

Influence of beam-induced wakefields on electron cloud buildup simulations

Sarah Wang^{1,2}, Konstantinos Paraschou², Lotta Mether²

¹ University of British Columbia,

² CERN BE-ABP-CEI Department

Abstract

Electron clouds form inside the vacuum chamber of the Large Hadron Collider when high-intensity, closely spaced proton bunches are being stored. Simulations with the PyECLOUD software can accurately predict the distributions of electrons when it has reached dynamic equilibrium. However, measurements based on the bunch-by-bunch energy loss indicate that the number of electrons do not grow as fast as in simulations. This project investigates whether the electromagnetic wakefields that are being induced by the interaction between protons and the resistivity of the chamber can influence the rate at which electrons multiply. We extend the PyECLOUD software capabilities to simulate the effect of wakefields forming in circular beam chambers, in the motion of electrons.

Keywords

electromagnetic wakefield; electron cloud; PyECLOUD.

1 INTRODUCTION

A beam can be defined by multiple single particles moving in the environment of a particle accelerator, which is characterized by electric and magnetic fields that control the movement of the beam. As the intensity of the beam increases, electromagnetic fields are self generated by the beam which can interfere with externally applied electric and magnetic fields. Subsequently, the beam can become unstable due to this interference. When a beam interacts electromagnetically with its surroundings to generate an electromagnetic field, it is known as a wake field. These wakefields can cause a chain effect as it affects the beam, which further perturbs the beams motion and generates more wakefields, resulting in a positive feedback loop known as collective instability [2]. Collective instability determines the ultimate performance of accelerator as it heavily contributes to beam loss. In this report, we investigate whether these wakefields can influence the buildup of electron clouds.

1.1 Multipactor effect

The multipactor effect is a phenomenon in radio frequency amplifier vacuum tubes where secondary electron emission can resonate with the oscillating electric field inside the beam, causing an electron avalanche [2]. An electron avalanche is the exponential multiplication of electrons due to free electrons accelerating quickly from the effects of the electric field, colliding with the pipe medium and causing electrons to be ionized, subsequently releasing more electrons, known as impact ionization [2]. Various factors can affect this including impact energy, number of electrons, angle, timing, etc. Ultimately, these effects can result in voltage breakdown and other deleterious effects.

1.2 Electron Clouds

Within an accelerator, closely spaced particle bunches often result in multipactor effects, which contribute to the formation of electron clouds (e-cloud). These effects also tend to affect positively charged particles more frequently, posing a significant performance limitation for proton beams. Circulating

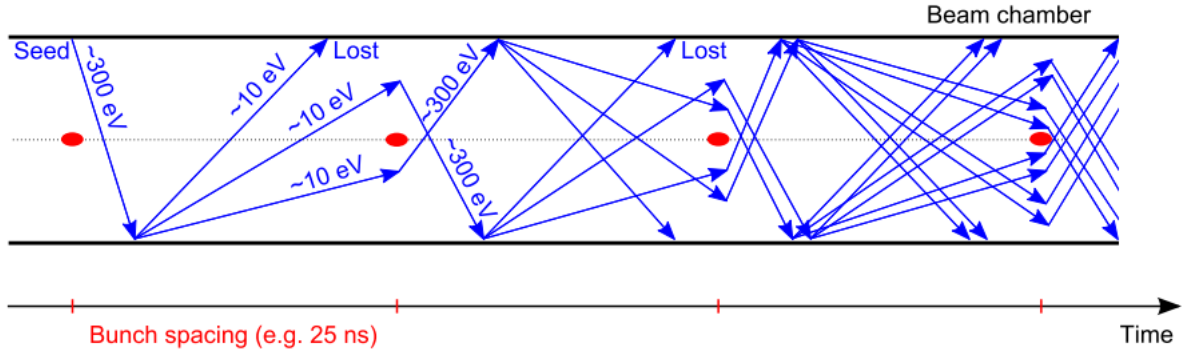


Figure 1: Mechanism of electron cloud formation [2]

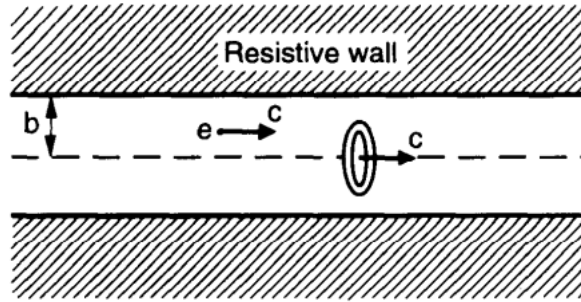


Figure 2: Beam ring possessing a multipole moment with $\cos m\theta$ distribution in a resistive beam pipe [1]

beam particles can produce electrons due to different mechanisms and are known as primary or seed electrons [2]. These seed electrons are attracted to the positively charged particle bunch and accelerated, causing them to crash into walls and emit secondary electrons. The secondary electrons in turn can be absorbed, reflected or produce more secondary electrons, resulting in an electron cloud [2].

1.3 Approximation of wakefields

Generally, in a perfectly smooth and conducting pipe, wake electromagnetic fields are not generated. However, if a pipe is not perfectly smooth, or it is smooth but non conducting, wakefields are generated behind the beam [1]. The beam is approximated as rigid and unaffected by the wake field it generates. In this paper, we investigate the simplest case of a smooth resistive pipe wall that is assumed to be infinitely thick.

The beam is also assumed to move at the speed of light and is given by the following distribution.

$$\rho_m = \frac{I_m}{\pi a^{m+1} (1 + \delta_{m0})} \delta(s - ct) \delta(r - a) \cos m\theta$$

where (r, θ, s) are the cylindrical coordinates with s in the direction of the beam, t is the time, a is the radius of the ring beam, and I_m is the m^{th} moment of the beam. The wakefield takes on a defined pattern and moves down the pipe with phase velocity c . In this project, we will only look at the simplest case of $m = 0$, where the beam is circular, but it is possible to use any type of distribution with enough moment terms used. The Maxwell equations for a test charge experience the EM fields at some fixed distance can be written in cylindrical coordinates and simplified to the following equations (for $m = 0$):

$$(E_r, E_s, B_\theta) = \cos m\theta \int_{-\infty}^{\infty} \frac{dk}{2\pi} e^{ikz} (\tilde{E}_r, \tilde{E}_s, \tilde{B}_\theta),$$

$$(E_\theta, B_r, B_s) = \sin m\theta \int_{-\infty}^{\infty} \frac{dk}{2\pi} e^{ikz} (\tilde{E}_\theta, \tilde{B}_r, \tilde{B}_s)$$

Where q is the elementary charge, c is the speed of light, σ is the conductivity and b is the radius of the pipe. Other assumptions are also utilized to determine that the field in the metal wall and outside are 0: the skin depth is much shorter than the pipe radius and the wall is infinitely wide [1]. Lastly, we make use of the long range approximation which is specified by the equation below:

$$|z| \gg b \left(\frac{c}{4\pi\sigma b} \right)^{1/3}$$

Essentially, the wakefield approximations will only hold in the range specified between the skin depth and long range approximation. Using these approximations, we derive the final equations in our simulation of the electron clouds.

$$E_s = \frac{q}{2\pi b} \sqrt{\frac{c}{\sigma}} \frac{1}{|z|^{3/2}} * \frac{1}{\sqrt{4\pi\epsilon_0}}$$

$$E_r = -\frac{3}{4} \frac{q}{2\pi b} \sqrt{\frac{c}{\sigma}} \frac{r}{|z|^{5/2}} * \frac{1}{\sqrt{4\pi\epsilon_0}}, \quad B_\theta = -\frac{3}{4} \frac{q}{2\pi b} \sqrt{\frac{c}{\sigma}} \frac{r}{|z|^{5/2}} * \sqrt{\frac{\mu_0}{4\pi}}$$

2 CODE AND SIMULATION

To accurately model e-cloud observations in the LHC, e-cloud simulations needed to be able to perform computationally extensive calculations including simulating thousands of particles in bunches as well as non uniform populations and lengths. E-cloud simulations also need to be able to handle less than ideal scenarios such as field free zones and dipoles, including quadrupoles and more complex setups. For this purpose, the original E-CLOUD code was developed. Newer code was eventually written to solve some of its issues; named after the original code, PyE-CLOUD utilized the same structures and physical models but was written in a more approachable and modern language, Python [2]. However, certain modifications were made as well including the numerical model – central modules such as MP size management, evaluation of electron space charge, MP tracker and interactions between electrons and the medium were overhauled [2].

2.1 PyE-CLOUD structure

PyE-CLOUD simulates e-cloud buildup in a thin disk at some point longitudinally along the beam pipe, where s is the coordinate along the pipe or the direction that the beam travels. In Figure 3, the workflow of the simulation is described. Seeds electrons from residual gas ionization and photoemissions are generated and grouped into MacroParticles or MPs to ease the computational load [2]. The electric field of the beam is evaluated at each MP location as well as the electron space charge contribution. The MP position and momentum can then be moved forward by one time step through integration of the equation of motion. At each time step, MPs can impact the wall, generating a model for the secondary emission process as well. This process determines the charge, energy and angle of the released electrons, and allows for the simulation to determine whether a re-scaling or creation of new MPs is necessary [2].

2.2 Wakefield class

PyE-CLOUD currently does not support the computation of wakefields, so it will be necessary to use the electric and magnetic fields derived above to model them in a new wakefield class. The wakefields are

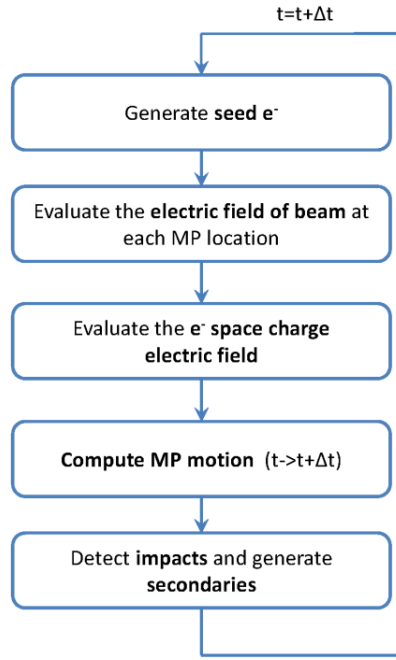


Figure 3: Flow chart of PyECLoud's main code loop [2]

calculated for all cylindrical components; however, the final electric and magnetic fields do not make use of the longitudinal component as it would require change to critical parts of the simulation. As mentioned above, the electric and magnetic fields are most easily derived in the cylindrical coordinate system first and converted to other coordinates afterwards. For a single particle, the below equations were derived:

$$E_s = A \frac{1}{|z|^{3/2}} * \frac{1}{\sqrt{4\pi\epsilon_0}}$$

$$E_r = -\frac{3}{4}A \frac{r}{|z|^{5/2}} * \frac{1}{\sqrt{4\pi\epsilon_0}}, \quad B_\theta = -\frac{3}{4}A \frac{r}{|z|^{5/2}} * \sqrt{\frac{\mu_0}{4\pi}}$$

where $A = \frac{q}{2\pi b} \sqrt{\frac{c}{\sigma}}$. However, for a particle bunch, it is necessary to sum all the electric and magnetic fields through an integral.

$$E_r = \int_{-\infty}^{\infty} \lambda(z') \frac{A}{|z - z'|^{5/2}} dz'$$

where $\lambda(z')$ represents the Gaussian profile for the bunch. The z_0 specified in the above integral represents the centre of each passing bunch. This integral represents the convolution of EM fields for any number of passing bunches. However, due to the divergence of the original functions for the electric and magnetic fields, it is necessary to represent them instead in terms of a piecewise function, where the section that is divergent can be instead "flattened" and set to the value of the wake function at an arbitrary constant value, alpha, which describes the z coordinate on the wake function. This is illustrated in Figure 4.

The effect of alpha can be observed in Figure 5 where the smaller alpha of 0.01 results in higher relative maximums for all the fields. This is in line with predictions as a smaller alpha means less of the original wake function is being cut off, resulting in higher convolutions near the maximums. The fields can also be seen to be more periodic rather than increasing.

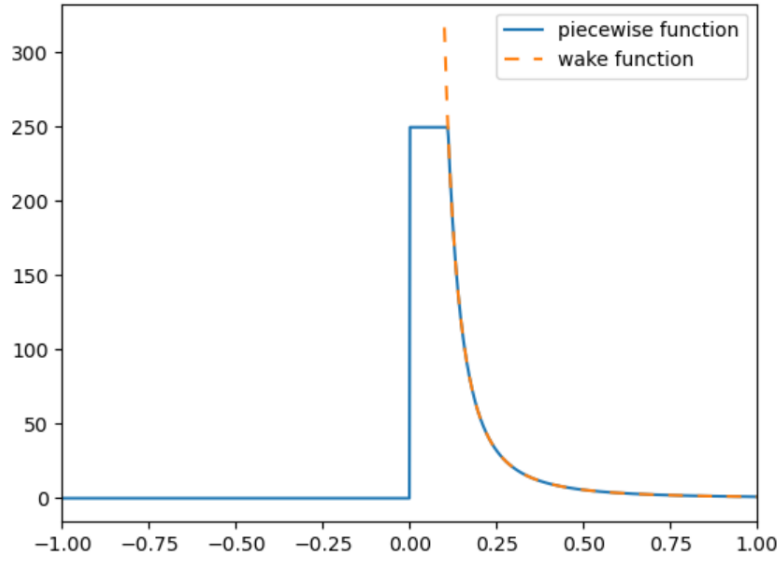
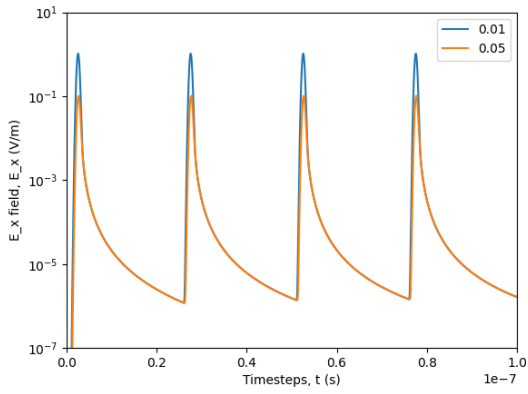
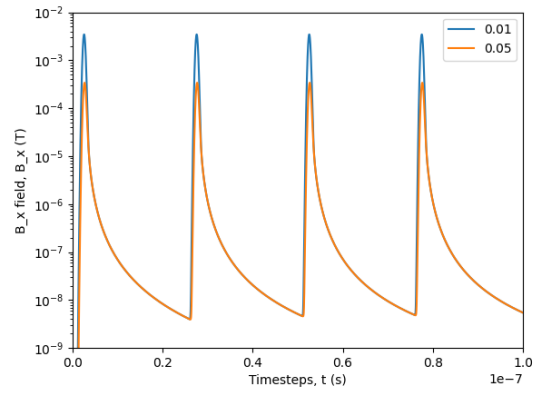


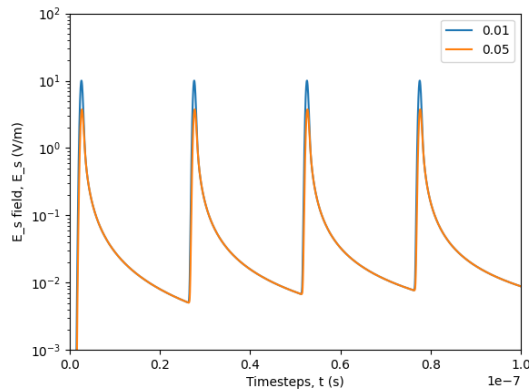
Figure 4: Piecewise function with an alpha of $z=0.05$



(a) E_x field



(b) B_x field



(c) E_s field

Figure 5: Comparison of multiple fields for $\alpha=0.01, 0.05$

2.3 Modification of other scripts

With the addition of the new *wakefield_class.py*, a flag for enabling wakefield functionality was created inside *machine_parameters.input* in addition to a resistivity variable, both of which were also added to *default_input_parameters.py*. Inside *init.py*, the wakefield class is initialized with input variables *beamtim.t* and *beamtim.lam_t_array*, which represent the time array and protons per time unit array respectively.

```
class wakefield:
```

```
    def __init__(self):
        self.e0 = sc.epsilon_0
        self.u0 = sc.mu_0
        self.conduct = 250 * 10**7 #conductivity (1/sigma*m)
        self.b = 0.044 #beam pipe radius (m)
        self.q = 1.6 * 10**11 * sc.elementary_charge #overall charge of beam (C)
        self.c = sc.speed_of_light #speed of light (m/s)
        self.x = np.linspace(0, 550, 10000)
        self.A = self.q/(2*np.pi*self.b)*np.sqrt(self.c/self.conduct)
        self.y = np.zeros_like(self.x)
        dx = self.x[1] - self.x[0]

        alpha = 0.05
        self.yw_32 = np.zeros_like(self.y)
        self.yw_52 = np.zeros_like(self.y)
        for i in range(len(self.y)):
            q = self.y[i] * dx # y is charge over time need to be multiplied by time
            self.yw_32 = self.yw_32 + q * self.z_32(self.x - self.x[i], alpha)
            self.yw_52 = self.yw_52 + q * self.z_52(self.x - self.x[i], alpha)

    def gaussian(self, x, mu, sigma):
        pw = np.piecewise(x, [np.abs((x-mu)/sigma) > 5],
            [0, lambda x: 1/(np.sqrt(2*np.pi) * sigma) * np.exp(-(x-mu)**2/(2*sigma**2))])
        return pw

    def z_52(self, z, alpha):
        pw = np.piecewise(z, [z <= alpha], [1./alpha**(5./2.), lambda x: 1./x**(5./2.)])
        pw[z <= 0] = 0
        return pw

    def E_r(self, x, y, z):
        func = np.interp(z, self.x, self.yw_52, left=0, right=0)
        return -3/4 * self.A * np.sqrt(x**2 + y**2) * 1/np.sqrt(4*np.pi*self.e0) * func
```

Lastly, the creation of the wakefields is done in *buildup_simulation.py*, where we define a function *_get_field_from_wakefields* that takes in the *x* and *y* cross sectional coordinates that are defined within the MP class. The function is later called and the electromagnetic wakefields produced are added to the overall electric and magnetic fields which includes those from the beams, space charge, etc. After running the simulation, it is possible to generate a variety of plots, the most important of which include a plot of the number of electrons in the chambers against time and one of the change in electric and/or magnetic fields within the beam at a specified location.

```

def _get_field_from_wakefields(self, MP_e, beamtim):
    z_curr = beamtim.tt_curr * sc.c
    x_mp = MP_e.x_mp[: MP_e.N_mp]
    y_mp = MP_e.y_mp[: MP_e.N_mp]
    Ex_n_wf = self.wf.E_x(x_mp, y_mp, z_curr)
    Ey_n_wf = self.wf.E_y(x_mp, y_mp, z_curr)
    Es_n_wf = self.wf.E_s(z_curr)
    Bx_n_wf = self.wf.B_x(x_mp, y_mp, z_curr)
    By_n_wf = self.wf.B_y(x_mp, y_mp, z_curr)

    return Ex_n_wf, Ey_n_wf, Es_n_wf, Bx_n_wf, By_n_wf

```

2.4 CST simulation

The accuracy of the wakefield capabilities was tested utilizing CST Studio, an electromagnetic field simulation software [3]. CST studio solves Maxwell's equations directly, serving as an extremely accurate comparison. In Figure 6, the plots produced by the PyECLoud software are plotted on top of the CST simulation. The blue and orange lines represent two different plots of the electric wake field using two different alphas. The red line is the electric field of the beam and the green is the simulated wake field by the CST simulation. At smaller times, we can see that the CST simulation differs from the PyECLoud simulation; the inaccuracies are due to the long range approximation used to derive the equations. However, the PyECLoud simulation does very well at larger times, meaning it is relatively accurate.

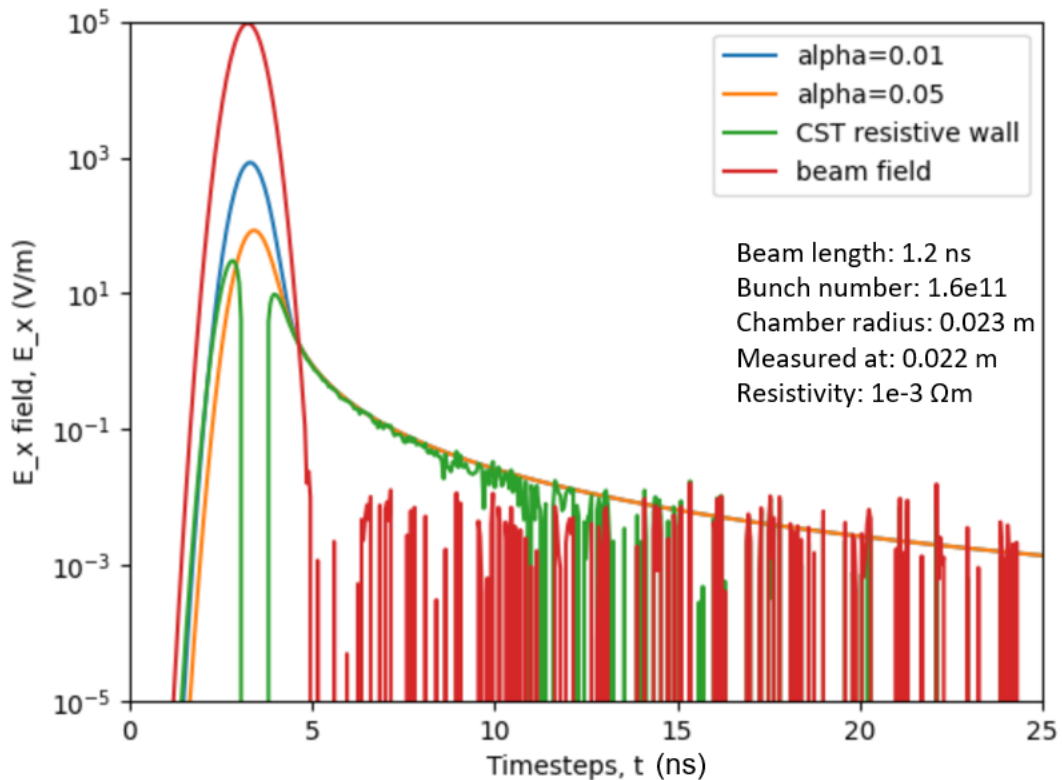


Figure 6: CST simulation

3 RESULTS AND DISCUSSION

After the wakefield capabilities were implemented, the simulation was run over a number of resistivities to test the effect of the electric and magnetic fields on the electrons. The parameters that were utilized for the simulations are given in Table 1. The number of electrons within the beam chamber is observed against time across two different radii (23 mm and 44 mm), three different resistivities (0, 4e1, and 4e2) and three different magnetic fields (drift, dipole, quadrupole).

Table 1: Parameters used for Figures 7, 8, 9 and A.1

Beam length (ns)	1
Number of protons/bunch	1.8e11
Pipe radius (mm)	23 44
Resistivities (Ωm)	0 (no wakefield) 4e1 4e2
Magnetic field	Drift (no magnetic field) Dipole Quadrupole
Alpha	0.05

The produced electric field points radially inwards, pushing the electrons outwards, which means there are two possible scenarios as a result: first, the electrons are able to reach the wall with enough energy to cause secondary electron emission due to the radial field pushing outwards, which would result in an increase in the final electron density. The second possible result is that the electrons hit the wall and are reabsorbed due to the electric field which ultimately causes a decrease in the final electron density. By observing the plots in Figure 7, the most likely scenario is the latter due to the fact that with increasing resistivity, the number of electrons decreases. This phenomenon is most clearly seen in the case with the larger radius, where there is a noticeable difference between the three different resistivities. The pattern is less obvious in the small radii, but still present with the slightly larger resistivity of 4e2 Ωm .

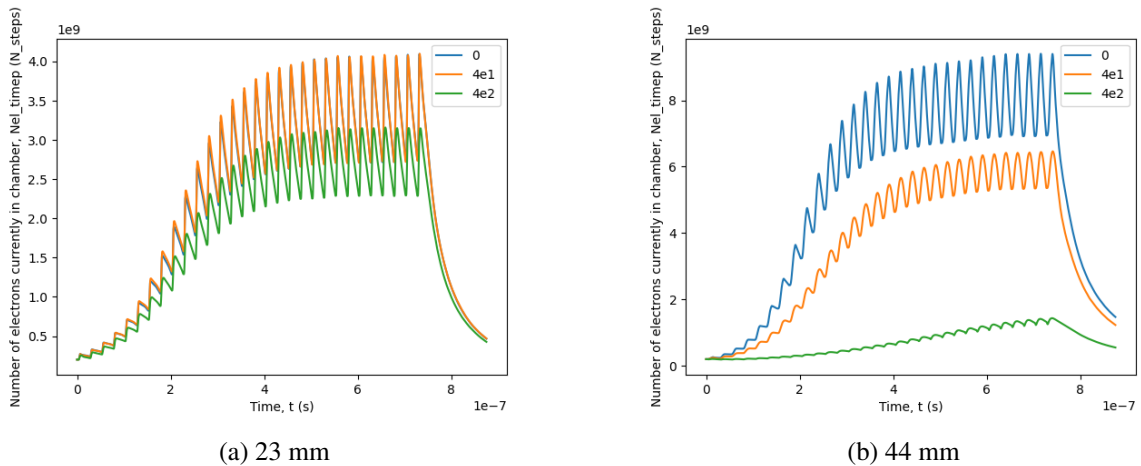


Figure 7: Dipole case

We observe an extremely similar trend in the quadrupole case as well despite the difference in topology and distribution of the electrons. Larger resistivities still result in lower electron multiplication,

and the effects are less evident in the case of the smaller radius.

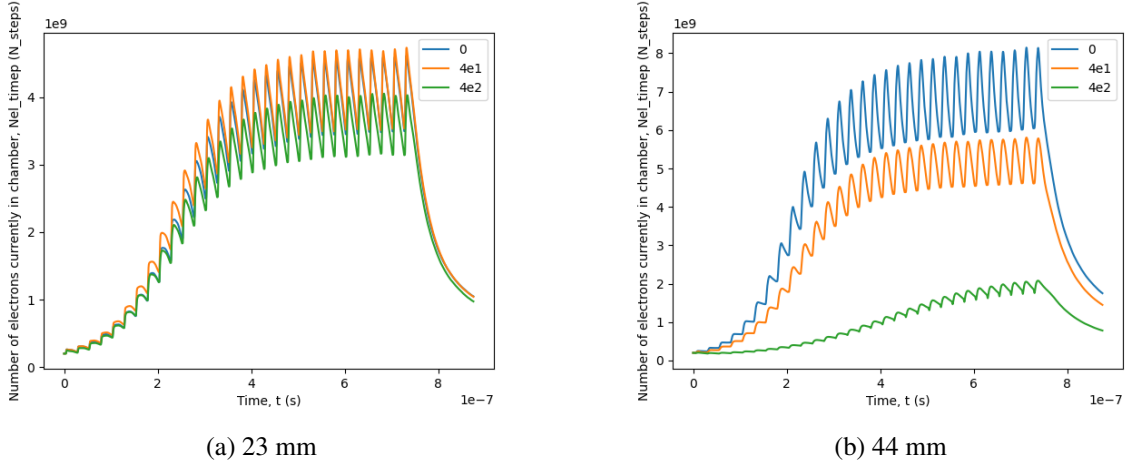


Figure 8: Quadrupole case

Lastly, for the drift case, we mostly see the same result in the case of the larger radii. However, an extremely unusual result is observed for the 23 mm radii, where with increased resistivity, we actually see increased electron growth. It is unclear why this occurs or differs from the other two resistivities.

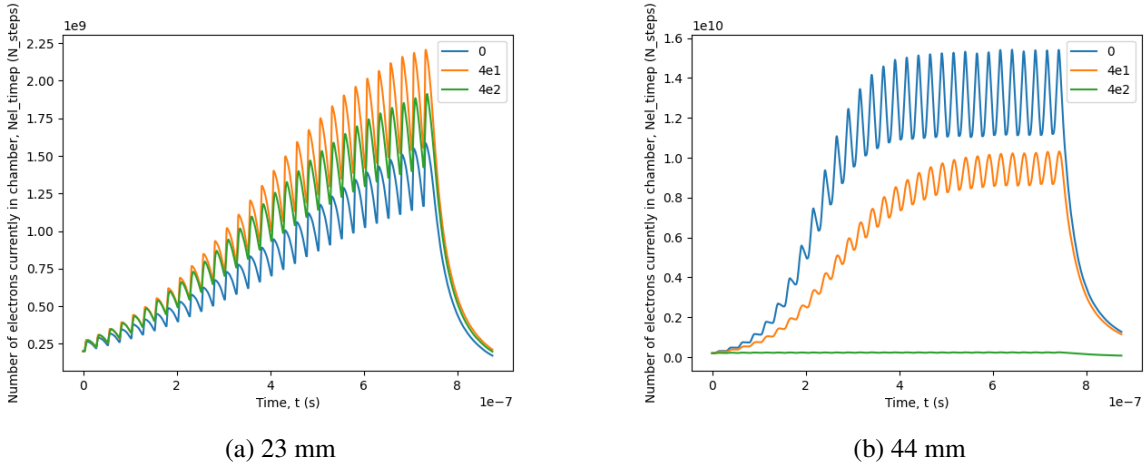


Figure 9: Drift case

4 CONCLUSION

The *wakefield_class.py* script allowed for the addition of long range resistive wall wakefield capabilities to the current capabilities of the PyELOUD software. The comparison to the CST simulation confirms the relative accuracy of the PyELOUD wakefield simulations at larger times. Comparisons at different radii and resistivities indicate that larger resistivity tends to mean less buildup, except for in the unusual case seen in the 23 mm radii for drift. Lastly, larger radii tend to demonstrate more apparent effects from change in resistivity

In the case of the LHC, the resistivity ($10^{-7}\Omega m$) is too low to see any significant wake field effects. However, some kicker magnets have materials with much larger resistivity, including the SPS MKPL which contains Cr_2O_3 with resistivity around $10^8\Omega m$, so there may be an effect in those cases.

Acknowledgements

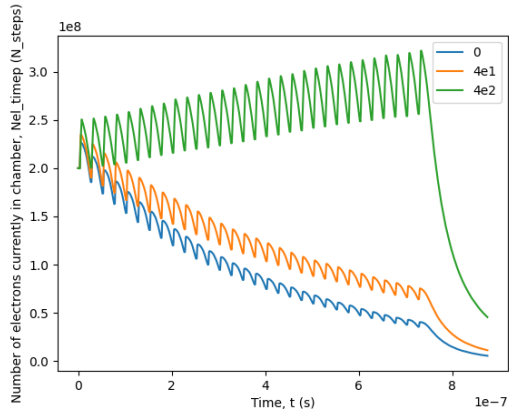
I am really grateful for all the support I received from the CERN BE-ABP-CEI department and in particular, my amazing supervisor Konstantinos Paraschou, who was an invaluable help to my project. Thanks to Chiara Antuono, Leonardo Sito, and Carlo Zannini for implementing the CST Studio simulations. I also thank the Institute of Particle Physics (IPP), the European Council for Nuclear Research (CERN), the Natural Sciences and Engineering Research Council of Canada (NSERC), for the monetary support that allowed me to travel to Geneva and participate in the CERN summer student program.

Bibliography

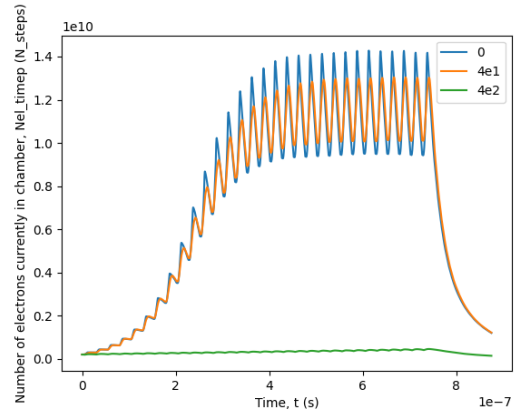
- [1] A.W. Chao, *Physics of Collective Beam Instabilities in High Energy Accelerators*, (Wiley, New York, 1993), Chap. 1-2.
- [2] G. Iadarola, *Electron cloud studies for CERN particle accelerators and simulation code development*, CERN, 2014.
- [3] SIMULIA, Dassault Systèmes, *CST Studio Suite*, San Diego: Dassault Systèmes, 2023.

Appendix

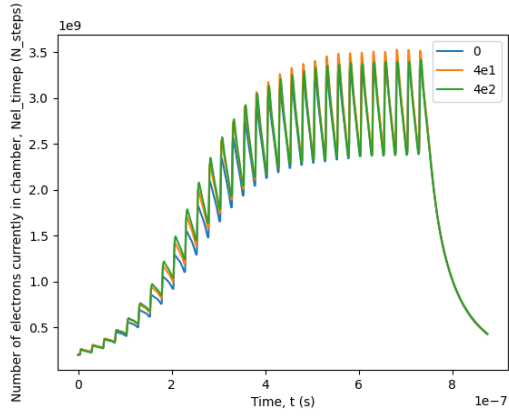
A Corresponding plots for higher beam intensity of $2.5e11$ protons



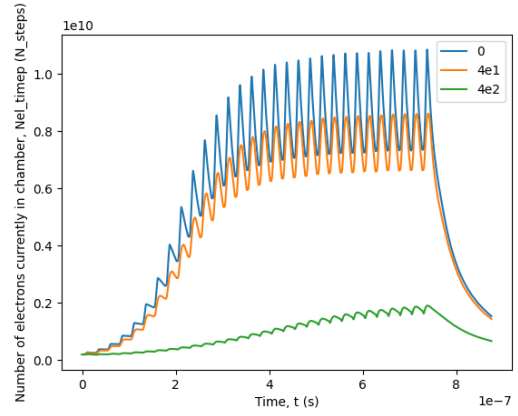
(a) Drift 23 mm



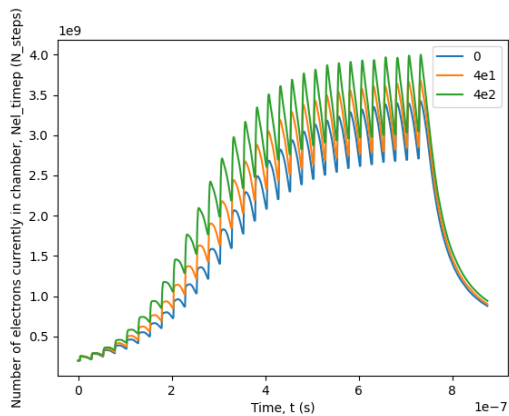
(b) Drift 44 mm



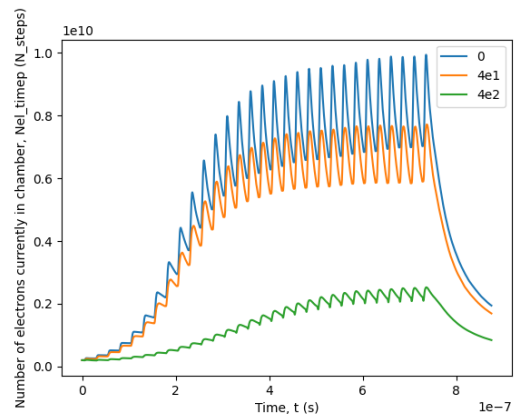
(c) Dipole 23 mm



(d) Dipole 44 mm



(e) Quadrupole 23 mm



(f) Quadrupole 44 mm

Figure A.1: 2.5e11 protons/bunch cases