Simulation of Ionisation with HEED and GEANT4

Piet Verwilligen

INFN Bari

TOF-PET FTM Simulation
September 22nd 2017, Bari

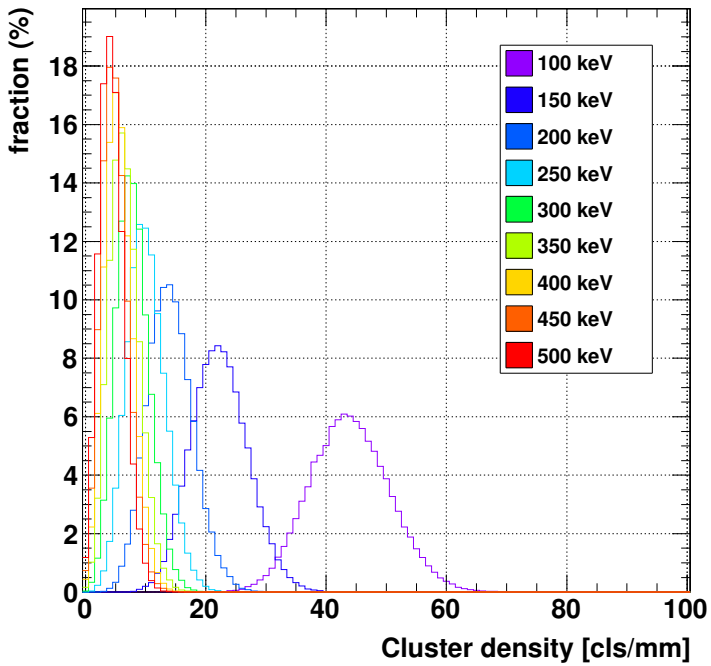
Tools to simulate e^- Ionisation

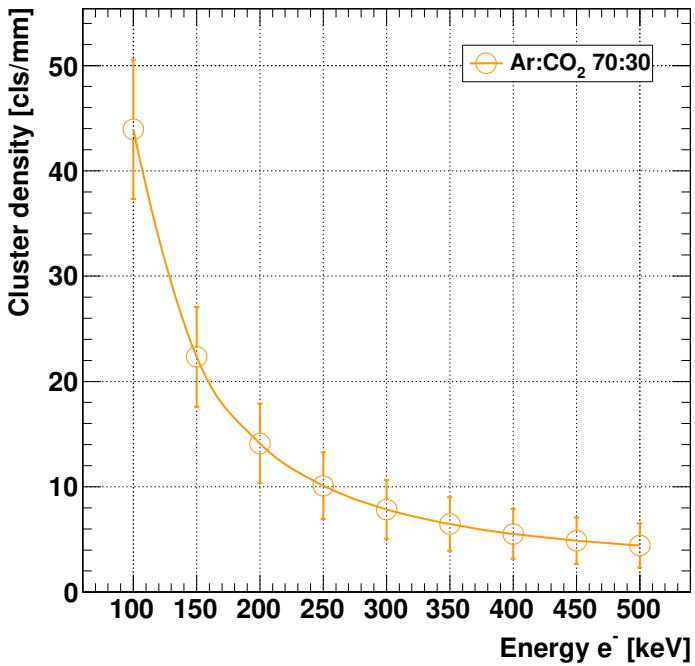
- **Heed with δ-e^--transport** *I. Smirnov*
  `TrackHeed` class in Garfield++
- **Heed without δ-e^--transport** *I. Smirnov, H .Schindler*
  δ-e^- simulated with Microscopic Tracking
  `TrackHeed` and `AvalancheMicroscopic` classes in Garfield++
- **Microscopic Tracking** *H. Schindler*
  `AvalancheMicroscopic` class in Garfield++
- **MIP / IMIP** *S. Biagi*
  `TrackElectron` class in Garfield++
- **Photo Absorption Ionisation (PAI) model** ...
  * implemented in **GEANT4:** `G4PAIPhotModel` (see e.g. TestEm8)
    http://geant4.web.cern.ch/geant4/UserDocumentation/Doxygen/examples_doc/html/ExampleTestEm8.html
  * implemented in **Garfield++:** `TrackPAI` class in Garfield++

- I had to solve a problem in HEED though ...

Today: Heed with δ-e^--transport

- **Ionisation of 100–500 keV e^- in 1 mm of Ar:CO_2 (70:30) mixture**

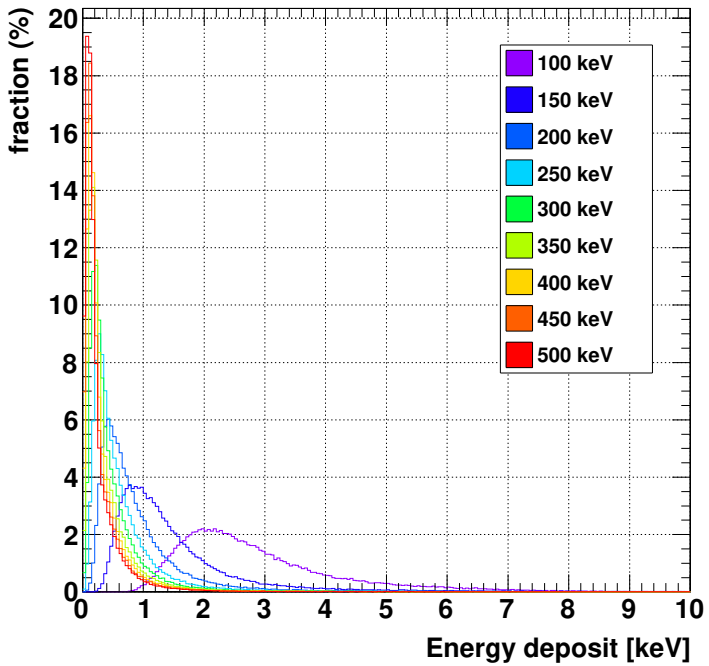




- Markers indicate MEAN and Error flags indicate RMS of distributions
- Will be reduced when using fitting to extract MEAN and STDEV

Today: Heed with δ-e^--transport

- **Energy Loss of 100–500 keV e^- in 1 mm of Ar:CO_2 (70:30) mixture**



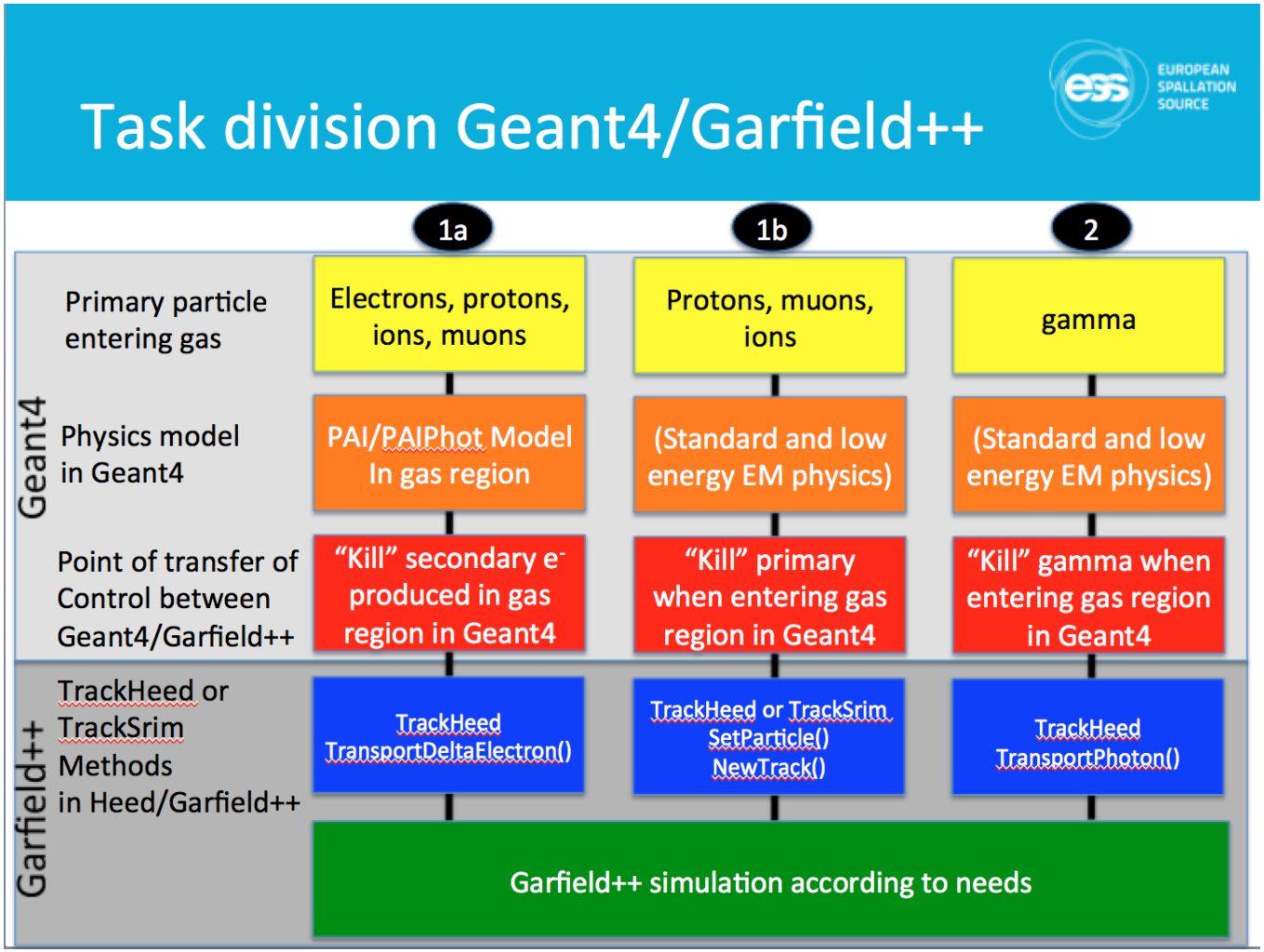
Next week(s):

- More cross check plots: electron range [cm], stopping power [MeV cm^2/g]
- Make plots for drift field of 3 kV/cm instead of 0.1 kV/cm
- Other gas mixtures: Ar:Isobutane (90:10), Ne:Isobutane (90:10)
- Other Ionisation simulations
- Estimates for Time Resolution
  - $1/n_{cls}v_{drift}$
  - d_{1stcls}/v_{drift}
  - Microscopic Tracking

Backup:

- Garfield++ - GEANT4 Interface
- Simulation of δ-e^- in HEED
- Mail to Garfield++/ heed users

Garfield++ - GEANT4 Interface



https://garfieldpp.web.cern.ch/garfieldpp/examples/geant4-interface/

Garfield++ - GEANT4 Interface

- `Relativistic charged particle enters thin absorber`
  Here two options exist: The Heed interface of Garfield++ can be used to create the primary ionization clusters. Heed uses the Photo-Absorption-Ionisation Model to create the ionisation clusters. The Photo-Absorption-Ionisation model also exists in Geant4 as G4PAIPhotModel for both light and heavy charged particles. So the alternative approach is to let Geant4 create the primary ionization clusters, and then take Garfield++ for the charge transport and the avalanches. How to use the G4PAIPhotModel is shown in example TestEm8.
- `Non-relativistic electron and/or very thick absorbers`
  Here it get's a little more tricky. Heed tracks the electron without Coulomb scattering. The stopping power of the electron is continuous and based on its initial energy. This works fine for Heed's intended purpose, the tracking of relativistic charged particles in thin absorbers. For slower electrons in thicker absorber (a use case not foreseen), it leads to unrealistically straight electron tracks and an incorrect energy loss. The Geant4 G4PAIPhotModel model, although also not intended for this use case, works better here and shows in combination with the Geant4 Coulomb scattering realistic tracks.
- `Non-relativistic heavy charged particle (α, ion) in gas`
  Either the Garfield++ SRIM interface (http://garfieldpp.web.cern.ch/garfieldpp/examples/srim/) or in Geant4 the G4PAIModel can be tried.
- `Very low energy (< 1 keV) electron in gas`
  The microscopic tracking of Garfield++ can be tried.
- `Gamma particle in gas`
  Heed can be used to transport the photon.

https://garfieldpp.web.cern.ch/garfieldpp/examples/geant4-interface/

Simulation of δ-e^- in HEED

Heed simulates the energy degradation of δ electrons and the production of secondary ("conduction") electrons using a phenomenological algorithm described in Ref. [18].

The asymptotic W value (eV) and the Fano factor of a `Medium` can be specified by the user by means of the functions

```
void Medium::SetW(const double w);
void Medium::SetFanoFactor(const double f);
```

If these parameters are not set, Heed uses internal default values. The default value for the Fano factor is $F = 0.19$.

The transport of δ electrons can be activated or deactivated using

```
void EnableDeltaElectronTransport();
void DisableDeltaElectronTransport();
```

If δ electron transport is disabled, the number of electrons returned by `GetCluster` is the number of "primary" ionisation electrons, i. e. the photo-electrons and Auger electrons. Their kinetic energies and locations are accessible through the function `GetElectron`.

If δ electron transport is enabled (default setting), the function `GetElectron` returns the locations of the "conduction" electrons as calculated by the internal δ transport algorithm of Heed. Since this method does not provide the energy and direction of the secondary electrons, the corresponding parameters in `GetElectron` are not meaningful in this case.

http://garfieldpp.web.cern.ch/garfieldpp/documentation/UserGuide.pdf

Mail to Garfield++/ HEED users

Piet Verwilligen <piet.verwilligen@cern.ch> Sep 20 (2 days ago)

to garfield-users

Dear Garfield Users

I would like to use HEED to simulate the ionization of electrons with a kinetic energy of a few hundred keV. Although these electrons are not relativistic, they are not classic either and are somewhat mildly relativistic. Now I do not know what exactly is the definition of "fast" used by Igor Smirnov in the paper describing HEED++ [1], but I would believe the electrons I am interested in are "fast".

Now when simulating the ionization (cluster density, dE/dx etc) of e.g. 100 keV electrons in a 1mm or 1cm thick gas volume (with small drift field of 100V/cm) I occasionally ran into an abort of the program. When running 100k events, I nearly hit it all the time, when simulating fewer events I was more lucky with the random numbers and the program finished fine.

Now I wanted to know why the HEED++ was aborting and whether my use case is a valid use case (since electrons are light and since they are not relativistic). On the one hand I read on the GEANT-Garfield++ interface page [2] that my case is borderline since HEED does not simulate the Coulomb scatterings. On the other hand I see that in the HEED++ paper [1] in Figure 1 that HEED++ is used to simulate the electron range down to electron energies of 200 eV, which is way below my use case. In the Garfield++ manual I also read that HEED can track (somehow) delta-electrons.

Mail to Garfield++/ HEED users

Now to be clear, the error I got that initiated an abort of the program is this:

```
ERROR:

d1<= 0
d1=-0.00445221
FunNameStack: s_init=1 qname=0
File is /lustre/home/piet/Project_CSN5_MPGD_FaTimA/garfieldpp-20170703//Heed/wcpplib/math/
kinem.cpp , line number is 44
```

(full log here [3]). I went digging a bit into the code and I found that the origin of the problem is in the function "physics" of HeedParticle.cpp. For this specific event I got an energy loss that is larger than the kinetic energy of the particle [4], leading to an error in kinem.cpp in the function "theta_two_part". Now indeed for electrons very large energy loss can occur when traversing materials, and the probability that the electron is stopped is not negligible.

Now I can think of a rather simple solution of this item, for now I checked whether the energy loss is not larger than the kinetic energy of the particle, and if it is larger than the kinetic energy of the particle i set it to the kinetic energy of the particle minus epsilon, where epsilon is a small number (1eV) to avoid zeroes in the calculation of "theta_two_part" again.

I was wondering whether some people could tell me whether I am not (ab)using the program for the calculation of ionization of 100 keV electrons (maybe there are other problems under the hood that I am neglecting) and in case not, whether the easy trick I use is justified. I am wondering whether other people experienced a similar problem in the past. Any ideas on how to deal with this situation or debug HEED are highly appreciated.

Mail to Garfield++/ HEED users

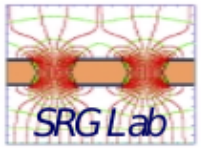
[4] some private printout of this event before the program abort:

```
HeedParticle :: physics :: Here it goes wrong :: theta_two_part (Ep0=0.520692, Ep1=0.506624,
Mp=0.510999, Mt=0.510999)
                           Ep0 = mass * c_squared + curr_kin_energy      = 5.68563e-06 *
89875.5 + 0.00969283
                           Ep1 = Ep0 - transferred_energy[qtransfer - 1] = 0.520692 - 0.0140679
                           Mp = mass * c_squared                        = 5.68563e-06 *
89875.5
                           Mt = electron_def.mass * c_squared           = 5.68563e-06 *
89875.5
kinem.cpp :: theta_two_part
             d0 = Ep0 * Ep0 - Mp2
             d1 = Ep1 * Ep1 - Mp2
ERROR:

d1<= 0
d1=-0.00445221
```

Main Contribution to Intrinsic σ_t: $(1/nv)$

The gas mixtures



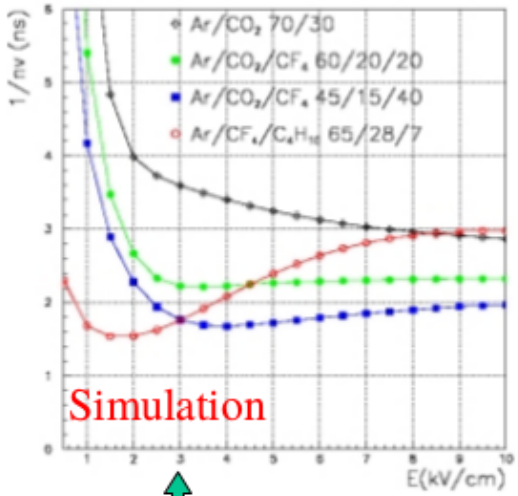
In the beam tests we studied 4 different gas mixtures:

1. Ar/CO_2 70/30;
2. $Ar/CO_2/CF_4$ 60/20/20;
3. $Ar/CO_2/CF_4$ 45/15/40;
4. $Ar/CF_4/C_4H_{10}$ 65/28/7;

Given

- n: the number of clusters per unit length;
- v: the electron drift velocity in the drift gap;

The 1/nv term is the main contribution to the intrinsic time resolution of this kind of detector.



Drift field 3 kV/cm

The $Ar/CO_2/CF_4$ 45/15/40 gas mixture should give the same time performance as the $Ar/CO_2/C_4H_{10}$ 65/28/7.

F. Murtas LNF/INFN

Frascati 28 November 2002

http://www.lnf.infn.it/seminars/talks/murtas_28_11_02.pdf