A Geant4/Garfield++ and Geant4/Degrad Interface for the Simulation of Gaseous Detectors

Dorothea Pfeiffer[a,b,*], Lennert De Keukeleere[c], Carlos Azevedo[d], Francesca Belloni[e], Stephen Biagi[f], Vladimir Grichine[g], Leendert Hayen[c], Andrei R. Hanu[h], Ivana Hřivnáčová[i], Vladimir Ivanchenko[b,j], Vladyslav Krylov[k,l], Heinrich Schindler[b], Rob Veenhof[b,m]

[a] *European Spallation Source (ESS AB),P.O. Box 176, SE-22100 Lund, Sweden*
[b] *CERN, CH-1211 Geneva 23, Switzerland*
[c] *Instituut voor Kern- en Stralingsfysica, KU Leuven, Belgium*
[d] *I3N - Physics Department, University of Aveiro, 3810-193 Aveiro, Portugal*
[e] *CEA Saclay, 91191 Gif-sur-Yvette, France*
[f] *Department of Physics, University of Liverpool, UK*
[g] *Lebedev Physical Institute of RAS, Moscow, Russia*
[h] *NASA Goddard Space Flight Center, Greenbelt, Maryland 20771, USA*
[i] *Institut de Physique Nucléaire, Université Paris-Sud, CNRS-IN2P3, Orsay, France*
[j] *Tomsk State University, Tomsk, Russia*
[k] *Laboratoire de l'Accélérateur Linéaire (LAL), Université Paris-Sud XI, CNRS/IN2P3, 91898 Orsay, France*
[l] *Taras Shevchenko National University of Kyiv (TSNUK), Kyiv, Ukraine*
[m] *Uludag University, 16059 Nülufer-Bursa, Turkey*

Abstract

For several years, attempts have been made to interface Geant4 and other software packages with the aim of simulating the complete response of a gaseous particle detector. The present paper illustrates different possibilities to interface Geant4 with two such packages, Garfield++ and Degrad. The basic idea is to use the Geant4 physics parameterization feature, and to implement a Garfield++ or Degrad based detector simulation as an external model. With the Geant4/Degrad interface, detailed simulations of the X-ray interaction in gaseous detectors, including shell absorption by photoelectric effect, subsequent Auger shake-off, and fluorescence emission, become possible. The Geant4/Garfield++ interface can be used for photons and charged

*Corresponding author
Email address: `dorothea.pfeiffer@cern.ch` (Dorothea Pfeiffer)

particles of all kinetic energies. Depending on the particular physics case, either the Geant4 PAI model, the Heed PAI model or both Geant4 and Heed are responsible for primary ionization and the production of the conduction electrons. For the case in which the Geant4 PAI model is used in conjunction with the Heed PAI model, a more detailed analysis is performed. Parameters, such as the lower production cut of the PAI model and the lowest electron energy limit of the physics list have to be set correctly. The paper demonstrates how to determine these parameters with the help of the W value and Fano factor of the gas mixture. The simulation results of this Geant4/Heed PAI model interface are then verified against the results obtained with the standalone software packages.

Keywords: Gaseous Detectors, Monte-Carlo Simulation, Particle interactions, Software engineering, Geant4

1. Introduction

Geant4 (GEometry ANd Tracking) [1–3] is an object oriented C++ toolkit for the simulation of the passage of particles through matter. In particle physics, Geant4 is the most commonly used software package for Monte Carlo simulations. However, its application areas also include nuclear and accelerator physics, as well as studies in medical and space science. In recent years, the framework has been extended to include low energy applications, and several extensions from different fields have been added. The NXSG4 package [4], for example, adds functionality for Polycrystalline Neutron Scattering in Geant4, whereas the DNA project [5, 6] extends Geant4 with processes for the modeling of early biological damage induced by ionizing radiation at the DNA scale.

This paper will focus on the simulation of gaseous detectors and presents an approach to interface Geant4 with Garfield++ and Degrad, using the Geant4 parameterization feature [7]. The Geant4/Garfield++ interface, as described in this paper, has been integrated into the European Spallation Source Geant4 framework [8], and into Geant4 VMC [9]. In Geant4, the G4PAIModel (photo absorption ionization) [10] and the G4PAIPhotonModel (photo absorption ionization photon model) [11] were designed for the simulation of gaseous detectors. Both models are derived from the work of Allison and Cobb [12] and emerge from the assumption that the interaction of relativistic particles with a gas is predominated by soft collisions, in which a virtual photon is absorbed by the atom as a whole. As a result, the calculated energy loss in these models is mainly determined by the photon absorption cross section of the individual atoms. In addition, it gives rise to two main regions with respect to the energy of the ionization electrons: the resonance region, where low-energy conduction electrons are created, and the pseudo-free region, in which the ionization electron receives a large amount of energy and can therefore be described by Rutherford scattering on a nearly free electron [12].The difference between the G4PAIModel and the G4PAIPhotonModel is that the latter also produces photons. Although the resulting differences are not substantial, the G4PAIPhotonModel more accurately describes the spatial charge distribution [11][1].

Traditionally, the Geant4 PAI model was designed for fast charged particles in thin absorbers [13], but has recently been extended to low energy regions [14]. The PAI model is able to correctly describe the number and positions of the initial electron-ion pairs, but it does not include additional processes such as attachment, recombination, diffusion and fluorescence. Furthermore, although electromagnetic fields are implemented in Geant4, it is not possible to simulate the process of an electron avalanche or the signal created on a wire. This emphasizes the necessity for an interface between Geant4 and a software package that can simulate the above-mentioned features of gaseous detectors.

Dedicated to the simulation of gaseous detectors such as Micro Pattern Gaseous Detectors (MPGD) [15] is the Fortran software package Garfield [16]. The software has been ported to C++ by the name of Garfield++ [17], and makes extensive use of the ROOT framework [18] for data analysis and visualization. For several years, attempts have been made to interface Geant4 and Garfield [19], a task which has been now hugely facilitated by the availability of Garfield++. Garfield++ accepts two- and three-dimensional field maps computed by various finite element programs as a basis for its calculations of drift and avalanche processes. The finite element technique can handle complex electrode shapes as well as dielectrics. For the computation of electron transport properties in nearly arbitrary gas mixtures, Garfield++ includes an interface to Magboltz [20]. Furthermore, Heed [21], a PAI model implementation [22] similar to that in Geant4, can be used to create the

[1]Unless explicitly stated, the term PAI model in this paper stands for both G4PAIModel and G4PAIPhotonModel model

initial electron-ion pairs.

Even more details concerning ionization, excitation levels and vibrational modes can be obtained by Degrad, another Fortran program [23]. Degrad includes an accurate Auger cascade model for the interaction of photons, electrons and ionizing particles with gas mixtures in electric and magnetic fields. For X-rays, the software automatically simulates shell absorption by the photoelectric effect, Compton scattering or pair-production and the subsequent Auger, Coster-Kronig, Shake-off and fluorescence emission. Bremsstrahlung emissions by secondary electrons are also included [24]. Degrad is hence the most complete software package available on the market to simulate the interaction of particles with various gases.

2. Division of tasks between Geant4, Garfield++ and Degrad

In the present section, possible scenarios for interfacing the different software packages will be summarized and discussed. Next, implementation details for each of these cases will be described.

Features like the interaction of high-energy particles with matter, or complex detector configurations containing various non-gaseous materials, can neither be implemented in Garfield++ nor Degrad. Geant4 is thus always used for the primary particle generation, the interaction of the primary particle with the detector material, and the possible creation of secondary particles in the detector material. Garfield++ on the other hand deals with the drift of ions and electrons, the electron avalanche creation and amplification and finally the signal generation and readout of the signal. But for the first interaction of the primary particles with the gas, and the subsequent production of electron-ion pairs in the gas, several possibilities exist [2]. Figure 1 shows a schematic overview of the different options of interfacing Geant4, Garfield++ and Degrad for charged particles and photons. As Heed is not thread-safe, only the Geant4 sequential mode can be used in combination with Heed.

- *A) Geant4 for all particles*
  The PAI model is used in Geant4 to create all electron-ion pairs in the

[2]The term primary particle will be used in the following text for a particle that enters the gas volume before interacting with the gas. This particle can either be identical to the primary particle fired with the Geant4 particle gun, or it can be a secondary particle, that was produced in the detector material.

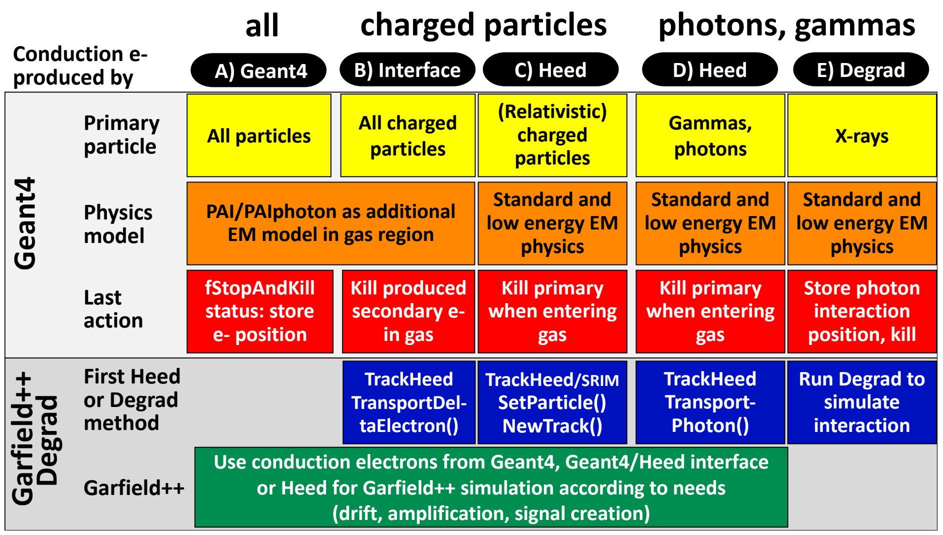

all
charged particles
photons, gammas
Conduction e- produced by
A) Geant4
B) Interface
C) Heed
D) Heed
E) Degrad
Geant4
Primary particle
All particles
All charged particles
(Relativistic) charged particles
Gammas, photons
X-rays
Physics model
PAI/PAIphoton as additional EM model in gas region
Standard and low energy EM physics
Standard and low energy EM physics
Standard and low energy EM physics
Last action
fStopAndKill status: store e- position
Kill produced secondary e- in gas
Kill primary when entering gas
Kill primary when entering gas
Store photon interaction position, kill
Garfield++ Degrad
First Heed or Degrad method
TrackHeed TransportDeltaElectron()
TrackHeed/SRIM SetParticle() NewTrack()
TrackHeed TransportPhoton()
Run Degrad to simulate interaction
Garfield++
Use conduction electrons from Geant4, Geant4/Heed interface or Heed for Garfield++ simulation according to needs (drift, amplification, signal creation)

Figure 1: Division of simulation tasks between Geant4, Garfield++ and Degrad for charged particles and photons. Depending on the use case, either Degrad, Geant4, Heed or the PAI models of both Geant4 and Heed together take care of the electron-ion-pair production.

gas. In case of charged particles, the PAI model can be used directly. If the primary particle is a photon, then first the standard EM physics is responsible for the interaction via Photoelectric effect, Compton scattering or Pair production. After the interaction, the PAI model deals with the produced electrons. When the track status of the ionization electron is fStopAndKill, its position is stored in the User SteppingAction or SensitiveDetector. As an alternative, if no high precision is required, the number and position of the electron-ion pairs can also be sampled from the energy deposition in the step, without actually creating any new electrons. At the stored position, Garfield++ then creates an electron-ion pair, to subsequently simulate the drift, amplification and signal creation. The disadvantage of this approach is that it is the only case where the Geant4 physics parametrization features cannot be used. Further, only the W value can be correctly reproduced, but not the Fano factor as will be shown in chapter 5. The advantage on the other hand is that the Geant4 multithreaded mode can be used, since only Garfield++ and not the non-thread safe Heed is used.

- *B) Geant4/Heed PAI model interface for all charged particles*
  This case applies to all charged particles independent of their kinetic energy. The PAI model is used in Geant4 to create only the primary ionization electrons in the gas, at which point Heed takes over. Heed is used to propagate all ionization electrons with a kinetic energy of a few keV and lower as δ-electrons. These δ-electrons further ionize the gas, until all electrons reach thermal energies. Since the Geant4 and the Heed PAI model are used together here, a verification of the results is necessary. The verification can be found in chapter 5.

- *C) Heed PAI model for relativistic charged particles*
  To simulate the interaction of relativistic protons, muons or electrons [3], the PAI model of Heed can be chosen. Heed takes over the control from Geant4 as soon as the primary particle enters the gas volume. In contrast to the PAI model in Geant4, however, Heed tracks the primary particle without Coulomb scattering. The stopping power of the primary is continuous and based on its initial energy, which works fine for Heed's intended purpose: the tracking of relativistic charged particles in thin absorbers. However, for slower particles in thicker absorbers, it leads to unrealistically straight tracks and an incorrect energy loss.

- *D) Heed for photons*
  The photon or gamma particle is killed in Geant4 as soon as it enters the gas volume. Heed then determines the appropriate cross section for the interaction, and takes care of the creation of electron-ion pairs. The approach works for all photons and gamma particles, but the physics implemented in Heed is limited compared to Degrad.

- *E) Degrad for photons*
  A very convincing use case for the Geant4/Degrad interface is the sim-

[3]For ions, Garfield++ also contains an SRIM [25] interface.

ulation of the photo electric effect for X-rays. In contrast to Heed, Degrad simulates the shell absorption by photoelectric effect and subsequent Auger shake-off and fluorescence emission. The program calculates the number of electrons and excitations after energy thermalization and gives the Fano factor for both the electrons and the excitations. Unfortunately, Degrad does neither determine the probability/cross section of the interaction, nor the correct location of the interaction. Therefore, Geant4 is used to determine the cross-section and the location of the interaction. In case the photoelectric effect takes place, Geant4 creates the photo electron. The position of creation is stored, after which the photo electron is killed in Geant4. At that point Degrad takes over, and re-simulates the photoelectric effect, as described above. The disadvantage of this approach is the increased run time of the simulation due to the use of Degrad.

Following the interface procedure, it is in all cases possible to continue the simulation in Geant4. The electron-ion pairs or optical photons produced in Garfield++ or Degrad, can be re-created as secondary particles in Geant4. The implementation for the different scenarios is described in more detail in the following section.

3. Implementation of the interface

To interface Geant4 with an external software package, the physics parametrization feature in Geant4 can be used. The general idea is to create a region, in which the user provides her own implementation of the physics and the detector response. This region, defined by the G4Region class, is created during the detector construction, and consists of one or more G4LogicalVolumes, often corresponding to sub detector volumes. The complete syntax is shown in listing 1 of Appendix A. To implement the parametrized physics model, the user has to create a new UserG4FastSimulationModel derived from G4VFastSimulationModel, and attach it to the region. It is possible to attach more than one UserG4FastSimulationModel to the same region. The user physics code is now used instead of Geant4 in this region, whereas for the remainder of the geometry, the Geant4 physics is still valid.

In the physics list of the program, a G4FastSimulationManagerProcess has to be created. The G4FastSimulationManagerProcess is the physics process of the parametrization, and it serves as interface between the Geant4 tracking and the user parametrization. At tracking time, the G4FastSimulation-ManagerProcess allows the user model to be triggered. In the AddParameterisation() method of the user physics list, the G4FastSimulationManager-Process must be added as a discrete process to the process list of the particles for which the model shall apply[4]. Listing 2 contains the template for the user physics list class.

The core part of the interface is the UserG4FastSimulationModel derived from G4VFastSimulationModel. The name G4VFastSimulationModel implies that the parametrized model is normally simpler and thus faster than the full Geant4 tracking. In the case of Garfield++ or Degrad, however, the parametrised model is more detailed. The G4VFastSimulationModel has three pure virtual methods, which must be overridden in the UserG4Fast-SimulationModel. The template of this class is shown in listing 3. The first method, IsApplicable(), must return true when the parametrization model should be applied to the particle under consideration. If this is not the case, the default Geant4 physics will be applied. The second method, ModelTrigger(), is called in every step along the track and should return true if the user-defined conditions of the track are fulfilled. Finally, the implementation of the parametrised model occurs in the DoIt()-method, following one of the scenarios described above in chapter 2. More on the implementation of the G4FastSimulationModel can be found in the Geant4 User guide for application developers[26].

The subsequent implementation steps are quite different for Garfield++ and Degrad and are therefore handled separately in the following sections 3.1 and 3.2, respectively.

3.1. Geant4 and Garfield++

In case of the Geant4/Garfield++ interface, the user implementation of the detector response is based on Garfield++. The G4Region or envelope corresponds here to a gas-filled detector volume [5]. For the use cases A (Geant4 PAI model) and B (Geant4/Heed PAI model interface), the PAI model has to be added as extra EM model to the user physics list (listing 2) and the lower production cut has to be set to the correct value (roughly between the minimum ionization potential and the W value of the gas mixture.) Further,

[4]As of geant4.10.4 it is also possible to implement this via a macro command

[5]Although Garfield++ also supports the simulation of Silicon detectors, it is predominantly a framework for the simulation of gaseous detectors.

the lowest electron energy has to be set to an appropriate value. More details about the lower production cut and the lowest electron energy limit will be provided in sections 4 and 5.

Depending on the use case, the two virtual methods IsApplicable() and ModelTrigger() of the G4FastSimulationModel should be adapted according to listing 4. In case B (Geant4/Heed PAI model interface), the Garfield++ model is valid for ionization electrons produced by the PAI model. The model is triggered when the kinetic energy drops below a user-defined threshold value. If alone Heed (cases C and D) is responsible for the creation of all the electron-ion pairs, the Garfield++ model is triggered as soon as the particle enters the gas volume regardless of its energy. The DoIt()-method displayed in listing 5 contains the actual Garfield++ model. The relevant particle is first killed in Geant4, and subsequently re-inserted and propagated in Garfield++. Here, Heed is used to generate the electron-ion pairs: The methods TransportDeltaElectron(), NewTrack() and TransportPhoton() are called in cases B, C and D, respectively. Since the NewTrack() method produces the electrons in clusters, these clusters need to be retrieved first, before extracting the individual electrons. In the other two cases the electrons can be retrieved directly. Subsequently, the electron drift and amplification can be simulated using the classes AvalancheMC and AvalancheMicroscopic according to the simulation needs. Details concerning these algorithms can be found in the Garfield user documentation [17].

A very useful application for the Geant4/Garfield++ interface is the simulation of secondary scintillation photons (electroluminescence) in gaseous detectors as shown in the bottom part of listing 5. The AvalancheMicroscopic class in Garfield++ is used to create the electron avalanche. The SetUserHandleInelastic() method accepts a user written callback function, that has access to the time, position and excitation levels in each simulation step. This information is stored in a data structure. In the case of pure Noble gases, if one assumes that each excitation will produce a secondary scintillation photon [27], the production of light is straightforward. For each element of the data structure filled in the callback function, a secondary optical photon is produced in Geant4 as shown in listing 6. The optical photon tracking will then be carried out by Geant4. For gas mixtures, a more detailed model should be adopted in Garfield++ in order to include the quenching of the Xenon excimers by the addmixture [28].

The user-defined G4VFastSimulationModel should contain a fourth public method where the Garfield++ related objects are initialized before the run is started. Features such as the detector geometry, the electro-magnetic field and preferred model for tracking have to be defined here, as demonstrated in listing 7. The geometry is added to the field component, and subsequently a sensor is created using the field component. The sensor class links the detector geometry with its properties (material, geometry, fields) to the transport classes such as, TrackHeed. Since the shape and size of the gas volume and the gas composition is available in the Geant4 DetectorConstruction class, the information from there can be used to create the detector geometry in the Garfield++ Model.

Listing 8 shows how to compile and link the Geant4/Garfield++ program. A CMakeLists.txt has to be created, in which the Garfield++ include and library directories are defined. The library flags -lGarfield -lgfortran have to be placed behind the library flags for the Geant4 libraries.

In case A, Garfield++ has to be directly interfaced to Geant4 in the UserSteppingAction or SensitiveDetector. In Geant4 the PAI model can either really produce all conduction electrons, or the number and positions of the electron-ion pairs can be sampled from the energy deposition in each step. The positions of the conduction electrons or electron-ion pairs are subsequently sent to Garfield++, where the detector simulation is continued. If the electrons are to be produced and not just sampled, the lower production cut of the Geant4 PAI model and the lowest electron energy have to be set to the correct values to produce the correct number of electrons, as discussed in section 4. Listing 9 illustrates how to store the last position of the electron track before it is stopped and killed. The sampling of conduction electrons from the energy deposition is shown in listing 10. The mean number of electron-ion pairs can be determined by the MeanNumberOfIonsAlongStep() method of the G4ElectronIonPair class for each step of the primary and secondary particles. Using the mean number of electron-ion pairs together with the user specified Fano factor, a random gamma function calculates the exact number of electron-ion pairs per step. The positions of these electron-ion pairs are sampled between the pre and the post step point of the step with the help of the G4UniformRand() distribution. It should be noted that this method is rather crude, especially for low energies as will be shown in chapter 4.2.

3.2. Geant4 and Degrad

The implementation of the Geant4/Degrad interface is based on a user-defined G4VFastSimulationModel 4, analogous to the Geant4/Garfield++

interface. During detector construction, a G4Region is created (listing 1). In the user physics list the AddParameterisation() function should be modified to include only the particles for which the model shall apply (listing 2). In case E, X-rays produce electrons via the photo-electric effect or Compton scattering in Geant4. Directly after their production, these electrons are killed in Geant4. Degrad then re-simulates the X-ray interaction at the time and location of the killed electrons. The parametrization in Geant4 is therefore not valid for X-rays, but for the electrons. In the IsApplicable() and ModelTrigger() methods of the G4VFastSimulationModel (listing 11), the model is enabled for electrons that have an energy higher than the thermalization energy. Finally, the DoIt() method contains the actual Degrad model and interfacing details. All parameters needed to run the simulation (gas composition, electric and magnetic field, particle type and kinetic energy) are stored in a parameter text file, which is read in by the Degrad binary. Prior to its call, the Degrad binary has to be compiled from the source code with the command "f95 degrad2.14.f -o degrad".

In the example a 5.9 keV Fe X-ray is simulated in pure Xenon. As Degrad is a Fortran software and writes its output to a result file, the simplest way to transfer information back to Geant4 is based on a file I/O chain: Geant4 runs Degrad with a parameter file and Degrad writes the simulation results to a text file. From this result file, Geant4 reads back the positions and production times of the ionization electrons. Degrad always assumes $(x0, y0, z0) = (0, 0, 0)$ as the interaction position of the photon, and simulates an infinite volume. Therefore, the positions of the ionization electrons have to be corrected, using the stored interaction position of the X-ray from Geant4. If the corrected positions are within the gas volume of the detector, the CreateSecondaryTrack() method of the G4FastTrack class is called to create new 7 eV (the thermalization energy set in Degrad) secondary electrons in Geant4. For a simulation of the light production, now the Garfield++ parametrization of the electroluminescence example could be used to produce optical photons.

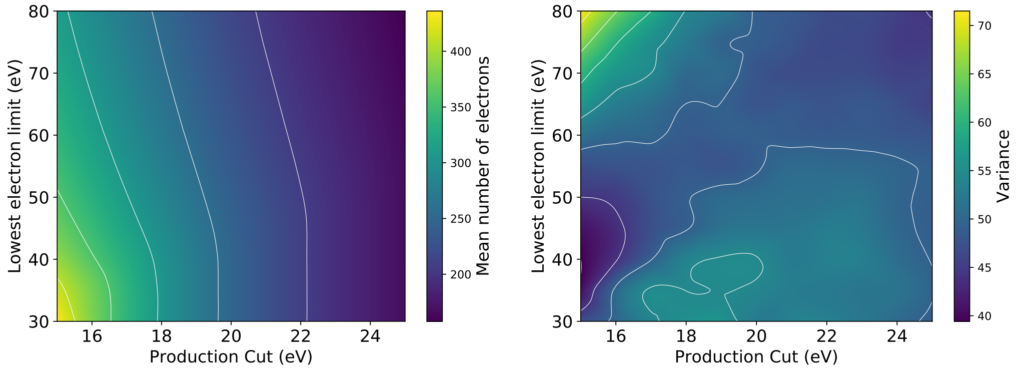
When choosing the "electron beam" option, Degrad is also able to simulate the interaction of electrons with arbitrary gas mixtures. Degrad always uses an infinite volume, and tracks the primary electron and all secondaries until they are thermalized. As a general tool to simulate electrons of all energies in various detector geometries, a Geant4/Degrad interface is thus not an option. Nevertheless, for fully contained low energy electrons it is useful to compare Degrad simulation results to Geant4 and Heed, as will be demonstrated in the following chapters.

4. Geant4 simulation parameters and their optimization

If a Geant4 PAI model simulation relies on the correct number of conduction electrons (e.g. case A), the correct setting of the lower production cut and the lowest electron energy limit is crucial.

4.1. Lower production cut and lowest electron energy

The lower production cut is the minimum energy transfer required to produce a new particle. The creation of secondary particles is not possible in an interaction, if the transferred energy is lower than the lower production cut. In this case, the energy is merely recorded as being deposited in the step. The lowest electron energy limit on the other hand is the energy threshold below which an electron is not tracked anymore. If the kinetic energy of an electron falls below the lowest electron energy limit value during a step, the full energy deposition independently of material [29] is enforced. All physics lists use a default value (e.g. 100 eV for the G4EMLivermore physics list), which can be modified by the user. Evidently, the two parameters influence only the production of secondaries and not the deposited energy, which remains stable.



(a) Mean number of electron-ion pairs (b) Variance of the distribution

Figure 2: Geant4 PAI photon model simulation of the mean number of electron-ion pairs and the variance of the electron-ion pair distribution for a 10 keV electron in a very large volume of He/isoButane 70/30.

Figure 2 illustrates the influence of the lower production cut and the lowest electron energy limit on the number of electron-ion pairs produced by the PAI photon model for a 10 keV electron in a very large volume of He/isoButane 70/30. As expected, figure 2a demonstrates that the mean number of electron-ion pairs rises with a decrease of the production cut and with a decrease of the lowest electron energy limit. For lower production cuts larger than 15 eV, the mean number of electrons stays stable for a range of lowest electron energy limits. The higher the lower production cut, the longer the range of stability. For the variance of the distribution on the other hand it is difficult to identify a clear behavior pattern for the two parameters, as indicates figure 2b.

4.2. W value and Fano factor

To determine the correct settings of the lower production cut and the lowest electron energy limit, the deposited energy obtained in the simulations has to be compared to the number of electron-ion pairs multiplied with the W value. Additionally, it is desirable to obtain the correct Fano factor. The relevant definitions of W value and Fano factor are summarized by references [30] and [31].

The W value, the mean energy needed to create an electron-ion pair, is defined as

$$W(T_0) = \frac{T_0}{N_I}, \tag{1}$$

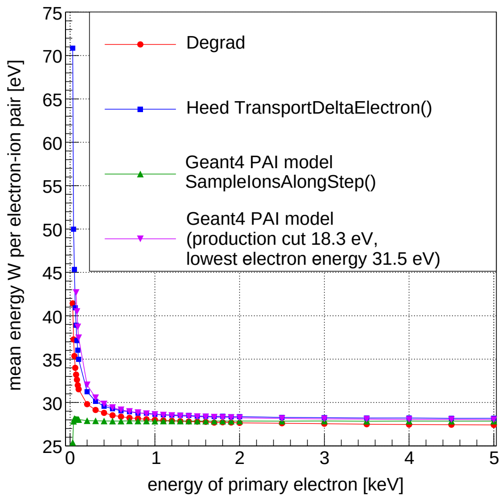
where N_i stands for the mean number of electron-ion pairs produced by a particle of initial energy T_0, dissipating its complete energy in the gas volume. For high particle energies, i.e. higher than 10 keV for electrons, the W value reaches a constant asymptotic value W_a. Below 1 keV electron energy, the cross section ratio of ionizing processes versus non-ionizing processes becomes smaller compared to the high energy region. A certain fraction of the dissipated energy is carried by electrons that are too low in energy to further ionize. With U representing the mean energy of these sub-ionization electrons, and E the energy of the impinging particle, the energy dependent expression for the W value can be written as

$$W(E) = \frac{W_a}{1 - \frac{U}{E}}. \tag{2}$$

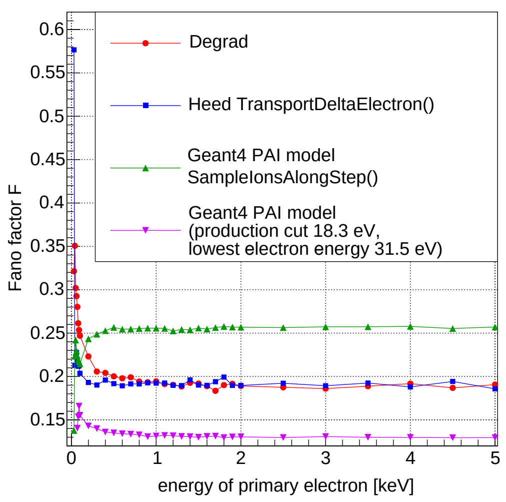
For high-energy particles in thin absorbers, a differential w value

$$w = \frac{dT}{dN} \tag{3}$$

is used. Since W(T_0) reaches a constant value W for energies of typically 10 keV and higher, w converges to a constant value equal to W_a for sufficiently high energies. In case of full absorption, the kinetic energy of the primary particle is thus identical to the number of electron-ion pairs N multiplied with the W value. For fast particles and thin absorbers, the number of electron-ion pairs N multiplied with the differential w value is equivalent to the deposited energy in the absorber. The Fano factor F is defined as the ratio between the variance and the mean of the electron-ion pair distribution. Figure 3 displays the W value and Fano factor simulated for fully contained 30 eV to 5 keV electrons in Ar/CO_2 70/30.



(a) Work function W



(b) Fano factor

Figure 3: Simulated W value and Fano factor for fully contained 30 eV to 5 keV electrons in Ar/CO_2 70/30.

With a lower production cut of 18.3 eV and a lowest electron energy limit of 30 eV, the Geant4 PAI model correctly reproduces the W value. The same applies to Heed and Degrad. W reaches the asymptotic value of about 28 eV for electron energies of 2 keV and larger, whereas for lower electron energies more energy is needed to create an electron-ion pair. The small divergence between the results from Degrad and the PAI model in Heed and Geant4, especially at low energies, can be explained by the approximations used in the calculations. The PAI model [22] relies on the assumption that a particle interacts only through the long range dipole part of the electromagnetic force, and consequently uses an approximation similar to the Bethe-Bloch formula [32]. The dipole approximation of the force is valid to about 1% for energies down to typically 1 keV for electrons or 2 MeV for protons. Below these energies, two problems arise. First, the Bethe Bloch formula requires phenomenological derived correction terms e.g. Anderson-Ziegler and Lindhard-Scharff. These terms are not included in the Heed PAI model. A second problem is that the non-dipole electromagnetic scattering becomes much larger below 1 keV electron energy and approaches 60% of the dipole scattering at 80 eV on atomic and molecular targets. The Degrad program accounts for both dipole and non-dipole scattering and is therefore more accurate below 1 keV electron energy.

The Fano factor simulation shows the expected asymptotic behavior in Degrad and Heed. Starting with higher values for lower energies, a value of 0.19 is reached for electron energies larger than 2 keV. With the Geant4 PAI model the correct Fano factor cannot be reproduced, the asymptotic value is too low with 0.13. The SampleIonsAlongStep() method of the G4ElectronIonPair class only reproduces the correct asymptotic W value. Since it always uses the same mean number of electron-ion pairs per step, the higher W value for electrons of less than 1 keV kinetic energy cannot be obtained. The sampled Fano factor is too high with a value of 0.25, in spite of using 0.19 as input Fano factor for the Gamma function that calculates the number of electron-ion pairs per step.

To summarize, figure 3 shows that Heed and Degrad reproduce the correct W value and Fano factor for fully contained electrons of 30 eV to 5 keV. With the correct settings of lower production cut and lowest electron energy limit, the Geant4 PAI model obtains the correct W value. It is not possible to simultaneously obtain the correct Fano factor, the variance of the electron-ion pair distribution is too small.

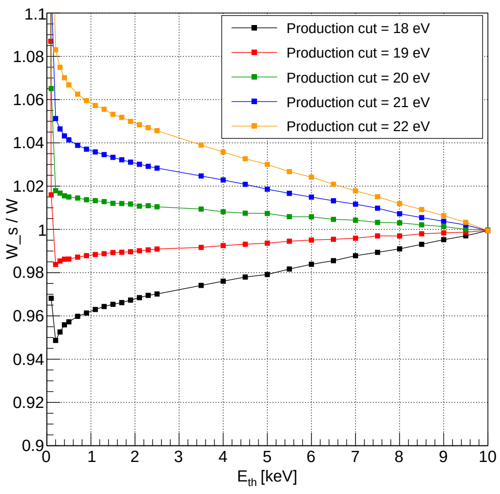
5. Verification

Case B (Geant4/Heed PAI model interface) is the only case where different physics models, here Geant4 and Heed, are used together. The PAI or PAI photon model of Geant4 is responsible for the primary ionization, and the TransportDeltaElectron() method of the Heed PAI model takes care of the

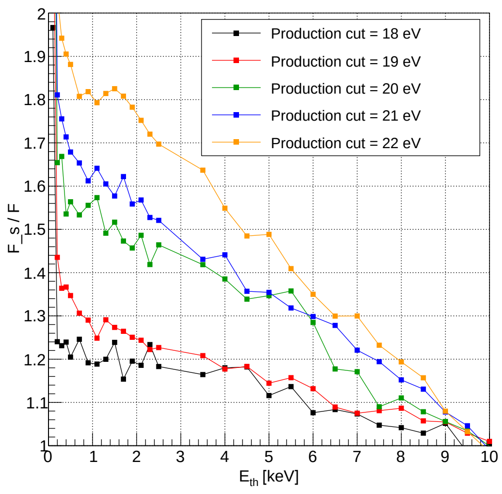
δ electron transport and produces electron-ion pairs. Extensive simulations have been carried out to verify the results in comparison to the standalone simulation programs. All simulations in this paper have been carried out at Normal Temperature and Pressure (NTP) without the presence of an electromagnetic field. G4EmLivermorePhysics was used as physics list (fixed lowest electron energy limit of 100 eV), and the PAI/PAI photon model was activated as additional EM physics model. For He/isoButane gas mixtures, the Geant4 10.4 PAI photon model was used, whereas the Geant4 10.3 PAI model was used for Ar/CO_2 gas mixtures. For Heed, the Garfield++ revision 541 was used. Since Heed is not thread-safe, the Geant4 sequential mode has been used for the simulations.

5.1. Lower production cut and transfer threshold

When using the Geant4/Heed PAI model interface, all electrons below a certain energy are killed in Geant4 and sent over to Heed, where they are treated as δ-electrons. This energy threshold is called transfer energy threshold. The effect of using different production cuts and transfer energy thresholds on the electron-ion pair distribution is shown in figure 4. For the W value (figure 4a) and the Fano factor (figure 4b), the ratios between the simulated value and the Heed reference value are plotted. For each of the plotted data points 10^4 electrons with an energy of 10 keV were released in a sufficiently large gas volume, filled with a 70/30 Helium-Isobutane mixture, to ensure full energy deposition. Subsequently, W was calculated using equation 1 and the corresponding Fano factor was obtained from the statistical mean and variance of the electron-ion pair distribution. Evidently, both the W value and Fano factor converge to the Heed values at a 10 keV transfer threshold, as the electrons are immediately transferred to Heed. For transfer thresholds below 100 eV, the simulated values rapidly increase towards infinity, as all electrons with a kinetic energy lower than this value are killed in the G4EMLivermore physics model. Looking at the W graph, an optimal production cut can be found somewhere in between 19 and 20 eV, regardless of the transfer threshold. The Fano factor, on the other hand, follows a steady increase over the entire energy range. Whereas for the standalone Geant4 PAI model the Fano factor was too small, the values obtained with the interface seem rather too large. Whereas Heed has a database of literature values for W, a default value of 0.19 for F is used for all gases. Hence, a value larger than 1 in figure 4b does not necessarily mean the interface produces a too large Fano factor. Additionally, a non-default Fano factor can be provided by the user in Heed, such that the desired values can be matched by the interface.



(a) W value: Simulated versus literature value

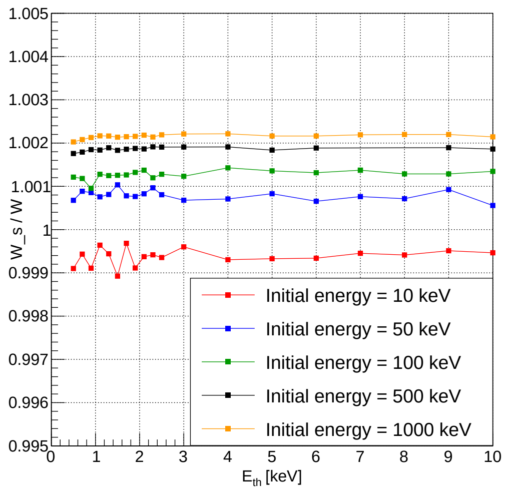


(b) Fano factor: Simulated versus literature value

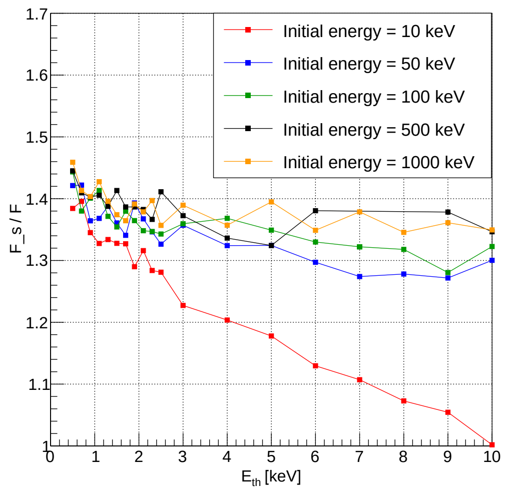
Figure 4: Influence of transfer threshold and lower production cut on W value and Fano factor for 10 keV electrons using the PAI photon model in He/isoButane 70/30.

Naturally, the results from figure 4 are futile if not the same optimal values are found for different energies of the primary. Using the optimal lower production cut of 19.45 eV, in figure 5 the W value and Fano factor were simulated for 10 keV to 1000 keV electrons. A relative difference of $\approx 0.1\%$ is observed for the obtained W values (figure 5a). The Fano factor, however, displays a clear discrepancy between the 10 keV and the higher primary energies, except for transfer thresholds below 2 keV, where the curves converge to comparable values. Therefore, it is recommended to set the threshold to values between 100 eV and 2 keV when using the G4EMLivermore physics. For standard EM physics (lowest electron energy limit of 1 keV), it should be between 1 keV and 2 keV.

The optimal lower production cut should also not vary much between different particle types and different particle energies for not fully contained particles. Figure 6a displays the deviation of N·w from the correct deposited energy in percent for 1 GeV alpha particles, electrons, mu+ and protons in Ar/CO_2 70/30. For Ar/CO_2 70/30, at a lower production cut of 21 eV, N·w



(a) W value



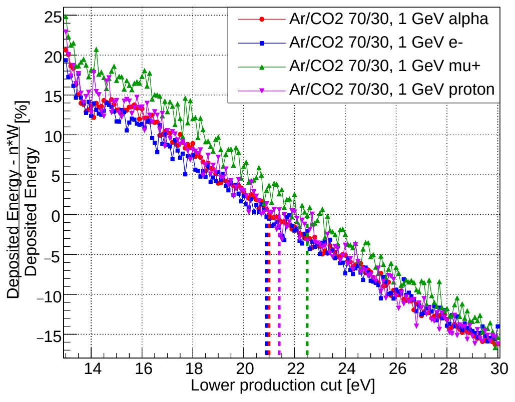
(b) Fano factor

Figure 5: Comparative study of the W value and Fano factor for primary electrons of 10 keV and 100 keV. An optimal production cut of 19.45 eV was used in all cases.

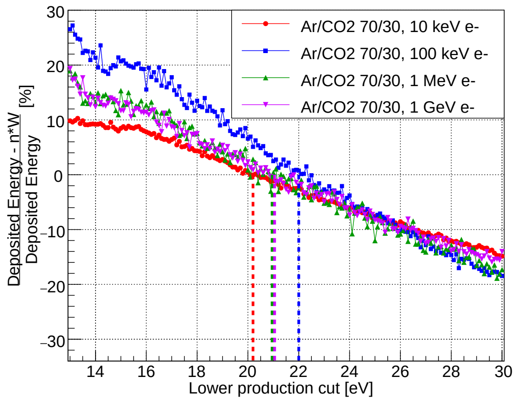
of the Geant4/Heed PAI model interface is equivalent to the deposited energy of the PAI model in Geant4 for alpha particles and electrons. For mu+ and protons, the best production cut is slightly higher with 21.5 eV and 22.5 eV. Nevertheless, even at a cut of 21 eV, N·w for 1 GeV proton is only 3% larger than the deposited energy. For the different electron energies of 10 keV to 1 GeV in figure 6b, the situation is similar. The best lower production cut for 1 MeV and 1 GeV electrons is 21 eV, for 100 keV electrons 22 eV and for 10 keV electrons 20 eV. At 21 eV, N·w for 100 keV electrons is only 3% higher than the deposited energy, whereas for 10 keV electrons it is 2% lower. For simulations in Ar/CO_2 70/30, 21 eV is therefore a good lower production cut. For other Argon and CO_2 gas mixtures, the best lower production cuts can be determined using the method shown in chapter 5.2.

5.2. Optimal lower production cut for different ratios

In a separate analysis, a parametrization of the optimal lower production cut as a function of mixture ratio was investigated. For this purpose, an approach similar to the determination of the W value of a gas mixture in Heed, was applied. Heed uses two assumptions to determine the W value. First, inter-species effects, such as Penning effect [33], are assumed to be negligible, which is not accurate for many mixtures with a small percentage



(a) 1 GeV particles



(b) 10 keV to 1 GeV electrons

Figure 6: Different 1 GeV particles and electrons of energies between 10 keV and 1 GeV in Ar/CO_2 70/30: Influence of the lower production cut of the PAI model on the deposited energy (n·w) in the Geant4/Heed PAI model interface. The figure shows the deviation from the correct deposited energy in percent.

of quench gas. Secondly, the total interaction cross section for the interaction of a charged particle with a molecule or atom in the gas is assumed to be proportional to the total charge of that molecule or atom. Although crude, these assumptions work quite well for most mixtures and a precision down to the percent level. Following this approach, the W value of a mixture is calculated as

$$W(f) = \frac{W_2 + (\frac{Z_1}{Z_2}W_1 - W_2)f}{1 + (\frac{Z_1}{Z_2} - 1)f}. \tag{4}$$

W_1 and W_2, and Z_1 and Z_2 are the W values and molecular charges of molecule 1 and 2, and f is the fraction of the molecule 1. As a first try, the same approach was used to determine the optimal production cut, resulting in the following expression

$$P(f) = \frac{P_2 + (\frac{Z_1}{Z_2}P_1 - P_2)f}{1 + (\frac{Z_1}{Z_2} - 1)f}, \tag{5}$$

with P_1 and P_2 as the optimal lower production cuts for the two pure gases. This model was tested for different mixtures of Helium and Isobutane and mixtures of Argon and CO2. The result of the simulations is shown in table 1. The first (third) column states the He (CO_2) fraction in the simulated mixture.

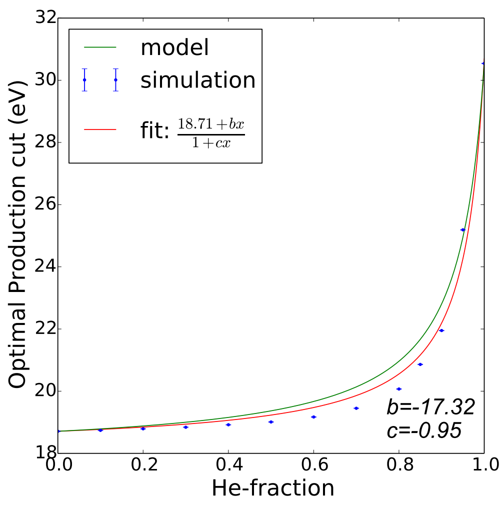
Table 1: Optimal lower production cut values for different ratios

	production cut (eV)	
f	He/isoButane (He)	Ar/CO_2 (CO_2)
0	18.71 ± 0.01	20.7 ± 0.01
0.1	18.74 ± 0.01	20.89 ± 0.01
0.2	18.79 ± 0.01	21.11 ± 0.01
0.3	18.84 ± 0.01	21.35 ± 0.01
0.4	18.92 ± 0.01	21.62 ± 0.01
0.5	19.01 ± 0.01	21.88 ± 0.01
0.6	19.17 ± 0.01	22.16 ± 0.01
0.7	19.45 ± 0.01	22.48 ± 0.01
0.8	20.07 ± 0.01	22.75 ± 0.01
0.85	20.86 ± 0.01	-
0.9	21.95 ± 0.01	23.12 ± 0.01
0.95	25.19 ± 0.01	-
1	30.54 ± 0.01	23.48 ± 0.01

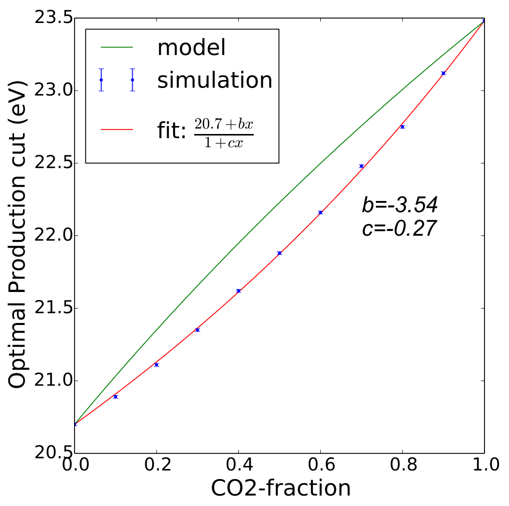
A comparison of the data from table 1 and the model described above is displayed in figure 7. In the case of both He/isoButane (figure 7a) and Ar/CO_2 (figure 7b), the model clearly overestimates the data by a few percent. In a second attempt to parametrize the optimal lower production cut, the following function

$$P(f) = \frac{a + bf}{1 + cf}, \tag{6}$$

was fitted to the data. In the fit, a is fixed to the 0% value from the table, and b and c are left as free parameters. The resulting fit (red) shows a clear improvement over the model (green). But the fitting approach requires additional effort, as a sufficient number of values need to be determined manually beforehand. Hence, the user may decide how precise W needs to be, and may even feed an experimentally determined W value to Heed, for which a production cut scan is necessary anyway.



(a) He/Iso



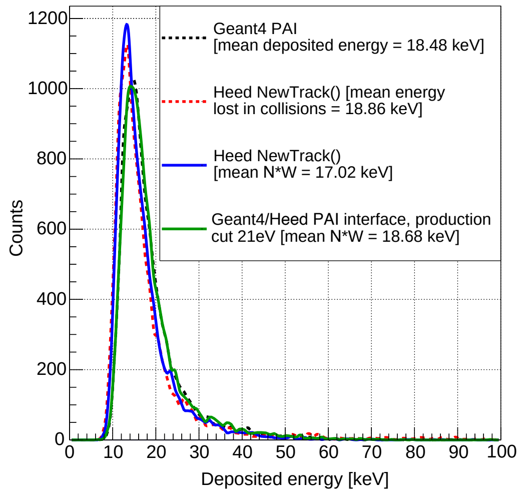
(b) Ar/CO2

Figure 7: Comparison of the data in table 1 and the model (equation 5). The green curve represents the model, the red curve corresponds to the fit according to equation 6 and the blue error bars show the simulated data.

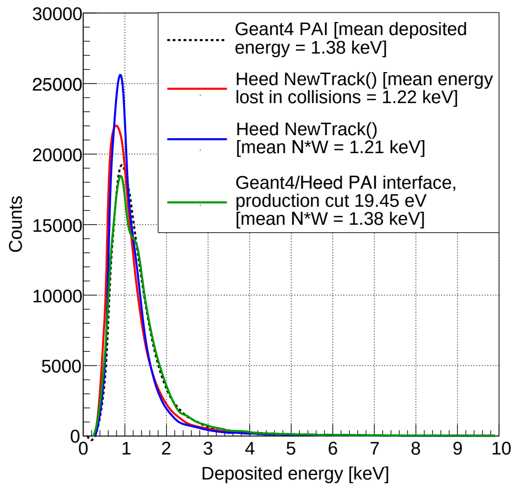
5.3. Deposited energy spectra

In the case of particles not fully contained in the gas volume, it is interesting to study the deposited energy spectra. For all simulations a transfer energy threshold of 2 keV has been used. Figure 8 displays the spectra for 10^6 1 GeV alpha particles in 1 cm of Ar/CO_2 70/30 and 10^6 1 MeV electrons in 1 cm of He/isoButane 70/30. The black dashed line shows the deposited energy spectrum of the Geant4 PAI or PAI photon model, and the green solid line the deposited energy spectrum ($N \cdot w$) simulated with the Geant4/Heed PAI model interface. Both curves have the same shape and the same mean value. From a Heed simulation using NewTrack(), the energy of the produced electrons clusters (red dashed curve) and $N \cdot w$ (blue solid curve) have been obtained. In a volume of limited dimensions, here 10 cm x 10 cm x 1 cm, the cluster energy in Heed is always larger than $N \cdot w$, since some of the delta electrons in the clusters have a long range, and are not fully contained in the gas volume. With respect to the deposited energy spectrum, the Geant4 PAI model, the Geant4/Heed PAI model interface and Heed deliver comparable results for relativistic particles.

For non-relativistic electrons, the situation is different. Figure 9 shows



(a) 1 GeV alpha particles in Ar/CO_2 70/30



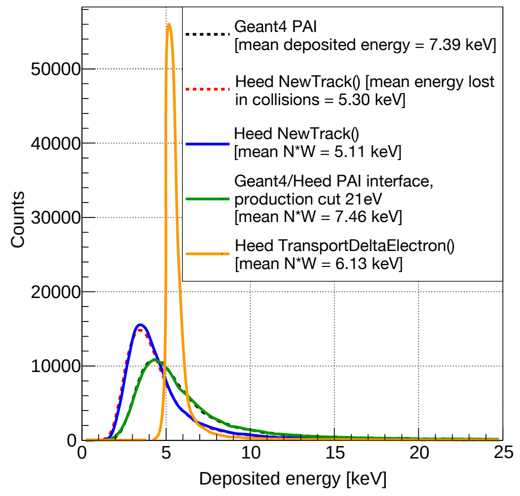
(b) 1 MeV e- in in He/isoButane 70/30

Figure 8: Relativistic particles: Deposited energy spectrum of 1 GeV alpha particle in Ar/CO_2 70/30 and 1 MeV electron in He/isoButane 70/30

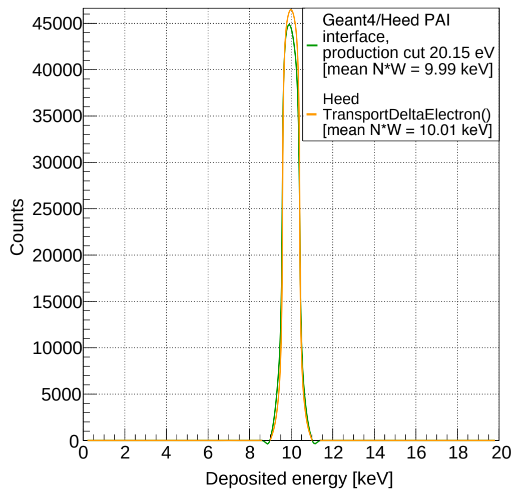
the deposited energy spectra for 100 keV electrons in Ar/CO_2 70/30 and 10 keV electrons in He/isoButane 70/30. For 100 keV electrons, the Heed NewTrack() and the Heed TransportDeltaElectron() functions produce very different spectra, which both are incorrect and give a mean deposited energy that is too low. The spectrum of the Geant4/Heed PAI model interface on the other hand is identical to the Geant4 PAI model spectrum. For 10 keV electrons, Heed NewTrack() cannot be used since it is not intended for electrons of this energy. Since the electrons are basically fully absorbed in the gas volume, these spectra can be compared to the results from section 5.1. Here, the optimal production cut was found by matching the W value calculated by the interface and Heed (figure 5a). Hence, it is not surprising that both spectra nicely overlap, with a slightly broader peak for the interface. This can be traced back to the Fano factor, which was 20% larger for the interface (figure 5b).

5.4. Spatial distribution of electron-ion pairs

The maximum energy of the δ electrons that are sent to the interface depends on the transfer energy threshold. A transfer energy threshold of 2 keV is equivalent to δ electrons with an energy of up to 2 keV. Simulations of the range of a 2 keV electron result in values of 160 μm for Degrad, whereas Heed



(a) 100 keV e- in Ar/CO_2 70/30



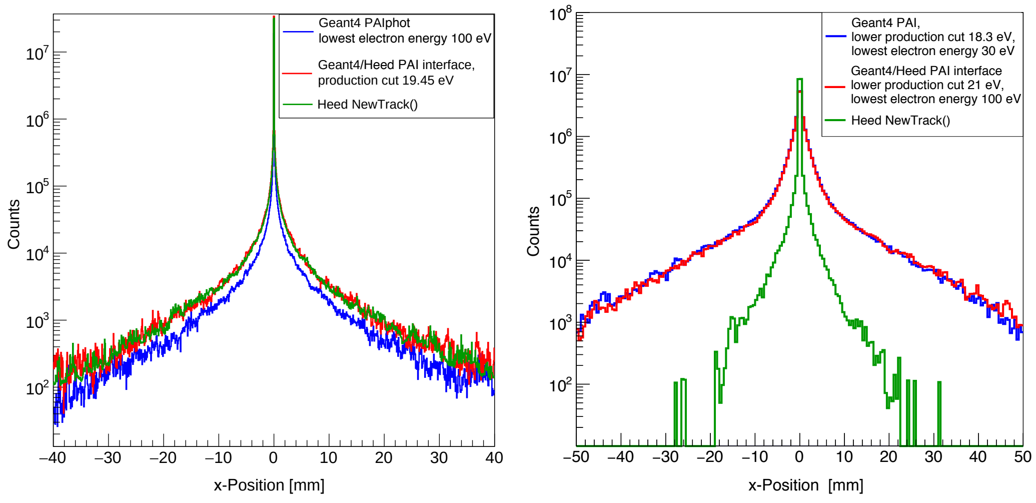
(b) 10 keV e- in He/isoButane 70/30

Figure 9: Slow electrons: Deposited energy spectrum of 100 keV electrons in Ar/CO_2 70/30 and 10 keV electrons in He/isoButane 70/30.

and Geant4 give 100 μm and 70 μm. Since Degrad includes the processes of photo-emission and absorption, it calculates a longer range than the other programs. To summarize, even with a transfer energy threshold of 2 keV, the total range and the track shape of the simulated primary particle will be similar for the interface and a stand-alone Geant4 PAI model simulation due to the short range of the δ electrons. The track shape will be dominated by the distribution of the primary ionization electrons created by the PAI or PAI photon model in Geant4.

Figure 10 shows the x position of all electron-ion pairs created by a 1 GeV electron in He/isoButane 70/30, and a 100 keV electron in Ar/CO_2 70/30. The momentum direction is along the z axis. For relativistic particles the spatial distribution of the electron-ion pairs agrees nicely between the Geant4/Heed PAI model interface and Heed as displayed in figure 10a. The Geant4 PAI photon model under-produces electrons, since the lowest electron energy limit was kept at the standard 100 eV value of the G4EmLivermorePhysics. For 100 keV electrons on the other hand, figure 10b indicates that Heed creates a distribution of electron-ion pairs that is too narrow. This can be traced back to the absence of Coulomb scattering in the NewTrack()-method of Heed. The Geant4 PAI model and the Geant4/Heed

PAI model interface, in contrast, do take this into account and therefore produce identical distributions. If one sets the lowest electron energy limit to 30 eV and uses a lower production cut of 18.3 eV, the standalone Geant4 PAI model simulation does not under-produce electrons any more.



(a) 1 GeV e- in He/isoButane 70/30 (b) 100 keV e- in Ar/CO_2 70/30

Figure 10: Spatial distribution of electron-ion pairs created by a 1 GeV electron in He/isoButane 70/30 and a 100 keV electron in Ar/CO_2 70/30. The plot shows the x coordinate of the electron-ion pairs.

5.5. Performance

Regarding the performance of the Geant4/Heed PAI model interface, the absolute time needed for a simulation depends of course on multiple factors, most importantly the performance of the computer hardware. Table 2 displays the performance of the following options: Geant4 PAI model (case A), Geant4/Heed PAI model interface (case B) and Heed PAI model (case C). Since Heed does not support multi-threading, single threaded programs where used. The program code was stripped down to include just the creation of the electron-ion pairs in the gas volume, not any drift or avalanche processes. The simulations were carried out several times on the same lxplus machine at CERN in direct successions, to exclude effects of different cluster usage. Averaged over 10 runs, the performance of the Geant4/Heed PAI model interface is the best with 2.24 s simulation time per 1000 events.

The Geant4 PAI model is slightly slower with 4.14 s per 1000 events, since the lowest electron energy limit has to be set to 30 eV, to obtain a sufficient number of electron-ion pairs. This slows the simulation down compared to the standard 100 eV limit of the G4EmLivermorePhysics. The Heed PAI model simulation takes the longest time with 64.2 s per 1000 events. The Heed NewTrack() method seems to be very slow compared to the Transport-DeltaElectron() method that is used by the interface.

Table 2: Comparison of simulation times for 10^5 1 GeV electrons in 1 cm of Ar/CO_2 70/30

Physics model in gas region	Simulation time per 1000 events
Geant4 PAI model (case A)	4.14 s
Geant4/Heed PAI model interface (case B)	2.24 s
Heed PAI model (case C)	64.20 s

6. Conclusion

The present paper demonstrates how to interface Geant4 and Degrad or Geant4 and Garfield++ using the G4VFastSimulationModel. The aim is a complete simulation of gaseous detectors. There are several possibilities to divide the task between Geant4 and Garfield++. As discussed, the preferred way for charged particles is to use the Geant4/Heed PAI model interface. The PAI model in Geant4 creates ionization electrons, which are then transferred to Heed when their kinetic energy falls below 1-2 keV. Using the W value and the Fano factor, a method was developed to determine the optimal lower production cut for the PAI model for different gas mixtures. With the optimal lower production cut, the correct deposited energy spectra and correct spatial charge distributions are obtained for different particle types and energies. The Geant4/Heed PAI model interface works even for non-relativistic and slow electrons, where the stand-alone Heed simulations fail. Its performance is better than that of standalone Geant4 or Heed. For photons, the Geant4/Garfield++ and Geant4/Degrad interfaces make it possible to simulate complex physics cases. As example it was shown how to simulate the interaction of 5.9 keV Fe X-rays with pure Xenon (photoelectric effect and subsequent production of optical photons via electroluminescence). To summarize, the Geant4/Garfield++ and Geant4/Degrad interfaces are very useful tools for detector simulations.

7. Acknowledgments

This work was partially funded by the EU Horizon 2020 framework, BrightnESS project 676548.

Appendix A. Appendix: Code listings

```
G4VPhysicalVolume* UserDetectorConstruction::Construct() {
  /* Geant4 in sequential mode*/
  G4Region* region = new G4Region("gasRegion");
  region->AddRootLogicalVolume(fLogicalVolumeGas);

  fModel = new GarfieldG4FastSimulationModel("UserModel", region);
}
```

Listing 1: G4Region implementation in the G4VUserDetectorConstruction class in Geant4 sequential mode

```
void UserPhysicsList::AddParameterisation() {
   /* Geant4 in sequential mode:
   Create G4FastSimulationManagerProcess*/
  G4FastSimulationManagerProcess* process =
      new G4FastSimulationManagerProcess("G4FSMP");
  auto it=GetParticleIterator();
  it->reset();
  while ((*it)()) {
    G4ParticleDefinition* p = it->value();
    G4ProcessManager* manager = p->GetProcessManager();
    G4EmConfigurator* config = G4LossTableManager::Instance()-> EmConfigurator();
    G4String name = "e-";
    if (p->GetParticleName() == name)   {
      /*Add G4FastSimulationManagerProcess*/
      manager->AddDiscreteProcess(fastSimProcess);
      /*Add PAI or PAI photon model*/
      G4PAIModel* pai = new G4PAIModel(p, "G4PAIModel");
      config->SetExtraEmModel(name,   "eIoni", pai, "gasRegion", 0.*eV, 1.*TeV, pai);
    }
  }
}

void UserPhysicsList::SetCuts() {
  /* Geant4 in sequential mode*/
  G4ProductionCutsTable::GetProductionCutsTable()->SetEnergyRange(cut*eV, 1.*TeV);
  G4EmParameters* emParams = G4EmParameters::Instance();
  emParams->SetLowestElectronEnergy(energy*eV);
}
```

Listing 2: Instantiation of the G4FastSimulationManagerProcess in a physics list in sequential Geant4. The code shows how to enable the PAI model and the Geant4/Garfield++ interface. From Geant4 10.2 on, the PAI model may be enabled via UI command instead, however this code will still work.

```
class UserFastSimulationModel : public G4VFastSimulationModel {

  public:
      /*Constructor and Destructor*/
      UserFastSimulationModel();
      ~UserFastSimulationModel();

      /*virtual methods*/

      /*return true if the G4FastSimulationModel has to be applied for a particle type*/
      virtual G4bool IsApplicable(const G4ParticleDefinition&);

      /*return true if conditions like kinetic energy of G4FastTrack are fulfilled*/
      virtual G4bool ModelTrigger(const G4FastTrack&);

      /*The parametrization, i.e. Garfield or Degrad related code, should be implemented here*/
      virtual void DoIt(const G4FastTrack&, G4FastStep&);
};
```
Listing 3: Class definition of the user-defined class derived from the G4VFastSimulationModel.

```
#include "GarfieldG4FastSimulationModel.hh"

G4bool GarfieldG4FastSimulationModel::IsApplicable(
const G4ParticleDefinition& particleType) {
  /*Case B*/
  return &def == G4Electron::ElectronDefinition();

  /*Case C*/
  return &def == G4Proton::ProtonDefinition();

  /*Case D*/
  return &def == G4Gamma::GammaDefinition();

  /*Electroluminescence example*/
  return &def == G4Electron::ElectronDefinition();
}

G4bool GarfieldG4FastSimulationModel::ModelTrigger(
    const G4FastTrack& fastTrack) {
  double ekin = fastTrack.GetPrimaryTrack()->
    GetKineticEnergy() / keV;

  /*Case B*/
  if(ekin < 2. * keV)   {return true;}

  /*Case C and D*/
  return true;

  /*Electroluminescence example*/
  if(ekin < 10. * eV) {return true;}

  return false;
}
```

Listing 4: G4FastSimulationModel

```cpp
void GarfieldG4FastSimulationModel::DoIt(const G4FastTrack& fastTrack, G4FastStep& fastStep) {
  /*Get world and local position from fast track, get momentum vector from fast track, get global time in ns, and x,y,z position in cm, get kinetic energy of fast track in eV*/
  double ekin_eV = fastTrack.GetPrimaryTrack()->GetKineticEnergy() / eV;

  /*Kill track in Geant4 - Garfield takes over*/
  fastStep.KillPrimaryTrack();

  /*Number of electrons produced in a collision*/
  int nc = 0;

  Garfield::TrackHeed* trackHeed = new Garfield::TrackHeed();
  trackHeed->EnableDeltaElectronTransport();

  /*Case B*/
  trackHeed->TransportDeltaElectron(x_cm, y_cm, z_cm, globaltime, eKin_eV, dx, dy, dz, nc);
  for (int cl = 0; cl < nc; cl++) {
    double xe, ye, ze, te;
    double ee, dxe, dye, dze;
    trackHeed->GetElectron(cl, xe, ye, ze, te, ee, dxe, dye, dze);
    /*Garfield++ simulation according to needs*/
  }

  /*Case C*/
  trackHeed->SetParticle("proton");
  trackHeed->SetKineticEnergy(ekin_eV);
  trackHeed->NewTrack(x_cm, y_cm, z_cm, globaltime, dx, dy, dz);
  double xcl, ycl, zcl, tcl, ecl, extra;
  while (trackHeed->GetCluster(xcl, ycl, zcl, tcl, nc, ecl, extra)) {
    for (int cl = 0; cl < nc; cl++) {
      double xe, ye, ze, te;
      double ee, dxe, dye, dze;
      trackHeed->GetElectron(cl, xe, ye, ze, te, ee, dxe, dye, dze);
      /*Garfield++ simulation according to needs*/
    }
  }

  /*Case D*/
  trackHeed->TransportPhoton(x_cm, y_cm, z_cm, globaltime, eKin_eV, dx, dy, dz, nc);
  for (int cl = 0; cl < nc; cl++) {
    double xe, ye, ze, te;
    double ee, dxe, dye, dze;
    trackHeed->GetElectron(cl, xe, ye, ze, te, ee, dxe, dye, dze);
    /*Garfield++ simulation according to needs*/
  }

  /*Electroluminescence example: Create the microscopic avalanche class*/
  Garfield::AvalancheMicroscopic * aval = new Garfield::AvalancheMicroscopic();
  /*Connect avalanche microscopic to sensor*/
  aval->SetSensor(sensor);
  /*Set the callback function*/
  aval->SetUserHandleInelastic(accessExcitationLevels);
  aval->AvalancheElectron(x0,y0,z0,t0,e0,0.,0.,0.);
  /*Loop over data structure filled in callback function, produce optical photons in Geant4
}

/*Define a global callback function*/
void accessExcitationLevels(double x, double y, double z, double t, int type, int level,Garfield::Medium * m) {
  /*Access the excitation levels, fill data
   structure with time and position of excitations*/
}
```
Listing 5: G4FastSimulationModel DoIt() method for Garfield++

```
void GarfieldG4FastSimulationModel::DoIt(const
   G4FastTrack& fastTrack, G4FastStep& fastStep) {
    /*Previously described steps*/

    /*Position of the Garfield++ particle*/
    G4ThreeVector position(x, y, z);

    /*Create new electron or optical photon*/
    G4DynamicParticle particle(G4Electron::
   ElectronDefinition(), G4RandomDirection(), eKin);
    G4DynamicParticle particle(G4OpticalPhoton::
   OpticalPhotonDefinition(),G4RandomDirection(),
   eKin);

    /*Set a number that is larger than the maximum
   number of expected secondaries*/
    fastStep.SetNumberOfSecondaryTracks(1000000);

    /*Create secondary electron*/
    fastStep.CreateSecondaryTrack(particle, position
   , time, true);

    /*Create secondary optical photon with random
   polarization*/
    fastStep.CreateSecondaryTrack(particle,
   G4RandomDirection(), position, time, true);
}
```

Listing 6: Production of secondary particles

```cpp
void GarfieldG4FastSimulationModel::Initialize() {
  /*Define the gas*/
  MediumMagboltz* gas = new MediumMagboltz();
  gas->SetComposition("ar", percentageAr, "co2",
  percentageCO2);
  gas->SetTemperature(temperature);
  gas->SetPressure(pressure);
  gas->LoadGasFile("ar_70_co2_30_1000mbar.gas");
  /*Set W value and Fano factor to literature or
   measured values*/
  gas->SetW(27.3090);
  gas->SetFanoFactor(0.1866);

  /*Create the geometry*/
  SolidBox* box = new SolidBox(0, 0., 0., dx, dy, dz
   );
  GeometrySimple* geo = new GeometrySimple();
  geo->AddSolid(box, gas);

  /*Make a component*/
  ComponentConstant* comp = new ComponentConstant();
  comp->SetGeometry(geo);
  comp->SetElectricField(-1000., 0., 0.);

  /*Make a sensor*/
  Sensor* sensor = new Sensor();
  sensor->AddComponent(comp);

  /*Create the track class*/
  TrackHeed* trackHeed = new TrackHeed();
  trackHeed->SetSensor(sensor);
  trackHeed->EnableDeltaElectronTransport();

  /*Create the drift algorithm*/
  AvalancheMC * drift = new Garfield::AvalancheMC();
  drift->EnableDiffusion();
  drift->DisableMagneticField();
  drift->SetSensor(sensor);

  /*Create the avalanche algorithm*/
  AvalancheMicroscopic * avalanche = new AvalancheMicroscopic();
  avalanche->SetSensor(sensor);
}
```

Listing 7: Setup of Garfield++ simulation

```
include(${Geant4_USE_FILE})
include_directories(${PROJECT_SOURCE_DIR}/include)
include_directories($ENV{GARFIELD_HOME}/Include)

link_directories($ENV{GARFIELD_HOME}/Library)
target_link_libraries(IonisationChamber ${Geant4_LIBRARIES} -lGarfield -lgfortran )
```

Listing 8: CMakeList.txt

```
void GarfieldSeppingAction::UserSteppingAction(const G4Step* theStep) {
  G4Track* track = theStep->GetTrack();
  G4double time = track->GetGlobalTime();
  G4ThreeVector wp = theStep->GetPostStepPoint()->GetPosition();
  G4ThreeVector p = theTouchable->GetHistory()->GetTopTransform().TransformPoint(wp);
  G4String name = track->GetDefinition()->GetParticleName();

  if(name == "e-" && track->GetStatus() == fStopAndKill) {
    /*Send position and time to Garfied++ to continue simulation*/
    SendElectronsToGarfield(p.getX(), p.getY(), p.getZ(), time);
  }
}
```

Listing 9: Storing of last positions of electron tracks in UserSteppingAction to generate electron-ion pairs for Garfield++

```
void GarfieldSeppingAction::UserSteppingAction(const
    G4Step* theStep) {
  double invFanoFactor = 1/0.19;
  G4double meanIon = G4LossTableManager::Instance()
   ->ElectronIonPair()->MeanNumberOfIonsAlongStep(
   theStep);
  /*Determine number of electron-ion pairs with the
   help of W and Fano*/
  G4int nIon =  G4lrint(G4RandGamma::shoot(meanion*
   invFanoFactor,invFanoFactor));

  /*Sample ionisation along step*/
  if(nIon > 0) {
    G4Track* fTrack = theStep->GetTrack();
    G4double t = fTrack->GetGlobalTime();
    G4ThreeVector p = theStep->GetPreStepPoint()->
   GetPosition();
    G4ThreeVector dp = theStep->GetPostStepPoint()->
   GetPosition() - p;
    for(G4int i=0; i<nIon; ++i) {
      G4ThreeVector nposw = p + dp*G4UniformRand();
      G4ThreeVector np = theTouchable->GetHistory()
   ->GetTopTransform().TransformPoint(nposw);

      /*Send position and time to Garfied++ to
   continue simulation*/
      SendElectronsToGarfield(np.getX(), np.getY(),
   np.getZ(), t);
    }
  }
}
```

Listing 10: Use of MeanNumberOfIonsAlongStep() in UserSteppingAction to generate electron-ion pairs for Garfield++

```
void DegradG4FastSimulationModel::DoIt(const
    G4FastTrack& fastTrack, G4FastStep& fastStep) {
  /*Start by killing the primary track*/
  fastStep.KillPrimaryTrack();

  /*Store the interaction position and time*/
  G4ThreeVector p = fastTrack.GetPrimaryTrack()->
    GetVertexPosition();
  G4double t = fastTrack.GetPrimaryTrack()->
    GetGlobalTime();

  /*Generate a random seed to Degrad*/
  G4int n = 54217137*G4UniformRand();
  G4String seed = G4UIcommand::ConvertToString(n);

  /* Write the Degrad input cards. Example for
    photon of 5.9 keV in pure Xenon (thermalization
    energy of 7.0 eV), in a 3000.0 V/cm electric
    field anti-parallel to the photon direction. The
    gas was considered to be at 20 degrees C and 760
    Torr. For details see Degrad source file */
  G4String degradCards="printf \"1,1,3,-1,"+seed+",
    5900.0,7.0,0.0\n
    7,0,0,0,0,0\n
    100.0,0.0,0.0,0.0,0.0,0.0,20.0,760.0\n
    3000.0,0.0,0.0,1,0\n
    100.0,0.5,1,1,1,1,1,1,1\n
    0,0,0,0,0,0\" > conditions.txt";
  G4int stdout=system(degradCards.data());

  /*Call Degrad with the input conditions file*/
  stdout=system("./degrad < conditions.txt");

  /*Here the user should write a simple code in C++
    just to read the Degrad ASCII file and get the
    electrons position and thermalization time.
    Remember that Degrad makes all interactions in (
    x0,y0,z0)=(0,0,0) and t0=0: the returned
```

```
   positions should be shifted according to
   interaction position and interaction time.
   Moreover, the Y and Z axes are swapped in Degrad
   relatively to Geant4. Distance units in Degrad
   are um and time is ps, whereas Geant4 uses mm and
    ns.*/
  G4ThreeVector ep;
  for (int electron=0;electron<
   NumberOfElectronsProduced; electron++) {
    ep.setX(posXDegrad*0.001+ip.getX());
    ep.setY(posZDegrad*0.001+ip.getY());
    ep.setZ(posYDegrad*0.001+ip.getZ());
    time=timeDegrad*0.001+t;

    /*Check if the electron position is inside the
   gas volume. Degrad does not use geometrical
   constraints*/
    G4Navigator* theNavigator =
   G4TransportationManager::GetTransportationManager
   ()->GetNavigatorForTracking();
    G4VPhysicalVolume* myVolume = theNavigator->
   LocateGlobalPointAndSetup(myPoint);
    G4String name=myVolume->GetName();
    /*gasName should be the name of the gas volume*/
    if (name.contains("gasName")) {
      G4DynamicParticle electron(G4Electron::
   ElectronDefinition(),G4RandomDirection(),
   thermalizationEnergy*eV);
      /*Create secondary electron*/
      G4Track *newTrack=fastStep.
      CreateSecondaryTrack(electron, ep,time,false);
      newTrack->SetTrackID(fastStep.
   GetNumberOfSecondaryTracks());
    }
  }
}
```

Listing 11: DoIt() method for Degrad

References

[1] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, F. Behner, L. Bellagamba, J. Boudreau, L. Broglia, A. Brunengo, H. Burkhardt, S. Chauvie, J. Chuma, R. Chytracek, G. Cooperman, G. Cosmo, P. Degtyarenko, A. Dell'Acqua, G. Depaola, D. Dietrich, R. Enami, A. Feliciello, C. Ferguson, H. Fesefeldt, G. Folger, F. Foppiano, A. Forti, S. Garelli, S. Giani, R. Giannitrapani, D. Gibin, J. G. Cadenas, I. González, G. G. Abril, G. Greeniaus, W. Greiner, V. Grichine, A. Grossheim, S. Guatelli, P. Gumplinger, R. Hamatsu, K. Hashimoto, H. Hasui, A. Heikkinen, A. Howard, V. Ivanchenko, A. Johnson, F. Jones, J. Kallenbach, N. Kanaya, M. Kawabata, Y. Kawabata, M. Kawaguti, S. Kelner, P. Kent, A. Kimura, T. Kodama, R. Kokoulin, M. Kossov, H. Kurashige, E. Lamanna, T. Lampén, V. Lara, V. Lefebure, F. Lei, M. Liendl, W. Lockman, F. Longo, S. Magni, M. Maire, E. Medernach, K. Minamimoto, P. M. de Freitas, Y. Morita, K. Murakami, M. Nagamatu, R. Nartallo, P. Nieminen, T. Nishimura, K. Ohtsubo, M. Okamura, S. O'Neale, Y. Oohata, K. Paech, J. Perl, A. Pfeiffer, M. Pia, F. Ranjard, A. Rybin, S. Sadilov, E. D. Salvo, G. Santin, T. Sasaki, N. Savvas, Y. Sawada, S. Scherer, S. Sei, V. Sirotenko, D. Smith, N. Starkov, H. Stoecker, J. Sulkimo, M. Takahata, S. Tanaka, E. Tcherniaev, E. S. Tehrani, M. Tropeano, P. Truscott, H. Uno, L. Urban, P. Urban, M. Verderi, A. Walkden, W. Wander, H. Weber, J. Wellisch, T. Wenaus, D. Williams, D. Wright, T. Yamada, H. Yoshida, D. Zschiesche, Geant4 - a simulation toolkit, Nuclear Instruments and Methods in Physics Research A 506 (506) (2003) 250–303. doi:10.1016/S0168-9002(03)01368-8.

[2] J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. A. Dubois, M. Asai, G. Barrand, R. Capra, S. Chauvie, R. Chytracek, G. A. P. Cirrone, G. Cooperman, G. Cosmo, G. Cuttone, G. G. Daquino, M. Donszelmann, M. Dressel, G. Folger, F. Foppiano, J. Generowicz, V. Grichine, S. Guatelli, P. Gumplinger, A. Heikkinen, I. Hrivnacova, A. Howard, S. Incerti, V. Ivanchenko, T. Johnson, F. Jones, T. Koi, R. Kokoulin, M. Kossov, H. Kurashige, V. Lara, S. Larsson, F. Lei, O. Link, F. Longo, M. Maire, A. Mantero, B. Mascialino, I. McLaren, P. M. Lorenzo, K. Minamimoto, K. Murakami, P. Nieminen, L. Pandola, S. Parlati, L. Peralta, J. Perl, A. Pfeiffer, M. G. Pia, A. Ribon, P. Rodrigues, G. Russo,

S. Sadilov, G. Santin, T. Sasaki, D. Smith, N. Starkov, S. Tanaka, E. Tcherniaev, B. Tome, A. Trindade, P. Truscott, L. Urban, M. Verderi, A. Walkden, J. P. Wellisch, D. C. Williams, D. Wright, H. Yoshida, Geant4 developments and applications, IEEE Transactions on Nuclear Science 53 (1) (2006) 270–278. doi:10.1109/TNS.2006.869826.

[3] J. Allison, K. Amako, J. Apostolakis, P. Arce, M. Asai, T. Aso, E. Bagli, A. Bagulya, S. Banerjee, G. Barrand, B. Beck, A. Bogdanov, D. Brandt, J. Brown, H. Burkhardt, P. Canal, D. Cano-Ott, S. Chauvie, K. Cho, G. Cirrone, G. Cooperman, M. Cortés-Giraldo, G. Cosmo, G. Cuttone, G. Depaola, L. Desorgher, X. Dong, A. Dotti, V. Elvira, G. Folger, Z. Francis, A. Galoyan, L. Garnier, M. Gayer, K. Genser, V. Grichine, S. Guatelli, P. Guéye, P. Gumplinger, A. Howard, I. Hrivnacova, S. Hwang, S. Incerti, A. Ivanchenko, V. Ivanchenko, F. Jones, S. Jun, P. Kaitaniemi, N. Karakatsanis, M. Karamitros, M. Kelsey, A. Kimura, T. Koi, H. Kurashige, A. Lechner, S. Lee, F. Longo, M. Maire, D. Mancusi, A. Mantero, E. Mendoza, B. Morgan, K. Murakami, T. Nikitina, L. Pandola, P. Paprocki, J. Perl, I. Petrovic, M. Pia, W. Pokorski, J. Quesada, M. Raine, M. Reis, A. Ribon, A. R. Fira, F. Romano, G. Russo, G. Santin, T. Sasaki, D. Sawkey, J. Shin, I. Strakovsky, A. Taborda, S. Tanaka, B. Tomé, T. Toshito, H. Tran, P. Truscott, L. Urban, V. Uzhinsky, J. Verbeke, M. Verderi, B. Wendt, H. Wenzel, D. Wright, D. Wright, T. Yamashita, J. Yarba, H. Yoshida, Recent developments in geant4, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 835 (2016) 186 – 225. doi:https://doi.org/10.1016/j.nima.2016.06.125.

[4] T. Kittelmann, M. Boin, Polycrystalline neutron scattering for Geant4: NXSG4, Computer Physics Communications 189 (2015) 114–118. doi:10.1016/j.cpc.2014.11.009.

[5] S. Incerti, A. Ivanchenko, M. Karamitros, A. Mantero, P. Moretto, H. Tran, B. Mascialino, C. Champion, V. Ivanchenko, M. Bernal, Z. Francis, C. Villagrasa, C. Baldacchino, P. Guèye, R. Capra, P. Nieminen, C. Zacharatou, Comparison of GEANT4 very low energy cross section models with experimental data in water, Medical Physics 37 (9) (2010) 4692–4708. doi:10.1118/1.3476457.

[6] M. Bernal, M. Bordage, J. Brown, M. Davidková, E. Delage, Z. E. Bitar, S. Enger, Z. Francis, S. Guatelli, V. Ivanchenko, M. Karamitros, I. Kyriakou, L. Maigne, S. Meylan, K. Murakami, S. Okada, H. Payno, Y. Perrot, I. Petrovic, Q. Pham, A. Ristic-Fira, T. Sasaki, V. Stepan, H. Tran, C. Villagrasa, S. Incerti, Track structure modeling in liquid water: A review of the geant4-dna very low energy extension of the geant4 monte carlo simulation toolkit, Physica Medica 31 (8) (2015) 861 – 874. doi:10.1016/j.ejmp.2015.10.087.

[7] Geant4 Collaboration, Chapter 5.2.6 Parameterization.
URL http://geant4.cern.ch/G4UsersDocuments/UsersGuides/ForApplicationDeveloper/html/TrackingAndPhysics

[8] T. Kittelmann, I. Stefanescu, K. Kanaki, M. Boin, R. Hall-Wilton, K. Zeitelhack, Geant4 based simulations for novel neutron detector development, Journal of Physics: Conference Series 513 (2) (2014) 022017.
URL http://stacks.iop.org/1742-6596/513/i=2/a=022017

[9] I. Hrivnacova, Geant4 VMC — ROOT a Data analysis Framework.
URL https://root.cern.ch/geant4-vmc

[10] Geant4 Collaboration, Photoabsorption Ionisation Model âĂŤ Physics Reference Manual 10.4 documentation.
URL http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/PhysicsReferenceManual/html/electromagnetic/energy{_}loss/pai.html?highlight=pai{#}id4

[11] V. Grichine, Ionization distribution near a relativistic particle track in gas, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 598 (2) (2009) 650 – 652. doi:10.1016/j.nima.2008.09.037.

[12] W. Allison, J. Bunch, J. Cobb, Relativistic charged particle identification by ionisation loss counters, Nuclear Instruments and Methods 153 (1) (1978) 65 – 67. doi:10.1016/0029-554X(78)90619-5.

[13] J. Apostolakis, S. Giani, L. Urban, M. Maire, A. Bagulya, V. Grichine, An implementation of ionisation energy loss in very thin absorbers for the GEANT4 simulation package, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers,

Detectors and Associated Equipment 453 (3) (2000) 597–605. `doi:10.1016/S0168-9002(00)00457-5`.

[14] V. Grichine, Recent validation and improvements of Geant4 standard EM package at low energies.
URL `https://indico.fnal.gov/event/4535/session/9/material/slides/0?contribId=93`

[15] M. Titov, L. Ropelewski, Micro-pattern gaseous detector technologies and rd51 collaboration, Modern Physics Letters A 28 (13) (2013) 1340022. `doi:10.1142/S0217732313400221`.

[16] R. Veenhof, Simulation of gaseous detectors.
URL `http://garfield.web.cern.ch/garfield/`

[17] H. Schindler, R. Veenhof, Garfield++ âĂŞ simulation of ionisation based tracking detectors.
URL `http://garfieldpp.web.cern.ch/garfieldpp/`

[18] R. Brun, F. Rademakers, Root - an object oriented data analysis framework, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 389 (1) (1997) 81 – 86, new Computing Techniques in Physics Research V. `doi:10.1016/S0168-9002(97)00048-X`.

[19] S. Guindon, R. Veenhof, A. Bellerive, Study of Drift Electrons Inside an ATLAS Muon Tube, Tech. rep. (2008).
URL `http://www.ipp.ca/pdfs/guindon{_}cern08.pdf`

[20] S. Biagi, Monte carlo simulation of electron drift and diffusion in counting gases under the influence of electric and magnetic fields, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 421 (1) (1999) 234 – 240. `doi:10.1016/S0168-9002(98)01233-9`.

[21] I. Smirnov, Modeling of ionization produced by fast charged particles in gases.
URL `http://ismirnov.web.cern.ch/ismirnov/heed`

[22] I. Smirnov, Modeling of ionization produced by fast charged particles in gases, Nuclear Instruments and Methods in Physics Research A 554 (2005) 474–493. `doi:10.1016/j.nima.2005.08.064`.

[23] S. Biagi, Transport of electrons in gas mixtures.
URL `http://magboltz.web.cern.ch/magboltz/`

[24] S. Biagi, Auger cascade model for electron thermalisation in gas mixtures produced by photons or particles in electric and magnetic fields.
URL `https://indico.cern.ch/event/245535/contributions/531797/attachments/420757/584265/cern.rd51.2013.pdf`

[25] J. Ziegler, SRIM-2003, Nuclear Instruments and Methods in Physics Research B 219-220 (2004) 1027–1036. `doi:10.1016/j.nimb.2004.01.208`.

[26] Geant4 Collaboration, Book For Application Developers.
URL `http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/fo/BookForAppliDev.pdf`

[27] C. Oliveira, H. Schindler, R. Veenhof, S. Biagi, C. Monteiro, J. dos Santos, A. Ferreira, J. Veloso, A simulation toolkit for electroluminescence assessment in rare event experiments, Physics Letters B 703 (3) (2011) 217 – 222. `doi:10.1016/j.physletb.2011.07.081`.

[28] C. Azevedo, D. González-Díaz, S. Biagi, C. Oliveira, C. Henriques, J. Escada, F. Monrabal, J. Gómez-Cadenas, V. Alvarez, J. Benlloch-Rodríguez, F. Borges, A. Botas, S. Cárcel, J. Cárcel, S. Cebrián, C. Conde, J. Díaz, M. Diesburg, R. Esteve, R. Felkai, e. L.M.P. Fern, P. Ferrario, A. Ferreira, E. Freitas, A. Goldschmidt, R. Gutiérrez, J. Hauptman, e. A.I. Hern, o. M. J.A. Hern, V. Herrero, B. Jones, L. Labarga, A. Laing, P. Lebrun, I. Liubarsky, N. Lopez-March, M. Losada, J. Martín-Albo, G. Martínez-Lema, A. Martínez, A. McDonald, C. Monteiro, F. Mora, L. Moutinho, J. M. Vidal, M. Musti, M. Nebot-Guinot, P. Novella, D. Nygren, B. Palmeiro, A. Para, J. Pérez, M. Querol, J. Renner, L. Ripoll, J. Rodríguez, L. Rogers, F. Santos, J. D. Santos, L. Serra, D. Shuman, A. Simón, C. Sofka, M. Sorel, T. Stiegler, J. Toledo, J. Torrent, Z. Tsamalaidze, J. Veloso, R. Webb, J. White, N. Yahlali, Microscopic simulation of xenon-based optical

tpcs in the presence of molecular additives, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 877 (2018) 157 – 172. doi:10.1016/j.nima.2017.08.049.

[29] S. Incerti, Important note on tracking.
URL https://twiki.cern.ch/twiki/bin/view/Geant4/LoweAtomicDeexcitation

[30] IAEA, Atomic and molecular data for radiotherapy and radiation research, Tech. rep. (1995).

[31] T. Doke, N. Ishida, M. Kase, Fano factors in rare gases and their mixtures, Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms 63 (4) (1992) 373 – 376. doi:10.1016/0168-583X(92)95207-8.

[32] D. Groom, N. Mokhov, S. Striganov, Muon Stopping Power and Range Tables 10 MeV - 100 TeV, Atomic Data and Nuclear Data Tables 78 (2) (2001) 183 – 356. doi:10.1006/adnd.2001.0861.

[33] F. Sauli, Gaseous Radiation Detectors. Fundamentals and Applications, Cambridge University Press, 2014.