CERN-ACC-2019-0029

July 2019
eric.wulff@cern.ch
giovanni.iadarola@cern.ch

Implementation and benchmarking of the Furman-Pivi model for Secondary Electron Emission in the PyECLOUD simulation code

E. G. T. Wulff and G. Iadarola
CERN, CH-1211 Geneva, Switzerland

Keywords: electron cloud, Furman-Pivi model, PyECLOUD, secondary electron emission

Summary

The Furman-Pivi model for the secondary electron emission process has been implemented in the PyECLOUD simulation code and can now be used in alternative to the ECLOUD model, developed for the LHC beam screen surface at the time of the LHC design.

In this document we describe the ECLOUD and the FP secondary emission models, and their implementation in PyECLOUD. We compare results of simulations performed using the two models and we crosscheck our simulation results against the POSINST code developed at LBNL (Berkeley).

Contents

1 Introduction

PyECLOUD is an open source macroparticle (MP) code that simulates electron cloud effects in particle accelerators [1, 2]. It is developed and maintained at CERN and is the evolution of the ECLOUD code, developed at CERN since the late '90s [3]. PyECLOUD is extensively used in e-cloud studies for the operating CERN accelerator complex, as well as for their upgrade projects and for other existing and planned particle accelerators.

A key part in the simulation of these phenomena is the modelling of the Secondary Electron Emission process, which takes place when an electron impacts on the beam chamber surface. The surface modelling implemented in PyECLOUD is inherited from the ECLOUD code and it is therefore often called "ECLOUD model" of secondary electron emission. It was developed at the time of the LHC design and construction, based on laboratory measurements performed on LHC beam screen surface samples [4].

Other similar simulation codes like POSINST [5] or openECLOUD [6] employ a different modelling of the surface behaviour, following the approach introduced by Furman and Pivi in [7]. These differences have made it difficult to compare results from the different codes, for example for code benchmarking purposes.

The Furman-Pivi (FP) model has been recently implemented also in the PyECLOUD simulation code and can be optionally used for e-cloud simulations instead of the ECLOUD model. This allows direct comparisons between the two models and a quantitative benchmarking against other simulation codes.

In this document we describe the ECLOUD and the FP secondary emission models, and their implementation in PyECLOUD. We compare results of simulations performed using the two models and we crosscheck our simulation results against the POSINST code.

2 Secondary Electron Emission

Secondary electron emission is the process in which electrons impacting on a surface cause the surface to emit more electrons. It is primarily described by a surface property called the Secondary Electron Yield (SEY), commonly denoted as δ, which is defined as the ratio between electron current emitted from the surface and the electron current impacting on the surface:

$$\delta(E_0, \theta_0) = \frac{I_{emit}}{I_{imp}}, \tag{1}$$

where δ is the SEY, E_0 is the energy of the impacting electrons and θ_0 the angle of incidence of the electrons with respect to the direction normal to the surface.

A model describing the secondary electron emission process for a certain surface needs to define:

- the dependence of δ on E_0 and θ_0;
- the energy distribution of the emitted electrons;
- the angular distribution of the emitted electrons.

The following two sections explain how the ECLOUD and the FP models handle these key points. Both models define different components of the emitted current and therefore of the SEY, which correspond to different physical mechanisms, as it will be detailed in the following.

2.1 The ECLOUD model

An overview of the ECLOUD model used in PyECLOUD will be presented here. For a more detailed description see [8].

In the ECLOUD model, the SEY is divided in two components:

$$\delta = \delta_e + \delta_{ts}, \tag{2}$$

where δ_e is associated to elastic interactions of the electrons with the surface, and δ_{ts} is associated to "true secondary" electrons.

The elastic component is independent of the angle of incidence, while it depends on the energy of the impacting electrons following the relation

$$\delta_e(E_0) = R_0 \left(\frac{\sqrt{E_0} - \sqrt{E_0 + E_e}}{\sqrt{E_0} + \sqrt{E_0 + E_e}} \right)^2, \tag{3}$$

which is chosen based on experimental data (R_0 and E_e are parameters defined on the basis of laboratory measurements [4].

The true secondary component is modelled by the following expression, which is found to be suited to fit experimental data [9]:

$$\delta_{ts}(E_0, \theta_0) = \delta_{ts}(\theta_0) D(E_0/E_{max}) \tag{4}$$

where:

$$D(x) = \frac{sx}{s - 1 + x^s}, \tag{5}$$

$$\delta_{ts}(\theta_0) = \delta_{ts}(\theta_0 = 0) e^{(1-\cos\theta_0)/2}, \tag{6}$$

$$E_{max}(\theta_0) = E_{max}(\theta_0 = 0)(1 + 0.7(1 - \cos\theta_0)). \tag{7}$$

$\delta_{ts}(\theta_0 = 0)$, $E_{max}(\theta_0 = 0)$ and s are again parameters, which are chosen based on laboratory measurements.

Typical SEY curves for the LHC beam screen surface are shown in Fig. 1. It can be noticed that at low energies, both δ_e and δ_{ts} contribute significantly to the total SEY, while at high energies the elastic component is negligible.

The SEY and its components can be defined in terms of the current hitting on and being emitted from the surface. In Fig. 2 the impacting current is divided into a component that is elastically scattered (I_{elas}) and one that penetrates into the material (I_{pen}) and interacts in a more complex way, generating true secondaries.

Using the currents in Fig. 2, the SEY components can be defined as

$$\delta_e = \frac{I_{elas}}{I_{imp}} \tag{8}$$

$$\delta_{ts} = \frac{I_{ts}}{I_{imp}} \tag{9}$$

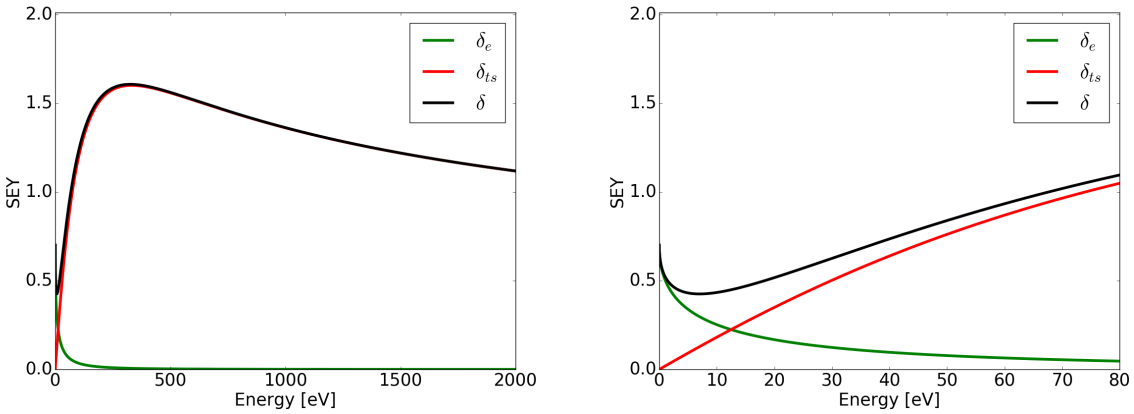

Figure 1: Left: the SEY components for the ECLOUD model. Right: zoom over the low-energy part of the curves.

It is computationally impossible to simulate electrons individually. Hence, electrons are grouped into macroparticles (MPs), each representing a certain number of electrons. In ECLOUD, when a MP collides with the chamber surface, it is simply rescaled according to the total SEY corresponding to the impinging angle and electron energy of the impacting MP.

The ratio between $\delta_e(E_0)$ and $\delta_{ts}(E_0, \theta_0)$ determines the probability of the event being treated as an elastic event or as a true secondary event. As shown in Fig. 3 the two categories of events are handled in the following way:

- For elastic events, the MP is emitted with the same energy with which it had impacted and with an angle that is opposite to the direction of impact;
- For true secondary events, the emission angle is generated according to a cosine distribution and the energy is generated according to a lognormal distribution, defined as:

$$\frac{dn_{true}}{dE} = \frac{1}{E\sigma_{true}\sqrt{2\pi}}\, e^{-\frac{(\ln(E)-\mu_{true})^2}{2\sigma_{true}^2}}, \tag{10}$$

  which is found to be appropriate to fit laboratory measurements. The behaviour of such a distribution is illustrated in Fig. 4.

A different scheme is also available in PyECLOUD, in which MPs undergoing elastic events are not rescaled, and the SEY is determined exclusively by rescaling MPs from the true secondary events only. Tests have shown no differences between simulations performed with the two schemes [10].

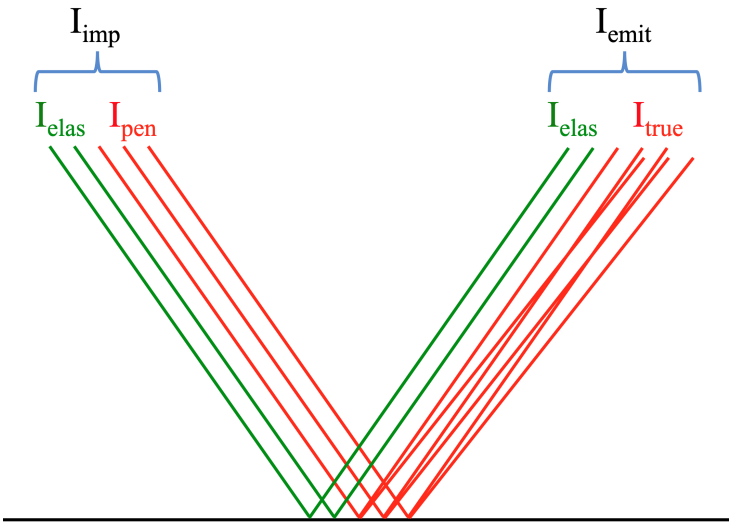

Figure 2: Decomposition of the current impacting on and emitted from a surface used in the ECLOUD model [10].

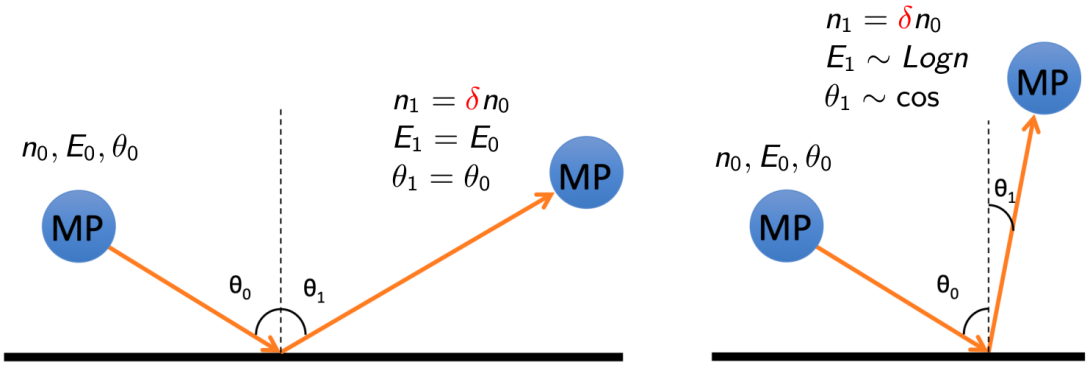

Figure 3: Sketch of elastic (left) and true secondary (right) collision events for the ECLOUD model [10].

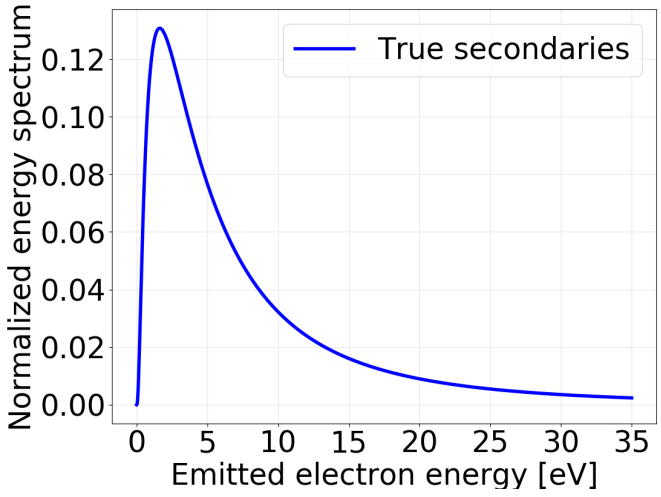

Figure 4: Energy distribution of the true secondary electrons for the ECLOUD model [8].

2.2 The Furman-Pivi model

The FP model is conceived for simulation codes in which all MPs have the same size. MPs are not rescaled and a probabilistic model is used to determine the number of MPs, n, generated by a collision event. The impacting MP is deleted if $n = 0$, preserved if $n \geq 1$, and new MPs are generated if $n \geq 2$. The model is summarized in the following. A more detailed description can be found in [7].

In the FP model, the SEY is divided into three components:

$$\delta(E_0, \theta_0) = \delta_e(E_0, \theta_0) + \delta_r(E_0, \theta_0) + \delta_{ts}(E_0, \theta_0), \tag{11}$$

where δ_e denotes the elastically backscattered electrons (equivalent to the elastic component of the ECLOUD model), δ_r denotes the "rediffused" electrons and δ_{ts} denotes the true secondaries. The rediffused electrons are emitted back by the surface after loosing some of their energy interacting with the material [7].

The SEY components are modelled using the following expressions:

$$\delta_e(E_0, \theta_0) = \delta_e(E_0, \theta_0 = 0)[1 + e_1(1 - \cos^{e_2}\theta_0)], \tag{12}$$
$$\delta_r(E_0, \theta_0) = \delta_r(E_0, \theta_0 = 0)[1 + r_1(1 - \cos^{r_2}\theta_0)], \tag{13}$$
$$\delta_{ts}(E_0, \theta_0) = \hat{\delta}(\theta_0) D(E_0/\hat{E}(\theta_0)), \tag{14}$$

with:

$$\delta_e(E_0, \theta = 0) = P_{1,e}(\infty) + [\hat{P}_{1,e} - P_{1,e}(\infty)]e^{-(|E_0 - \hat{E}_e|/W)^p/p}, \tag{15}$$
$$\delta_r(E_0, \theta = 0) = P_{1,r}(\infty)[1 - e^{-(E_0/E_r)^r}], \tag{16}$$
$$\hat{\delta}(\theta_0) = \hat{\delta}_{ts}[1 + t_1(1 - \cos^{t_2}\theta_0)], \tag{17}$$
$$\hat{E}(\theta_0) = \hat{E}_{ts}[1 + t_3(1 - \cos^{t_4}\theta_0)], \tag{18}$$

where $D(x)$ has been defined previously in Eq. 5 and e_1, e_2, r_1, r_2, $P_{1,e}(\infty)$, $P_{1,r}(\infty)$, $\hat{P}_{1,e}$, $\hat{E}_e$, E_r, W, p, r, t_1, t_2, t_3 and t_4, are model parameters.

The dependence on the incident electron energy for the three SEY components are shown in Fig. 5 (for a set of parameters defined in [7], based on measurements on copper samples).

As for the ECLOUD model, the SEY and its components can be defined in terms of electron currents as illustrated in Fig. 6, which shows that it is only for true secondary events that additional electrons are emitted. The three SEY components can be written as:

$$\delta_e = \frac{I_{elas}}{I_{imp}} \tag{19}$$

$$\delta_r = \frac{I_{re}}{I_{imp}} \tag{20}$$

$$\delta_{ts} = \frac{I_{true}}{I_{imp}} \tag{21}$$

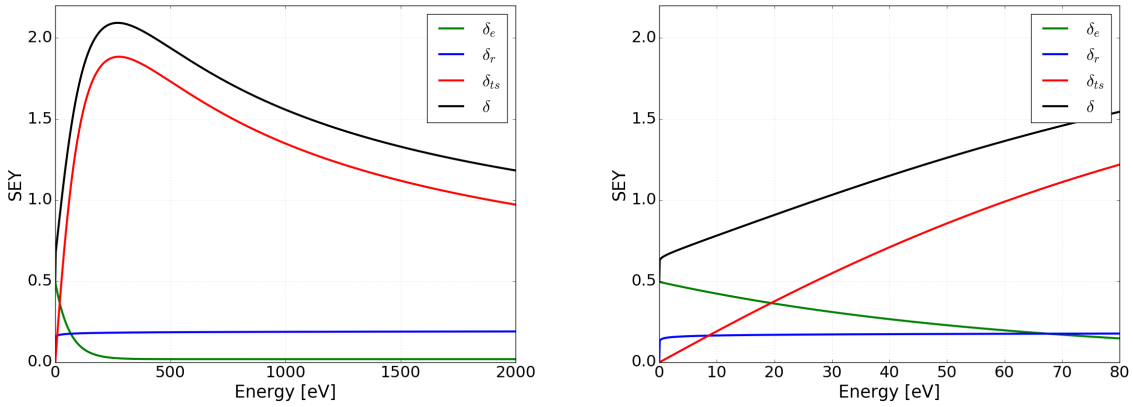

Figure 5: Left: the SEY components for the FP model. Right: zoom over the low-energy part of the curves.

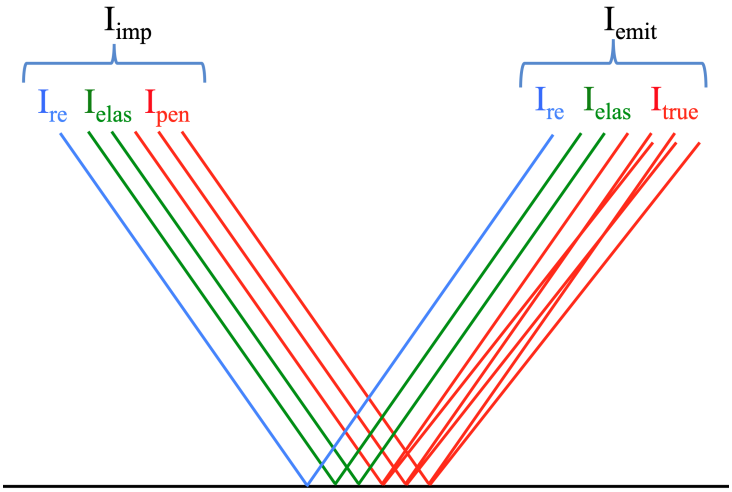

Figure 6: Decomposition of the current impacting on and emitted from a surface used in the FP model [7].

$$\delta = \delta_e + \delta_r + \delta_{ts} = \frac{I_{emit}}{I_{imp}} \ . \tag{22}$$

It is convenient for the implementation to introduce another quantity, which Furman and Pivi call the "SEY per penetrated current", δ'_{ts}, defined as

$$\delta'_{ts} = \frac{I_{true}}{I_{pen}} = \frac{\delta_{ts}}{1 - \delta_e - \delta_r} \ . \tag{23}$$

The electrons emitted after elastic interactions are emitted with the following energy

spectrum (which has its maximum at the energy of the impacting electron)[1]:

$$f_{1,e} = \frac{2e^{-(E-E_0)^2/2\sigma_e^2}}{\sqrt{2\pi}\sigma_e \mathrm{erf}(E_0/\sqrt{2}\sigma_e)} , \quad (0 \le E \le E_0). \tag{24}$$

Equation 24 is the one reported in [7] but there has been a modification in the POSINST code after the publishing of the paper. In order to make the distribution narrower for lower energies, σ_e is modified depending on E_0 according to

$$\sigma_e := \sigma_e - 1.88 + 2.5[1 + \tanh(0.01(E_0 - 150))] . \tag{25}$$

This modification can be optionally used also in PyECLOUD. It will be used in particular in Sec. 5 when comparing PyECLOUD against POSINST.

The rediffused electrons are emitted with the following energy spectrum:

$$f_{1,r} = \frac{q+1}{E_0^{q+1}} E^q , \quad (0 \le E \le E_0). \tag{26}$$

The energy spectrum for the true secondary electrons depends on the number of electrons generated in the event:

$$f_{n,ts} = A_n E^{p_n - 1} e^{-E/\epsilon_n}, \quad (0 \le E \le E_0), \tag{27}$$

where p_n and ϵ_n are model parameters that depend on the number of emitted electrons for a particular event. Electrons are never emitted with energy larger than the impacting one.

The number of emitted electrons for a certain event is in turn a random variable which is generated according to a Poisson distribution:

$$P'_{n,ts} = \frac{\delta'^n_{ts}}{n!} e^{-\delta'_{ts}}. \tag{28}$$

This distribution ensures that the expectation on the number of electrons emitted from a true secondary event is given by δ'_{ts}:

$$\langle n \rangle = \sum_{n=1}^{\infty} n P'_{n,ts} = \delta'_{ts}. \tag{29}$$

A Binomial distribution for n can also be optionally used, as described in [7].

The functions $f_{1,e}$ and $f_{1,r}$ are illustrated in Fig. 7. Also shown is the weighted average $f_{avg,ts} = \sum_{n=1}^{M} n P'_{n,ts} f_{n,ts}(E)$.

The FP model uses a cosine emission-angle distribution for all emitted electrons.

[1]Here all energy distributions are defined normalized to one, so that they coincide with the energy probability distribution function for each event type. Note that in [7] different normalization choices are used.

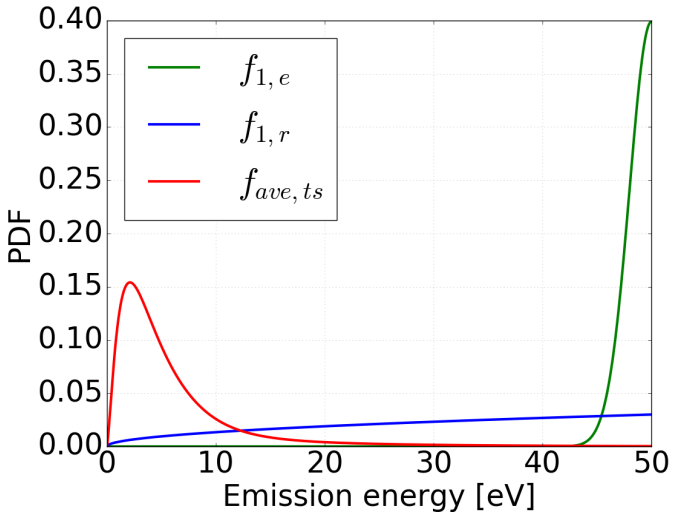

Figure 7: Emission energy distributions for the three components of the secondary electron current in the FP model. The case of impacting energy $E_0 = 50$ eV is considered. All curves are normalised to have area one.

2.2.1 Implementation in PyECLOUD

The following algorithm has been programmed in PyECLOUD to implement the FP model:

1. When a MP impacts the wall the SEY components are evaluated using Eqs. 12-14;
2. The event type is decided according to the following probabilities:

$$\begin{aligned} \mathrm{P}(\text{elastic}) &= \delta_e \ , \\ \mathrm{P}(\text{rediffused}) &= \delta_r \ , \\ \mathrm{P}(\text{true secondary}) &= 1 - \delta_e - \delta_r \ . \end{aligned} \tag{30}$$

3. The number of emitted MPs is decided:
   - for elastic or rediffused event types, always one MP is generated (in fact the impacting MP is re-used);
   - for true secondary events, the number of emitted MPs is generated according to the Poisson distribution function defined in Eq. 28 (or optionally by using a Binomial distribution);
4. The energy of the emitted MP is generated using the appropriate distribution among those defined by Eqs. 24-27 (the detailed algorithm for this step is described in Appendix A);
5. All MPs are emitted with an angle with respect to the normal to the surface that is generated according to a cosine distribution.

This numerical implementation of the described stochastic model differs in some aspects from the one implemented in POSINST and described in [7]. It has been chosen in order to allow for an easier integration within PyECLOUD and to be compatible with the a-posteriori reconstruction of the surface properties (see Sec. 3).

3 Consistency checks on the Furman-Pivi implementation

To ensure that the SEY components have been implemented correctly, tests were performed extracting the corresponding curves from a running simulation. In these tests electrons were shot at a surface and the emitted electrons were counted and grouped according to their event type. The SEY components were then calculated as the ratio between the emitted current of the corresponding event type and the impacting current. The extracted SEY components match very well the corresponding analytical expressions, as shown in Fig. 8. The same method was used to crosscheck the energy distributions of the emitted electrons as shown in Fig. 9.

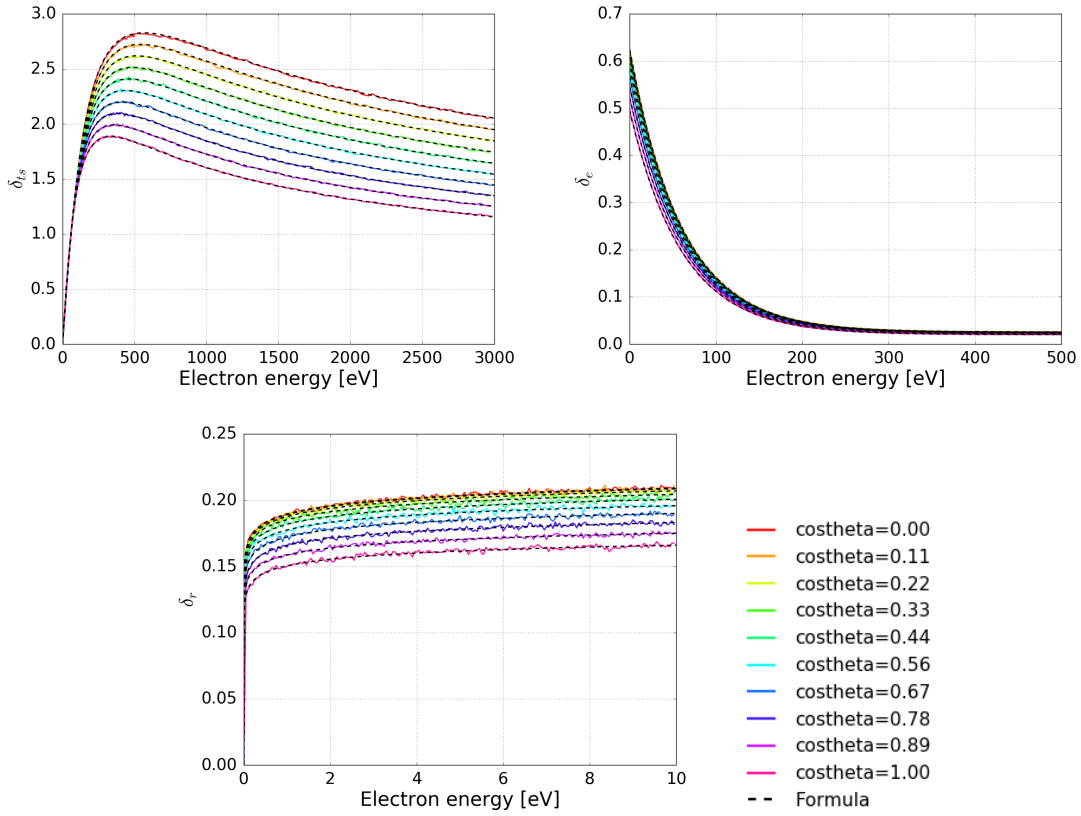

Figure 8: SEY curves extracted using Monte Carlo experiments for different angles of incidence and using impinging electron energy $E_0 = 50$ eV.

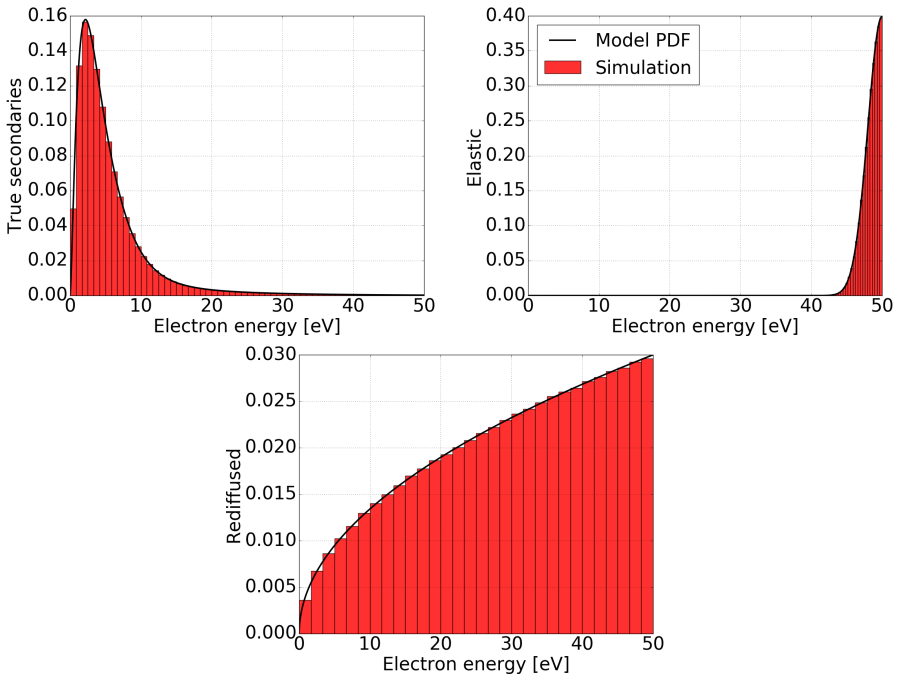

Figure 9: Normalized emission energy spectra extracted using Monte Carlo experiments for normal incidence and using impinging electron energy $E_0 = 50$ eV.

4 Comparison between ECLOUD and Furman-Pivi model

In order to compare the simulation results of the two SEY modules in a meaningful way they should both be representative of the same physical surface.

For this purpose the parameters in the FP model were chosen in order to obtain the same SEY component curves and the same emitted energy distribution as for the ECLOUD model of the LHC beam screen surface (defined by the parameters in Tab. 1). The corresponding FP parameters are reported in Tab. 2.

The rediffused electrons were set to zero, as they are not present in the ECLOUD model. In the case of δ_{ts} the fit could be done exactly, since the same formula is used in both models. In PyECLOUD, the elastically scattered electrons have equal energy to the impacting electrons, E_0. Therefore the parameters of $f_{1,e}$ in the FP model were simply chosen in order to make the distribution very narrow. This was done by choosing $\sigma_e << 1$ eV.

The method of non linear least squares [11] was used to fit δ_e and $f_{n,ts}$. In the FP model there are several energy distributions for the true secondaries, depending on the number of emitted electrons. We chose to fit their average to the ECLOUD distribution. The average PDF can be written as

$$f_{avg,ts} = \sum_{n=1}^{M} nP_{n,ts} f_{n,ts}(E) = \sum_{n=1}^{M} nA_n E^{p_n - 1} e^{-E/\epsilon_n}. \tag{31}$$

Table 1: ECLOUD parameters for LHC beam screen surface.

Parameter	PyECLOUD name	Value
	Elastic electrons	
R_0	`R0`	0.7
E_e (eV)	`E0`	150
	True secondary electrons	
s	`s_param`	1.35
$\hat{\delta}_{ts}$	`deltaTSHat`	1.6
$\hat{E}_{ts}$ (eV)	`Emax`	332
	`E_th`	35
σ_{true}	`sigmafit`	1.0828
μ_{true}	`mufit`	1.6636
	`secondary_angle_distribution`	`"cosine_3D"`
	Other	
	`switch_no_increase_energy`	0
	`switch_model`	`"ECLOUD"`

To reduce the number of free parameters and make the fit simpler, the maximum of each of the terms in the sum was forced to match exactly the maximum of the ECLOUD distribution. The maximum of each $f_{n,ts}$ can be found by taking the derivative and finding its zero:

$$\begin{aligned} f_{n,ts}(E) &= A_n E^{p_n-1} e^{-E/\epsilon_n} \\ \frac{d}{dE}(f_{n,ts}(E)) &= A_n \left((p_n-1)E^{p_n-2} - \frac{1}{\epsilon_n} E^{p_n-1} \right) e^{-E/\epsilon_n} = 0 \Rightarrow \\ & (p_n-1)E^{p_n-2} - \frac{1}{\epsilon_n} E^{p_n-1} = 0 \Rightarrow \\ & E_{\max} = \epsilon_n (p_n - 1). \end{aligned} \tag{32}$$

This provides a relationship between the p_n and the ϵ_n parameters, effectively halving the number of parameters to fit:

$$\epsilon_n = \frac{E_{\max}}{p_n - 1}. \tag{33}$$

The fitted SEY curves and energy distributions are shown in Fig. 10.

4.1 Buildup simulation results

Simulation studies were performed for the test cases defined in Tab. 3, in order to compare the two models. The simulation results are compared in Figs. 11 and 12. Very good agreement is found between the two models both for the calculated heat load and for the electron current impinging on the walls. This shows that the differences in the way the two models handle the generation of the secondaries have no significant impact on the simulation results.

Table 2: FP model parameters for the LHC beam screen surface defined in Tab. 1.

Parameter	PyECLOUD name	Value
	Backscattered electrons	
$P_{1,e}(\infty)$	`p1EInf`	0.00216
$\hat{P}_{1,e}$	`p1Ehat`	0.710
$\hat{E}_e$ (eV)	`eEHat`	0
W (eV)	`w`	46.0
p	`p`	0.469
σ_e	`sigmaE`	10^{-4}
e_1	`e1`	0
e_2	`e2`	2
	`use_modified_sigmaE`	`False`
	Rediffused electrons	
$P_{1,r}(\infty)$	`p1RInf`	-
E_r eV	`eR`	-
r	`r`	-
q	`q`	-
r_1	`r1`	-
r_2	`r2`	-
	True secondary electrons	
$\hat{\delta}_{ts}$	`deltaTSHat`	1.0 - 1.8
$\hat{E}_{ts}$	`eHat0`	332
s	`s`	1.35
t_1	`t1`	0.676
t_2	`t2`	0.767
t_3	`t3`	0.7
t_4	`t4`	1
m	`M_cut`	10
	`choice`	`"poisson"`
	`conserve_energy`	`True`

Parameters	**Values**
p_n	1.22, 1.66, 1.22, 1.10, 4.28, 1.02, 1.02, 1.02, 29.93, 1.02
ϵ_n	7.44, 2.47, 7.45, 16.36, 0.50, 79.62, 66.04, 70.81, 0.06, 79.89

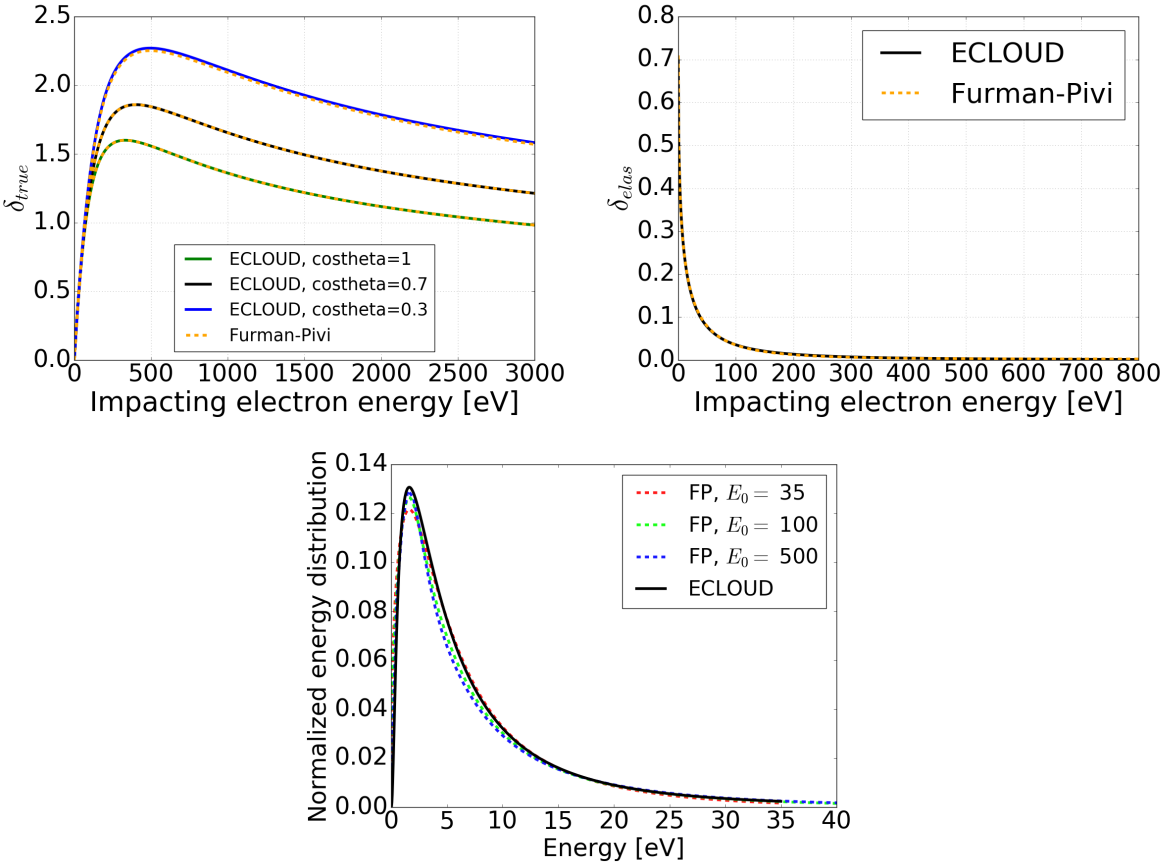

Figure 10: ECLOUD and FP SEY curves and true secondary energy distribution for the LHC beam screen surface defined in Tab. 1.

Table 3: Simulation parameters used in the PyECLOUD simulation study.

Parameter	Value
Beam energy	450 GeV
Bunch population	$0.5 \times 10^{11} - 2.2 \times 10^{11}$ p$^+$/bunch (scanned)
Bunch spacing	25 ns
Number of bunches	2760
Magnetic field	0.53 T (dipole)
Chamber	23 mm $\times$ 18 mm (rectangular)
$\hat{\delta}_{ts}$	1.0 - 1.8 (scanned)

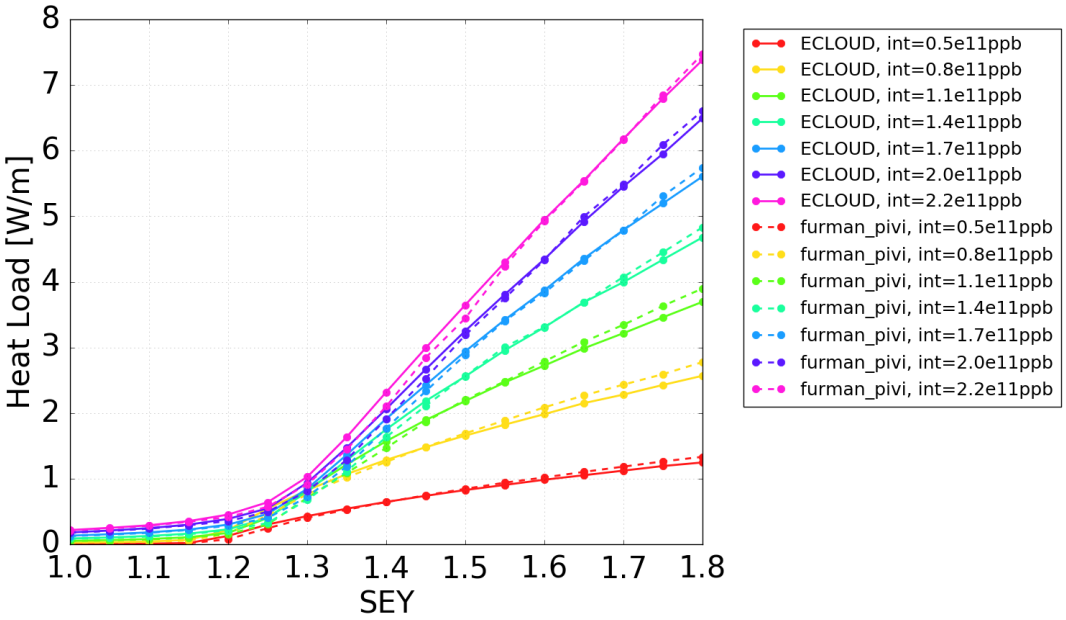

Figure 11: Comparison between ECLOUD and FP model: heat load as a function of the SEY.

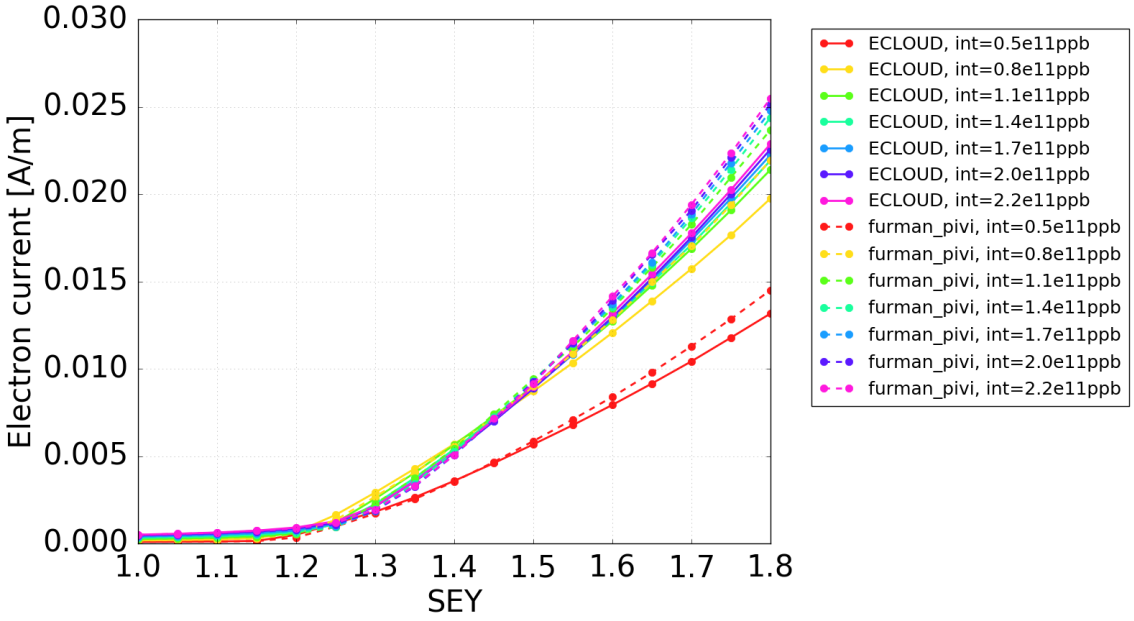

Figure 12: Comparison between ECLOUD and FP model: current of impacting electrons as a function of the SEY.

5 Comparisons against POSINST

In order to have an independent validation for the FP implementation and for PyECLOUD in general, comparisons were carried out against the POSINST code developed at LBNL, which also employs the FP model [5].

5.1 Simulations without rediffused electrons

A first set of simulations was performed using the simulation parameters in Tab. 2 and 3. The bunch population has been kept fixed at 1.1×10^{11} p^+/bunch.

The heat loads and electron currents obtained by these simulations are shown in Figs. 13 and 14. The agreement between the two codes is in general very good.

All the main observables from the simulations are very similar for the two codes, in particular the e-cloud risetime and saturation level (see Fig. 15), the electron distribution within the chamber (see Fig. 16), the energy spectrum of the impacting electrons (see Fig. 17) and the instantaneous electron flux onto the chamber's wall (see Fig. 18). The POSINST simulations tend to be more affected by particle noise.

The simulation run times have also been compared and are presented in Tab. 4.

Table 4: Average simulation run times.

	Average run time
PyECLOUD-ECLOUD	47m
PyECLOUD-FP	41m
POSINST	1h34m

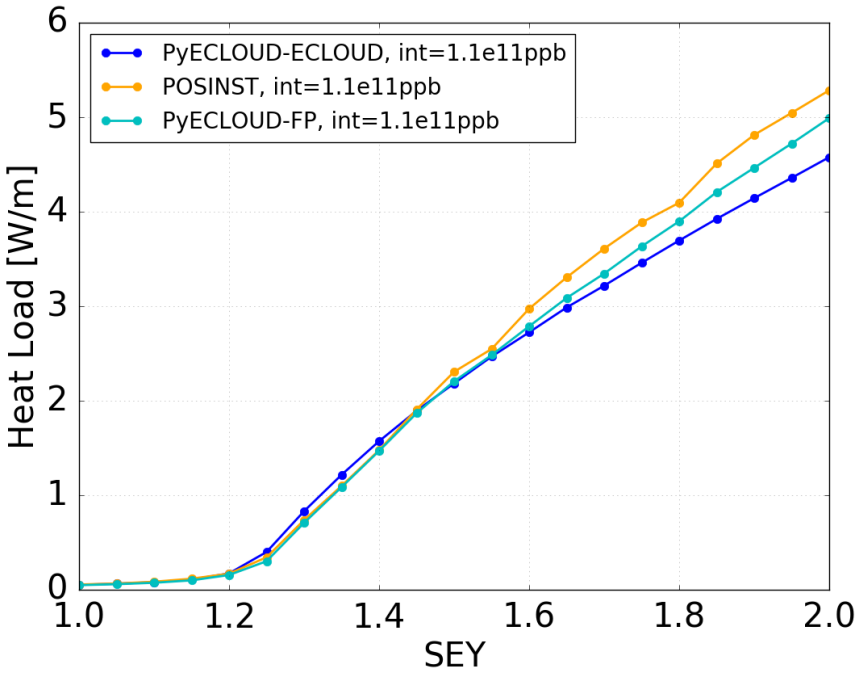

Figure 13: Comparison between PyECLOUD and POSINST: heat load as a function of the SEY (no rediffused electrons are present in these simulations.

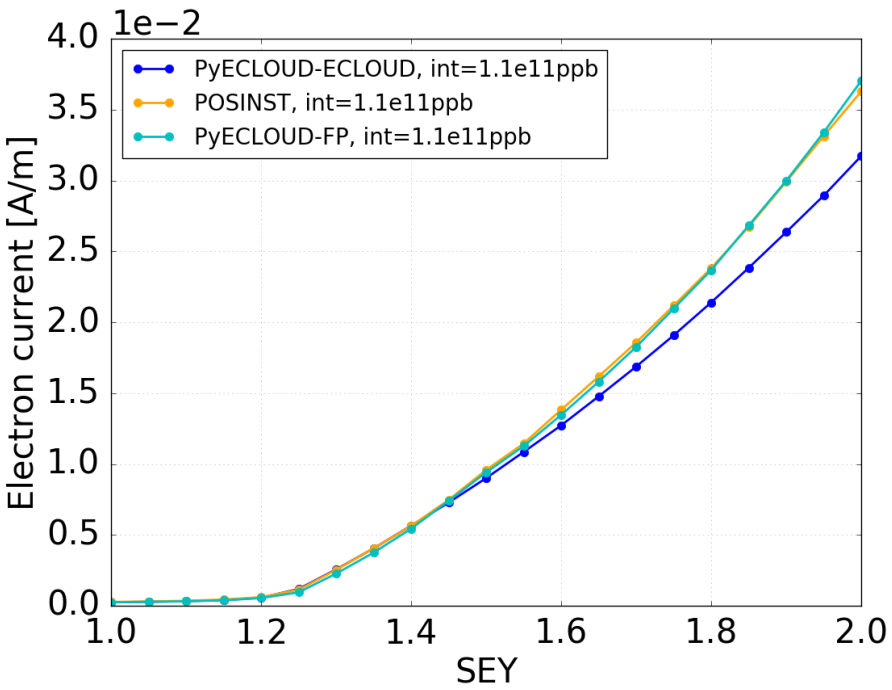

Figure 14: Comparison between PyECLOUD and POSINST: current of impacting electrons as a function of the SEY (no rediffused electrons are present in these simulations).

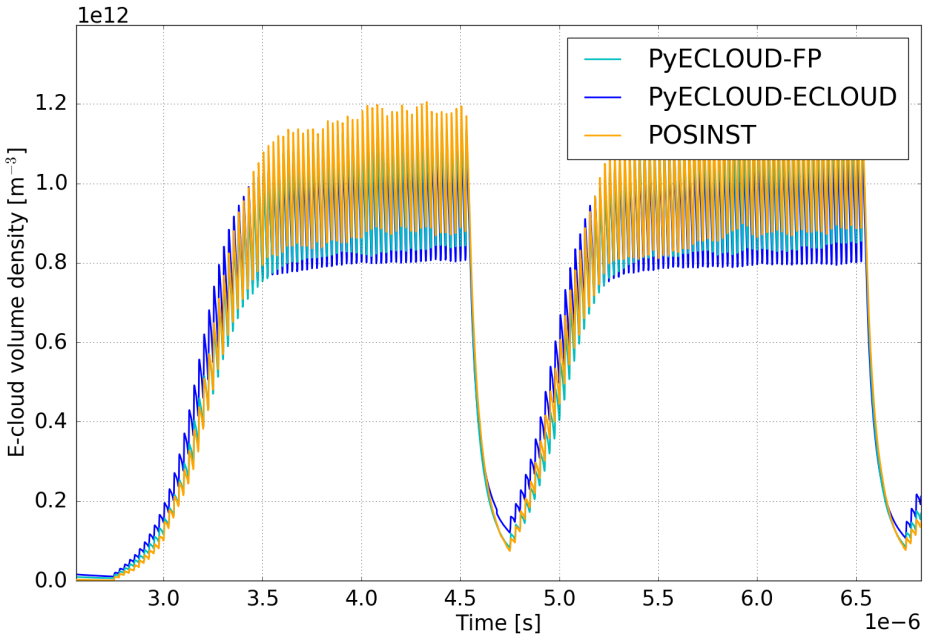

Figure 15: Evolution of the average e-cloud density in the chamber (1.1×10^{11} p$^+$/bunch, $\delta_{ts} = \hat{\delta}_{ts} = 1.5$, no rediffused electrons are present in this simulation).

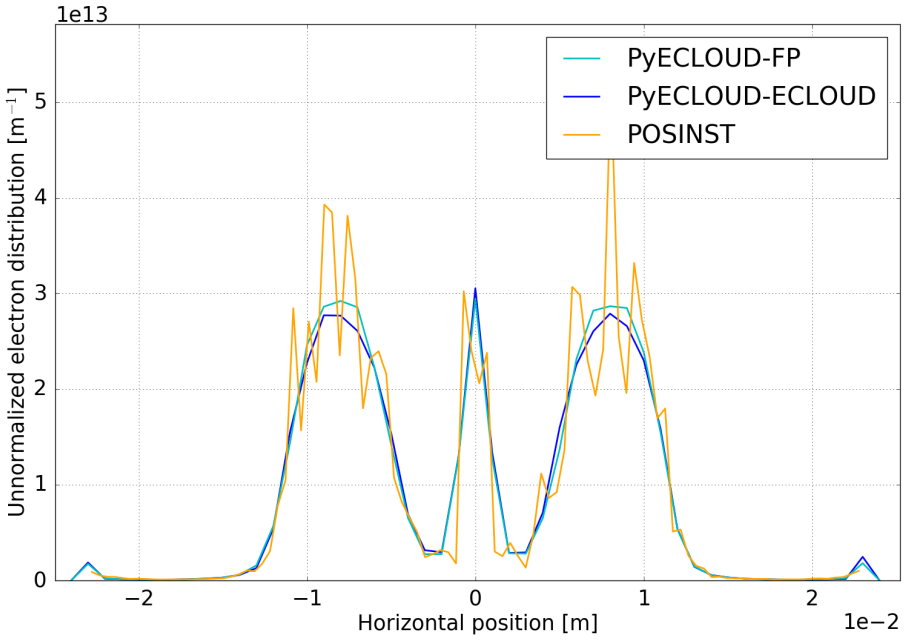

Figure 16: Horizontal distribution of the electrons inside the chamber (1.1×10^{11} p$^+$/bunch, $\delta_{ts} = \hat{\delta}_{ts} = 1.5$, no rediffused electrons are present in this simulation).

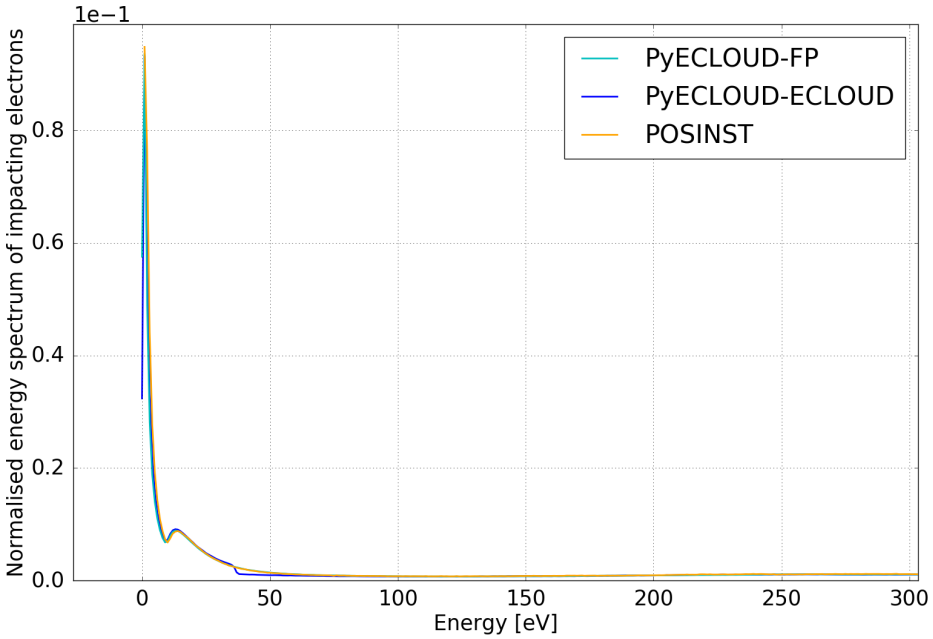

Figure 17: Energy spectrum of electrons impacting on the chamber's wall (1.1×10^{11} p$^+$/bunch, $\delta_{ts} = \hat{\delta}_{ts} = 1.5$, no rediffused electrons are present in this simulation).

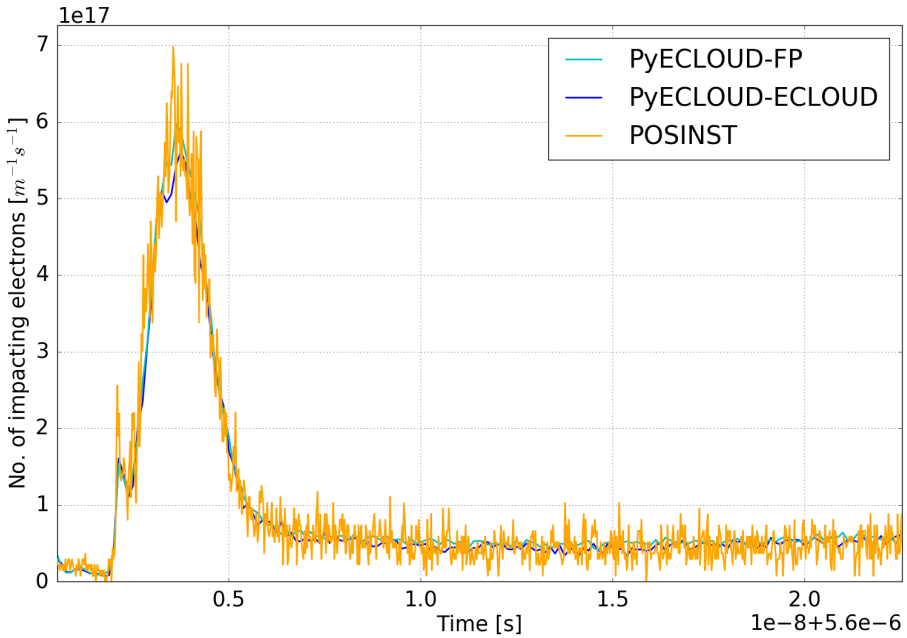

Figure 18: Electron bombardment rate (1.1×10^{11} p$^+$/bunch, $\delta_{ts} = \hat{\delta}_{ts} = 1.5$, no rediffused electrons are present in this simulation).

5.2 Simulations including rediffused electrons

In the simulations presented in Sec. 5.1, no rediffused electrons were present. In order to test also this part of the implementation we performed another set of simulations using the surface parameters in Tab. 5 (taken from [7]).

The heat loads and electron currents from these simulations are shown in Figs. 19 and 20 and other observables are shown in Figs. 21-24. Very good agreement is found,

Table 5: Model parameters from [7] used to compare PyECLOUD and POSINST including the effect of rediffused electrons.

Parameter	PyECLOUD name	Value
	Backscattered electrons	
$P_{1,e}(\infty)$	`p1EInf`	0.02
$\hat{P}_{1,e}$	`p1Ehat`	0.496
$\hat{E}_e$ (eV)	`eEHat`	0
W (eV)	`w`	60.86
p	`p`	1
σ_e	`sigmaE`	2
e_1	`e1`	0.26
e_2	`e2`	2
	`use_modified_sigmaE`	`False`
	Rediffused electrons	
$P_{1,r}(\infty)$	`p1RInf`	0.2
E_r eV	`eR`	0.041
r	`r`	0.104
q	`q`	0.5
r_1	`r1`	0.26
r_2	`r2`	2
	True secondary electrons	
$\hat{\delta}_{ts}$	`deltaTSHat`	1.0 - 2.0
$\hat{E}_{ts}$	`eHat0`	276.8
s	`s`	1.54
t_1	`t1`	0.66
t_2	`t2`	0.8
t_3	`t3`	0.7
t_4	`t4`	1
m	`M_cut`	10
	`choice`	`"poisson"`
	`conserve_energy`	`True`

Parameters	Values
p_n	2.5, 3.3, 2.5, 2.5, 2.8, 1.3, 1.5, 1.5, 1.5, 1.5
ϵ_n	1.5, 1.75, 1.0, 3.75, 8.5, 11.5, 2.5, 3., 2.5, 3.0

confirming that the implementation of the rediffused electrons is correct.

The simulation run times have been compared also for this set of simulations and are presented in Tab. 6.

Table 6: Average simulation run times.

	Average run time
PyECLOUD-FP	1h5m
POSINST	3h23m

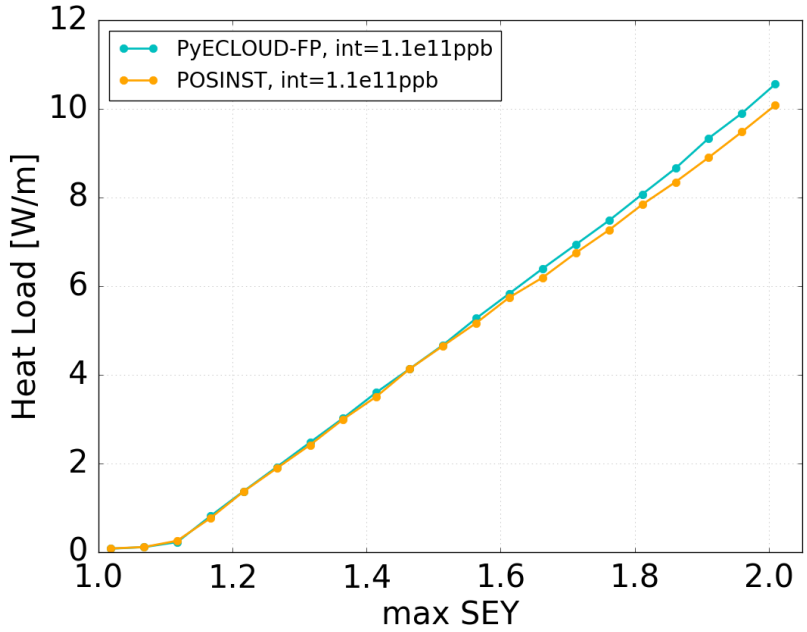

Figure 19: Comparison between PyECLOUD and POSINST: heat load as a function of the SEY (rediffused electrons are present in these simulations). Note that the SEY indicated in the figure is max $\left[\hat{\delta}_{ts}(E) + \delta_e(E) + \delta_r(E)\right]$.

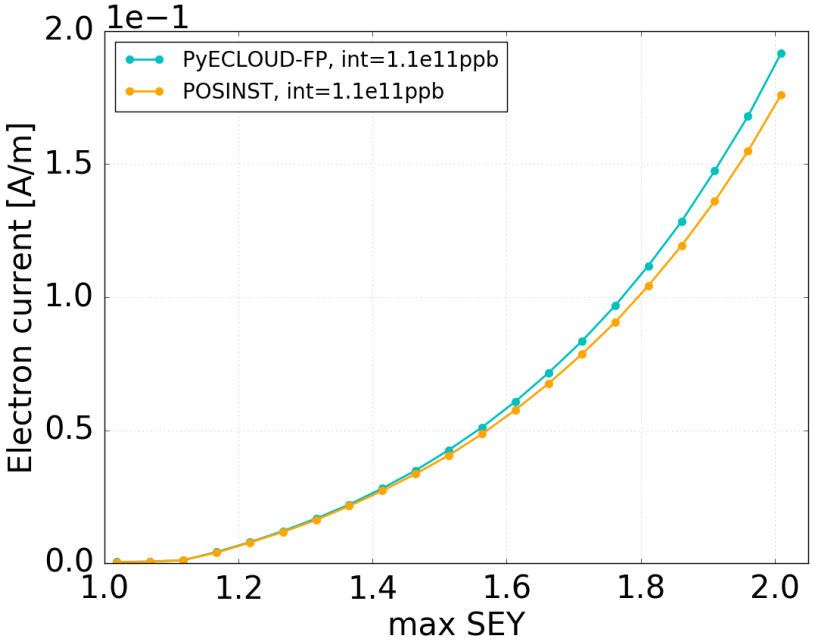

Figure 20: Comparison between PyECLOUD and POSINST: current of impacting electrons as a function of the SEY (rediffused electrons are present in these simulations). Note that the SEY indicated in the figure is max $\left[\hat{\delta}_{ts}(E) + \delta_e(E) + \delta_r(E)\right]$.

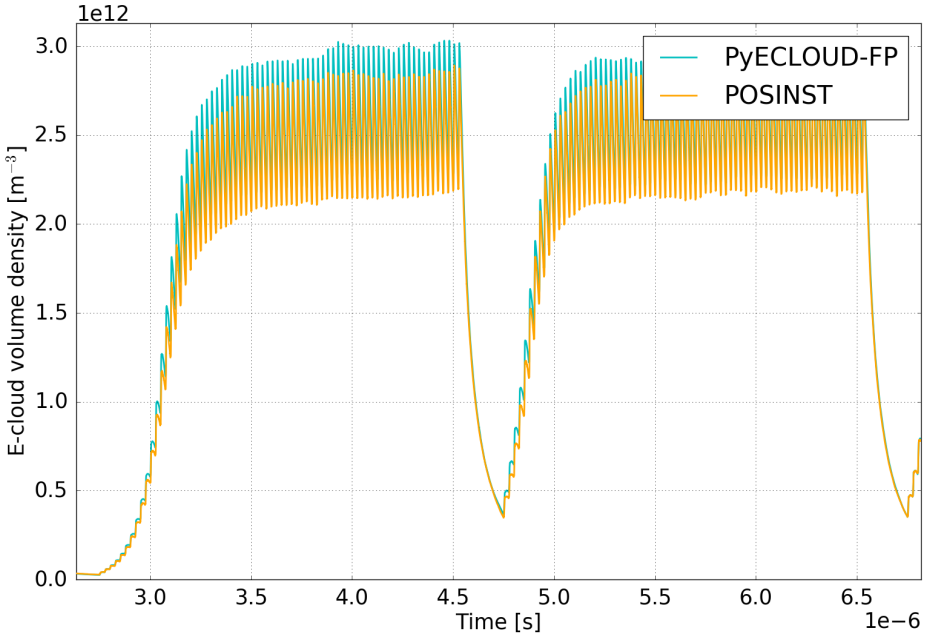

Figure 21: Evolution of the average e-cloud density in the chamber (1.1×10^{11} p$^+$/bunch, max $\left[\hat{\delta}_{ts}(E) + \delta_e(E) + \delta_r(E)\right] = 1.5$, rediffused electrons are present in this simulation).

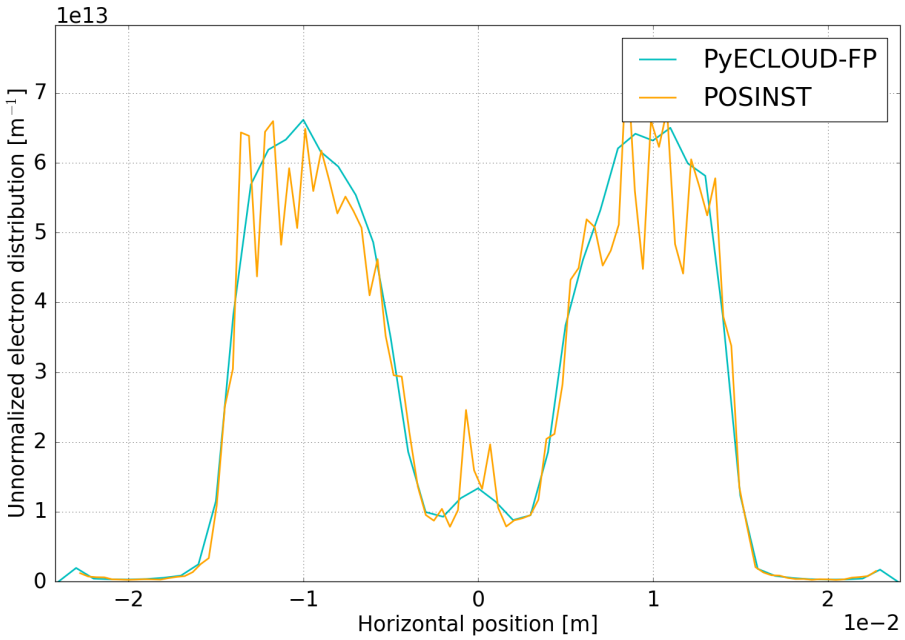

Figure 22: Horizontal distribution of the electrons inside the chamber (1.1×10^{11} p$^+$/bunch, max $\left[\hat{\delta}_{ts}(E) + \delta_e(E) + \delta_r(E)\right] = 1.5$, rediffused electrons are present in this simulation).

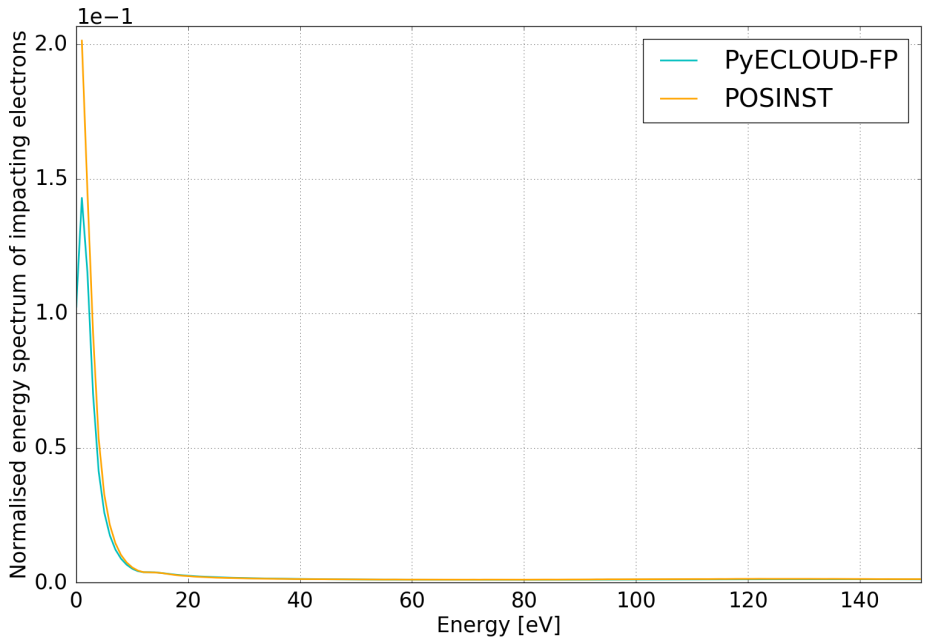

Figure 23: Energy spectrum of electrons impacting on the chamber's wall (1.1×10^{11} p$^+$/bunch, max $\left[\hat{\delta}_{ts}(E) + \delta_e(E) + \delta_r(E)\right] = 1.5$, rediffused electrons are present in this simulation).

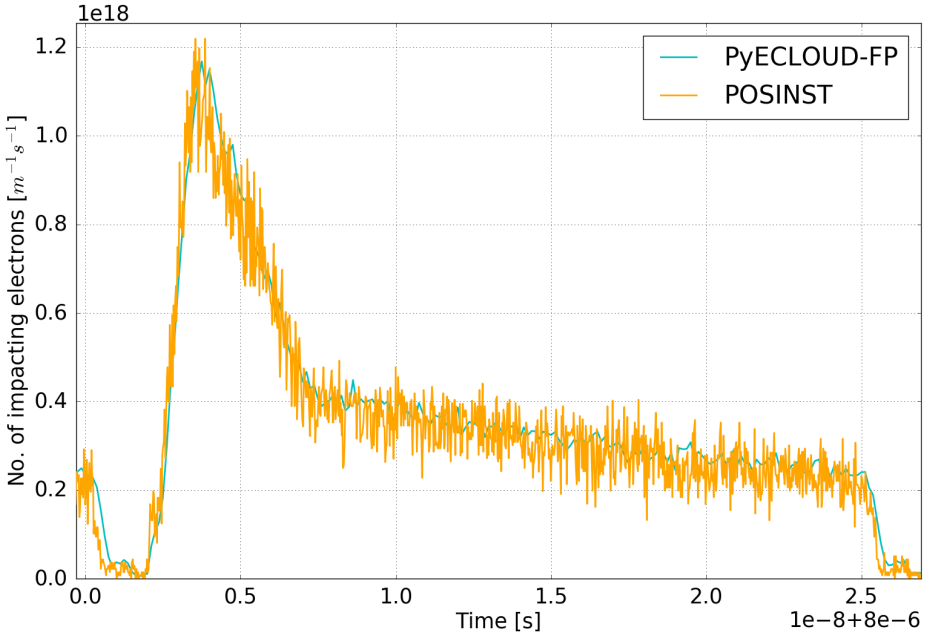

Figure 24: Electron bombardment rate (1.1×10^{11} p$^+$/bunch, max $\left[\hat{\delta}_{ts}(E) + \delta_e(E) + \delta_r(E)\right] = 1.5$, rediffused electrons are present in this simulation).

6 Conclusions

The FP model of secondary electron emission has been implemented in the PyECLOUD code and has been compared against the ECLOUD model implemented at CERN.

Simulation results were found to agree well when the two models are representative of the same physical surface.

PyECLOUD simulations could also be compared against the POSINST code, which also employs the FP model. A very good agreement has been found in all tested scenarios. For the chosen simulation parameters, POSINST simulations take significantly longer than PyECLOUD simulations and tend to be more affected by numerical noise.

Acknowledgements

The authors would like to thank G. Arduini, P. Dijkstal, M. Furman, L. Mether, E. Métral, G. Penn, G. Rumolo, G. Skripka, and J-L. Vay for their important input and support to the presented studies.

Appendices

A Energy generation for emitted electrons

A.1 Elastically backscattered electrons

In the FP model, the energy distribution of the backscattered electrons is:

$$f_{1,e} = \frac{2e^{-(E-E_0)^2/2\sigma_e^2}}{\sqrt{2\pi}\sigma_e \mathrm{erf}(E_0/\sqrt{2}\sigma_e)} \ , \quad (0 \leq E \leq E_0) \tag{34}$$

where the normalization is such that

$$\int_0^{E_0} f_{1,e} dE = 1 \ . \tag{35}$$

To use the method of inverse transform sampling [12] in order to generate energies according to this distribution, we need to find the inverse of the corresponding cumulative distribution function (CDF):

$$F_{CDF,e}(E) = \int_0^{E} f_{1,e}(E')dE'. \tag{36}$$

To do that, we group all constants with respect to E into a single constant, A_e so that

$$f_{1,e} = A_e e^{-(E-E_0)^2/2\sigma_e^2} \ , \quad A_e = \frac{2}{\sqrt{2\pi}\sigma_e \mathrm{erf}(\frac{E_0}{\sqrt{2}\sigma_e})} \Rightarrow \tag{37}$$

$$F_{CDF,e} = A_e \int_0^{E} e^{-(E'-E_0)^2/2\sigma_e^2} dE' \ . \tag{38}$$

Remembering that $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}$ and making the variable change $\frac{E-E_0}{\sqrt{2}\sigma_e} = t$ allows us to write

$$\begin{aligned} F_{CDF,e} &= A_e \int_{\frac{-E_0}{\sqrt{2}\sigma_e}}^{\frac{E-E_0}{\sqrt{2}\sigma_e}} \sqrt{2}\sigma_e e^{-t^2} dt \\ &= \sqrt{2}\sigma_e A_e \left(\int_{\frac{-E_0}{\sqrt{2}\sigma_e}}^{0} e^{-t^2} dt + \int_0^{\frac{E-E_0}{\sqrt{2}\sigma_e}} e^{-t^2} dt \right) \\ &= \frac{\sqrt{2}\sigma_e A_e}{(2/\sqrt{\pi})} \left(\mathrm{erf}(\frac{E_0}{\sqrt{2}\sigma_e}) + \mathrm{erf}(\frac{E-E_0}{\sqrt{2}\sigma_e}) \right), \end{aligned} \tag{39}$$

where we have made use of the fact that the error function is an odd function. Now, plugging in the expression for A_e from Eq. 37 and simplifying we get

$$F_{CDF,e} = \frac{2}{\sqrt{2\pi}\sigma_e \mathrm{erf}(\frac{E_0}{\sqrt{2}\sigma_e})} \frac{\sqrt{2}\sigma_e}{(2/\sqrt{\pi})} \left(\mathrm{erf}(\frac{E_0}{\sqrt{2}\sigma_e}) + \mathrm{erf}(\frac{x - E_0}{\sqrt{2}\sigma_e}) \right) \Rightarrow$$

$$F_{CDF,e}(E) = 1 + \frac{\mathrm{erf}(\frac{E-E_0}{\sqrt{2}\sigma_e})}{\mathrm{erf}(\frac{E_0}{\sqrt{2}\sigma_e})} \ . \tag{40}$$

Finally, inverting Eq. 40 gives

$$F^{-1}_{CDF,e}(x) = E_0 + \sqrt{2}\sigma_e \mathrm{erf}^{-1} \left((E-1)\mathrm{erf}(\frac{E_0}{\sqrt{2}\sigma_e}) \right) \ . \tag{41}$$

Emission energies for backscattered electrons are then obtained by generating uniformly distributed random numbers and plugging them into Eq. 41.

A.2 Rediffused electrons

In the FP model the energy distribution for the rediffused electrons is

$$f_{1,r} = \frac{q+1}{E_0^{q+1}} E^q \ , \quad (0 \leq E \leq E_0), \tag{42}$$

which is normalized so that

$$\int_0^{E_0} f_{1,r} dE = 1 \ . \tag{43}$$

Again, we need to find the corresponding CDF

$$F_{CDF,r}(E) = \int_0^E f_{1,r}(E')dE' = \int_0^E \frac{q+1}{E_0^{q+1}} E'^q dE' = \frac{x^{q+1}}{E_0^{q+1}}. \tag{44}$$

The inverse of Eq. 44 is given by

$$F^{-1}_{CDF,r}(x) = E_0 x^{1/(q+1)}. \tag{45}$$

Emission energies for rediffused electrons are then obtained by generating uniformly distributed random numbers and plugging them into Eq. 45.

A.3 True secondary electrons

The energy of the true secondary electrons in the FP model follow distributions in the form:

$$f_{n,ts} = A_n E^{p_n - 1} e^{-E/\epsilon_n} \ . \tag{46}$$

where A_n is a normalization factor chosen so that:

$$\int_0^{E_0} f_{n,ts} dE = 1 \ . \tag{47}$$

We want to find an expression for the corresponding CDF:

$$F_{CDF,ts}(x) = \int_0^x f_{n,ts} dE = A_n \int_0^x E^{p_n - 1} e^{-E/\epsilon_n} dE \ . \tag{48}$$

Introducing the following variable change makes further manipulations simpler.

$$t = E/\epsilon_n \iff \epsilon_n t = E \quad \implies \quad \epsilon_n dt = dE \tag{49}$$

Using Eq. 49, the CDF of Eq. 48 can now be written as

$$F_{CDF,ts}(x) = A_n \int_0^{x/\epsilon_n} (\epsilon_n t)^{p_n - 1} e^{-1} \epsilon_n dt = A_n \epsilon_n^{p_n} \int_0^{x/\epsilon_n} t^{p_n - 1} e^{-t} dt \tag{50}$$

This can be written as:

$$F_{CDF,ts}(x) = A_n \epsilon_n^{p_n} \Gamma(p_n) P(p_n, x/\epsilon_n), \tag{51}$$

where Γ is the Gamma function and $P(z, x)$ is the normalised incomplete lower beta function defined as [7]:

$$P(z, x) = \frac{1}{\Gamma(z)} \int_0^x t^{z-1} e^{-t} dt \ . \tag{52}$$

The inverse of the CDF can be written as:

$$F_{CDF,ts}(x)^{-1} = \epsilon_n P^{-1} \left(p_n, x/A_n \epsilon_n^{p_n} \Gamma(p_n)\right), \tag{53}$$

where $P^{-1}(z, x)$ is the inverse (in x) of the incomplete lower gamma function defined in Eq. 52.

Emission energies for true secondary electrons are then obtained by generating uniformly distributed random numbers and plugging them into Eq. 53.

When emitting multiple true secondary MPs for a single impact, the sum of the energies of all emitted MPs should not exceed the one of the impacting MP. An option has been implemented in PyECLOUD that checks this condition and regenerates the MPs in case the condition is not fulfilled.

References

[1] PyECLOUD. https://github.com/PyCOMPLETE/PyECLOUD.

[2] G. Iadarola, E. Belli, K. Li, L. Mether, A. Romano, and G. Rumolo. Evolution of Python Tools for the Simulation of Electron Cloud Effects. In *Proceedings, 8th International Particle Accelerator Conference (IPAC 2017): Copenhagen, Denmark, May 14-19, 2017*, page THPAB043, 2017.

[3] G. Rumolo and F. Zimmermann. Practical User Guide for ECloud. Technical Report SL-Note-2002-016-AP, CERN, Geneva, May 2002.

[4] R. Cimino, I. Collins, M. A. Furman, M. Pivi, F. Ruggiero, G. Rumolo, and F. Zimmermann. Can low-energy electrons affect high-energy physics accelerators? *Physical Review Letters*, 93, 06 2004.

[5] POSINST. https://blast.lbl.gov/blast-codes-posinst/.

[6] openECLOUD. https://github.com/openecloud/openecloud.

[7] M. A. Furman and M. T. F. Pivi. Probabilistic model for the simulation of secondary electron emission. *Phys. Rev. ST Accel. Beams*, 5:124404, Dec 2002.

[8] G. Iadarola. *Electron Cloud Studies for CERN Particle Accelerators and Simulation Code Development*. PhD thesis, Università degli Studi di Napoli Federico II, Napoli Italy, and European Organization for Nuclear Research (CERN), Geneva, Switzerland, March 2014. CERN-THESIS-2014-047.

[9] Proceedings ECLOUD'02. *CERN Yellow Report No. CERN-2002-001*, 2002.

[10] E. G. T. Wulff. Low energy electron modelling in PyECLOUD. Presentation at the e-cloud meeting 22 Feb. 2019, https://indico.cern.ch/event/790352/.

[11] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[12] L. Devroye. Random variate generation in one line of code. In *Proceedings Winter Simulation Conference*, pages 265–272, Dec 1996.