1–8. DOI: 10.1109/COBEP.2017.8257396.

[70] *Semiconductor devices.* en. URL: https://aps.ee.ethz.ch/research/devices.html (visited on 04/17/2024).

[71] Corinna Martinella, Ruben Garcia Alia, Roger Stark, et al. “Impact of Terrestrial Neutrons on the Reliability of SiC VD-MOSFET Technologies”. In: *IEEE Transactions on Nuclear Science* (Mar. 2021). DOI: 10.1109/TNS.2021.3065122.

[72] Pengwei Li et al. “Analysis of the influence of single event effects on the characteristics for SiC power MOSFETs”. In: *2017 Prognostics and System Health Management Conference (PHM-Harbin)*. 2017, pp. 1–4. DOI: 10.1109/PHM.2017.8079302.

[73] K. Niskanen et al. “Proton Irradiation-Induced Reliability Degradation of SiC Power MOSFET”. In: *IEEE Transactions on Nuclear Science* 70.8 (Aug. 2023), pp. 1838–1843. ISSN: 0018-9499, 1558-1578. DOI: 10.1109/TNS.2023.3242829. URL: https://ieeexplore.ieee.org/document/10038573/ (visited on 05/05/2024).

[74] *Particle Therapy Research Center (PARTREC).* en-GB. URL: https://umcgresearch.org/w/partrec (visited on 04/21/2024).

[75] Harry Kiewiet. *AGOR-Facility for IRradiation of Materials.* Customer information. Apr. 2015.

[76] *Description | PIF | Paul Scherrer Institut (PSI).* en. URL: `https://www.psi.ch/en/pif/description` (visited on 04/03/2024).

[77] Mukul Sarkar and Albert Theuwissen. *A Biologically Inspired CMOS Image Sensor.* en. Vol. 461. Studies in Computational Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. ISBN: 9783642349003 9783642349010. DOI: `10.1007/978-3-642-34901-0`. URL: `https://link.springer.com/10.1007/978-3-642-34901-0` (visited on 04/13/2024).

[78] *a2A3840-13gmBAS | Basler AG.* en. URL: `https://www.baslerweb.com/en/shop/a2a3840-13gmbas/` (visited on 04/12/2024).

[79] *Basler - Expert for Computer & Machine Vision.* en. Mar. 2024. URL: `https://www.baslerweb.com/en/` (visited on 04/21/2024).

[80] *pylon Software for image acquisition.* en. URL: `https://www.baslerweb.com/en/software/pylon/` (visited on 05/09/2024).

[81] Josua Florczak et al. "Calibration of the Deposited Energy in CMOS Imagers for Particle Detection on Nanosatellites". In: *IEEE Transactions on Nuclear Science* 70.8 (2023), pp. 1966–1972. DOI: `10.1109/TNS.2023.3274876`.

Appendix A

G4SEE input macro

In the following an example of G4SEE input macro for 500 μm-thick diamond detector irradiated by 14.04 MeV neutrons examples is shown. The biasing for neutron inelastic interactions is enabled in the SV. The HP model is used, since the neutron energy is below 20 MeV. Deposited energy is being scored for ten thousand primary particles in one thread. The command for enabling the NRESP71 model can be seen in this macro. Except for the standard scoring, commands for the detailed scoring are also defined.

```
#
# G4SEE Single Event Effect simulation toolkit
# ============================================
# SPDX-FileCopyrightText: © 2022 CERN for the benefit of the G4SEE
 ↪  Collaboration <https://cern.ch/g4see>
# SPDX-License-Identifier: CC-BY-4.0
#
#############################################
### Input macro file for G4SEE simulation ###
#############################################
/run/numberOfThreads                1
/run/printProgress                  1000

##############
### Material
# define diamond material           NAME       DENSITY      ELEMENTS
/SEE/material/addMaterial           Diamond    3.52 g/cm3   C  1

### Geometry
# BULK                              MATERIAL   WIDTH unit  THICK unit
 ↪  BIAS
/SEE/geometry/Bulk                  Diamond    4.5 mm      500 um
 ↪  true

# SV                                POS. unit  WIDTH unit  THICK unit
 ↪  BIAS
```

```
/SEE/geometry/SV                    0 0 0 mm   4.5 4.5 mm  499 um
 ↪  true

# BEOL                              MATERIAL   WIDTH unit  THICK unit
 ↪  BIAS    NAME
/SEE/geometry/BEOL/addLayer         G4_Au      4.5 mm      250 nm
 ↪  false   Gold
/SEE/geometry/BEOL/addLayer         G4_AIR     55 mm       13.4 mm
 ↪  false   Air1
/SEE/geometry/BEOL/addLayer         G4_Al      55 mm       1 mm
 ↪  false   Al
/SEE/geometry/BEOL/addLayer         G4_AIR     55 mm       20 cm
 ↪  false   Air2

##############
### Biasing
/SEE/biasing/biasParticle           neutron
/SEE/biasing/biasProcess            neutronInelastic
/SEE/biasing/biasProcess            nCapture
/SEE/biasing/biasFactor             1000

##############
### Physics
/SEE/physics/addPhysics             G4HadronElasticPhysicsHP
/SEE/physics/addPhysics             G4HadronPhysicsFTFP_BERT_HP
/SEE/physics/listProcesses

# Extra model for neutron-carbon interactions
/process/had/particle_hp/use_NRESP71_model true

##############
### Initialize
/run/initialize

#############
### Visualization
#/control/execute vis.mac

##############
### Primaries (General Particle Source)
/gps/verbose                        0
/gps/particle                       neutron
/gps/ene/type                       Gauss
/gps/ene/mono                       14.04 MeV
/gps/ene/sigma                      148 keV
/gps/pos/centre                     0 0 21.34 cm
/gps/pos/type                       Point
```

```
/gps/ang/type                       iso
/gps/ang/mintheta                   0 deg
/gps/ang/maxtheta                   0.5801 deg
/gps/ang/minphi                     0 deg
/gps/ang/maxphi                     360 deg

##############
### Scoring
# Standard scoring - Energy deposited
/SEE/scoring/addScoring             1       Edep
/SEE/scoring/setHistogram           lin     0 MeV     20 MeV  100
/SEE/scoring/addScoring             2       Edep
/SEE/scoring/setHistogram           log     100 eV    20 MeV  100
/SEE/scoring/dumpHistogramsAfter    1000

# Detailed scoring
/SEE/scoring/detailed               True
/SEE/scoring/detailed/printPrimary  True
/SEE/scoring/detailed/setCSVFormat  True
/SEE/scoring/detailed/addTrack      True
/SEE/scoring/detailed/addParent     True
/SEE/scoring/detailed/addEkin       True
/SEE/scoring/detailed/addPosition   True
/SEE/scoring/detailed/addMomentum   True
/SEE/scoring/detailed/addEdep       True
/SEE/scoring/detailed/addNdep       True
/SEE/scoring/detailed/addZ          True
/SEE/scoring/detailed/addA          True
/SEE/scoring/detailed/addVolume     True
/SEE/scoring/detailed/addCounts     True
/SEE/scoring/detailed/addProcess    True
/SEE/scoring/detailed/addEexc       True

##############
### Run
/run/beamOn                         10000
```

Appendix B

Dose scoring

Below it can be seen a main part of the G4SEE source code, where dose scoring is implemented. The `SenstiveVolume` and `RunAction` class are shown. In the `ProcessHits` function in `SensitiveVolume` class the total, the total ionizing (TID) and the non-ionizing (TNID) doses are summed per event. In the `AddEdep` function in `RunAction` class these values are added to the run. At the end of a run, in `EndOfRunAction` class the mass of the SV is obtained and with the corresponding energies deposited the doses are calculated.

```
// G4SEE Single Event Effect simulation toolkit
// ============================================
// SPDX-FileCopyrightText: © 2022 CERN for the benefit of the G4SEE
 ↪  Collaboration <https://cern.ch/g4see>
//
// This software is distributed under the terms of the GNU General
 ↪  Public License version 3
// (GPL Version 3) or any later version, copied verbatim in the file
 ↪  "LICENSES/GPL-3.0-or-later.txt".
//
// In applying this license, CERN does not waive the privileges and
 ↪  immunities granted to it
// by virtue of its status as an Intergovernmental Organization or
 ↪  submit itself to any jurisdiction.
//
// This software uses Geant4 developed by Members of the Geant4
 ↪  Collaboration (https://cern.ch/geant4).
//
// SPDX-License-Identifier: GPL-3.0-or-later
//
// Author: Dávid Lucsányi (CERN) <david.lucsanyi@cern.ch>
//         Eva Fialová <eva.fialova@cern.ch>
//
##############################
### Sensitive Volume class ###
##############################
```

```
...
G4bool SensitiveVolume::ProcessHits(G4Step* aStep,
 ↪  G4TouchableHistory*)
{

// get total, ioizing and non-ionizing Edep
G4double edep = aStep->GetTotalEnergyDeposit();
G4double nedep = aStep->GetNonIonizingEnergyDeposit();
G4double iedep = edep - nedep;
G4Track* track = aStep->GetTrack();

G4String process;
const G4VProcess* creatorproc = track->GetCreatorProcess();
if(creatorproc)
{
    process = creatorproc->GetProcessName();
}
else
{
    process = "primary";
}

ParticleHit* newHit = new ParticleHit();

G4int eventid = static_cast<TrackInformation*>(track->GetUserInforma⌋
 ↪  tion())->GetEventID();

// track level information
newHit->SetEventID(eventid);
newHit->SetTrackID(track->GetTrackID());
newHit->SetParentID(track->GetParentID());
newHit->SetName(track->GetParticleDefinition()->GetParticleName());
newHit->SetZ(track->GetParticleDefinition()->GetAtomicNumber());
newHit->SetA(track->GetParticleDefinition()->GetAtomicMass());
newHit->SetProcess(process);
newHit->SetVolume(track->GetLogicalVolumeAtVertex()->GetName());
// step level information
newHit->SetEdep(edep);
newHit->SetNdep(aStep->GetNonIonizingEnergyDeposit());
newHit->SetEkin(aStep->GetPreStepPoint()->GetKineticEnergy());
newHit->SetPos(aStep->GetPreStepPoint()->GetPosition());
newHit->SetMom(aStep->GetPreStepPoint()->GetMomentum());

G4double weight = track->GetWeight();
newHit->SetWeight(weight);

if(fWeightPerEvent != 1. && weight != fWeightPerEvent)
```

```
    fWeightPerEvent = std::min(fWeightPerEvent, weight);
else
    fWeightPerEvent = weight;

// SUM PER EVENT
fEdepPerEvent += edep;
fNEdepPerEvent += nedep;
fIoEdepPerEvent += iedep;
//  newHit -> Print();

// Get hit for total accounting
auto hitTotal = (*fHitsCollection)[fHitsCollection->entries() - 1];

// Add values
hitTotal->Add(edep, nedep, iedep);

return true;

}



##############################
###    Run Action class    ###
##############################

/////////////////////////////////////////////////////
RunAction::RunAction(const G4String& outputDir) :
        G4UserRunAction(),
        fDoseScoring(false),
        fOutputDir(outputDir)
{

// add new units for dose scoring
//
const G4double milligray = 1.e-3 * gray;
const G4double microgray = 1.e-6 * gray;
const G4double nanogray = 1.e-9 * gray;
const G4double picogray = 1.e-12 * gray;
new G4UnitDefinition("milligray", "milliGy", "Dose", milligray);
new G4UnitDefinition("microgray", "microGy", "Dose", microgray);
new G4UnitDefinition("nanogray", "nanoGy", "Dose", nanogray);
new G4UnitDefinition("picogray", "picoGy", "Dose", picogray);

fRunMessenger = new RunMessenger(this);

if(fDoseScoring)
```

```
    CreateFile();

#if defined(G4MULTITHREADED) && defined(APP_MULTITHREADED)
        fTHREAD = G4Threading::G4GetThreadId();
#else
        fTHREAD = 0;
#endif

}

...

/////////////////////////////////////////////////////
void RunAction::BeginOfRunAction(const G4Run* aRun)
{

G4int run_number = aRun->GetRunID();
G4cout << "### Run " << run_number << " start." << G4endl;

fEdep = 0;
fNEdep = 0;
fIoEdep = 0;

fDose = 0;
fNDose = 0.;
fIoDose = 0.;

fNumberEv = 0.;

// inform the runManager to save random number seed
// G4RunManager::GetRunManager()->SetRandomNumberStore(false);

}

/////////////////////////////////////////////////////
void RunAction::EndOfRunAction(const G4Run* aRun)
{

G4cout << "Number of events = " << aRun->GetNumberOfEvent() <<
 ↪  G4endl;
fNumberEv = aRun->GetNumberOfEvent();

// Compute dose = total energy deposit in a run
//
G4double edep = fEdep;
G4double nedep = fNEdep;
G4double ioedep = fIoEdep;
```

```
// get the mass of SV
const DeviceConstruction* deviceConstruction = static_cast<const
 ↪  DeviceConstruction*>(
    G4RunManager::GetRunManager()->GetUserDetectorConstruction());
G4double mass = deviceConstruction->GetScoringVolume()->GetMass();

G4double dose = edep / mass;
G4double ndose = nedep / mass;
G4double iodose = ioedep / mass;

fDose = dose;
fNDose = ndose;
fIoDose = iodose;

if(fDoseScoring)
    CreateFile();

}

/////////////////////////////////////////////////////
void RunAction::AddEdep(G4double edep, G4double nedep, G4double
 ↪  ioedep)
{

fEdep += edep;
fNEdep += nedep;
fIoEdep += ioedep;

}
```

Appendix C

LET scoring

Below it can be seen a main part of the G4SEE source code, where LET scoring is implemented. The SenstiveVolume class is shown. There the LET2 value computes the LET from the G4EmCalculator class, LET3 computes the LET dividing the particle deposited energy by its track length per each step. The LET value per event is computed in the EndOfEvent() function, dividing deposited energy summed over an event by the primary particle track length for that event.

```
// G4SEE Single Event Effect simulation toolkit
// ============================================
// SPDX-FileCopyrightText: © 2022 CERN for the benefit of the G4SEE
↪  Collaboration <https://cern.ch/g4see>
//
// This software is distributed under the terms of the GNU General
↪  Public License version 3
// (GPL Version 3) or any later version, copied verbatim in the file
↪  "LICENSES/GPL-3.0-or-later.txt".
//
// In applying this license, CERN does not waive the privileges and
↪  immunities granted to it
// by virtue of its status as an Intergovernmental Organization or
↪  submit itself to any jurisdiction.
//
// This software uses Geant4 developed by Members of the Geant4
↪  Collaboration (https://cern.ch/geant4).
//
// SPDX-License-Identifier: GPL-3.0-or-later
//
// Author: Dávid Lucsányi (CERN) <david.lucsanyi@cern.ch>
//         Eva Fialová <eva.fialova@cern.ch>
//
##############################
### Sensitive Volume class ###
##############################
...
```

```
/////////////////////////////////////////////////////
G4bool SensitiveVolume::ProcessHits(G4Step* aStep,
 ↪  G4TouchableHistory*)
{

// energy deposit
G4double edep = aStep->GetTotalEnergyDeposit(); /// keV;

// track
G4Track* track = aStep->GetTrack();
G4StepPoint* preStepPoint = aStep->GetPreStepPoint();
G4String volume = track->GetLogicalVolumeAtVertex()->GetName();

G4ThreeVector test =
    aStep->GetPreStepPoint()->GetPosition() -
     ↪  aStep->GetPostStepPoint()->GetPosition();

G4ThreeVector boundaryPoint;

if(preStepPoint->GetStepStatus() == fGeomBoundary)
{
    boundaryPoint = preStepPoint->GetPosition();
}

// Track end point coordinates
G4ThreeVector track_position_SV = track->GetPosition();
// Entering point of sv coordinates
G4ThreeVector x_SV = boundaryPoint - track_position_SV;
// Track length in SV
G4double track_SV = sqrt(x_SV.x() * x_SV.x() + x_SV.y() * x_SV.y() +
 ↪  x_SV.z() * x_SV.z());

...

// step length
G4double stepLength = 0.;
if(aStep->GetTrack()->GetDefinition()->GetPDGCharge() != 0.)
{
    stepLength = aStep->GetStepLength();
}

// LET per step length
G4double LET3 = edep / stepLength;

// LET from G4Em calculator
G4EmCalculator emCal;
```

```
G4double ekin = aStep->GetPreStepPoint()->GetKineticEnergy();
G4String particle =
 ↪  track->GetParticleDefinition()->GetParticleName();
G4Material* material = aStep->GetTrack()->GetMaterial();
G4String mat = material->GetName();
G4double LET1 = emCal.ComputeTotalDEDX(ekin, particle, mat) / (keV /
 ↪  um);

G4String process;
const G4VProcess* creatorproc = track->GetCreatorProcess();
if(creatorproc)
{
    process = creatorproc->GetProcessName();
}
else
{
    process = "primary";
}

ParticleHit* newHit = new ParticleHit();

G4int eventid = static_cast<TrackInformation*>(track->GetUserInforma⌋
 ↪  tion())->GetEventID();

// track level information
newHit->SetEventID(eventid);
newHit->SetTrackID(track->GetTrackID());
newHit->SetParentID(track->GetParentID());
newHit->SetName(track->GetParticleDefinition()->GetParticleName());
newHit->SetZ(track->GetParticleDefinition()->GetAtomicNumber());
newHit->SetA(track->GetParticleDefinition()->GetAtomicMass());
newHit->SetProcess(process);
newHit->SetVolume(track->GetLogicalVolumeAtVertex()->GetName());

// step level information
newHit->SetEdep(edep);
newHit->SetLET1(LET1);
newHit->SetLET3(LET3);
newHit->SetNdep(aStep->GetNonIonizingEnergyDeposit());
newHit->SetEkin(aStep->GetPreStepPoint()->GetKineticEnergy());
newHit->SetPos(aStep->GetPreStepPoint()->GetPosition());
newHit->SetMom(aStep->GetPreStepPoint()->GetMomentum());

G4double weight = track->GetWeight();
newHit->SetWeight(weight);

if(fWeightPerEvent != 1. && weight != fWeightPerEvent)
```

```
    fWeightPerEvent = std::min(fWeightPerEvent, weight);
else
    fWeightPerEvent = weight;

// SUM PER EVENT
fEdepPerEvent += edep;

// FINAL TRACK LENGTH AT THE END OF EVENT
if(track->GetParticleDefinition()->GetParticleName() ==
 ↪  fInitialParticleName)
{
    fTrack = track_SV;
}

fHitsCollection->insert(newHit);

return true;

}

/////////////////////////////////////////////////////
void SensitiveVolume::EndOfEvent(G4HCofThisEvent*)
{

G4int entries = fHitsCollection->entries();
std::vector<ParticleHit*>* vec = fHitsCollection->GetVector();
ParticleHit* hit = NULL;

/// STANDARD SCORING
std::map<G4String, Scoring*>::iterator it;
for(it = fScoringMap.begin(); it != fScoringMap.end(); ++it)
{
    Scoring* fScoringElement = it->second;

    /// STANDARD SCORING: Edep per event scoring
    if(fScoringElement->IsEdepScoringEnabled())
    {
        fScoringElement->AddToHistogram(fEdepPerEvent, 1. *
         ↪  fWeightPerEvent);
    }

    /// STANDARD SCORING: LET per Event scoring
    if(fScoringElement->IsLETScoringEnabled())
    {
        fScoringElement->AddToHistogram(fEdepPerEvent / fTrack, 1. *
         ↪  fWeightPerEvent);
    }
```

```
///  STANDARD SCORING: LET G4EMCalculator
if(fScoringElement->IsLET2ScoringEnabled())
{
    std::set<G4int> fParticleAlreadyScored;
    for(G4int i = 0; i < entries; i++)
    {
        hit = vec->at(i);
        G4String particleName = hit->GetName();

        if(particleName.compare(fScoringElement->GetLETScoringPa⌋
         ↪  rticle()) ==
         ↪  0)
        {
            G4int trackID = hit->GetTrackID();
            if(fParticleAlreadyScored.find(trackID) ==
             ↪  fParticleAlreadyScored.end())
            {
                fScoringElement->AddToHistogram(hit->GetLET2(),
                 ↪  1. * hit->GetWeight());
                fParticleAlreadyScored.insert(trackID);
            }
        }
    }
}

// STANDARD SCORING: LET per step
if(fScoringElement->IsLET3ScoringEnabled())
{
    std::set<G4int> fParticleAlreadyScored;
    for(G4int i = 0; i < entries; i++)
    {
        hit = vec->at(i);
        G4String particleName = hit->GetName();
        if(particleName.compare(fScoringElement->GetLETScoringPa⌋
         ↪  rticle())==0)
        {
            G4int trackID = hit->GetTrackID();
            if(fParticleAlreadyScored.find(trackID) ==
             ↪  fParticleAlreadyScored.end())
            {
                fScoringElement->AddToHistogram(hit->GetLET3(),
                 ↪  1. * hit->GetWeight());
                fParticleAlreadyScored.insert(trackID);
            }
        }
    }
```

```
        }

        ...
    }

    }
```