ELECTRON CLOUD SIMULATIONS WITH PyECLOUD

G. Iadarola*, Università di Napoli Federico II , Napoli, Italy and CERN, Geneva, Switzerland
G. Rumolo, CERN, Geneva, Switzerland

Abstract

PyECLOUD is a newly developed code for the simulation of the electron cloud (EC) build-up in particle accelerators. Almost entirely written in Python, it is mostly based on the physical models already used in the ECLOUD code but, thanks to the implementation of new optimized algorithms, it exhibits a significantly improved performance in accuracy, speed, reliability and flexibility. PyECLOUD simulations have been already broadly employed for benchmarking the EC observations in the Large Hadron Collider (LHC). Thanks to the new feature of running EC simulations with bunch-by-bunch length and intensity data from machine measurements, the scrubbing process of the LHC beam pipes could be reconstructed from heat load measurements in the cryogenic dipoles. In addition, PyECLOUD simulations also provide the estimation of the bunch-by-bunch energy loss, which can be compared with the measurements of the stable phase shift.

INTRODUCTION

The Electron Cloud (EC) has been recognized as a possible limitation to the performances of the Large Hadron Collider (LHC). [1] In 2011, a one week scrubbing run was enough to lower the Secondary Electron Yield (SEY) of the LHC beam screens to values which allow an almost EC free operation with 50ns bunch spacing.

On the other hand Machine Development (MD) sessions with 25ns beams showed that a severe EC is still developing with this bunch spacing, the main observables being the heat load on the cryo-magnets, the dynamic pressure rise as well as detrimental effects on the beam, namely fast EC induced instabilities, which could be avoided with high chromaticity settings, and slower incoherent effects as particle losses, emittance growth and energy losses, especially on the last bunches of the injected trains.[2]

Analysis and predictions on the EC formation rely most entirely on numerical simulations. CERN has a long experience in the EC build-up simulation, mostly carried out with the ECLOUD code, developed and maintained at CERN since 1997. [3]

Unfortunately this code, written in FORTRAN 77, resulted not very suitable to be adapted to the length and the irregular structure of the beams employed in the MD sessions.

Therefore we have decided to write a fully reorganized code, in a newer and more powerful language, considering that the initial effort would be compensated by a significantly increased efficiency in future developments and debugging.

The new code has been called PyECLOUD, since it is almost entirely written in Python and is largely based on the physical models of the ECLOUD code. On the other hand, several features have been modified, and in some cases completely redesigned, with respect to ECLOUD, with substantial improvements in terms of reliability, accuracy, speed and usage flexibility. [4]

PyECLOUD

As ECLOUD, PyECLOUD is a 2D macroparticle (MP) code, where the electrons are grouped in MPs in order to reduce the computational burden.

The dynamics of the MP system is simulated following the stages sketched in Fig. 1.

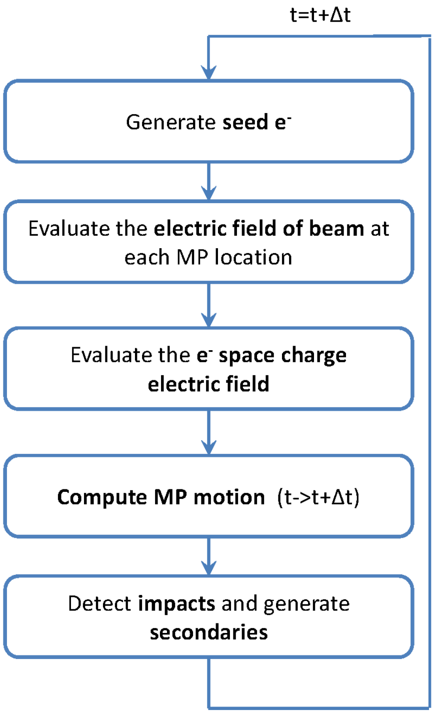

Figure 1: Flowchart representing PyECLOUD main loop.

At each time step Δt, *seed* electrons, due to residual gas ionization and/or to synchrotron radiation induced photoemission from the walls, are generated with the same time evolution of the beam and with transverse position and momentum determined by theoretical or empirical models.

Then the electric field acting on each MP is evaluated: the field of the beam is precomputed on a suitable rectangular grid, loaded from file and obtained at each MP location by a linear (4 points) interpolation; the *space charge*

---

[*] giovanni.iadarola@cern.ch

ISBN 978-3-95450-116-8

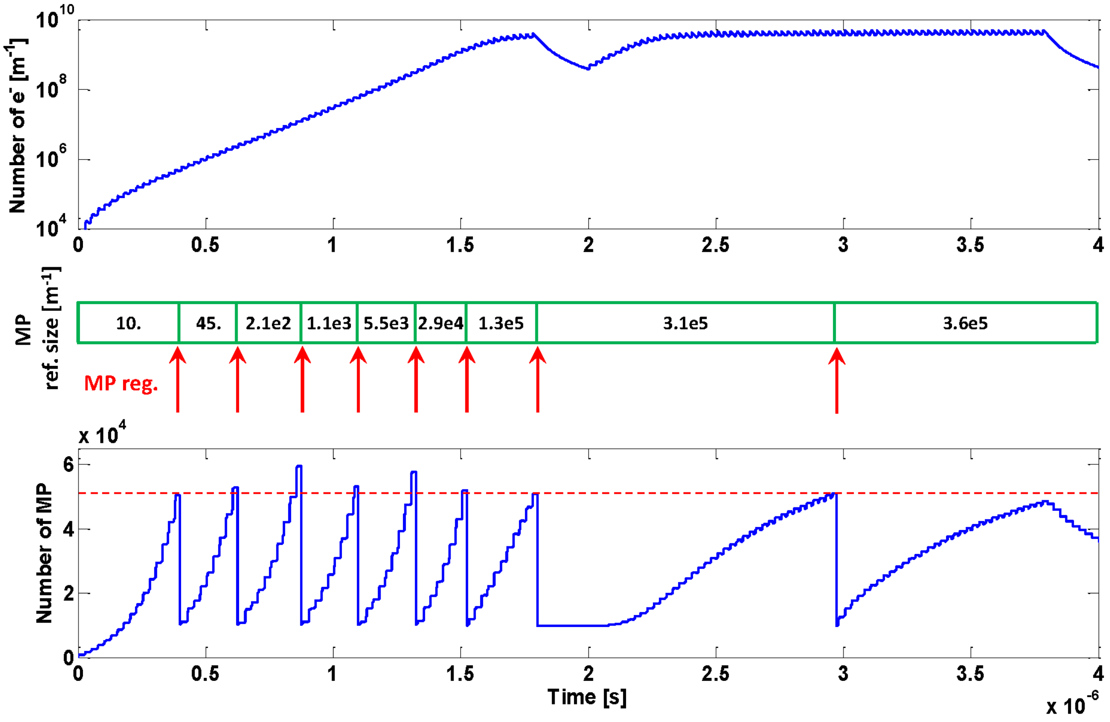

Figure 2: Top: evolution of the number of electrons in the beam pipe for a 25ns LHC beam in the SPS (2 trains of 72 bunches); middle: evolution of the reference MP size; bottom: evolution of the number of MPs, the regeneration threshold is highlighted in red.

contribution of the electron system itself is calculated by a classical Particle in Cell (PIC) algorithm, where the finite difference method is employed to solve the electrostatics equation with perfectly conducting boundary conditions.

Once the total electric field at each MP location is known, MP positions and momenta are updated by integrating the dynamics equation; at this stage the presence of an externally applied dipolar magnetic field can also be taken into account.

At each time step, a certain number of MPs can hit the wall. In these cases a proper model of the secondary emission process is employed to generate charge, energy and angle of the emitted electrons. According to the size of the emitted charge, a rescaling of the impinging MP can be performed or new MPs can be emitted.

MP Size Management

One of the peculiarities of the EC build-up process is the fact that, due to the multipacting effect, the electron number can spread several orders of magnitude (see Fig. 2 - top). As a consequence, it is impossible to choose a MP size which is suitable for the entire simulation, allowing both a satisfactory description of the phenomena and a computationally affordable number of MPs. The MP size management in PyECLOUD has been significantly improved with respect to ECLOUD and will be briefly described in this subsection.

MP sizes are not rigidly imposed but are determined by ”decisions” taken during the execution. For this purpose a reference MP size N_{ref}, adaptively changed during the simulation, is employed to control the number of electrons per MP. In particular:

- The size of MPs generated by *seed* mechanisms is exactly N_{ref};
- When a MP hits the wall, it is simply rescaled according to the SEY if the emitted charge is less than $1.5N_{ref}$, otherwise ”true” secondary MPs are generated so that the resulting MP size is as close as possible to N_{ref};
- Once per bunch passage, a cleaning procedure is performed, which deletes the MPs with charge smaller than $10^{-4}N_{ref}$.

N_{ref} is changed whenever the total number of MPs becomes larger than a certain threshold (typical value $\sim 10^5$), which means that the computational burden is becoming too high. When this happens, a *regeneration* of the set of MPs is performed, by the following procedure (see Fig. 2):

- Each MP is assigned to a cell of a uniform grid in the 5-D space (x, y, v_x, v_y, v_z) obtaining an approximation of the phase space distribution of the electron gas;

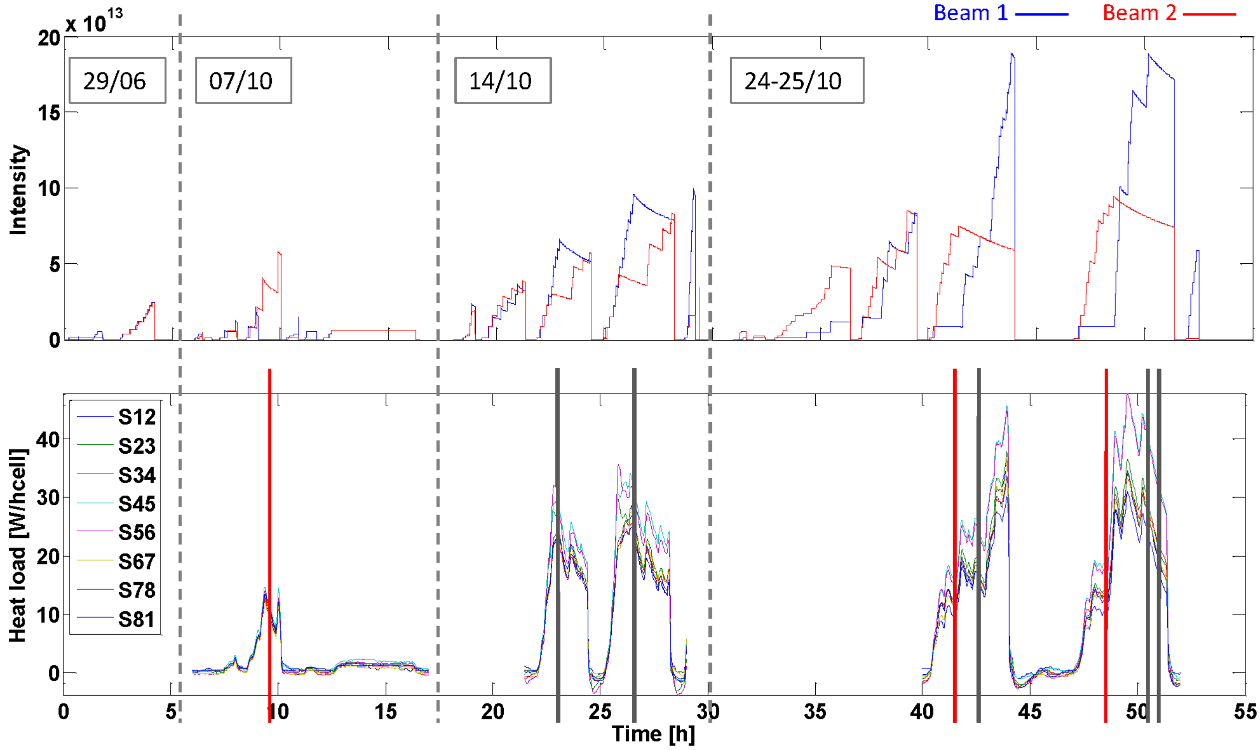

Figure 3: Beam intensity and heat load in the eight arcs of the LHC during the MD sessions with 25ns beams. The vertical bars represent the measurement points used to compare heat load with electron cloud simulations. The red vertical bars correspond to the measurement points in which only (or mainly) beam 2 was in the machine.

- The new N_{ref} is chosen in order to get a target number of MPs (typically 5-10 times smaller than the regeneration threshold), which still allows an accurate simulation but with a more reasonable computational effort;
- A new set of MPs, having the new reference size, is generated according to the computed distribution.

Several numerical test have shown that the errors on the total charge and the total energy which are introduced by this procedure, are never larger than 1-2%.

SIMULATIONS FOR THE LHC ARCS

PyECLOUD simulations have been extensively used to benchmark and analyze EC observations in the LHC. [4, 5, 6] In particular they have been employed for the estimation of the SEY of the copper coated beam screen of the cryogenic arcs.

The cryogenic system, provides data on the total power dissipated (in W/half-cell) on the beam screens of both beams 1 and 2. The heat load evolution during the MD sessions with 25ns in 2011 are shown in Fig. 3.

We know that the EC in the beam chamber at a certain time, and consequently also its effects, strongly depends on the beam structure at the same time. Therefore, in order to compare the heat load data with the simulation results, we have taken several cuts in time (marked as vertical lines in Fig. 3) and carried out EC simulations using the correct beam structures (measured bunch lengths and bunch intensities) at those times for both beam 1 and beam 2. This is crucial for a reliable estimation of the SEY, since, due to EC effects on the beam, both bunch intensities and bunch lengths are different along the train and strongly changing with time.

PyECLOUD simulations were run scanning the maximum of the SEY function ($\delta_{\max}$), so that the curves of the simulated heat loads $\Delta W^{(i)}_{\mathrm{b1-sim}}(\delta_{\max})$ and $\Delta W^{(i)}_{\mathrm{b2-sim}}(\delta_{\max})$, could be produced for each i-th measurement point. The electron reflectivity at zero energy was fixed to the value of 0.7 for all simulated cases. The solution $\delta^{*(i)}_{\max}$ for the i-th heat load measurement considered is then found from the equation:

$$\Delta W^{(i)}_{\mathrm{b1-sim}}(\delta_{\max}) + \Delta W^{(i)}_{\mathrm{b2-sim}}(\delta_{\max}) = \Delta W^{(i)}_{\mathrm{meas}} \quad (1)$$

This procedure allows to draw the evolution of $\delta_{\max}$ of the beam screens reported in Fig. 4. From this curve we can see that $\delta_{\max}$ was about 2.28 at the beginning of the scrubbing run, then it had already decreased to 2.1 by the time the first 25ns beams were first injected into the LHC, and to 1.52 thanks to the scrubbing tests with 25ns beams. Comparing these values with the EC threshold evaluated from simulations (red line in Fig. 4) it is possible to identify the

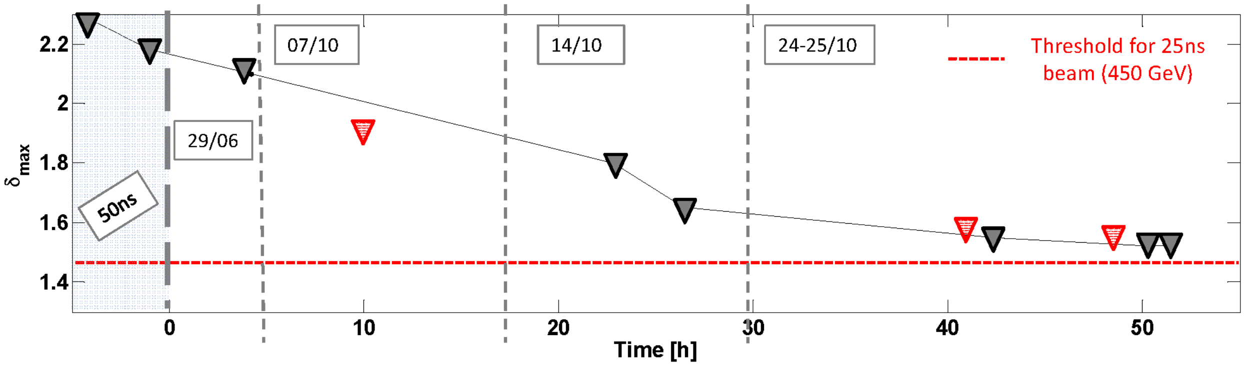

Figure 4: Estimated evolution of δ_{max} on the inner surface of the beam screen in the dipole chambers. The first two points correspond to the measurements done with the 50ns beams during and after the scrubbing run, while the other points corresponding to the vertical bars of Fig. 3.

further amount of scrubbing that is necessary to mitigate the EC effects at the injection energy.

The three points marked with red vertical bars in Fig. 3 and red triangles in Fig. 4 correspond to situations in which the total heat load could be only (or mainly) attributed to beam 2, allowing a separated estimation of δ_{max} for the beam screen of beam 2. They show that, while it seems plausible that at the beginning of the 25ns MDs the beam screen of beam 2 was more quickly scrubbed than that of beam 1, the conditioning status of the two screens has become later equalized, the last two points for beam 2 being hardly distinguished from the values obtained from the total heat load.

PyECLOUD also provides the calculation of the bunch-by-bunch energy loss per turn. This is based on a simple balance on the energy of the electron cloud. The difference between the total energy of the electrons (electrostatic and kinetic) before and after the bunch passage plus the energy lost in electron-wall collisions during the bunch passage represents the net energy transfered from the bunch to the electrons — and therefore, lost by the bunch. The bunch-by-bunch energy loss per turn thus calculated can be directly compared with the one estimated from stable phase shift measurements from the RF system [7].

The data acquired at the time of the last measurement from Fig. 3 have been plotted in Fig. 5 together with the simulated energy loss, as resulting from the PyECLOUD simulation. The best fit for the measured data has been found with $\delta_{\mathrm{max}} = 1.5$ (very close to the value estimated from the heat load) and a 10% uncaptured beam present in the gaps between trains and also in the abort gap.

The absolute values, as well as both the intra-batch and batch-to-batch trend, seem to be very well caught by the simulation. Furthermore, when zooming on single batches (Fig. 6), we can see that the simulations can successfully reproduce the measurements down to a surprisingly high level of detail.

The EC distribution calculated by PyECLOUD simulations (with δ_{max} estimated from heat load measurements) has been also employed for beam dynamics simulations with the HEADTAIL code in order to benchmark fast instabilities observed on 25ns beams in the LHC with low chromaticity settings. Details on the simulation method and results can be found in [6]. The good agreement obtained on the onset of the instability along the batch gives a further indication that our model and simulation code provide a good description of the EC formation in the LHC.

CONCLUSIONS

PyECLOUD is a new EC build-up simulation code, which has been developed for reliable and efficient benchmarking and analysis of the EC observations in LHC. The consistence of simulation results with heat load, stable phase-shift and instability observations in the machine provide strong indications on the reliability of the employed models and numerical solutions.

ACKNOWLEDGMENTS

The authors would like to express their gratefulness to G. Arduini, H. Bartosik, C. Bhat, V. Baglin, R. De Maria, O. Dominguez, M. Driss Mensi, J. Esteban-Muller, K. Li, H. Maury Cuna, G. Miano, E. Métral, H. Neupert, G. Papotti, E. Shaposhnikova, M. Taborelli, L. Tavian, C. Y. Vallgren, and F. Zimmermann for the support they provided in the code development, the simulation work and the machine data analysis as well as for valuable comments and discussions.

REFERENCES

[1] O. Bruning *et al.*, “LHC Design Report”, CERN, 2004

[2] G. Rumolo *et al.*, “LHC experince with different bunch spacings in 2011 (25, 50 and 75ns)”, Proceedings of the LHC Performance Workshop - Chamonix 2012, 6-10 February 2012, Chamonix, France

[3] G. Rumolo *et al.*, “Simulation of the electron-cloud build up and its consequences on heat load, beam stability, and diagnostics”, Phys. Rev. ST Accel. Beams 4 (2001), 1

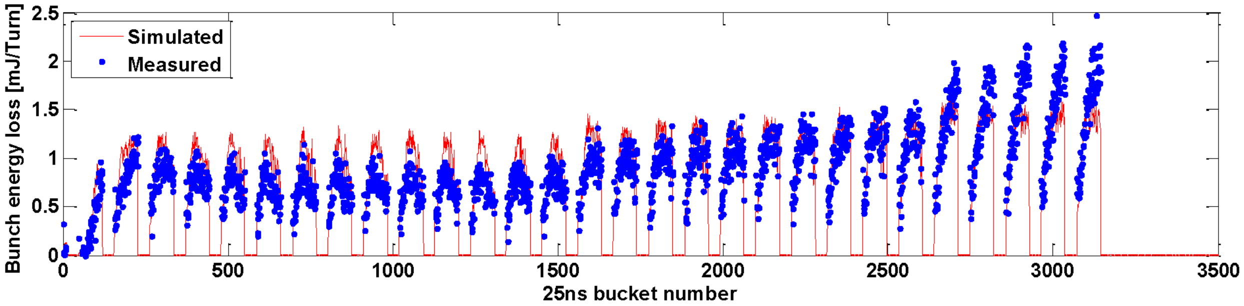

Figure 5: Bunch by bunch energy loss (for beam 1) at the time of the last measurement in Fig. 3.

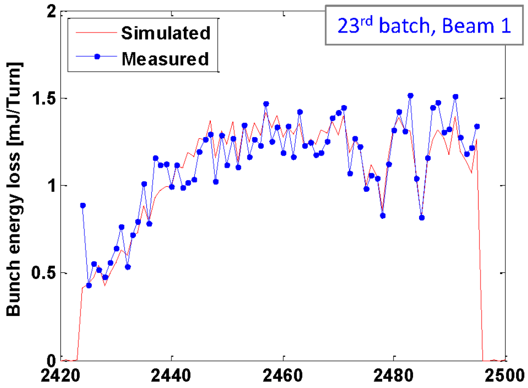

Figure 6: Close up on a selected batch in Fig. 5.

[4] G. Iadarola and G. Rumolo, “PyECLOUD and build-up simulations at CERN”, Proceedings of the ECLOUD12 workshop, 5-9 June 2012, La Biodola, Isola d’Elba, Italy

[5] H. Maury Cuna *et al.*, “Simulation of electron-cloud heat load for the cold arcs of the Large Hadron Collider”, Proceedings of the 3rd International Particle Accelerator Conference 2012, 20 - 25 May 2012, New Orleans, USA

[6] H. Bartosik *et al.*, “Benchmarking HEADTAIL with e-cloud observations with LHC 25ns beam”, Proceedings of the ECLOUD12 workshop, 5-9 June 2012, La Biodola, Isola d’Elba, Italy

[7] J. Esteban Muller *et al.*, “Synchronous Phase Shift at LHC”, Proceedings of the ECLOUD12 workshop, 5-9 June 2012, La Biodola, Isola d’Elba, Italy

ISBN 978-3-95450-116-8