

A MONTE-CARLO SOFTWARE TOOLS FOR THE CHARACTERIZATION OF GAS MIXTURES IN VARIOUS DETECTOR

MICHELE RENDA^{1,2}

¹Department of High Energy and Elementary Particle Physics,
“Horia Hulubei” National Institute for Physics and Nuclear Engineering,
Reactorului 30, RO-077125, P.O.B. MG-6, Măgurele, Romania

²Faculty of Physics, University of Bucharest,
Atomiștilor 405, RO-077125, P.O.B. MG-11, Măgurele, Romania
E-mail: michele.renda@cern.ch

Received December 6, 2017

Abstract. In this article we present the conceptual design of a new software tool for the estimation of the electron swarm characteristics in gas mixture under intense static electric and magnetic fields. We present the results of the preliminary performance tests that were performed with simulations.

Key words: software tool, gaseous detectors, simulation.

1. INTRODUCTION

Gas detectors have a major role in high energy physics for the construction of large particle detectors. Their operating principle is quite simple: when an incident particle interacts with a molecule of the gas mixture, some electrons are extracted and drifted forward to some electrodes where they induce an electric signal. The electric field maintains the electron drift and may introduce an intrinsic amplification.

A challenging task for the simulation of these detectors is represented by the determination of the gas characteristics like the mean free path, the first Townsend coefficient, drift speed, etc. Due to the numerous processes involved - elastic and inelastic collisions, ionizations, attachment, the analytic approach can provide satisfactory results only when special conditions are met, *i.e.* when the elastic processes are prevailing over the inelastic ones.

A valid approach implies solving of the Boltzmann transport equation [1]: because this approach is not trivial, the standard solution truncates the energy distribution expansion after the first two terms of the Legendre polynomials. This approximation produces valid results when the gas is composed of noble gases at moderate electric fields, where the inelastic scattering is absent [2]. Increasing the number of terms of the Legendre polynomials will improve the accuracy of the results at the cost of a higher computational time. For example, adding a third term will improve the accuracy of drift velocity from a value $\approx 2\%$ [3] to a value better than 1% [4].

A multi term solution of the Boltzmann equation was developed by Ness [5]. Using this method, it is possible to characterize the behavior of electron swarms in neutral gases under the influence of crossed electric and magnetic static fields.

There are several numeric methods for solving the Boltzmann equation [6]. Using a combination of adaptive grids and approximate fast Fourier transform, they provide us with the solution of the discontinuous distribution function. This approach succeeds to produce results comparable with those obtainable by experiments.

We designed a software tool able to produce a full swarm characterization in gas mixture using Monte-Carlo technique. The correctness and the performances of this tool were tested and validated using simulation data.

The paper is organized as follows: in section 2 is presented the Monte-Carlo method for the determination of the gas mobilities, in section 3 are presented the existing solutions, and in section 4 and 5 our implementation.

2. MONTE CARLO METHODS

A different approach allows calculating results with a higher accuracy than with analytical methods. As input, it only requires the cross sections tables for elastic and inelastic processes of each gas component. The major advantage of this approach is that it can handle gases with both elastic and inelastic interaction and with arbitrary electric and magnetic static fields. The main weak point, the simulation of a large set of interactions, can be mitigated using a parallel computing implementation.

If we have a gas mixture we can set [7, 8] the free time between two collisions t_c using a pseudo-random number $R \in [0, 1)$ according the formula:

$$\ln \left(\frac{1}{1-R} \right) = \int_0^{t_c} \nu(t) dt \quad (1)$$

where, $\nu(t)$ is the collision frequency at a given time. The expression of $\nu(t)$ can be determined knowing all cross sections of each gas components:

$$\nu(\varepsilon) = \sqrt{\frac{2\varepsilon}{m_e}} \sum_{k=1}^N n_k \left(\sigma_k(\varepsilon) + \sum_{i=1}^{J_k} \sigma_{ki}(\varepsilon) \right) \quad (2)$$

where σ_k is the elastic scattering cross section for the k component σ_{ki} is the i -th inelastic scattering cross section and n_k is the component molecular number volume density.

The main difficulty of equation 1 is that we need to numerically solve the integral for every value of R . An optimization is to use the “null-collision” technique [8]. This solution allows us to avoid the integration of equation 2 for each value of R replacing $\nu(\varepsilon)$ in equation 1 with a constant value ν' , getting a simple relation

between R and t :

$$t_c = -\frac{\ln(1-R)}{\nu'} \quad (3)$$

This substitution is valid only if a number of null collisions, $\nu' - \nu(\varepsilon)$, which does not alter the particle direction and speed, are added into the simulation [8].

The determination of the value of ν' is not a trivial task: the value must be always than real $\nu(t)$ for all accessible particle energies but a too high value could cause an excessive number of null collisions and affect the computation performance. The algorithm presented by Skullerud [8] can be used to determine the optimal value of ν' for a given setup.

3. EXISTING SIMULATION TOOLS

Magboltz [9] software tool was developed by Stephen Biagi and is currently used to model electrons motion in gas avalanche radiation detectors [4]. This software is using different approaches to have a good accuracy on the drift velocity: using a solution of the Boltzmann transport equation, truncated to the third term of Legendre polynomials, it was able to calculate the drift velocity with an accuracy better than 1 %. Unfortunately, this solution was not performing very well for calculating the Lorentz angle, with deviations up to 10 %. To overcome this problem the Monte Carlo integration technique was adopted by Magboltz: this solution produces the desired result accuracy and it is only limited by the provided computation time.

Magboltz is currently used in high energy physics experiments to optimize and design new classes of gas detectors, and represent a valuable resource. However some weak points limit its versatility:

- Being written in Fortran it is very difficult to interface it with software written in other languages
- There is no support for vectorization and multi-threading processing which could help to drastically reduce the computation time and provide more accurate result
- It can use only the gas cross section existing in its distribution and it is not possible to use external data.

4. CONCEPTUAL DESIGN FOR A NEW SOFTWARE TOOL

In this section, we will present the requirements for a new software tool, which, using updated technologies, will improve the existing weak points of the existing implementations and also adding new features. We will present a list of requirements that were imposed to our design.

4.1. DEVELOPMENT TOOLS

Despite `Fortran` is still widely used in the scientific community, it is now being replaced with object oriented languages. These languages allow writing a more maintainable and reusable code. `C++` was chosen as the main programming language because it allows developing an efficient implementation. The solution must be available as a shared library, allowing an immediate integration with any existing `C++` program. It should be properly documented with a completed code documentation (*via* an automatic document building tools like `Doxygen` [10]) and the library building should use a common building tool like `CMake` [11].

4.2. CROSS SECTION DATABASES

The solution chosen by `Magboltz` was to include the cross section databases inside the program distribution. Indeed, this was a good solution, allowing the user to become immediately productive. However, this choice is limiting the possibility to use external or self-made databases, limiting the versatility of the platform. A better implementation should allow the user to load its own cross section database. `LXCat` [12] is a well know archive of gas cross sections (including the database used in `Magboltz`): it consists of a web application giving the user the possibility to browse the existing databases and perform comparisons between them. A very important feature of `LXCat` is the possibility to export the data into `XML` format. Therefore, the user should be able to read the `XML` files and to perform the simulation using custom cross sections.

4.3. STATIC ELECTRIC AND MAGNETIC FIELDS

An important requirement is the capability to manage static electric and magnetic fields with arbitrary direction. Good performances could be archived using analytic expressions but these expressions would become quite complex. A workaround is a change of coordinate from the global coordinate system (x, y, z) to a local one (x', y', z') such that the \hat{z}' versor is now directed in the same direction as the magnetic field \vec{B} .

In this new coordinate system the expression of electron velocity $v(t)$ is:

$$\begin{aligned} v'_x(t) &= \left(v'_{0x} - \frac{E'_y}{B'_z} \right) \cos(\omega_0 t) + \left(v'_{0y} + \frac{E'_x}{B'_z} \right) \sin(\omega_0 t) + \frac{E'_y}{B'_z} \\ v'_y(t) &= \left(v'_{0y} + \frac{E'_x}{B'_z} \right) \cos(\omega_0 t) - \left(v'_{0x} - \frac{E'_y}{B'_z} \right) \sin(\omega_0 t) - \frac{E'_x}{B'_z} \\ v'_z(t) &= v'_{0z} + \frac{q_0}{m_0} E'_z t \end{aligned} \quad (4)$$

and the electron position

$$\begin{aligned}
s'_x(t) &= s'_{0x} + \left(v'_{0y} + \frac{E'_x}{B'_z}\right) \left(\frac{1}{\omega_0}\right) + \frac{E'_y}{B'_z} t + \left(v'_{0x} - \frac{E'_y}{B'_z}\right) \left(\frac{1}{\omega_0}\right) \sin(\omega_0 t) \\
&\quad - \left(v'_{0y} + \frac{E'_x}{B'_z}\right) \left(\frac{1}{\omega_0}\right) \cos(\omega_0 t), \\
s'_y(t) &= s'_{0y} - \left(v'_{0x} - \frac{E'_y}{B'_z}\right) \left(\frac{1}{\omega_0}\right) - \frac{E'_x}{B'_z} t + \left(v'_{0y} + \frac{E'_x}{B'_z}\right) \left(\frac{1}{\omega_0}\right) \sin(\omega_0 t) \quad (5) \\
&\quad + \left(v'_{0x} - \frac{E'_y}{B'_z}\right) \left(\frac{1}{\omega_0}\right) \cos(\omega_0 t), \\
s'_z(t) &= s'_{0z} + v'_{0z} t + \frac{q_0 E'_z}{2m_0} t^2,
\end{aligned}$$

where

$$\omega_0 = \frac{q_e}{m_e} B'_z \quad (6)$$

is the cyclotron frequency of the electron inside the magnetic field, \vec{E}' and \vec{B}' are the electric and magnetic field, \vec{v}'_0 the particle initial velocity and \vec{s}'_0 the initial position.

It is important to notice that because all the vectors from equations above were expressed in the local coordinate system, they need to be transformed to the laboratory coordinate system.

5. THE PROTOTYPE

Based on the requirements presented in the previous section, we developed a prototype to test the performance of our conceptual design. Due to the magnitude of the workload needed to implement a robust and versatile solution, it was decided to split the software too into three separated components: The first component, present in section 5.1, is used to perform vector operations performing the dimensional check at compile time. The second component, presented in section 5.2, perform the parsing of existing cross section tables. The last component, described in section 5.3, perform the actual simulation. This organization was chosen to encourage other researchers to reuse the created libraries in their simulation, with the intention to produce a set of specialized libraries for the high energy physics community.

5.1. UNIVEC

The first component developed is a library which takes care of vector arithmetic. The common solution for vector arithmetic used in high energy physics is

CLHEP [13, 14]. This library contains a mature set of classes used for vector calculation, random number generation, matrix handling and much other. It is currently widely used at CERN and it is integrated into Geant4.

One of the instruments used in physics to cross-check the correctness of an expression is the dimensional analysis. Despite this instrument is not able to find out all possible errors, it helps finding a wide class of mistakes that could occur. The programming language currently available does not have support for dimensional checking in their core library. For C++ there is an external library, Boost Units [15], which allows a zero-overhead static dimensional checking at compile time.

This library is mature and reliable but it does not provide vector arithmetic functionalities. Therefore, we developed a vector library with dimensional type checking and we published it at: <https://github.com/micrenda/univec>

In Table 1, we present the classes developed to create and operate on unit-aware 2D and 3D vectors. There is the support for the standard C++ operators (like +, *, etc.) to perform vector operations and an utility class was created to switch from a set of coordinate to another to adapt to the problem symmetry.

Table 1

Vector classes implemented in Univec

	2D	3D
Cartesian	VectorC2D	VectorC3D
Spherical	VectorP2D	VectorS3D
Cylindrical	-	VectorY3D

5.2. ZCROSS

Another very important aspect that was considered was the cross-section data set to be used in the calculation. We decided to avoid using a hard-coded table as in Magboltz but to give the possibility to load databases from existing collections. In our first implementation, we decided to focus on LXCat [12] and we developed a library able to parse the XML file exported from LXCat homepage. An intermediate XML format (with associated XSD schema) was created and a set of classes were created to be able to manage total and differential cross sections. In Figure 1 are shown the cross section plots for the $e+\text{CO}_2$ process from Itikawa database [16].

The code we developed code was validated parsing the databases existing on LXCat and is available at <https://github.com/micrenda/zcross>. We decided to publish it as a separate project to promote its use by third party software with needs to read cross sections files.

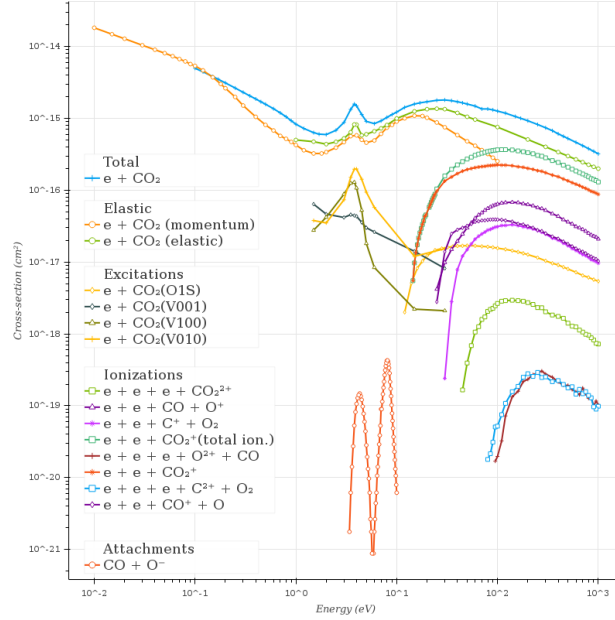


Fig. 1 – Plots of the cross-section for $e+\text{CO}_2$ process from the Itikawa database [16] divided by interaction type.

5.3. BETABOLTZ

The third package described in this article, available at <https://github.com/micrenda/betaboltz>, performs a microscopic simulation of the electron motion in gases of any arbitrary gas mixture, pressure, temperature and static electric and magnetic fields. It performs both detailed avalanche simulations and statistical aggregations to determine parameters like drift velocities, diffusion and Townsend coefficients.

The layout of the developed solution is very simple, the core component implements the common functionalities such as managing the movement of electrons or ions, calculating the interaction frequency $\nu(\epsilon)$ and selecting interaction types. A set of “handlers” are registered and are called when certain conditions are met, as shown on Table 2. The end user can choose to freely implement the handler extending the class `BaseHandler` or just use some existing implementations for common operations like saving to CSV (`ExportCSVHandler`) or calculate drift attributes (`SwarmHandler`). In Figure 2, we present an example of the simulation of the avalanche generated by a single electron under a static electric field: we show how we can track the microscopic evolution of an electron swarm and how we can extract information about the processes taking place.

In Figure 3 we present the determination of the electron drift velocity in CF_4 .

Table 2

Handlers in Betaboltz.

Name	Description
onTableSelection	Executed at the first step of the run when the cross section tables are selected.
onRunStart	Executed when starting a new run.
onRunEnd	Executed when completing a run.
onEventStart	Executed when starting a new event.
onEventEnd	Executed when completing an event.
onBulletCreate	Executed when withing an event a new bullet, an electron or an ion, is created, by initial setup or by ionization.
onBulletCollision	Executed when withing an event there is a collision between a bullet and a gas molecule.
onBulletDestroy	Executed when a bullet, an electron or a ion, is destroyed, due to the end of the event or by attachment.

This gas has a large Ramsauer dip in the elastic cross-section, providing a high drift velocity and fast detector response. The calculation of the drift velocity and diffusion in CF_4 is used by Biagi *et al.* [4] as a benchmark for Magboltz.

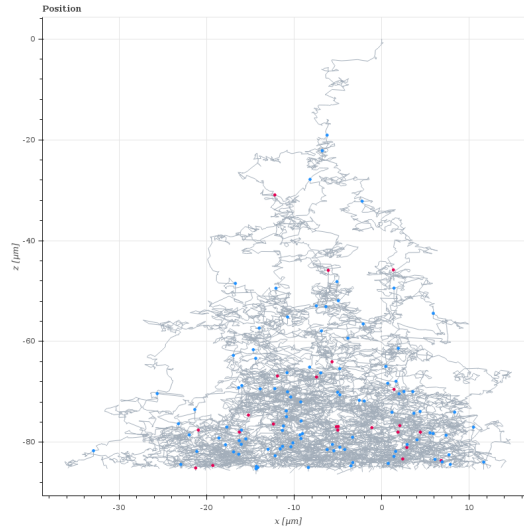


Fig. 2 – (Color online). Electron avalanche amplification under static electric field $E_z = 40 \text{ kV/cm}$. Gas mixture is composed of Ar and CO_2 with mass ratio 93 : 7 and density 1.66575 mg/cm^3 . Red dots represent CO_2 ionization while blue dots refer to Ar ionization events. Cross section data for Ar and CO_2 were taken respectively from Biagi [9] and Itikawa [16] databases via LXCat[12].

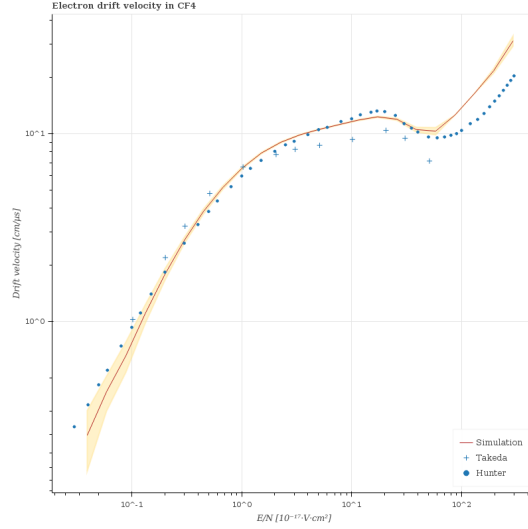


Fig. 3 – (Color online). Drift velocity of an electron in CF_4 with mass density 3.72 mg/cm^3 . Blue circles represent the experimental drift velocities determined by Hunter *et al.* [17], while blue crosses are the experimental values by Takeda and Ikuta [18]. The yellow area represents the standard deviation of the value obtained by our simulation. Cross section data are taken from Bondage database [19, 20].

6. CONCLUSION

In this article we presented a brief review of the theoretical framework of the Monte-Carlo gas analysis. The existing solutions were analyzed and we propose a new software tool. The developed components were tested, and made publicly available. A substantial effort is required to have a full implementation of Betaboltz, including the validation of the results against existing experimental data. First result obtained for swarm in Ar:CO_2 were presented and give us a robust feedback to continue the development of this software tool.

Acknowledgements. This work was supported by the research project ATLAS CERN-RO. We would like to thank our ATLAS group colleagues, in particular Călin Alexa, for guidance and support.

REFERENCES

1. L. Boltzmann, *Vorlesungen über gastheorie*, vol. **1**, (JA Barth, 1896).
2. G. Schultz, J. Gresser, *Nuclear Instruments and Methods* **151**(3), 413–431 (1978); DOI:10.1016/0029-554X(78)90151-9.
3. I. D. Reid, *Australian Journal of Physics* **32**(3), 231–254 (1979); DOI:10.1071/ph790231.
4. S. F. Biagi, *Nuclear Instruments and Methods* **A421**, 234–240 (1999); DOI:10.1016/S0168-9002(98)01233-9.

5. K. F. Ness, Journal of Physics D: Applied Physics **27**(9), 1848 (1994); DOI:10.1088/0022-3727/27/9/007.
6. A. Heintz, P. Kowalczyk, R. Grzhibovskis, Journal of Computational Physics **227**, 6681–6695 (2008); DOI:10.1016/j.jcp.2008.03.028.
7. H. B. Milloy, R. W. Crompton, J. A. Rees, A. G. Robertson, Australian Journal of Physics **30**(1), 61–72 (1977); DOI:10.1071/ph770061.
8. H. R. Skullerud, Journal of Physics D: Applied Physics **1**(11), 1567 (1968); DOI:10.1088/0022-3727/1/11/423.
9. *Magboltz - transport of electrons in gas mixtures*, <http://magboltz.web.cern.ch/magboltz/>.
10. D. Van Heesch, *Doxygen: Source code documentation generator tool* (2008), www.doxygen.org.
11. *Cmake*, <https://cmake.org/>.
12. L. C. Pitchford, L. L. Alves, K. Bartschat, S. F. Biagi, M.-C. Bordage *et al.*, Plasma Processes and Polymers **14**(1-2) (2017); DOI:10.1002/ppap.201600098.
13. L. Lönnblad, Computer Physics Communications **84**, 307–316 (1994); DOI:10.1016/0010-4655(94)90217-8.
14. *Clhep - a class library for high energy physics*, <http://proj-clhep.web.cern.ch/proj-clhep/>.
15. M. C. Schabel, S. Watanabe, *Boost. units 1.1.0* (2003).
16. Y. Itikawa, Journal of Physical and Chemical Reference Data **31**(3), 749–767 (2002); DOI:10.1063/1.1481879.
17. S. R. Hunter, J. G. Carter, L. G. Christophorou, Physical Review A **38**(1), 58–69 (1988); DOI:10.1103/PhysRevA.38.58.
18. A. Takeda, N. Ikuta, Journal of the Physical Society of Japan **62**(7), 2368–2381 (1993); DOI:10.1143/JPSJ.62.2368.
19. M. C. Bordage, P. Ségur, A. Chouki, Journal of Applied Physics **80**(3), 1325–1336 (1996); DOI:10.1063/1.362931.
20. M.-C. Bordage, P. Ségur, L. G. Christophorou, J. K. Olthoff, Journal of Applied Physics **86**(7), 3558–3566 (1999); DOI:10.1063/1.371258.