PyECLOUD Reference Manual

Giovanni Iadarola,
Eleonora Belli, Philipp Dijkstal,
Lotta Mether, Annalisa Romano,
Giovanni Rumolo, Eric Wulff
CERN - Geneva, Switzerland

PyECLOUD reference manual

This document describes the main input and output parameters for the PyECLOUD code for the simulation of the electron cloud buildup in particle accelerators.

1 Input files

1.1 Simulation parameters

This input file has to be named **simulation_parameters.input**.

Other input filenames. The following variables specify the names of the other three input files which define the physical model of the simulation.	
machine_param_file	Name of the machine parameter file.
secondary_emission_parameters_file	Name of the secondary emission configuration file.
beam_parameters_file	Name of the beam parameter file (in case of a multiple beam simulation this is the master beam).

Secondary beams The code can simulate the EC buildup in the presence of more than one circulating beam. For this purpose a list of secondary beam files (one for each beam) has to be provided. In the presence of secondary beams, the master beam determines the length of the simulation, the energy used for the calculation of the bending field, and the bunch spacing used for regeneration and saving purposes.	
secondary_beam_file_list	(optional – default = []) List of names of secondary beam parameter files.

Log and progress files	
logfile_path	Path of a text file that reports some synthetic information on the ongoing simulation (Passage number, number of MPs and number of electrons in the chamber).
progress_path	Path of a text file that reports the simulation progress in percent.

Time sampling The simulated time interval is defined by the length of the beam profile (number of bunch passages) specified in the beam description.	
Dt	[s] Simulation time step. This input is not considered if beam profile is imported from file.

t_end	[s] Extra time interval after the specified beam profile. Only coasting beam present in this interval. The value of this variable can also be negative in order to stop the simulation before the end of the specified profile.

Negligible beam linear density	
lam_th	[beam part/m] When the linear density of the beam is below this value, primary electrons are not generated and beam forces on the electrons are neglected.

MP management settings	
N_mp_max	Size of the arrays allocated to store the MP sizes, positions and velocities. MP regeneration settings must be set such that this number of MPs is never exceeded.
N_mp_regen	MP regeneration threshold. When the total number of MPs exceeds N_mp_regen a regeneration of the set of MPs is applied and a new Nref is chosen in order to get a target number of MPs. The check is performed at each t=n*b_spac with n an integer.
N_mp_regen_low	Lower MP regeneration threshold. When the total number of MPs becomes smaller than N_mp_regen_low a regeneration of the set of MPs is applied and a new Nref is chosen in order to get a target number of MPs. The check is performed at each t=n*b_spac with n an integer.
t_ON_regen_low	For t< t_ON_regen_low the lower MP regeneration check is not performed.
N_mp_after_regen	Target number of MPs obtained after regeneration.
fact_split	MP split factor; when secondary emission occurs, if the size of the emitted charge is smaller than fact_split*Nref, the impacting MP is simply rescaled according to the SEY. Otherwise new MPs are generated in order to keep the MP size as close as possible to Nref.
fact_clean	MP clean factor. MPs smaller than fact_split*Nref are deleted from the MP set. This test is performed at each t=n*b_spac with n an integer.

nel_mp_ref_0	[e-/m] Initial MP size.
Nx_regen, Ny_regen, Nvx_regen, Nvy_regen, Nvz_regen	Number of mesh points in each dimension for the generation of the uniform grid for the 5-D phase space used for the MP regeneration.
regen_hist_cut	Defines a threshold value for the phase space density, below which no electrons are generated.
N_mp_soft_regen	(optional: by default soft regeneration is not applied, the presence of these inputs enables it) Soft regeneration threshold. The check is performed at each t=n*b_spac with n an integer. N_mp_soft_regen should be smaller than N_mp_regen.
N_mp_after_soft_regen	(optional: by default soft regeneration is not applied, the presence of these inputs enables it) Target number of MPs obtained after a soft regeneration.
N_mp_async_regen, N_mp_after_async_regen	(optional: by default asynchronous soft regeneration is not applied, the presence of these inputs enables it) Parameters for asynchronous soft regeneration (the simulation does not wait for the next bunch passage to regenerate).

Space charge parameters	
Dt_sc	[s] Time step for the update of the electron space charge field map.
Dh_sc	[m] Grid size for the space charge PIC solver.
t_sc_ON	[s] Electron space charge forces are neglected for t<t_sc_ON – normally t_sc_ON=0.
sparse_solver	(optional) choices are 'klu', 'scipy_slu' [default]
PyPICmode	(optional) Options are 'FiniteDifferences_ShortleyWeller' [default], 'ShortleyWeller_WithTelescopicGrids', 'FiniteDifferences_Staircase'
Dh_electric_energy	[m] (optional – default = None) Grid size used to compute stored electric energy. If None is given, electric energy is not computed.
flag_em_tracking	(optional – default = False) Forces generated by electrons are computed using electromagnetic potentials instead of quasi-electrostatic approximation (for more information see `https://indico.cern.ch/event/840676/contributions/3532754/`).

flag_reinterp_fields_at_substeps	(optional – default = False) If true, field maps from beams and clouds are interpolated at each Boris substep.

Multigrid parameters To be used with PyPICmode = 'ShortleyWeller_WithTelescopicGrids'	
f_telescope	Magnification factor between grids (it must be always $0 < f < 1$).
target_grid	Target grid parameters. It is a dictionary containing: 'x_min_target', 'x_max_target', 'y_min_target', 'y_max_target' and 'Dh_target'.
N_nodes_discard	Number of nodes at the edge of the internal grids which are discarded in the field interpolation (going to coarser grid).
N_min_Dh_main	Minimum size of the first internal grid in the Δh of the main grid.

Saving settings	
filen_main_outp	Main output filename or path. (optional - default 'Pyecltest.mat')
save_only	Define subset of output variables to be saved, other variables are discarded. (optional - default None)
dec_fact_out	Output is saved only for every nth time step.
save_mat_every	Output file .mat is saved only for every nth bunch passage. (optional - default 1)
Dx_hist	[m] Defines the binning for the saving of the horizontal histograms.
r_center	[m] Radius of a circle around the (0, 0) point (center of the chamber) for the calculation of the local central density.
Dt_En_hist	[s] Time step used to store the energy spectrum of the electrons impacting on the chamber.
Nbin_En_hist	Number of bins used for the energy histogram.
En_hist_max	[eV] Maximum energy in energy spectrum. Larger energies are attributed to the last bin of the histogram.
flag_En_hist_seg	(optional – default False) If enabled, energy histograms per segment are saved.
flag_lifetime_hist	If set to True enables lifetime histogram (optional – default=False).

If **flag_lifetime_hist** is set to True, the variables **Dt_lifetime_hist**, **Nbin_lifetime_hist** and **lifetime_hist_max** need to be set.	
Dt_lifetime_hist	[s] Time step used to store the histogram of the electrons lifetime.
Nbin_lifetime_hist	Number of bins used for the histogram of the electrons lifetime.
lifetime_hist_max	[s] Maximum lifetime in the histogram of the electrons lifetime. Larger lifetimes are attributed to the last bin of the histogram.
flag_cos_angle_hist	Save the cosine of the incident angle of impacting electrons. (optional – default True)
cos_angle_width	Width of the histogram bins. (optional – default 0.05)
flag_movie	(optional – default=0) (1 $\Rightarrow$ On, 0 $\Rightarrow$ Off) Saves files with the electron density distribution at each space charge evaluation.
flag_sc_movie	(optional– default=0) (1 $\Rightarrow$ On, 0 $\Rightarrow$ Off) Saves files with the electron space charge electric field distribution at each space charge evaluation.
save_mp_state_time_file	(optional – default: no MP state is saved) [s] List of instants at which MP positions and velocities are saved on file. The list can also be provided as a .mat file.
flag_detailed_MP_info	(optional – default=0) (1 $\Rightarrow$ On, 0 $\Rightarrow$ Off) Enables save of MP number at each time step.
flag_hist_impact_seg	(optional – default=0) (1 $\Rightarrow$ On, 0 $\Rightarrow$ Off) If the chamber is a polygon, enables saving the number of electrons which impacts onto each segment of polygon.
flag_verbose_file, **flag_verbose_stdout**	(optional – default=False) True/False toggles to save detailed information of pathological impacts in output file and stdout respectively.
dec_fac_secbeam_prof	(optional – default=1) Decimation factor to decrease the number of points of the longitudinal beam profile of secondary beams.

el_density_probes	(optional – default: no probes) (List of Python dictionaries) Probes for electron density evaluation. For each of them it is necessary to indicate the position (x,y) and radius of the circle (r_obs).
save_simulation_state_time_file	(optional – default: no simulation state is saved) [s] List of instants at which the full simulation state is saved on file (pickle files). The list can also be provided as a .mat file.
checkpoint_folder	(optional – default: no checkpoints are saved) Path to folder where checkpoints are saved.
checkpoint_DT	(optional – default: no checkpoints are saved) Time interval in seconds between checkpoints being saved.
copy_main_outp_folder	(optional – default: output is not copied) Path to folder for storing backups of the output. Backups are needed to restart a simulation from a checkpoint.
copy_main_outp_DT	(optional – default: output is only copied if a checkpoint is saved) Enables backing up the output file in regular intervals.
x_min_hist_det, x_max_hist_det, y_min_hist_det, y_max_hist_det, Dx_hist_det	(optional – default: no detailed histogram is saved) [m] Defines a region of the chamber where a detailed horizontal distribution histogram is saved (_det stands for detailed).
step_by_step_custom_observables, pass_by_pass_custom_observables, save_once_custom_observables	(optional – default: no custom output) Custom observable definition. See example for detailed usage.

Cloud particles	
cloud_mass	(optional – default=electron mass) [kg] Mass of cloud particles.
cloud_charge	(optional – default=electron charge) [C] Charge of cloud particles.

Additional clouds Simulations with multiple clouds can be enabled with the following input parameter. See Section 1.5 for a detailed description of this simulation mode.	
additional_clouds_file_list	(optional – default=[]) List of names of additional cloud files.

SEY Extraction of the SEY curves can be disabled.	
extract_sey	(optional – default=True) If set to False, it disables the extraction the SEY curves.

Extraction of electron emission energy distribution	
extract_ene_dist	(optional – default=False) If set to True, secondary emission energy distributions are extracted and saved.
ene_dist_test_E_impact_eV	The impacting energy used in the energy distribution extraction. [eV]
Nbin_extract_ene	The number of bins used in the extracted energy distributions.
factor_ene_dist_max	Defines the maximum energy included in the extracted energy distribution. The maximum energy is factor_ene_dist_max * ene_dist_test_E_impact_eV

1.2 Machine Parameters

Chamber profile description	
chamb_type	(optional – default = 'ellip') Possible settings: 1. chamb_type = 'ellip' for elliptical chamber profile. 2. chamb_type = 'rect' for rectangular chamber profile. 3. chamb_type = 'polyg' or 'polyg_cython' for polygonal chamber profile.
When chamb_type = 'ellip' or 'rect' the following input variables must be provided:	
x_aper, y_aper	Horizontal and vertical semi-apertures of the transverse chamber profile.
When chamb_type = 'polyg' the following input variable must be provided:	
filename_chm	Name of file containing horizontal and vertical vertexes of the chamber profile.
filename_chm_photoem	Name of file containing horizontal and vertical vertexes of the chamber profile that is used for the photoemission mode 'per_segment'. The chamber file must also contain the cumulative distribution function of the emission probability for each segment. The vertexes of this chamber must lie on the edges of the main chamber.
flag_assume_convex	[optional] Default: True
flag_counter_clockwise_chamb	[optional] Default: True. Only needed for the photoemission model 'per_segment'. It specifies the order in which the vertexes are defined.

Tracking and magnetic field (the tracking algorithm has to be chosen according to the magnetic field conditions).

track_method	(optional – default = 'StrongBdip') Possible settings: 1. track_method = 'StrongBdip' for vertical uniform magnetic field. 2. track_method = 'StrongBgen' for a generic transverse magnetic field map. 3. track_method = 'BorisMultipole' for arbitrary multipoles with Boris electron tracker.
When track_method = 'StrongBdip' the following input variables must be provided:	
B	[T] Value of the vertical uniform magnetic field. If B=-1 the magnetic field is calculated from the total length of the bending magnets (bm_totlen) and from the beam energy.
bm_totlen	[m] Total length of dipoles inside the accelerator (it must be provided when B=-1).
When track_method = 'StrongBgen' the following input variables must be provided:	
B_map_file	Name of a .mat file containing the transverse field map. The case of a quadrupole magnet with unit gradient is embedded in the code and can be used setting: B_map_file = 'analytic_quadrupole_unit_grad'.
Bz_map_file	Name of a .mat file containing the longitudinal field map. ??? The case of a quadrupole magnet with unit gradient is embedded in the code and can be used setting: B_map_file = 'analytic_quadrupole_unit_grad'.
fact_Bmap	(optional – default = 1.) Scaling factor applied to the magnetic field map.
B0x, B0y, B0z	(optional – default = 0) [T] Uniform magnetic fields added to the map.
B_zero_thrld	[T] Value below which the local magnetic field is approximated with 0.
When track_method = 'BorisMultipole', N_sub_steps and B_multip must be provided, while B_skew is optional:	
N_sub_steps	Number of tracking sub-steps per time step.
B_multip	Magnetic momenta. Higher order multipoles and skew magnets are supported, as well as compound magnets. The formalism in PyECLOUD follows this formula: $B_y + iB_x = \sum_{n=0}^{\infty} \frac{1}{n!} (b_n + ib'_n) (x + iy)^n$

	B_multip specifies $b_n = \frac{\partial^n B_y}{\partial x^n}$. For example: to simulate a dipole with B=8.3 T set B_multip = [8.3]; to simulate a sextupole with 100 T/m^2 set B_multip = [0., 0., 100.].
B_skew	The skew parameters can be included through B_skew. For a skew quadrupole with a gradient of 12 T/m, set B_multip to [0.,0.] and B_skew to [0.,12.]. B_skew specifies $b'_n = \frac{\partial^n B_x}{\partial x^n}$ and defaults to *None*.
B0x, B0y, B0z	(optional – default = 0) [T] Uniform magnetic fields added to the map.

Optics parameters (can be provided also in the beam parameter file(s) independently for the different beams) — not needed if transverse beam size is directly defined in the beam parameter files.	
betafx, betafy	[m] Horizontal and vertical beta functions at the simulation section.
Dx, Dy	(optional – default = 0) [m] Horizontal and vertical dispersion functions at the simulation section.

Residual gas ionization parameters (if the following input parameters are omitted primary electron generation by residual gas ionization is not enabled).	
gas_ion_flag	(optional – default=0) (1 ⇒ On, 0 ⇒ Off) Enables primary electron generation by residual gas ionization.
P_nTorr	[nTorr] Pressure in the vacuum chamber.
sigma_ion_MBarn	[MBarn] Ionization cross section of the residual gas.
Temp_K [K]	Temperature of the residual gas.
unif_frac	Fraction of primary electrons uniformly distributed inside the chamber. The remaining part is generated according to the transverse distribution of the beam.
E_init_ion	[eV] Initial energy of the generated electrons.
t_ion	???

Photoemission parameters (if generation by photoemission is not desired, the following parameters can be omitted).

photoem_flag	(optional – default=0) (0 ⇒ Off, 1 ⇒ traditional, 2 or 'from_file' ⇒ Angular distribution from file, 3 or 'per_segment' ⇒ Part of chamber definition) Enables primary electron generation by photoemission.
k_pe_st	[m^{-1}] Number of photoelectrons to be generated per proton and per unit length.
energy_distribution	Can be one of: • 'gaussian': $p(E) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{(E-\mu)^2}{2\sigma^2}\right)$ • 'log-normal': $p(E) = \frac{1}{E\sigma\sqrt{2\pi}} e^{\frac{(\log(E)-\mu)^2}{2\sigma^2}}$ • 'rect': $p(E)$ uniform for $\mu - \sigma < E < \mu + \sigma$ • 'mono': $E = \mu$ • 'lorentz': $p(E) = \frac{1}{\pi} \frac{\sigma}{\sigma^2 + (E-\mu)^2}$ The Lorentz and normal distributions are cut at 0 to prevent negative energies of photoemitted electrons.
e_pe_sigma	[eV] σ from above.
e_pe_max	[eV] μ from above.
out_radius	[m] Radius of a circle external to the chamber.
phem_resc_fact	Rescaling factor on the position vector of the generated photoelectrons.
photoelectron_angle_distribution	See secondary_angle_distribution in the secondary emission parameters.
Photoemission parameters for photoem_flag = 1 **A part of photoelectrons is generated in the corner of the beam screen on direct impact of photons (non-reflected). The reflected photons generated elsewhere.**	
inv_CDF_refl_photoem_file	Name of a .mat file providing the inverse of the Cumulative Distribution Function (CDF) for the angular distribution of the reflected photons. If inv_CDF_refl_photoem_file = 'unif_no_file' the reflected photons have uniform angular distribution and no file needs to be provided.
refl_frac	Fraction of photoelectrons generated by reflected photons.
x0_refl, y0_refl	[m] Impact poiny of primary (not reflected) photons. x0_refl can be set to 'left' or 'right', in that case the point is computed automatically.

alimit	[rad] Extent (1σ) of the Gaussian angular distribution of photoelectrons generated by non-reflected photons.
Photoemission parameters for photoem_flag = 2 or 'from_file' **The coordinates of all generated photoelectrons is specified from a file**	
inv_CDF_all_photoem_file	See inv_CDF_refl_photoem_file, but for all photons. If it is set to 'unif_no_file', the photoelectron generation is uniform over the whole surface. Example scripts to generate a file with the correct format can be found in the folder `PyECLOUD/other/photoemission_angular_distribution`.
Uniform initial distribution Simulation starts with electrons uniformly distributed in the chamber (if the following input parameters are omitted this feature is not enabled).	
init_unif_flag	(optional – default=0) ($1 \Rightarrow$ On, $0 \Rightarrow$ Off) Enables uniform electron distribution at the beginning of the simulation.
Nel_init_unif	Initial number of electrons.
E_init_unif	[eV] Initial energy of the generated electrons.
x_max_init_unif, x_min_init_unif, y_max_init_unif, y_min_init_unif	[m] Edges of the region where the electrons are generated.

1.3 Beam parameters

Basic definitions	
m0_part	(optional – default=proton mass) [Kg] Mass of beam particles.
q_part	(optional – default=proton charge) [C] Charge of beam particles.
energy_eV	[eV] Total energy of the beam particles.
Dp_p	(optional – default=0) Momentum spread (r.m.s.) of the beam. This parameter is not used if the transverse beam size is directly provided (vars. sigmax, sigmay).
nemittx, nemitty	[m] Normalized transverse emittance of the beam. These parameters are not used if the transverse beam size is directly provided (vars. sigmax, sigmay).
x_beam_pos, y_beam_pos	(optional – default=0) [m] Transverse position of the beam within the vacuum chamber.
sigmax, sigmay	(optional – default=-1) [m] Transverse geometrical beam size. If sigmax, sigmay=-1 the beam size is calculated through the energy, the normalized emittance, the dispersion and the momentum spread of the beam.

Transverse electric field of the beam	
beam_field_file	Name of a .mat file containing the map of the transverse beam electric field. If beam_field_file = -1 or beam_field_file = 'computeFD' the beam field map is computed using the embedded Finite Difference (FD) Poisson solver. If beam_field_file='computeBE' the electric field is computed using the Bassetti-Erskine formula and the image terms for the elliptical boundary conditions (this option is available only when the chamber profile is elliptical). If beam_field_file='compute_FDSW_multigrid' the beam field map is computed using the embedded Finite Difference (FD) Poisson solver in the multigrid case.
save_beam_field_file_as	(optional – default: the beam field map is not saved) Name of the file where the employed field map is saved.

When beam_field_file = -1 or beam_field_file = 'computeFD' the following parameter must be provided:	
Dh_beam_field	Grid size for the FD solver and for the saved field map.
When beam_field_file = 'computeBE' the following parameters must be provided:	
Nx, Ny	Number of points of the employed field map.
nimag	Number of image terms.
When beam_field_file = 'compute_FDSW_multigrid' the following parameters must be provided:	
Dh_beam_field	Grid size for the FD solver and for the saved field map.
f_telescope_beam	Magnification factor between grids (it must be always $0 < f < 1$)
target_grid_beam	Target grid parameters. It is a dictionary containing: 'x_min_target', 'x_max_target', 'y_min_target', 'y_max_target' and 'Dh_target'.
N_nodes_discard_beam	Number of nodes at the edge of the internal grids which are discarded in the field interpolation (going to coarser grid).
N_min_Dh_main_beam	Minimum size of the first internal grid in the Δh of the main grid.

Beam longitudinal profile	
b_spac	Bunch spacing. It is used to generate the beam profile when it is defined in the form of a bunched beam. It is also the time interval for cleaning and regeneration of the MP set, as well as for saving.
fact_beam	Rescaling factor applied to the beam profile.
coast_dens	[p/m] Coasting beam density added to the beam profile.
flag_bunched_beam	Two possible settings: 1. flag_bunched_beam = 0 the beam profile is loaded from file. 2. flag_bunched_beam = 1 the beam profile is generated from a filling pattern.
When flag_bunched_beam = 1 the following parameters must be provided:	
sigmaz	[m] Bunch length (1σ).

t_offs	[s] Delay in the longitudinal profile (n.b. if t_offs=0 only half of the first bunch is simulated).
filling_pattern_file	Name of a file providing the intensities of the different bunches (zeros for empty slots). The data can be provided also in form of a python list with no need to provide an input file, e.g. filling_pattern_file=4*(72*[1.]+8*[0.]).
When flag_bunched_beam = 0 the following parameter must be provided:	
beam_long_prof_file	Name of a .mat file providing the longitudinal beam profile. This file will also define the time step used for the simulation.

1.4 Secondary emission parameters

Choice of the model	
switch_model	(optional – default=0) Different secondary emission models: 1. switch_model = 0 or 'ECLOUD' for the model used in ECLOUD. 2. switch_model = 'ACC_LOW' for a more accurate treatment of low energy impacts. 3. switch_model = 'ECLOUD_nunif' for the model used in ECLOUD, with sey info in the chamber shape file. 4. switch_model = 'ECLOUD_nunif_charging' for the model used in ECLOUD and charging effects, with sey and charging info in the chamber shape file. 5. switch_model = 'cos_low_ene' for cosine low energy dependence. 6. switch_model = 'flat_low_ene' for flat low energy dependence. 7. switch_model = 'furman_pivi' for the Furman-Pivi model of secondary electron emission. 8. switch_model = 'from_file' for interpolation of a user-specified curve.

Secondary Electron Yield These are the parameters for the 'ECLOUD' model, as described in G. Iadarola's thesis.	
Emax	Energy corresponding to the maximum SEY.
s_param	Shape parameter of the true SEY curve.
del_max	Maximum of the SEY curve.
R0	Weight of the reflected electron component.
E0	Shape parameter of the reflected SEY curve.
E_th	Maximum energy for true secondary electrons.
sigmafit	Sigma parameter of the lognormal distribution.
mufit	Mu parameter of the lognormal distribution.

Secondary Electron Yield These are the parameters for the 'furman_pivi' model.	
E_th	Maximum energy for true secondary electrons.
furman_pivi_surface	A Python dict containing all parameters needed for the Furman-Pivi model. The parameters are listed and described below.
General parameters. To be specified in furman_pivi_surface.	1. 'use_modified_sigmaE': Boolean, modifies the σ_e parameter from the FP paper in the same way as done in POSINST. 2. 'use_ECLOUD_theta0_dependence': Boolean, use same incident angle dependence of SEY as in ECLOUD. 3. 'use_ECLOUD_energy': Boolean, use same secondary electron energy distribution as in ECLOUD. Also makes elastic electrons use the ECLOUD emission angle. 4. 'conserve_energy': Boolean, ensures the sum of emitted electron energies in a single event is not greater than the impacting electron energy. 5. 'exclude_rediffused': Boolean, sets $\delta_r = 0$. 6. 'choice': 'poisson' or 'binomial', choice of probability distribution for $P'_{n,ts}$ 7. 'M_cut': int, the maximum number of macroparticles that can be emitted in a single event. 8. 'p_n': numpy array containing M_cut floats. Corresponds to p_n in FP paper. 9. 'eps_n': numpy array containing M_cut floats. Corresponds to ϵ_n in FP paper.

Parameters for backscattered electrons. To be specified in furman_pivi_surface.	1. 'p1EInf': float, $P_{1,e}(\infty)$. 2. 'p1Ehat': float, $\hat{P}_{1,e}$. 3. 'eEHat': float, $\hat{E}_e$. 4. 'w': float, W. 5. 'p': float, p. 6. 'e1': float, e_1. 7. 'e2': float, e_2. 8. 'sigmaE': float, σ_e. 9. "theta_e_max': float, value to which the incidence angle is limited when computing the angular dependence for backscattered electrons.
Parameters for rediffused electrons To be specified in furman_pivi_surface	1. 'p1RInf': float, $P_{1,r}(\infty)$. 2. 'eR': float, E_r. 3. 'r': float, r 4. 'q': float, q. 5. 'r1': float, r_1. 6. 'r2': float, r_2. 7. "theta_e_max': float, value to which the incidence angle is limited when computing the angular dependence for rediffused electrons.

Parameters for true secondaries. To be specified in furman_pivi_surface.	1. 'deltaTSHat': float, $\hat{\delta}_{ts}$. 2. 'eHat0': float, $\hat{E}_{ts}$. 3. 's': float, s. 4. 't1': float, t_1 5. 't2': float, t_2 6. 't3': float, t_3 7. 't4': float, t_4

Other parameters	
sey_file	Needed for secondary emission model 'from_file'. Path to a file that specifies the reflected and true secondary emission yields for different energies. An example can be found in the subfolder sey_files in the PyECLOUD code.
flag_costheta_delta_scale	If enabled the SEY curve associated to true secondary electrons is scaled as a function of the angle of the impacting electron. Default is TRUE.
flag_costheta_Emax_shift	If enabled the SEY curve associated to true secondary electrons is shifted (chenge of energy scale) as a function of the angle of the impacting electron. Default is TRUE.
switch_no_increase_energy	(Default=0) 1. switch_no_increase_energy = 0 : Off. 2. switch_no_increase_energy = 1 : No longer in use. 3. switch_no_increase_energy = 2 : Ensure that the true secondaries do not have more energy than the corresponding impacting electrons.
thresh_low_energy	Maximum energy for which the energy check is performed.
scrub_en_th	Minimum energy of scrubbing electrons (for scrubbing current estimations).
secondary_angle_distribution	Can be 'cosine_2D', 'cosine_3D' or 'normal_emission'. For new electrons, θ describes the angle between the surface normal and their initial velocity vector. $\mathrm{d}n/\mathrm{d}\theta = \cos\theta$ or $\mathrm{d}n/\mathrm{d}\theta = \cos\theta\sin\theta$ in the 2D or 3D cases, respectively. More info here. The more accurate is the 3D cosine distribution which takes into account the surface element $\sin\theta$ in spherical coordinates. For 'normal_emission', $\theta = 0$.

1.5 Simulations with multiple clouds

The code provides the option of including more than one cloud in the simulation. Each cloud is influenced by the other clouds through their space charge. Cross-ionization processes between different clouds can be defined. These options can be enabled by providing the following additional information.

In the **simulation_parameters.input** file:	
additional_clouds_file_list	(optional – default=[]) List of names of additional cloud files.
cross_ion_definitions	(optional – default={ }) Dictionary that describes the desired cross-ionization processes (more details can be found in `https://indico.cern.ch/event/835473/contributions/3525107/`).

Each cloud file should contain a subset of the input parameters specified in the main input files, as listed below. The following parameters are required in the cloud input file.

Mandatory cloud parameters	
cloud_mass	See Simulation parameters section.
cloud_mass	
gas_ion_flag	See Machine parameters section.
photoem_flag	
init_unif_flag	
init_unif_edens_flag	
switch_model	See Secondary emission parameters section.

In addition, a cloud file can contain the following optional parameters. Any optional parameter that is not specified in the cloud input file takes its value from the main input, and gets the default value if the parameter is not given in the main input files.

Optional cloud parameters	
MP management settings	See the Simulation parameters section for the detailed list of parameters and their usage.
Saving settings	
Log and progress files	
N_sub_steps	See under "Tracking and magnetic field" in the Machine parameters section.
Residual gas ionization parameters	See the Machine parameters section for the detailed list of parameters and their usage.
Photoemission parameters	
Uniform initial distribution	

Uniform initial density	
Secondary emission parameters	As listed in the Secondary emission parameters section.

Each cloud produces its own .mat output file, as described in the following chapter. The file is named according to the main output file name and the cloud file name.

2 Output

The main PyECLOUD output is in a single .mat file (by default called 'Pyecltest.mat'). It can be opened in python, for example using the myloadmat_to_obj module in the main PyECLOUD folder, or usin the scipy.io module.
The file contains arrays of different shape. The time steps in the simulation are spaced by Dt from the simulation parameters input file. However the output is only saved every nth time step, where n is specified by dec_fact_out. The number of these steps is denoted N_step in the following.

Variables saved at every time step		
t(N_steps)	[s]	The time at all time steps where the output is saved.
En_emit_eV_time(N_steps)	[eV]	Combined energy of all electrons emitted since the last step.
En_imp_eV_time(N_steps)	[eV]	Combined energy of all electrons impacted since the last step.
En_kin_eV_time(N_steps)	[eV]	Combined kinetic energy of all electrons currently in the chamber.
Nel_emit_time(N_steps)		Number of electrons emitted since the last step.
Nel_imp_time(N_steps)		Number of electrons impacted since the last step.
Nel_timep(N_steps)		Number of electrons currently in the chamber.
cen_density(N_steps)	[m^{-3}]	Electron density in a sphere around the center of the chamber with radius r_center.
lam_t_array(N_steps)	[m^{-1}]	Current line density of the beam charge.

N_pass is defined as the number of bunch passages. N_bins is the number of bins in the horizontal plane used to produce related histograms. N_en is the number of bins in impact energy used to produce related histograms.

Variables saved at each bunch passage		
t_hist(N_pass)	[s]	The time at which data for the bunch passages are saved.
En_hist		Energy spectrum of the impacting electrons (electron per time interval and per energy bin). The last bin contains all electrons falling outside the selected enegy interval.
all_Ekin_hist		Kinetic energy histogram of all the electrons in the chamber (electron per time interval and per energy bin). The last bin contains all electrons falling outside the selected enegy interval.
t_En_hist	[s]	The time at which the electron energy spectrum is saved.

En_g_hist	[eV]	Center of the energy bins used to save the energy spectrum.
lifetime_hist		Histogram of the electrons lifetime for each bunch passage. The last bin contains all electrons falling outside the selected time interval.
lifetime_g_hist	[s]	Center of the time bins used to save the lifetime histogram.
xg_hist(N_bins)	[m]	The x coordinates corresponding to related histogram bins.
N_mp_corrected_pass(N_pass)		Number of times a MP tracking error occurred.
N_mp_impact_pass(N_pass)		Number of MPs that impacted.
N_mp_pass(N_pass)		Number of current MPs.
N_mp_ref_pass(N_pass)		Current reference MP size.
energ_eV_impact_hist(N_pass, N_bins)		Horizontal histogram of the energy deposited on the walls by the electrons.
nel_hist(N_pass, N_bins)		Horizontal histogram of the electron positions.
nel_impact_hist_scrub(N_pass, N_bins)		Horizontal histogram of the electron impacts with an energy higher than scrub_en_th.
nel_impact_hist_tot(N_pass, N_bins)		Horizontal histogram of all electron impacts.

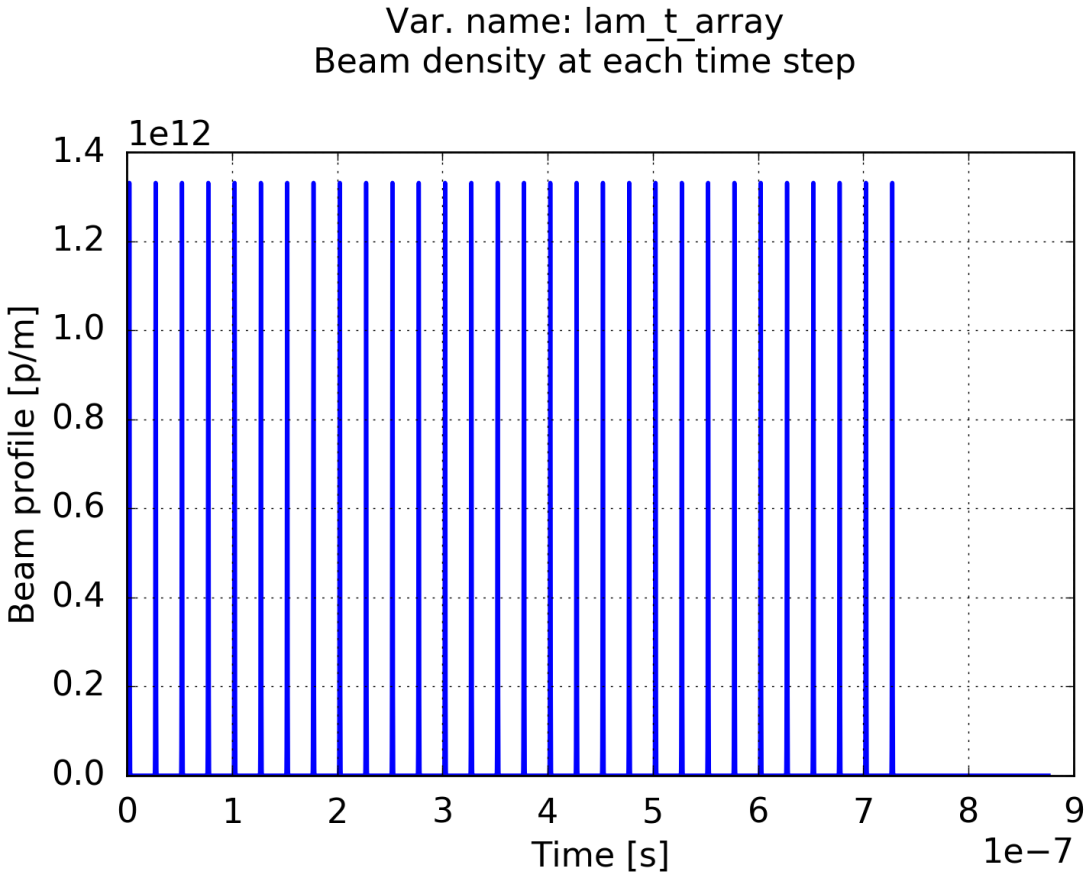
Other variables		
dec_fact_out		Timestep output is only saved every nth time step, where n corresponds to dec_fact_out.
b_spac	[s]	Bunch spacing used for simulation internals.
t_sc_video	[s]	The times at which the space charge forces are recomputed.
U_sc_eV		Not used.
N_mp_time		Only used when flag_detailed_MP_info. Save the total number of macroparticles at each time step.
nel_hist_impact_seg		Number of electrons that impacted on each chamber segment. Only used if flag_hist_impact_seg is set.
energ_eV_impact_seg	[eV]	Combined energy of all impacted electrons on each chamber segment. Only used if flag_hist_impact_seg is set.
t_sec_beams	[s]	Time where information is stored on secondary beams.
sec_beam_profiles [m^{-1}]		Charge profile of secondary beams.

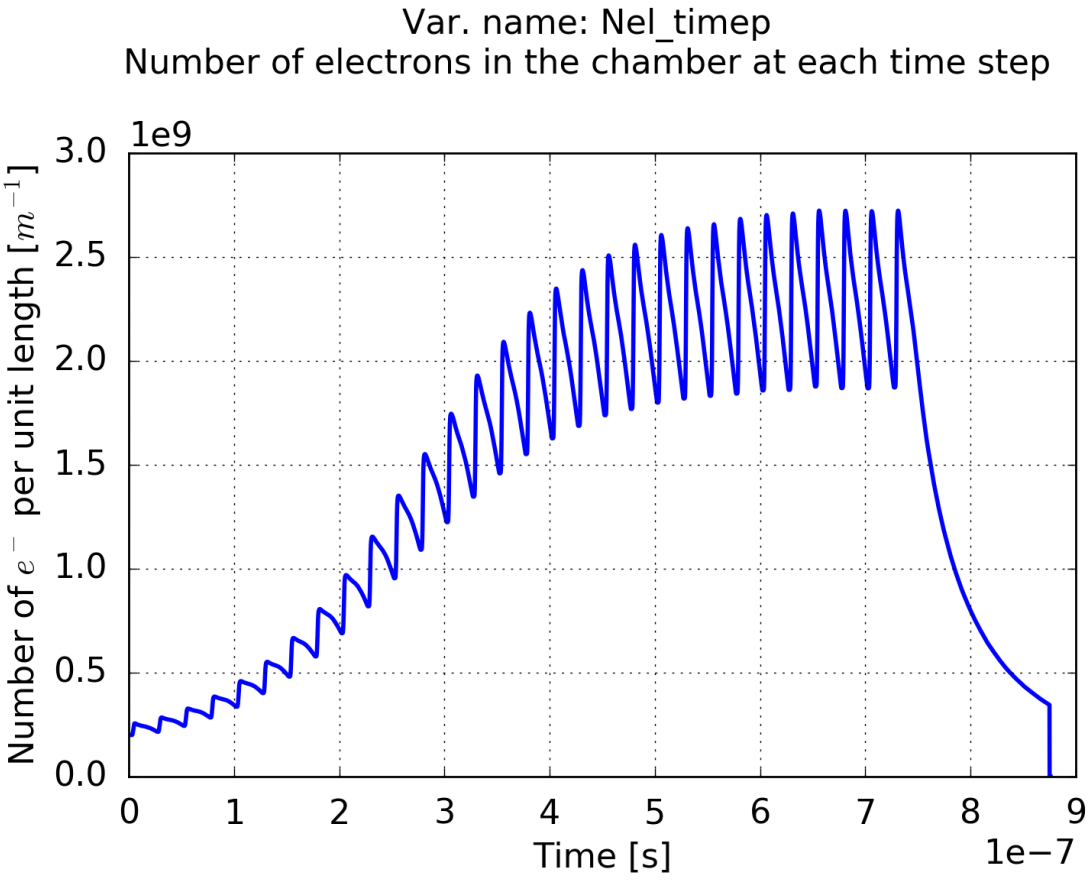
el_dens_at_probes		[m^{-3}]Electron densities at locations specified with el_density_probes.
x_el_dens_probes	[m]	x coordinates of electron density probes
y_el_dens_probes	[m]	y coordinates of electron density probes
r_el_dens_probes	[m]	Radii used to measure electron density probes
nel_hist_det		Detailed electron histogram. Only used if flag_hist_det is set.
nel_hist_det		Coordinates corresponding to the bins of nel_hist_det. Only used if flag_hist_det is set.
cos_angle_hist		The cosine of the impact angles of all electrons.
xg_hist_cos_angle		The bins corresponding to the cos_angle_hist.

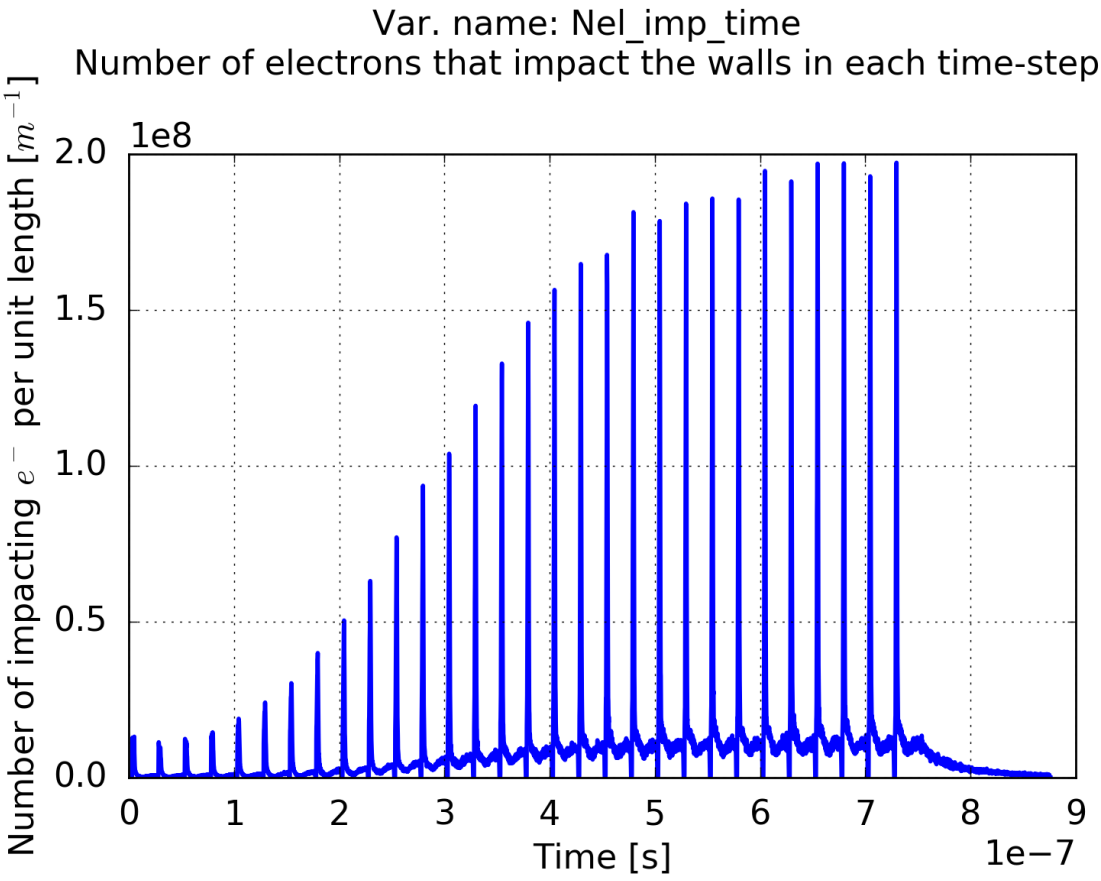
Example plots, for one of the simulations in the PyECLOUD test suite, are produced by the script:

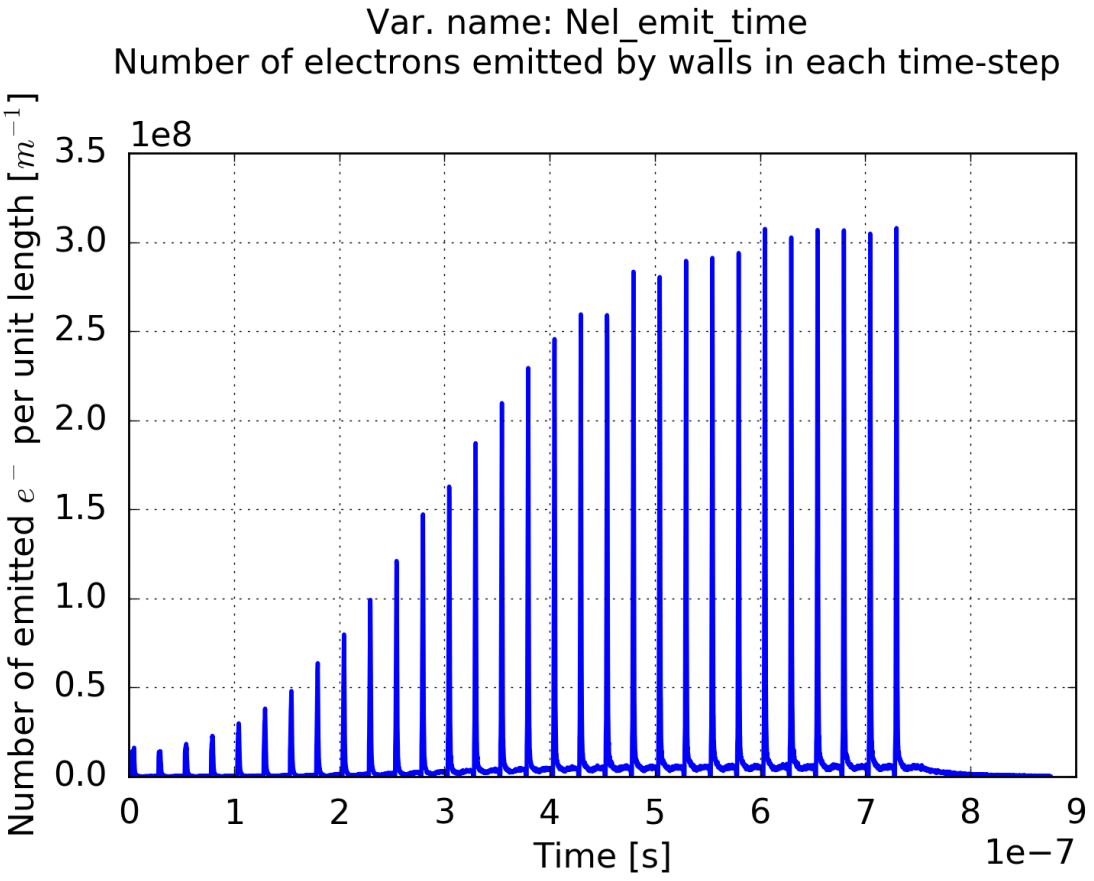
doc/example/000_plot_main_output.py

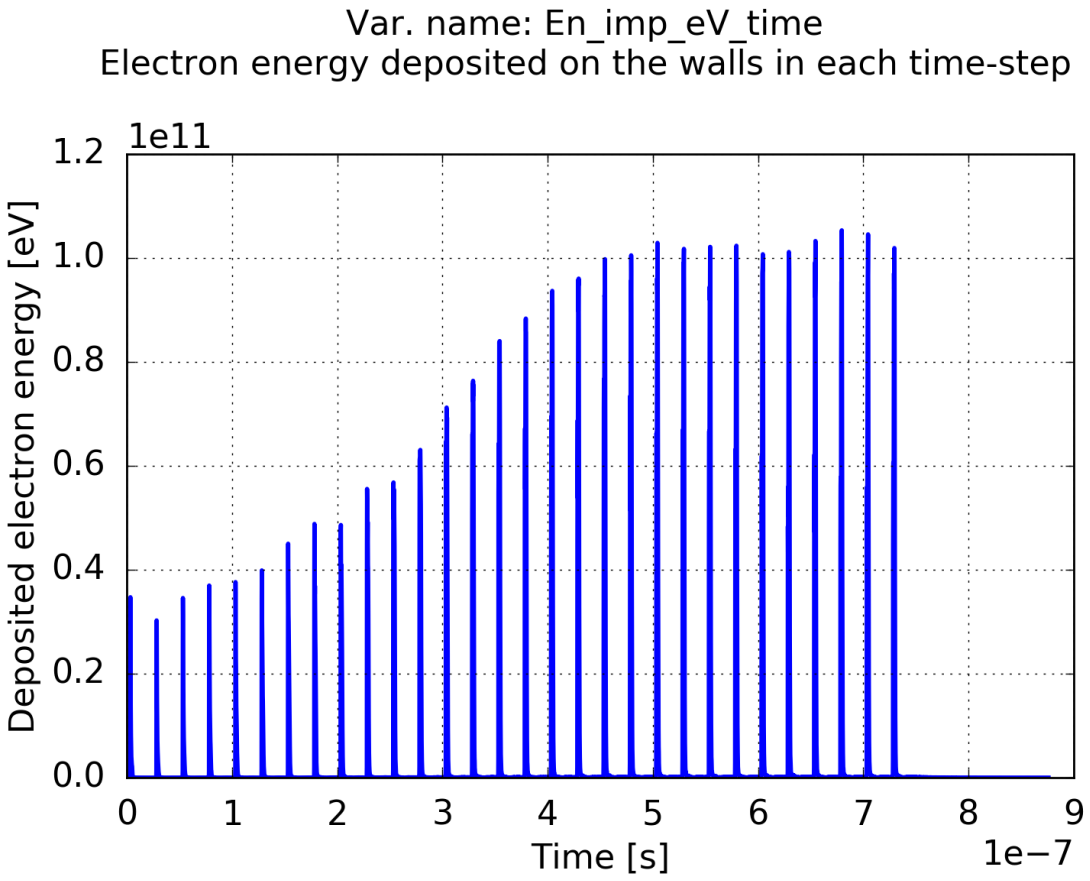
and reported in the following pages (with some description).

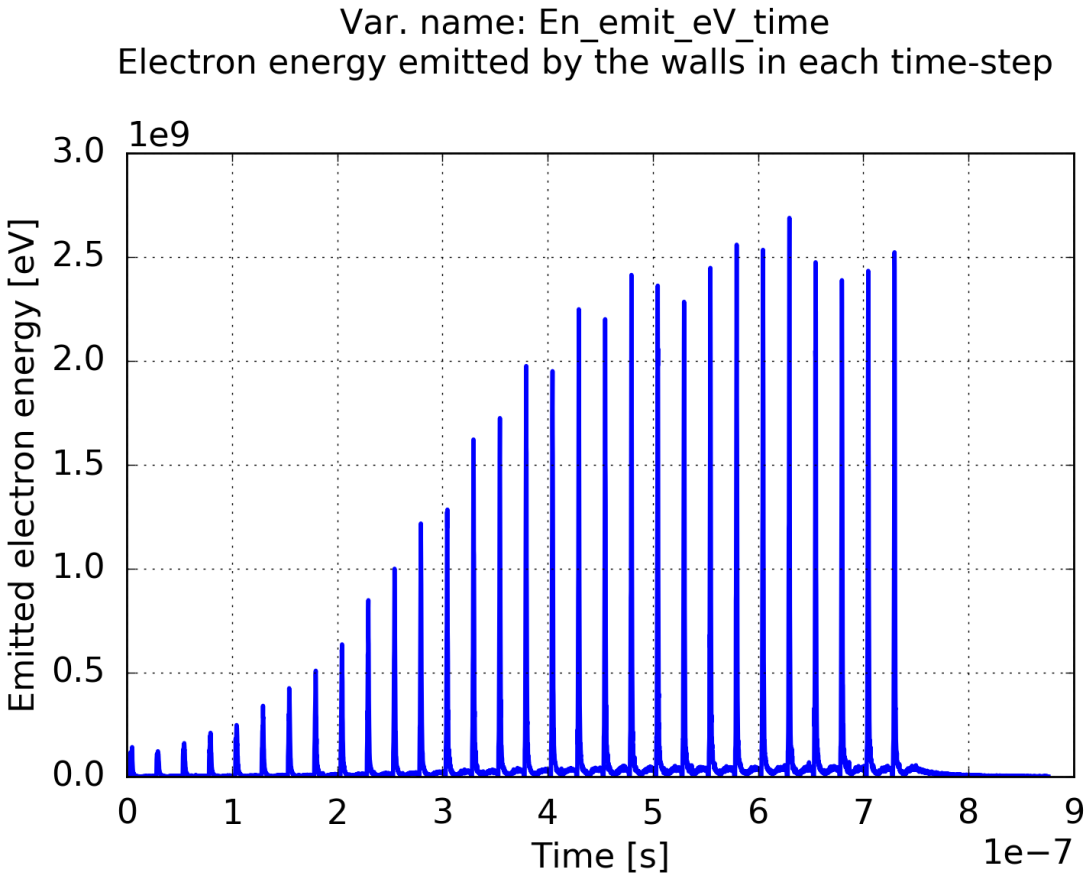

Var. name: lam_t_array
Beam density at each time step
1e12
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Beam profile [p/m]
0
1
2
3
4
5
6
7
8
9
Time [s]
1e−7

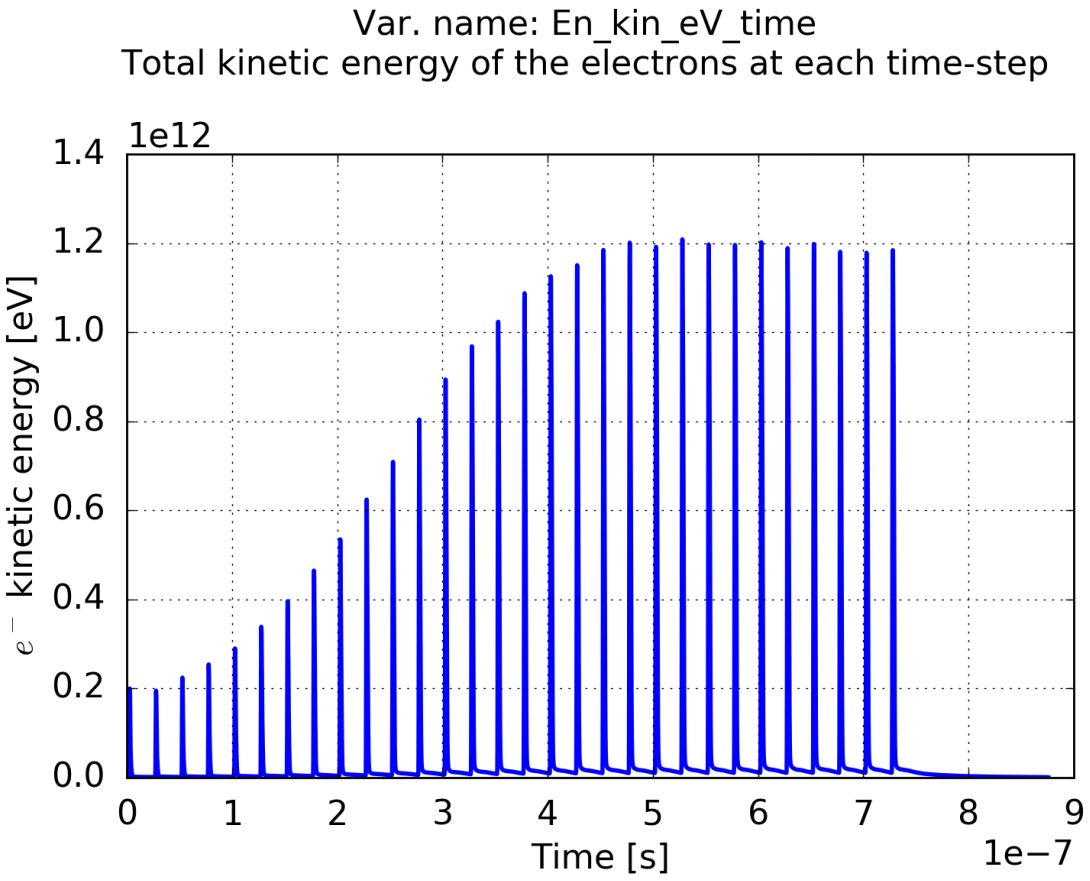

Var. name: Nel_timep
Number of electrons in the chamber at each time step
1e9
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Number of e^- per unit length [m^{-1}]
0
1
2
3
4
5
6
7
8
9
Time [s]
1e−7

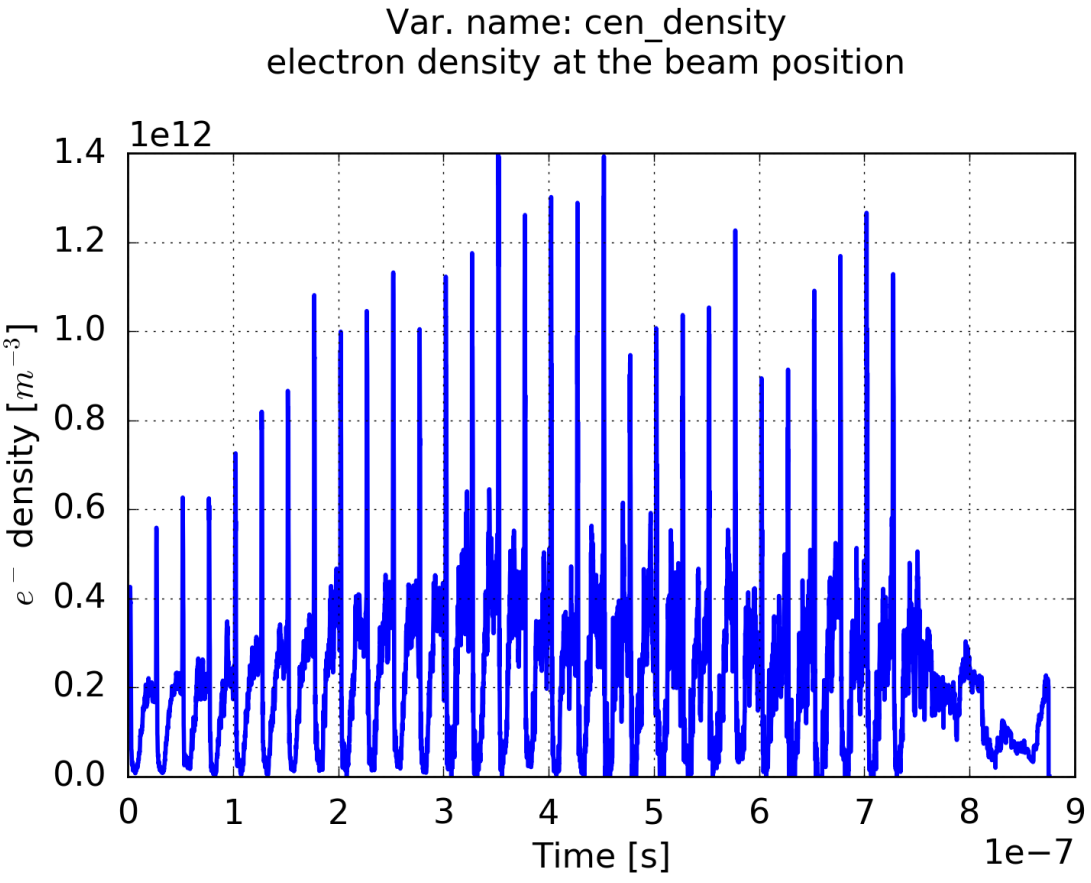

Var. name: Nel_imp_time
Number of electrons that impact the walls in each time-step
1e8
2.0
1.5
1.0
0.5
0.0
Number of impacting e^- per unit length [m^{-1}]
0
1
2
3
4
5
6
7
8
9
Time [s]
1e−7

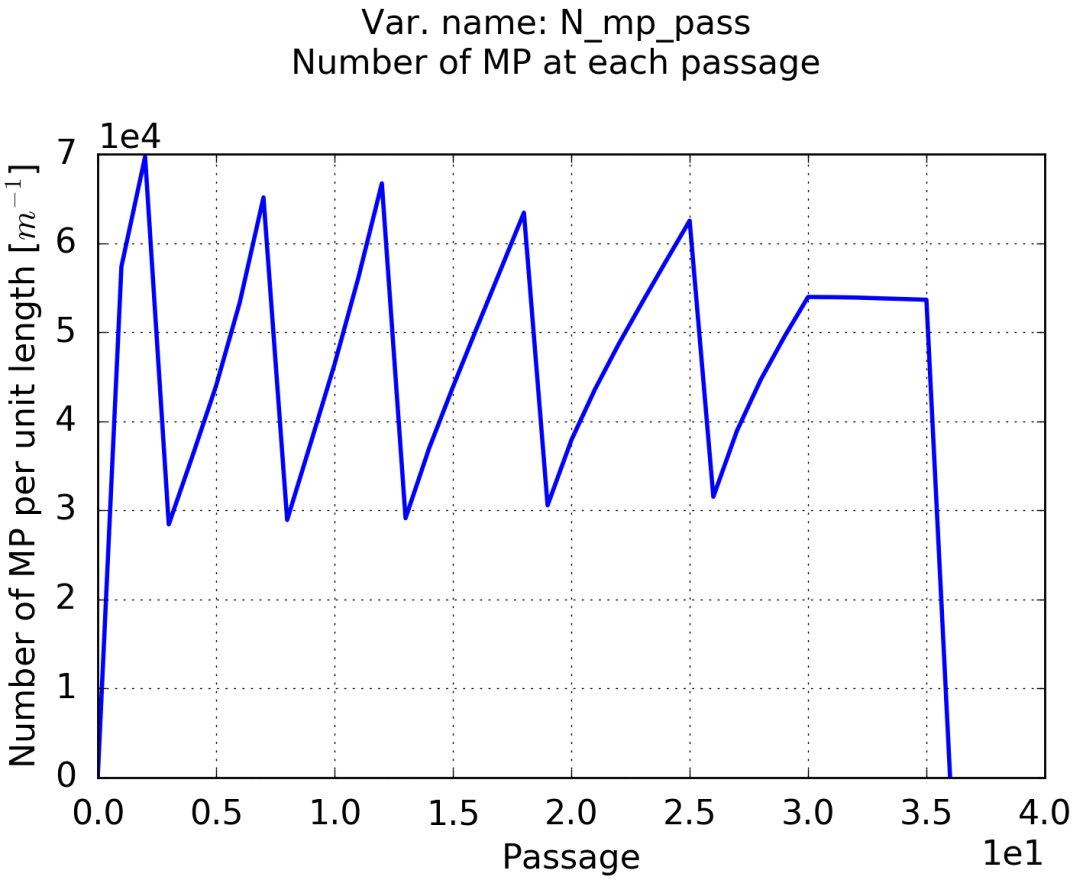

Var. name: Nel_emit_time
Number of electrons emitted by walls in each time-step
1e8
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Number of emitted e^- per unit length [m^{-1}]
0
1
2
3
4
5
6
7
8
9
Time [s]
1e−7

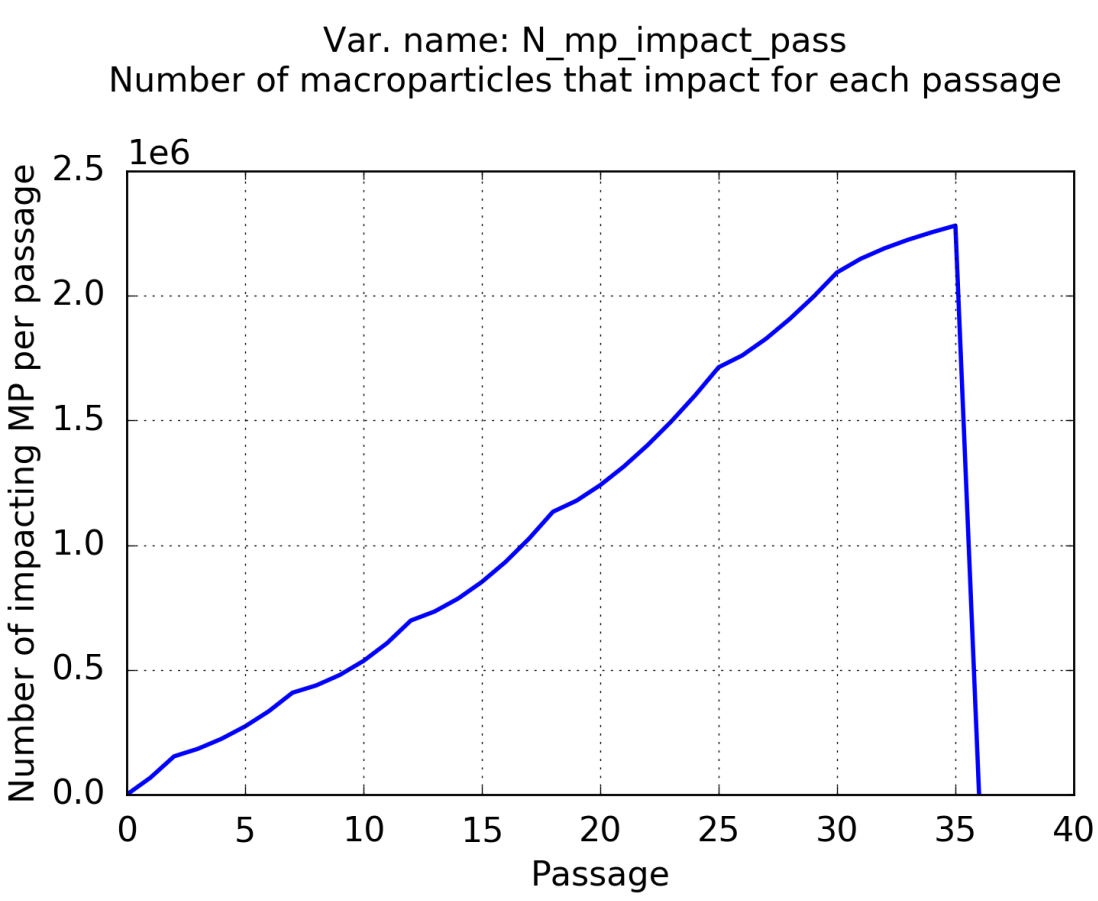

Var. name: En_imp_eV_time
Electron energy deposited on the walls in each time-step
1e11
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Deposited electron energy [eV]
0
1
2
3
4
5
6
7
8
9
Time [s]
1e−7


Var. name: En_emit_eV_time
Electron energy emitted by the walls in each time-step
1e9
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Emitted electron energy [eV]
0
1
2
3
4
5
6
7
8
9
Time [s]
1e−7

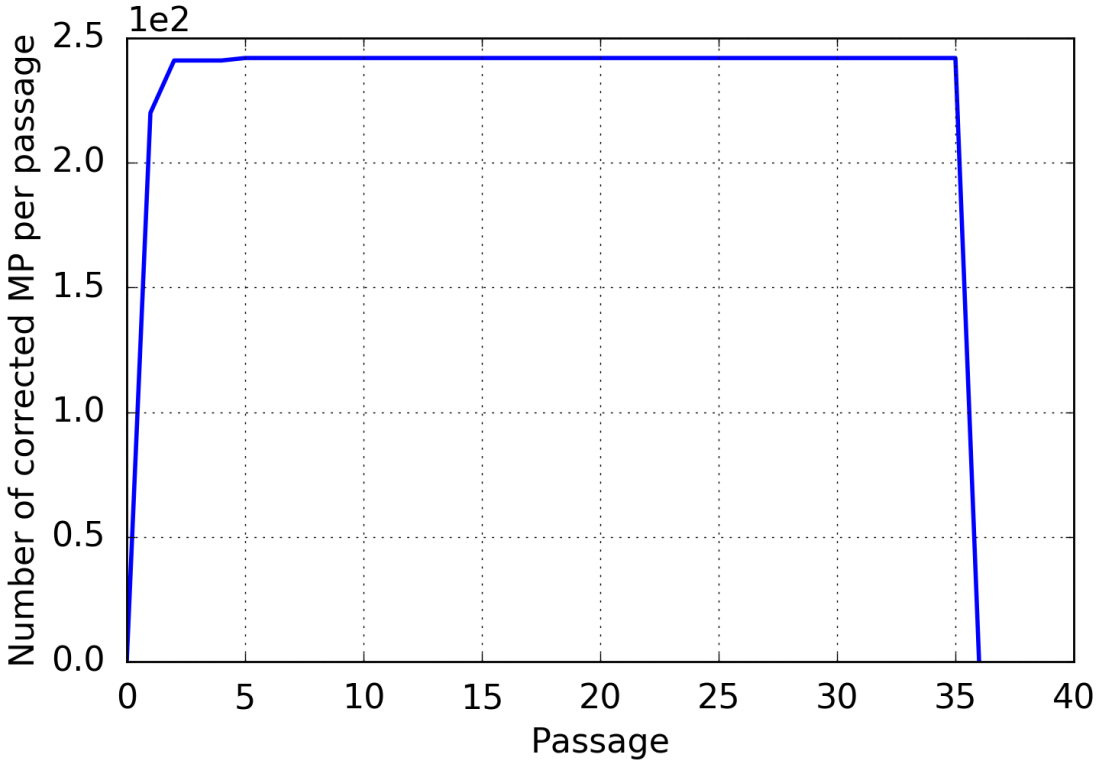

Var. name: En_kin_eV_time
Total kinetic energy of the electrons at each time-step
1e12
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
e^- kinetic energy [eV]
0
1
2
3
4
5
6
7
8
9
Time [s]
1e−7


Var. name: cen_density
electron density at the beam position
1e12
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
e^- density [m^{-3}]
0
1
2
3
4
5
6
7
8
9
Time [s]
1e−7

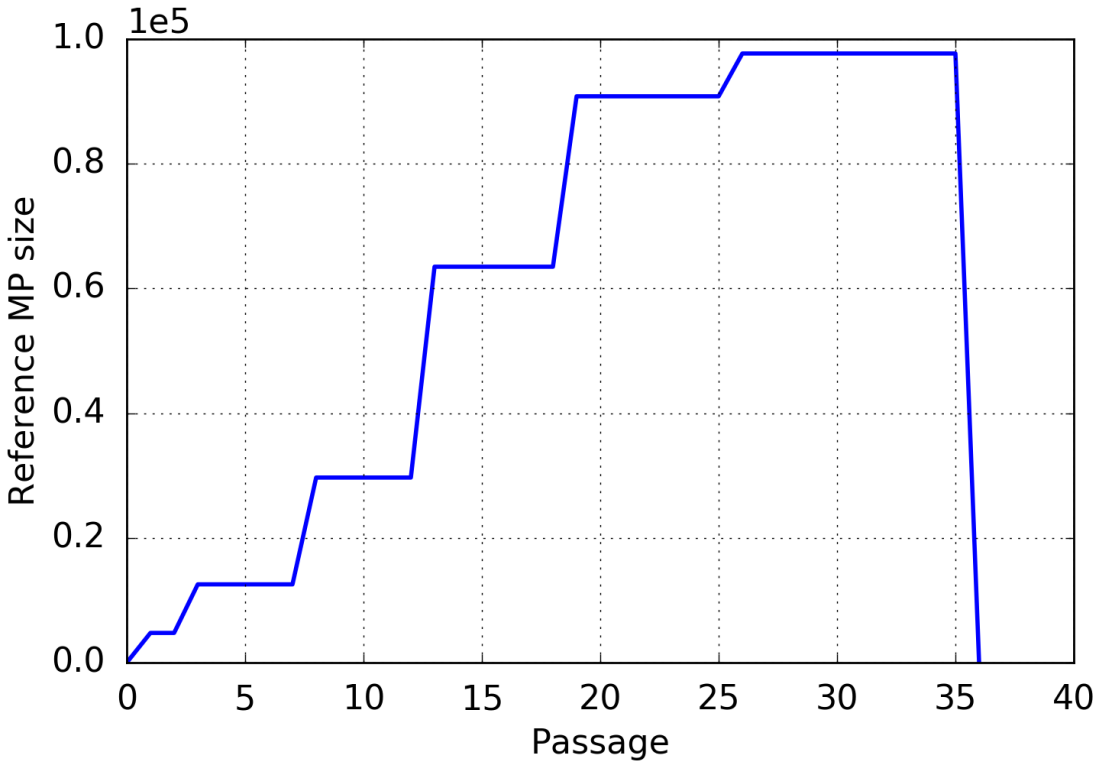

Var. name: N_mp_pass
Number of MP at each passage
1e4
7
6
5
4
3
2
1
0
Number of MP per unit length [m^{-1}]
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1e1
Passage

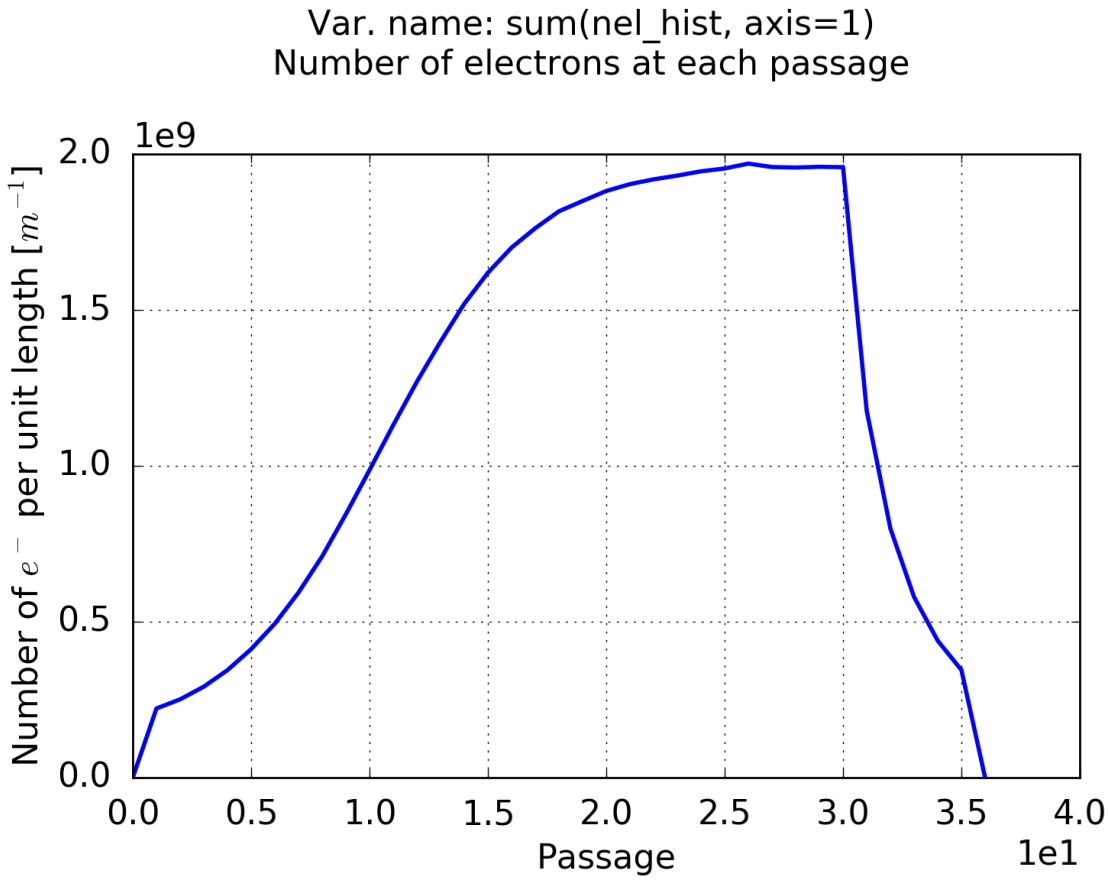

Var. name: N_mp_impact_pass
Number of macroparticles that impact for each passage
1e6
2.5
2.0
1.5
1.0
0.5
0.0
Number of impacting MP per passage
0
5
10
15
20
25
30
35
40
Passage

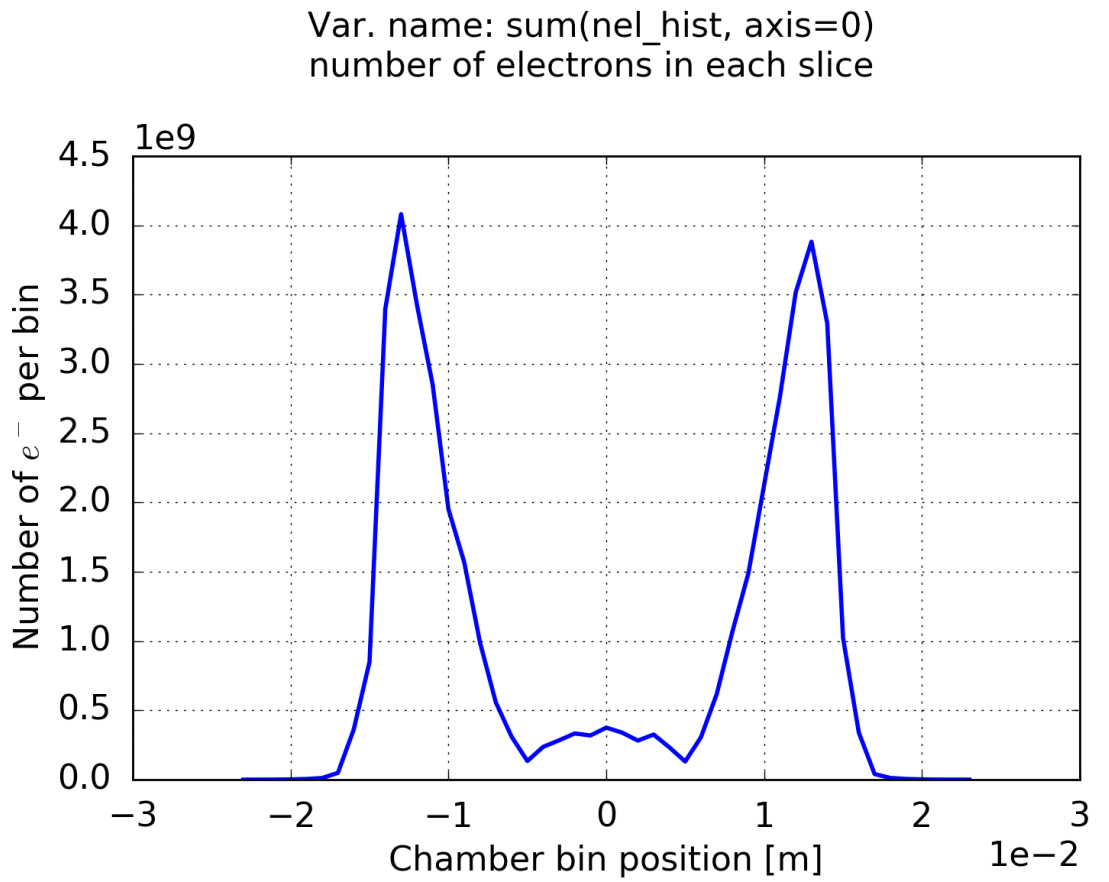
Var. name: N_mp_corrected_pass
Number of macroparticles for which fallback algorithm is used.

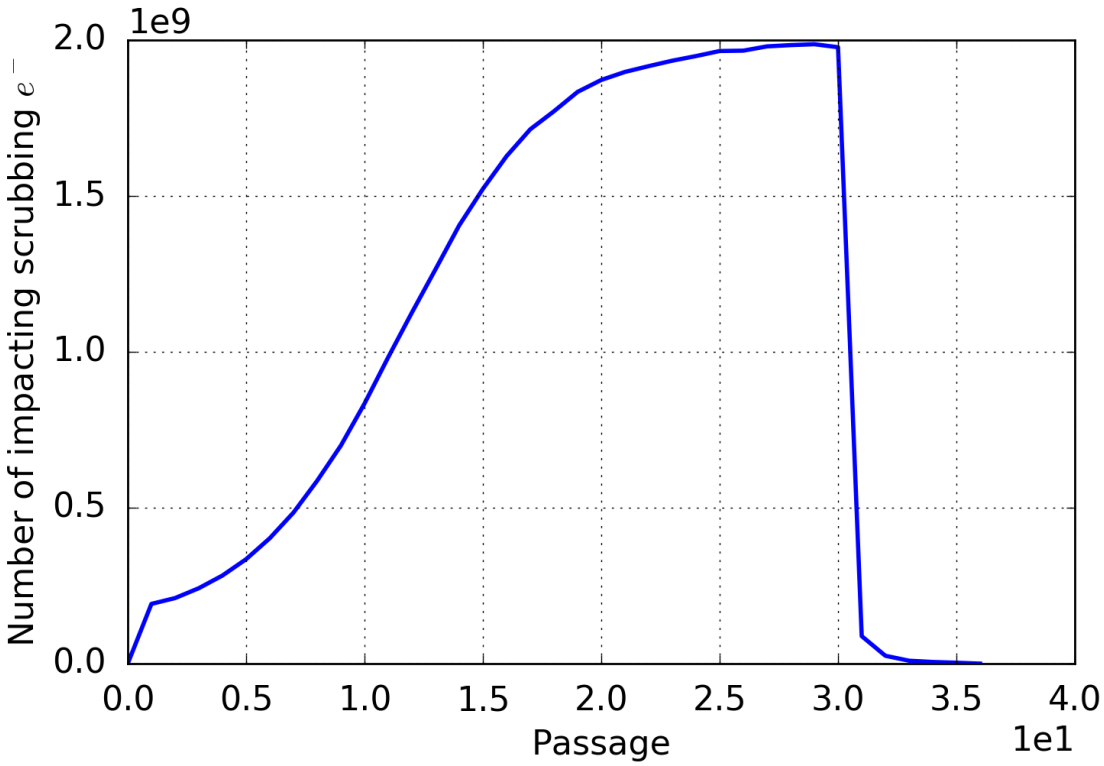


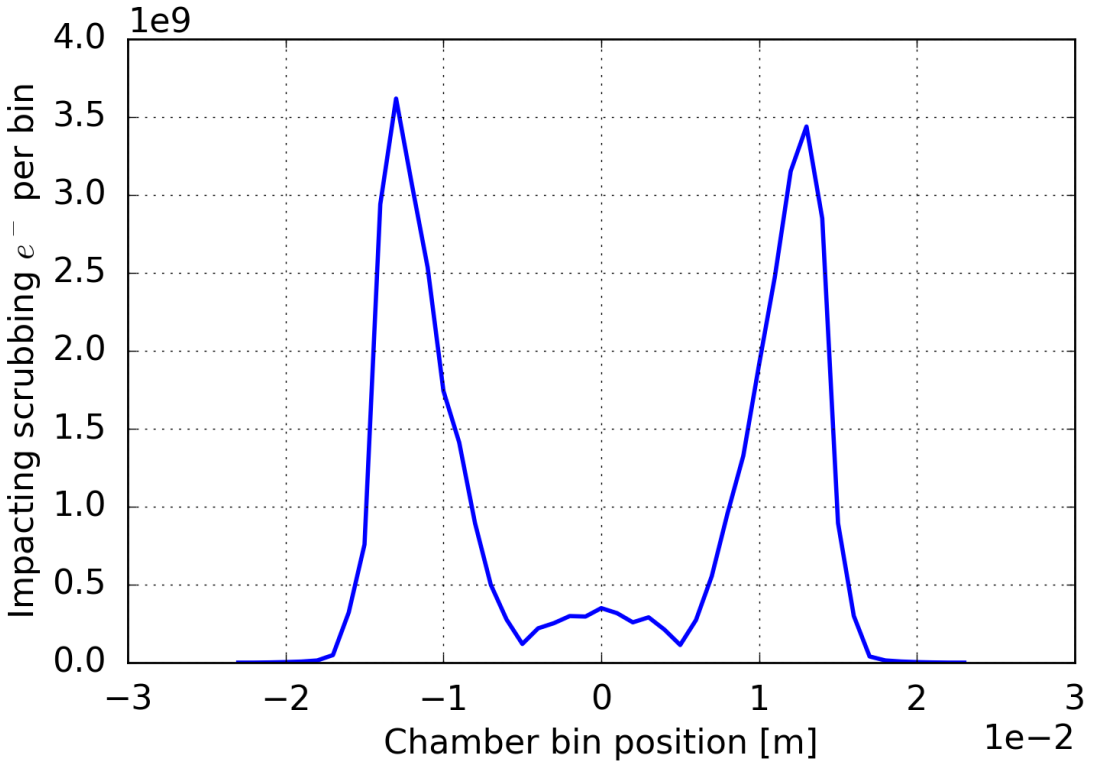
Var. name: N_mp_ref_pass
Reference macroparticle size at each passage

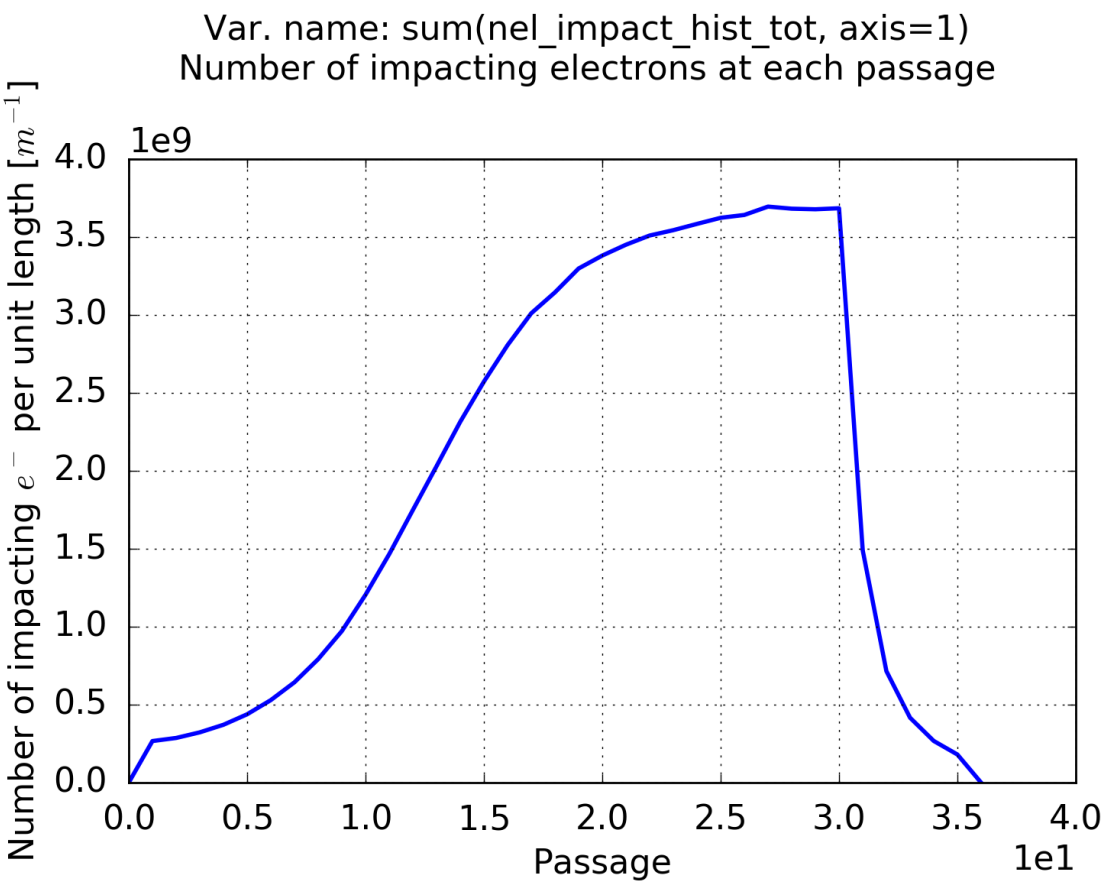


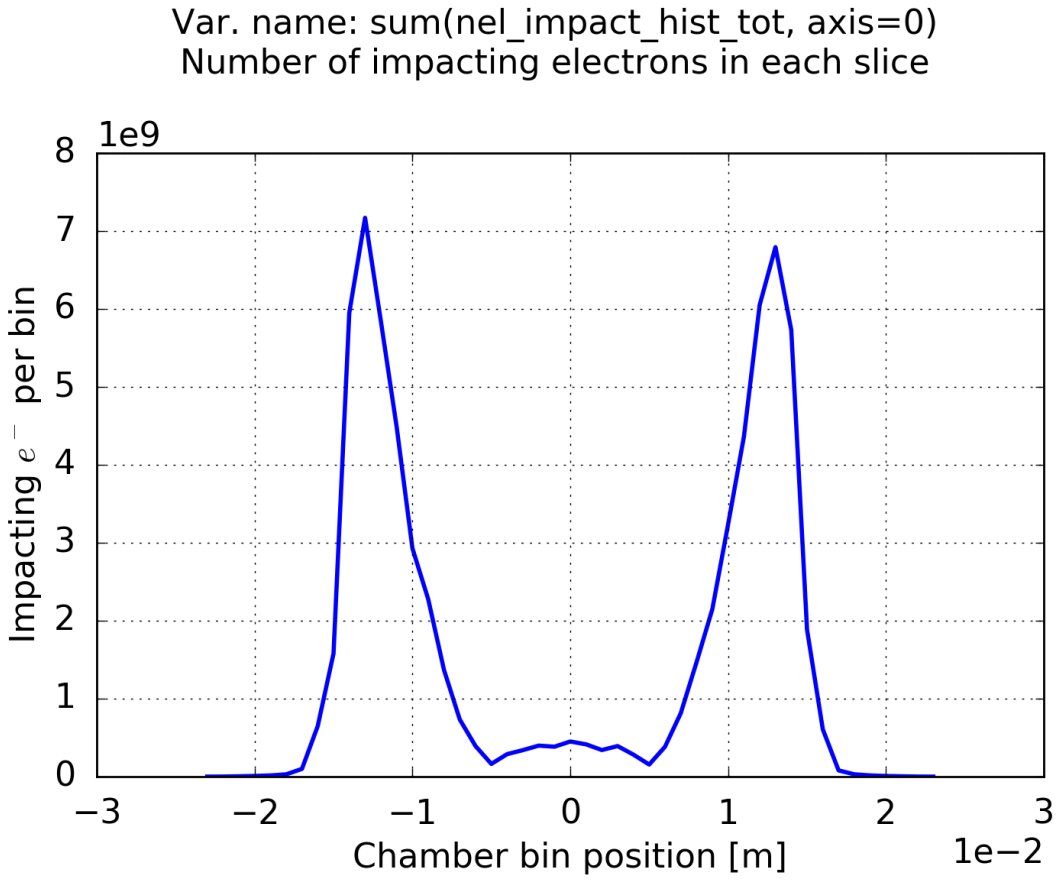

Var. name: sum(nel_hist, axis=1)
Number of electrons at each passage
1e9
2.0
1.5
1.0
0.5
0.0
Number of e^- per unit length [m^{-1}]
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Passage
1e1

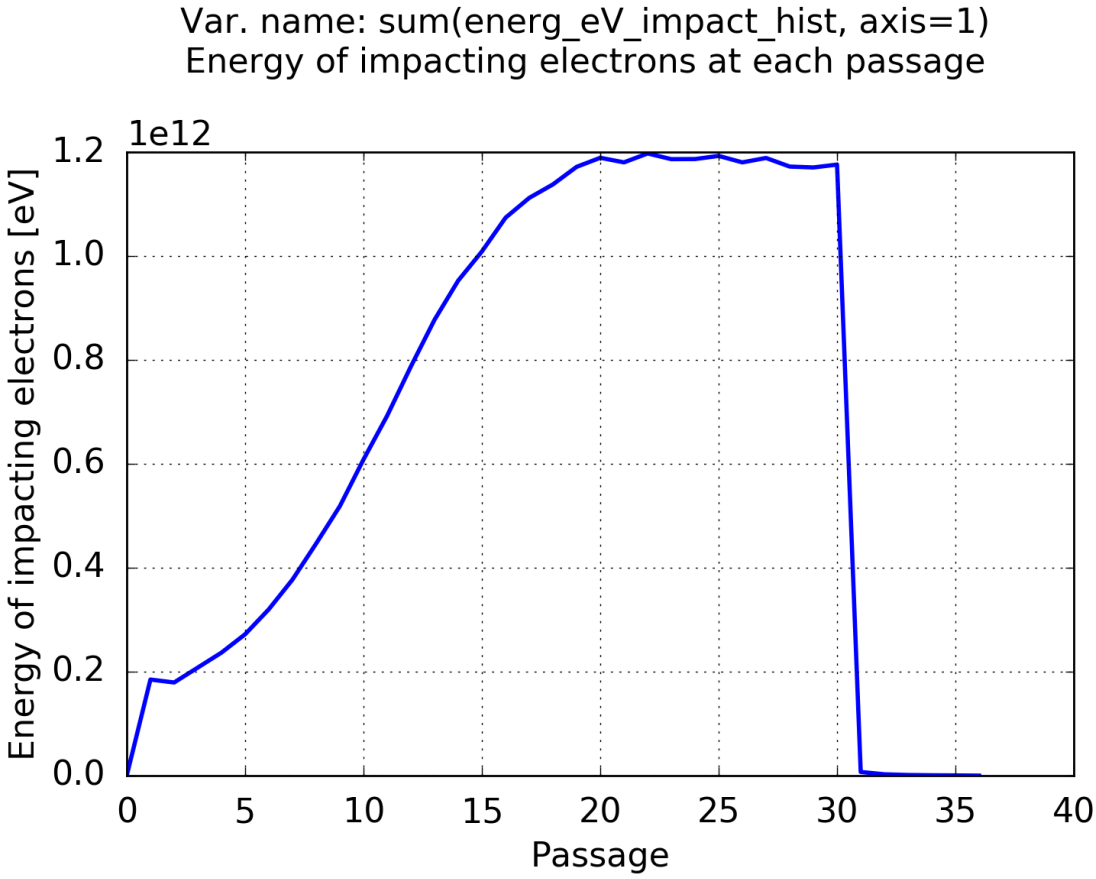

Var. name: sum(nel_hist, axis=0)
number of electrons in each slice
1e9
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Number of e^- per bin
−3
−2
−1
0
1
2
3
Chamber bin position [m]
1e−2

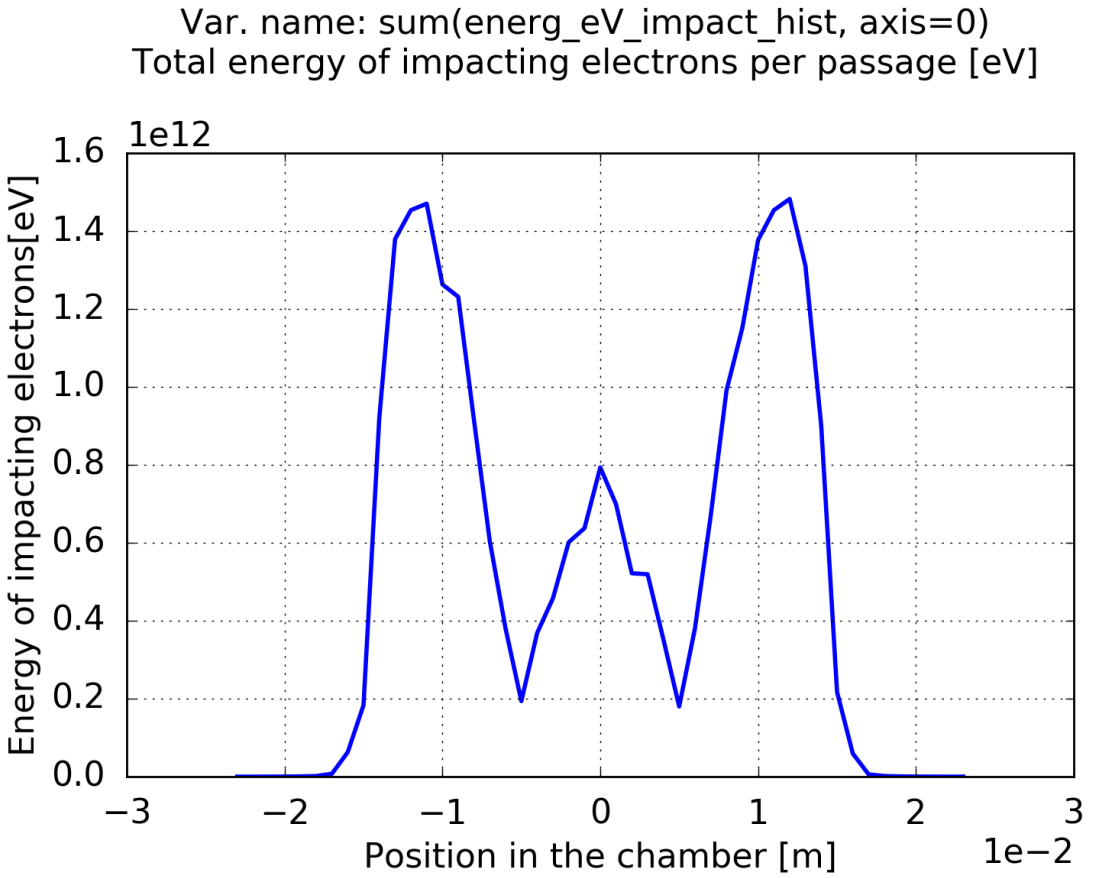

Var. name: np.sum(ob.nel_impact_hist_scrub, axis=1)
Number of impacting scrubbing electrons [E>E_scrub]
1e9
2.0
1.5
1.0
0.5
0.0
Number of impacting scrubbing e^-
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Passage
1e1


Var. name: sum(nel_impact_hist_scrub, axis=0)
Impacting scrubbing electrons in each slice
1e9
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Impacting scrubbing e^- per bin
−3
−2
−1
0
1
2
3
Chamber bin position [m]
1e−2


Var. name: sum(nel_impact_hist_tot, axis=1)
Number of impacting electrons at each passage
1e9
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Number of impacting e^- per unit length [m^{-1}]
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Passage
1e1


Var. name: sum(nel_impact_hist_tot, axis=0)
Number of impacting electrons in each slice
1e9
8
7
6
5
4
3
2
1
0
Impacting e^- per bin
−3
−2
−1
0
1
2
3
Chamber bin position [m]
1e−2


Var. name: sum(energ_eV_impact_hist, axis=1)
Energy of impacting electrons at each passage
1e12
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Energy of impacting electrons [eV]
0
5
10
15
20
25
30
35
40
Passage


Var. name: sum(energ_eV_impact_hist, axis=0)
Total energy of impacting electrons per passage [eV]
1e12
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Energy of impacting electrons[eV]
−3
−2
−1
0
1
2
3
Position in the chamber [m]
1e−2