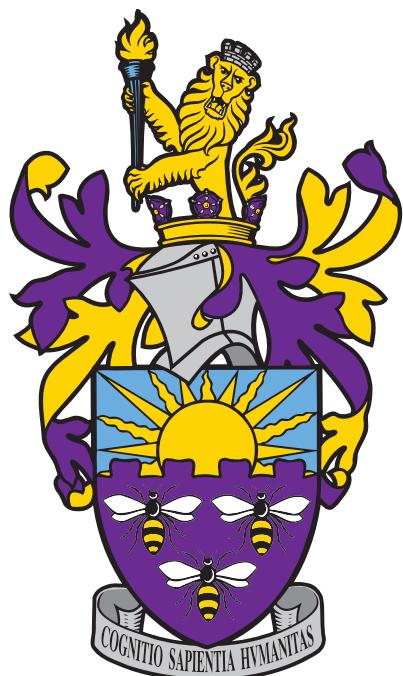


# A compact plasma beam dump for next generation particle accelerators

OSCAR JAKOBSSON



School of Physics and Astronomy  
*Cockcroft Accelerator Group*  
THE UNIVERSITY OF MANCHESTER  
Manchester, United Kingdom 2019

A compact plasma beam dump for next generation particle accelerators  
Oscar Jakobsson  
School of Physics and Astronomy  
The University of Manchester

## **Abstract**

Keywords: LWFA, PWFA, collective deceleration, plasma beam dump, EuPRAXIA

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Simulations of varying density profiles</b>	<b>3</b>
2.1	Plasma simulations . . . . .	3
<b>3</b>	<b>Semi-analytical model for beam dynamics</b>	<b>5</b>
3.1	Single-particle dynamics . . . . .	5
3.1.1	Longitudinal and transverse wakefields . . . . .	5
3.1.2	Particle-push algorithm . . . . .	6
3.1.3	Semi-analytical model . . . . .	8
3.2	Beam dynamics . . . . .	11
3.2.1	Static beam approximation . . . . .	11
3.2.2	Dynamic beam density . . . . .	14
<b>4</b>	<b>Ongoing work and concluding remarks</b>	<b>17</b>
	<b>Bibliography</b>	<b>17</b>

# 1

## Introduction

This report details the work conducted in the second half of a Master's project in the Novel Accelerator group at the University of Manchester. This work has not been done in isolation. Work has been done in collaboration with the members of the plasma wakefield acceleration group. However, the results and ideas presented in chapter two however are mine unless otherwise stated. The algorithm presented in chapter 3 has been developed in close collaboration with Dr. Bonatto. Having said that, the idea, general structure and development of the algorithm is of my working. In this chapter, care has been taken to emphasize the work done, as well as helpful input and crucial insights, by Dr. Bonatto.

## 1. Introduction

# 2

## Simulations of varying density profiles

### 2.1 Plasma simulations

## 2. Simulations of varying density profiles

---

# 3

## Semi-analytical model for beam dynamics

The details of PIC codes was described in detail in the previous report, see chapter 2 of the previous report for further details. Conventional PIC codes will simulate millions of macroparticles and update all their positions while considering their mutual Coloumb interactions. These also include collisions effects, ionisation, synchrotron and bremsstrahlung and many additional physical effect. For our purposes however we simply want a quick model to give us a rough idea of the behaviour of these varying density schemes that we are considering.

### 3.1 Single-particle dynamics

#### 3.1.1 Longitudinal and transverse wakefields

The equations governing the behaviour of the plasma under the influence of an electron beam were derived from first principles in the previous report. We include them below in the form most commonly used in today's literature. The angular plasma frequency

$$\omega_p = \sqrt{\frac{4\pi e^2 n_p}{m_e}} \quad (3.1)$$

gives the frequency of oscillations of the plasma waves as a beam passes through the plasma. The longitudinal force can be shown to be given by

$$\frac{E_{zb}(\xi, r)}{E_0} = -k_p^3 \int_{\infty}^{\xi} d\xi' \cos [k_p (\xi - \xi')] \int_0^{\infty} dr' r' I_0(k_p r_{<}) K_0(k_p r_{>}) n_b(\xi', r') / n_0 \quad (3.2)$$

where  $k_p = \omega_p/c$  is the plasma wavenumber. The longitudinal (displacement variable)/(convolution variable)  $\xi'$  in our Green's function is integrated across the interval  $\xi' \in [\infty, \xi]$  to emphasise the fact that the wakefield at a position  $\xi$  behind the beam can only be caused by interactions of particles in front of that positions due to causality. The Panofsky-Wenzel theorem states that the transverse field corresponding to a given longitudinal field is given by

$$W(\xi, r) = \int_{\infty}^{\xi} \frac{\partial E_z(\xi, r, z)}{\partial r} dz \quad (3.3)$$

which upon differentiation and integration yields

$$\frac{W_{\perp}(\xi, r)}{E_0} = -k_p^3 \int_{\infty}^{\xi} \sin(k_p(\xi - \xi')) d\xi' \int_0^{\infty} \frac{\partial}{\partial r} (I_0(k_p r_{<}) K_0(k_p r_{>})) \frac{n_b(\xi', r') / n_0}{\partial r'} r' dr' \quad (3.4)$$

Since the aim of this model is to model the beam dynamics in a varying plasma density we need to include its effect on the induced wakefield, as described by equations 3.2 and 3.4. The plasma

density only enters the above equations through the wavenumber  $k_p$ , which both scales the overall amplitude of the wakefield as well as determines the rate of oscillations and decay of the wakefield through its inclusion in the trigonometric and Bessel functions, respectively. Hence by letting the plasma density  $n_p \mapsto n_p(z)$  as a function of the propagation distance  $z$  the wavenumber will vary at each longitudinal position as well,  $k_p \mapsto k_p(z)$ , which in turn adds a propagation distance dependence in the wakefield equations:  $E_z(\xi, r, z)$  and  $W_\perp(\xi, r, z)$ . These expressions can be solved numerically in software such as Mathematica, but the calculation is time-consuming and often approximations are used. Before going into detail about how we approximate and compute these expressions it is necessary to describe the algorithm in which aim to use them to understand the degree of approximation necessary.

### 3.1.2 Particle-push algorithm

The first simplification used in this algorithm is to consider the dynamics of a test particle inside the bunch under the influence of the wakefield set up by the electron bunch as it propagates through a plasma. We assume that the shape and density of the beam is static as it propagates through the plasma, thereby removing the need to constantly having to update the beam density  $n_b$  in the wakefield integrals. Furthermore, we assume that the only forces acting on the test-particle are the electric fields of the wakefields themselves, thereby ignoring the Coulomb repulsions within the bunch, synchrotron and betatron radiation etc. Hence to determine the dynamics of we simply need to place it at a position  $(\xi_i, r_i)$  inside the bunch and its wakefield, with an energy equal to that of the particle in the bunch, and calculate the forces acting on it at that positions. Then, by choosing a sufficiently short time-step  $\Delta t$  that we allow these forces to act on the particle, we can calculate a new positions  $(\xi_{i+1}, r_{i+1})$  that the particle will have moved to during this time. By repeating this process we should be able to determine how this particle will behave as it gets decelerated and falls behind the bunch. By further increasing the density at each iteration we should be able to investigate what happens when this particle enters the defocusing region.

This approach is inspired by the particle-push and field-solver algorithms underlying all PIC codes. These algorithms, described in detail in chapter 2 of the previous report, only act as inspiration for this model and are not necessary to understand this model. The method by which we perform the iterations in this algorithm are described in this section.

Firstly, a position  $(\xi_0, r_0)$  inside the bunch is chosen as a starting point for our test particle. The energy of this particle is then set to equal that of the particles in the beam, e.g 1 GeV to allow for comparison with the simulations in chapter 2. We then set the size of the time-step  $\Delta t$  used to discretize the particle motion. To calculate the particles new position  $(\xi_1, r_1)$  after time  $\Delta t$ , or more generally any new position  $(\xi_{i+1}, r_{i+1})$  given a current position  $(\xi_i, r_i)$ , we first need to calculate the longitudinal and transverse electric field at this position,  $E_z(\xi_i, r_i, z)$  and  $W_\perp(\xi_i, r_i, z)$ , where  $z$ , the distance the beam has travelled through the plasma  $z = i \cdot v_{beam} \Delta t$  is used to determine the density at that position given a density profile  $n_p(z)$ . The speed  $v_{beam}$  of the particles in the ultra-relativistic beam is given by kinetic energy  $T_{beam} = (\gamma - 1)m_e c^2$  as

$$v_{beam} = c \sqrt{1 - \left(1 + \frac{T_{beam}}{m_e c^2}\right)^{-2}} \quad (3.5)$$

which is very close to the speed of light for our 1 GeV bunch, but nevertheless must be calculated as it is crucial for subsequent steps of the algorithm. The same equation can be used to updated the longitudinal speed of the test particle from  $v_z(i)$  to  $v_z(i+1)$  if the updated kinetic  $T_e(i+1) = T_e(i) + \Delta T_e(i)$  can be found. The kinetic energy change of the particle  $\Delta T_e(i) = T_e(i+1) - T_e(i)$  between iterations  $i$  and  $i+1$  is given by work done on the particle by the longitudinal wakefield during time  $\Delta t$ . This work, in turn, is given by integrating the force exerted on the particle over

the distance  $v_z(i)\Delta t$  that the particle moves during the iteration. However, since we assume that the force is constant in each iteration we have that

$$\Delta T_e(i) = -eE_z(\xi_i, r_i, z_i) v_z(i)\Delta t \quad (3.6)$$

which we use to updated the longitudinal speed to

$$v_z(i+1) = c\sqrt{1 - \left(1 + \frac{T_e(i) + \Delta T_e(i)}{m_e c^2}\right)^{-2}} \quad (3.7)$$

Even though reference is made to the co-moving variable  $\xi$  in the above calculation, the calculation itself is performed in the lab frame where the beam has an energy of 1 GeV etc. However, to calculate at what longitudinal position  $\xi_{i+1}$ , with respect to the beam and its wakefield, the particle will end up after time  $\Delta t$  we now move into the co-moving reference frame. Given that we know the force acting on the particle and the initial velocity, the updated position is given by relativistic equivalent of the standard  $x = vt + at^2/2$  formula. From relativistic kinematics we found that this is given by

$$\xi_{i+1} = \xi_i + \frac{v_z(i) - v_0}{1 - v_z(i)v_0/c^2}\Delta t - \frac{c^2}{a_z} \left(1 - \sqrt{1 + \frac{a_z^2 \Delta t^2}{c^2}}\right) \quad (3.8)$$

where the second term gives the relative velocity  $v'_\xi$  between the particle and the wakefield in the co-moving frame and  $a_z$  is the acceleration in the longitudinal direction. If the particle is moving relativistic in co-moving frame, i.e when falling behind the bunch the speed of the particle relative to the wakefield is  $-c$ , the relativistic acceleration expression needs to be used. The force in the co-moving (primed) frame

$$\mathbf{F}' = \frac{d\mathbf{p}'}{dt} \quad (3.9)$$

now needs to include the relativistic momentum  $\mathbf{p}' = m_e\gamma(\mathbf{v}')\mathbf{v}'$ , where  $|\mathbf{v}'| = \sqrt{v'^2_\xi + v_r^2}$  is the speed of the particle in the co-moving frame, which gives that the acceleration in the co-moving frame

$$\mathbf{a}' = \frac{1}{m_e\gamma(|\mathbf{v}'|)} \left( \mathbf{F}' - \frac{(\mathbf{v}' \cdot \mathbf{F}')\mathbf{v}'}{c^2} \right) \quad (3.10)$$

Taking the longitudinal component of this then gives the longitudinal relativistic acceleration

$$a_z(i) = -\frac{eE_z(\xi, r, z)}{m_e\gamma(|\mathbf{v}'|)^3} \quad (3.11)$$

which due to the gamma term is only of concern when the relative velocity between particle and wakefield in the co-moving frame is large. Using this expression in equation 3.8 we now update the co-moving position to  $\xi_{i+1}$ , thereby completing the longitudinal push.

Due to the rapid variation in the radial position and velocity the kinetic energy approach has proven inaccurate for too long time steps, since  $v_r$  might be negative in one iteration and positive in the next, providing inaccurate descriptions of the distance the transverse wakefield has exerted force on the electron. For this reason we compute the

$$\Delta v_r(i) = \frac{a_r(i)\Delta t}{\sqrt{1 + \frac{a_r^2(i)\Delta t^2}{c^2}}} \quad (3.12)$$

where the transverse acceleration is given by the transverse component of equation XX.

$$a_z(i) = -\frac{eE_z(\xi_i, r_i, z_i)}{m_e\gamma(|\mathbf{v}'|)^3} \quad (3.13)$$

and the updated velocity is given by relativistic velocity addition

$$v_r(i+1) = \frac{v_r(i) + \Delta v_r(i)}{\sqrt{1 + \frac{v_r(i)\Delta v_r(i)}{c^2}}} \quad (3.14)$$

The updated radial positions is then given by the radial equivalence of equation XX.

$$r_{i+1} = r_i + v_r(i)\Delta t - \frac{c^2}{a_r(i)} \left( 1 - \sqrt{1 + \frac{a_r^2(i)\Delta t^2}{c^2}} \right) \quad (3.15)$$

By placing a test particle at positions in the beam we can then track the behaviour of this particle under the influence of the wakefield described by the the linear fluid model as the density changes. This of course neglects all Coloumb repulsion for particles in the beam, synchrotron radiation emission etc. However, to model the dynamics of this particle under the extremely strong electric field set up by the wakefield one needs to use a small step size in the algortihm to avoid pushing the particle too far or accelerating it for too long. To calcualte these forces the wakefield integrals need to be solved at each iteration step, which is a computationally expensive procedure that would ultimately ruin this algorithm if not simplified substantially.

### 3.1.3 Semi-analytical model

In order to calculate the wakefield induced by a bi-Gaussian beam with density

$$n_b(\xi, r) = n_{b_0} e^{-\frac{\xi^2}{2\sigma_\xi^2} - \frac{r^2}{2\sigma_r^2}} \quad (3.16)$$

we need to compute the wakefield integrals 3.2 and 3.4. These integral can be separated into integration over the co-moving variable  $\xi$  and the radial distance  $r$ . For a Gaussian beam a closed form expression for the  $\xi$  integral in equation 3.2 can be found

$$\begin{aligned} E_{zb}(\xi) &= \int_{-\infty}^{\xi} d\xi' \cos[k_p(\xi - \xi')] e^{-\frac{\xi'^2}{2\sigma_\xi^2}} \\ &= \frac{1}{2} \sqrt{\frac{\pi}{2}} k_p n_{b_0} \sigma_\xi e^{-\frac{1}{2} k_p (k_p \sigma_\xi^2 + 2i\xi)} \left( \operatorname{erfc}\left(\frac{\xi - ik_p \sigma_\xi^2}{\sqrt{2}\sigma_\xi}\right) + e^{2ik_p \xi} \operatorname{erfc}\left(\frac{\xi + ik_p \sigma_\xi^2}{\sqrt{2}\sigma_\xi}\right) \right) \end{aligned} \quad (3.17)$$

where, as is often the case for integration involving Gaussians, the complementary error function [ref]

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-u^2} du \quad (3.18)$$

appears in the expression. A close form expression is however not possible for the radial integral in equation 3.2:

$$E_{zb}(r) = \int_0^\infty dr' r' \left[ I_0(k_p r) K_0(k_p r') \Theta(r' - r) + I_0(k_p r') K_0(k_p r) \Theta(r - r') \right] e^{-\frac{r^2}{2\sigma_r^2}} \quad (3.19)$$

since the convolution of the modified Bessel functions with the Gaussian radial function can only be evaluated numerically, at least as far as the author is aware. To achieve a closed form expression of this integral we approximate the radial Gaussian by expanding it to second order such that

$$n_b(r) = \left( 1 - \frac{r^2}{2\sigma_r^2} \right) \theta(\sqrt{2}\sigma_r - r) + \mathcal{O}(r^4) \quad (3.20)$$

where the Heavieside function  $\theta(\sqrt{2}\sigma_r - r)$  prevents negative densities. This approximation yields a closed form expression when evaluated in equation 3.19, which we then multiply by expression 3.17 as well as add a density-profile dependence to the plasma wavenumber,  $k_p \mapsto k_p(z)$ , to give our approximate longitudinal wakefield:

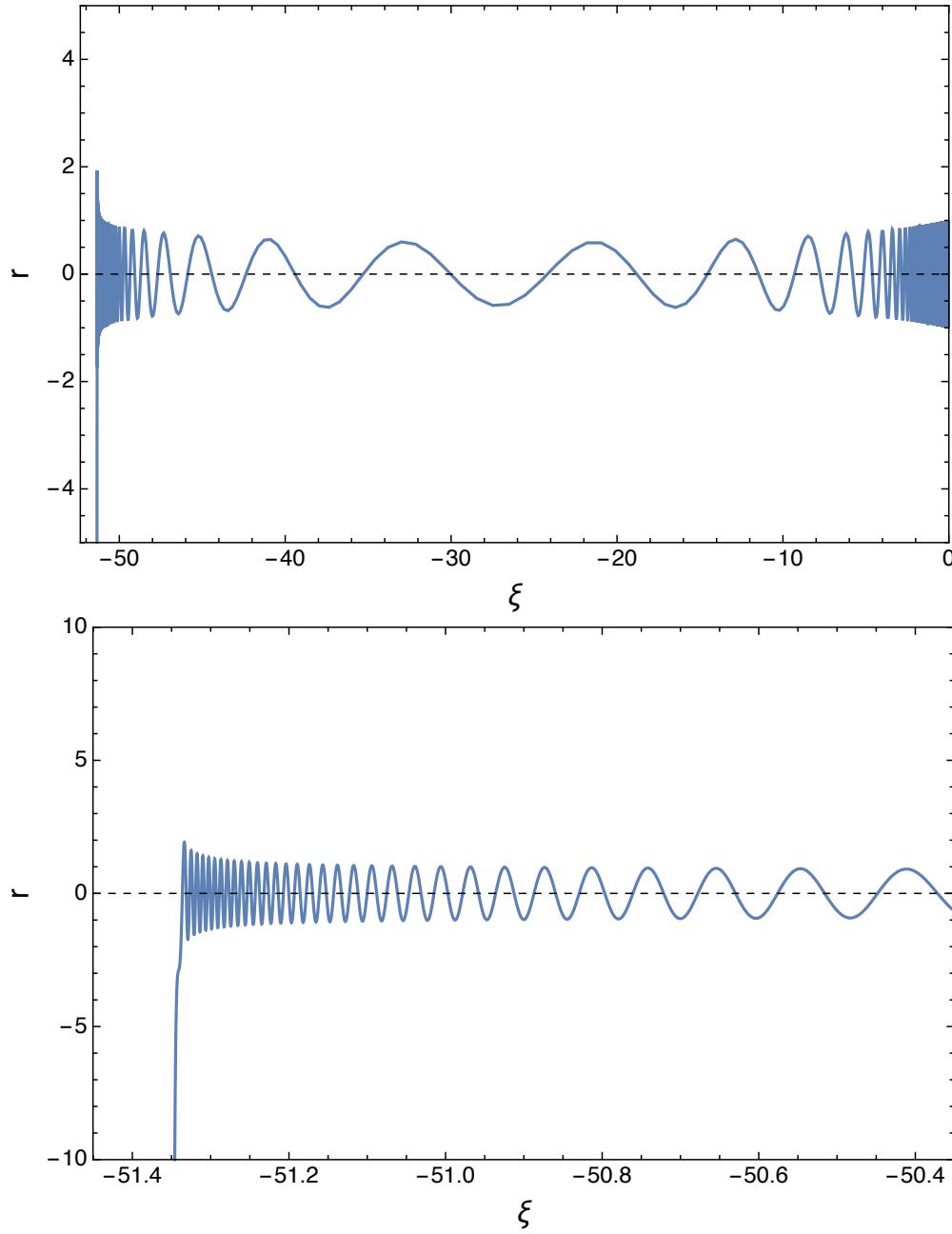
$$E_z(\xi, r, z) = \frac{1}{2} \sqrt{\frac{\pi}{2}} k_p^3(z) n_{b_0} \sigma_\xi e^{-\frac{1}{2} k_p(z)(k_p(z)\sigma_\xi^2 + 2i\xi)} \left( \operatorname{erfc}\left(\frac{\xi - ik_p(z)\sigma_\xi^2}{\sqrt{2}\sigma_\xi}\right) + e^{2ik_p(z)\xi} \operatorname{erfc}\left(\frac{\xi + ik_p(z)\sigma_\xi^2}{\sqrt{2}\sigma_\xi}\right) \right) \\ \left( -\frac{\theta(\sqrt{2}\sigma_r - r) (k_p(z)^2(r^2 - 2\sigma_r^2) + 4k_p(z)^2\sigma_r^2(K_0(k_p(z)r)I_2(\sqrt{2}k_p(z)\sigma_r)))}{2k_p(z)^4\sigma_r^2} \right) \\ \left( \frac{2K_0(k_p(z)r)I_2(\sqrt{2}k_p(z)\sigma_r)}{k_p(z)^2} + \frac{(4k_p(z)^2\sigma_r^2(I_0(k_p(z)r)K_2(\sqrt{2}k_p(z)\sigma_r)) + 4)}{2k_p(z)^4\sigma_r^2} \right) \quad (3.21)$$

where  $I_2$  and  $K_2$  are the second-order modified Bessel functions of the first and second kind, respectively. To find the transverse wakefield  $W_{perp}$  in this approximation we again use the Panofsky-Wenzel theorem and equation 3.3. The full expression is lengthy and contained in Appendix XXX for your amusement. Despite the length and complexity of these expressions, they evaluate significantly faster than the original integrals. Our algorithm can therefore make efficient use of them to allow a large number of to be performed at a reasonable time. To investigate to what accuracy this model can describe single-particle dynamics inside a wakefield as well as ejection of particles due to defocusing, we set up the model to run with the same parameters as those in simulation XXX of chapter 2. We then place a test particle at a position ( $\xi = 0, r = 1\mu\text{m}$ ) to represent a particle close to the middle of the beam. The radial offset of one micron is chosen to give a non-zero transverse force, which is zero at precisely  $r = 0$ . We also set the kinetic energy of the test particle  $T_e(i = 0) = 1\text{GeV}$ . The step-size is set to  $\Delta t = 1\text{fs}$  which given an electric force on the order of  $\text{GeV/m}$  corresponds to a particle radial displacement of  $< 0.1\mu\text{m}$ , this ensures that the test particle is not pushed past the center of the beam in the first iteration, where the transverse force changes direction. We then proceed to iteratively update  $\xi_i \rightarrow \xi_{i+1}$  and  $r_i \rightarrow r_{i+1}$  using equations 3.8 and 3.15, respectively. These in turn compute the longitudinal and transverse wakefields using the approximate expressions in equations 3.21 and 3.29. Finally, we perform run the model using two density profiles: a uniform density, from which we should expect no transverse ejection, and the density profile

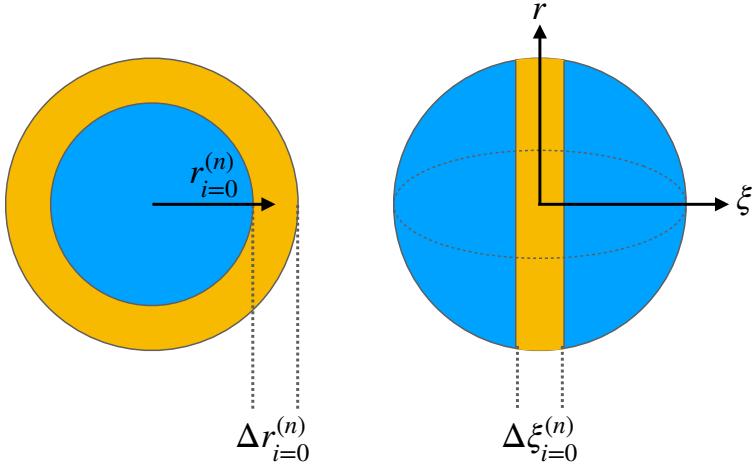
$$n_p(z) = \begin{cases} n_0 & , \quad z \leq z_c \\ n_0 \frac{c_1}{(c_2 - z)^2} & , \quad z > z_c \end{cases} \quad (3.22)$$

which was used in the simulation, such that  $k_p(z)$  gets updated after each iteration as described in section 3.1.2 [Particle-push algorithm]. By storing the energy and position of the particle after each iteration we can then plot the trajectory of the particle as it falls behind the beam, the results of which is shown in figure 3.1. Where we can see that since the test particle starts in a region of focusing field it oscillates around the axis of propagation, until it has fallen behind the beam long enough to reach the defocusing region, where it subsequently gets expelled. [Add stuff about the energy plot and energy of ejected particle] By following a particle through this algorithm all the way until it falls behind the beam and ends up getting defocused, as a result of an increasing density and approaching defocusing region, we can get a prediction of the energy of the ejected

[Add plots for different radial starting positions to learn what determines ejection energy after the beam dynamics section instead.]



**Figure 3.1:** Put constant density and increasing density three plots (radialfull, radialzoom,energy) next to each other in two columns. Maybe re-do in python and overall wakeifled to see how far in defocusing region the particle gets ejected. Add lines on full plot to indicate zoomed in second plot. Also add plot of kinetic energy above these two.



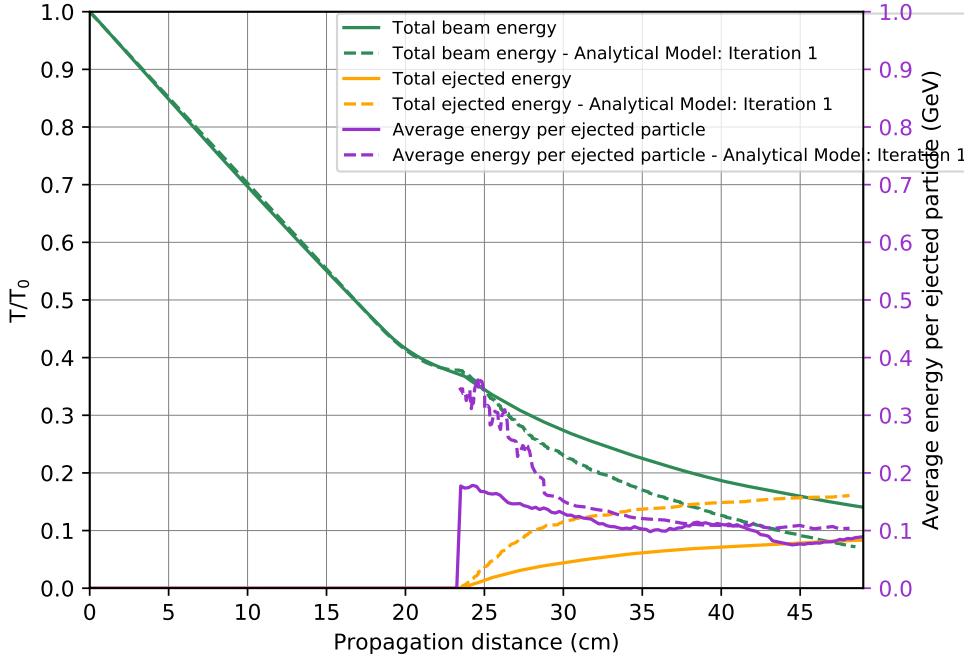
## 3.2 Beam dynamics

### 3.2.1 Static beam approximation

In this section we outline how the model for single-particle dynamics can be extended to extend to model the behaviour of a full beam. This approach assumes that the density of the beam,  $n_b(\xi, r)$ , used to induce the wakefield is constant as it propagates through the varying plasma density is constant. Firstly, the single-particle algorithm is run for multiple test particles starting at various positions within the beam. The combined dynamics of these particles can then be used to represent the kinematics of the beam itself. However, before doing so the contribution of each test particle to the beam must be considered. This is because the beam we are modelling is a bi-Gaussian as given by equation 3.16. Consequently, a test particle placed far away from the center of the beam needs to represent fewer electrons in the beam than a test particle close to the center of the beam, where the density is significantly higher. Furthermore, since we want to utilize the cylindrical symmetry of our equations to the fullest, a test particle at as starting position  $(\xi_{i=0}, r_{i=0})$  should represent all particles at that position around the beam axis. Hence each test particle has an associated volume, given by an annular cross-section of a sphere of circumference  $2\pi r$ , multiplied by thickness its radial and longitudinal thickness, as shown in figure XXX. The contribution of this volume to the overall beam properties is then weighted in proportion to its placement in the bi-Gaussian beam to give the relative number of particles that this test particle represents. Consequently, the relative number of particles that the  $n^{\text{th}}$  test-particle represents is given by

$$N \left( \xi_{i=0}^{(n)}, r_{i=0}^{(n)} \right) = 2\pi r_{i=0}^{(n)} \times \Delta r_{i=0}^{(n)} \times \Delta \xi_{i=0}^{(n)} \times \exp \left[ -\frac{(\xi_{i=0}^{(n)})^2}{2\sigma_\xi^2} - \frac{(r_{i=0}^{(n)})^2}{2\sigma_r^2} \right] \quad (3.23)$$

where  $(\xi_{i=0}^{(n)}, r_{i=0}^{(n)})$  is the test-particle's starting position, which is weighted by the bi-Gaussian accordingly. The volume of each test particle is given by the circumference of the annular cross section  $2\pi r_{i=0}^{(n)}$ , multiplied by the radial and longitudinal widths  $\Delta r_{i=0}^{(n)} = (r_{i=0}^{(n+1)} - r_{i=0}^{(n)})$  and  $\Delta \xi_{i=0}^{(n)} = (\xi_{i=0}^{(n+1)} - \xi_{i=0}^{(n)})$ , respectively. With these weighting factors in place we then choose a distribution of test particles evenly along the beam. For instance, by placing test particles in the ranges  $-2\sigma_\xi \leq \xi \leq 2\sigma_\xi$  and  $0 < r \leq 2\sigma_r$  we cover 91.1% of the volume of the bi-Gaussian beam. Having run the single-particle algorithm for these test particles we can then access the energy of the test particle at each propagation distance  $z$ . These are then summed together and weighted according to their starting positions to give the energy of the pseudo-beam as a function of propagation



**Figure 3.2:** Comparison of results for semi-analytical model in the static beam approximation and 2D EPOCH simulation. [Add average energy per ejected particle as well.]

distance  $z$ :

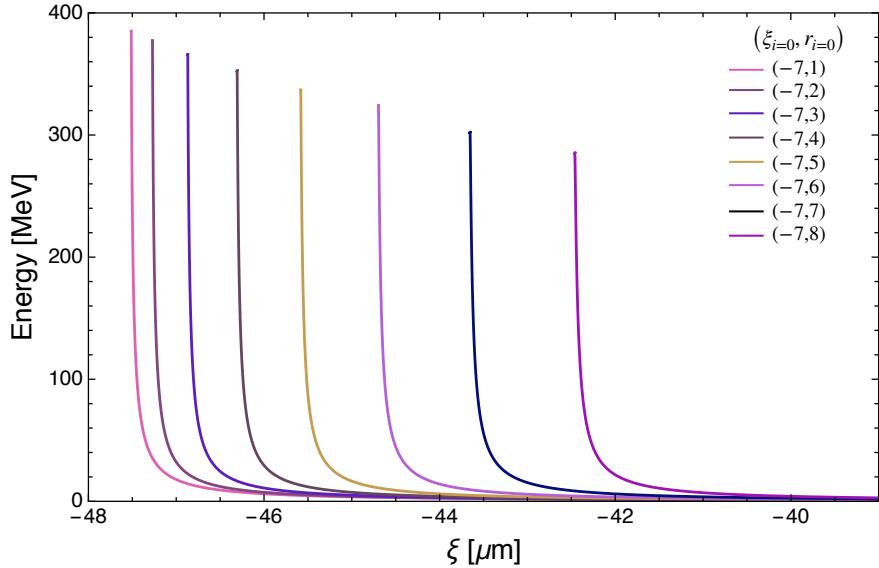
$$T_{beam}(z) = \sum_{n=0}^N T_e^{(n)}(z) \times N(\xi_{i=0}^{(n)}, r_{i=0}^{(n)}) \quad (3.24)$$

where the kinetic energy of the test particles after each iteration,  $T_e^{(n)}(z)$ , is set to zero if the particle has been ejected at a position  $z_{eject} < z$ . Similarly, the total ejected energy can be calculated by storing the position and energy of the particle when it is ejected:

$$E_{eject}^{Tot}(z) = \sum_{n \in n_{eject}(z)} T_e^{(n)}(z_{eject}) \times N(\xi_{i=0}^{(n)}, r_{i=0}^{(n)}) \quad (3.25)$$

were  $n_{eject}(z)$  is the set of particles ejected at a position  $z_{eject} < z$ , which are again appropriately weighted by their starting position. It should be emphasized that the thickness  $\Delta\xi_{i=0}^{(n)}$  is again given by the difference in position between successive test-particles and not, as the above equation might suggest, by the difference in position between successive ejected test-particles.

Figure 3.2 shows the results for running this model with the same simulation parameters as in figures XXX, using longitudinal and radial starting positions in the range  $-2\sigma_\xi \leq \xi_{i=0}^{(n)} \leq 2\sigma_\xi$  and  $-2\sigma_r \leq r_{i=0}^{(n)} \leq 2\sigma_r$ , where  $\sigma_\xi = \sigma_r = 5\mu\text{m}$  and a constant step-sizes  $\Delta\xi_{i=0}^{(n)} = \Delta r_{i=0}^{(n)} = 1\mu\text{m}$ . Furthermore, the radial-ejection limit is set to  $40\mu\text{m}$  to match the size of the simulation window. At this point, it is instructive to go through these results in detail. We can see that the model follows the simulation results up to the point when the density increase and particle ejection begins. This is as expected given that at this point all particles in the beam are still highly-relativistic, ensuring that no particles have yet fallen behind. Nevertheless, the excellent agreement between the model and simulation is interesting in view of the deformation of the beam already present at this stage, an example of which is shown in figure XXX. Once the beam starts losing particles, however, the



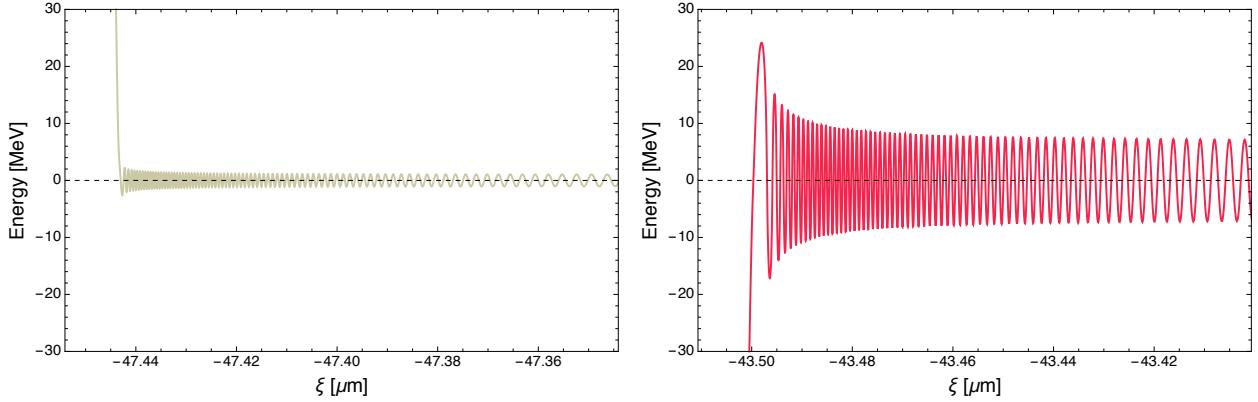
**Figure 3.3:** Radial energies,  $\xi_i$

predictions from the model deviate from the results of the simulation. The fact that the simulation predicts a higher final beam energy is understandable given that the simulated beam starts losing particles at this stage, which leads to fewer plasma electrons displaced by the remaining beam and consequently an induced wakefield with a lower electric field amplitude in comparison to the unchanged beam in our model. This weaker wakefield, in turn, leads to a comparatively slower rate of energy extraction and thereby a higher overall beam energy.

In view of this disparity between the modelled and simulated beam, it is interesting that the results for the average energy per ejected particle in figure XXX appear to exhibit the exact opposite behaviour, namely that the difference between the model and simulations starts out large and then decreases. In the model, the first set of ejected particles with higher than expected energies is found to be composed of test particles starting close to the axis of propagation, at  $r_{i=0}^{(n)} = 1 \mu\text{m}$ , and close to the positions of maximum electric field amplitude,  $\xi_{i=0}^{(n)} \approx \xi_{E_{\max}}$ , which appears behind the bunch as shown in figure XXX. The reason for the latter of these is simply since particles in this region gets decelerated the strongest, and hence reach the defocusing region first. The relation between the ejected energy of a particle and its radial starting position can be understood from figures XXX and XXX. The effect that a particle's initial radial position in the beam has on its subsequent energy when ejected behind the beam is made clearer by figures XXX1 and XXX2. The first of which shows the end of the energy spectrum for particles placed at  $\xi_{i=0}^{(n)} = -7 \mu\text{m}$  for  $1 \mu\text{m} \leq r_{i=0}^{(n)} \leq 8 \mu\text{m}$ . Figure XXX2 shows the path of the test particles in figure XXX1 starting at  $r = 1$  and  $r = 6$ , respectively.

(amplitude of oscillations in XXX2, weaker decelerating field, defocusing region is closer  
(if we look at the profile of the transverse field (add plot) we can see how the amplitude of oscillations contribute to the ejection energy, small oscillations low field, longer to get defocused, and also the longitudinal force is stronger. Large oscillations: stronger defocusing )

Finally, even if we disregard the high energy of the first set of ejected particles, the total ejected energy in figure XXX is higher in the model than in the simulation. Similar to the beam energy, this is due to the wakefield in our model not weakening as the beam propagates. Since each test particle represents a volume, this leads to regions of our pseudo-beam getting decelerated to non-relativistic velocities, and subsequently its particles ejected, at a rate that is higher than the beam in the simulation. For a more accurate model it is clear that we need to address the issue of particle-loss.



**Figure 3.4:** abc

### 3.2.2 Dynamic beam density

The model presented above is able to compute the dynamics of a particle, and by extension the dynamics of a beam, at a reasonable computational time due to the approximations we have used. In particular, the fact that closed-form expressions are found for the wakefield integrals in equations 3.2 and 3.4 means that these computationally expensive integrals only need to be evaluated once at the start of the algorithm. The importance of maintaining this feature restricts the range of improvements we can introduce to the model. One of these restrictions is that we can not alter the functional shape of the beam, since this would require re-integration of the wakefields. Instead, we choose to maintain the bi-Gaussian shape of the beam but add a functional dependence to the amplitude of the closed-form expressions for the longitudinal and transverse wakefields. Since we assume the beam shape is constant, this amplitude should only vary due to the total number of particles in the bi-Gaussian part of the beam, since the plasma response from individual particles falling behind is negligible. We therefore need to specify a way of determining when a test-particle has fallen behind the beam. Initial tests were done with strict beam boundaries, whereby the contribution of the  $n^{\text{th}}$  test-particle to the total number of particles was set to zero once it had fallen behind to a position  $\xi_i^{(n)} < -5\sigma_\xi$ , for instance. This however lead to successive sharp drops in the amplitude of the wakefield, which in turn induced instabilities in the particles' radial oscillations which ultimately ejected particles before entering the defocusing region.

to this end, we reduce the influence of each test-particle smoothly

This is the same as saying that some of the particles that the test-particle represents stay at that positions.

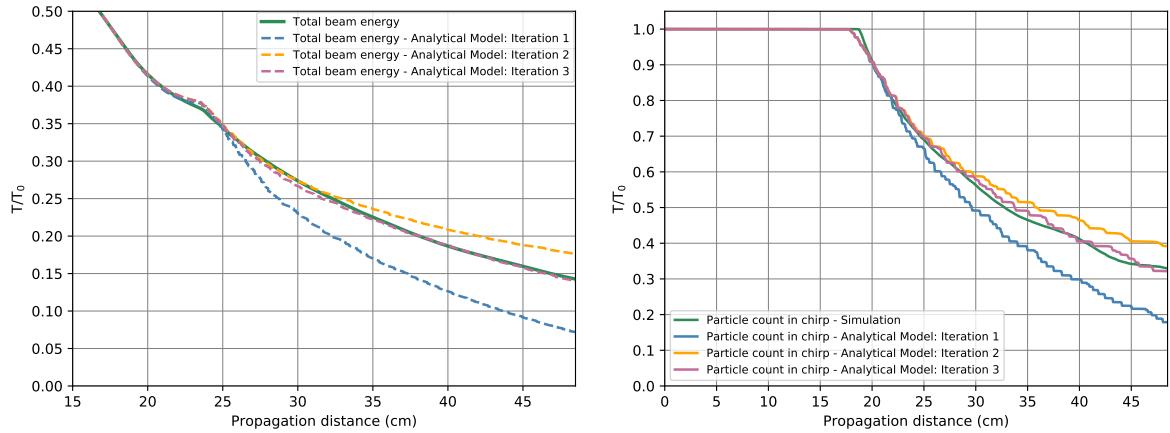
[Skip plot of iteration 1 and simulation only and go straight to saying that this introduces the concept of a global iterations in our algorithm]

$$N_{\text{chirp}}(z) = \sum_{n=0}^N T_e^{(n)}(z) \times N(\xi_{i=0}^{(n)}, r_{i=0}^{(n)}) \times \exp\left[-\frac{(\xi_i^{(n)}(z) - \xi_{i=0}^{(n)})^2}{2\sigma_\xi^2}\right] \quad (3.26)$$

$$E_z^{(j+1)}(\xi, r, z) = N_{\text{chirp}}^{(j)}(z) \times E_z(\xi, r, z) \quad (3.27)$$

$$W_\perp^{(j+1)}(\xi, r, z) = N_{\text{chirp}}^{(j)}(z) \times W_\perp(\xi, r, z) \quad (3.28)$$

where  $j$  is the number of global iterations.



Plot maximum electric field from model and epoch.

Add loss of particle calculation and iteration of self-updating of algorithm without having to redo integral for beam density.

Use gaussian for particle loss to model that the influence of a particle on the induced wakefield diminishes gradually as it falls through the bunch. Better than sharp particle count cut if fallen behind chirp.

$$\begin{aligned}
 W(\xi, r, z) = & \frac{ie^{-\frac{1}{2}k_p(k_p\sigma\xi^2+2i\xi)}n_{b_0}\sqrt{\frac{\pi}{2}}\sigma\xi\delta(r-\sqrt{2}\sigma r)\operatorname{erfc}\left(\frac{\xi-ik_p\sigma\xi^2}{\sqrt{2}\sigma\xi}\right)r^2}{4\sigma r^2} \\
 & - \frac{ie^{ik_p\xi-\frac{k_p^2\sigma\xi^2}{2}}n_{b_0}\sqrt{\frac{\pi}{2}}\sigma\xi\delta(r-\sqrt{2}\sigma r)\operatorname{erfc}\left(\frac{ik_p\sigma\xi^2+\xi}{\sqrt{2}\sigma\xi}\right)r^2}{4\sigma r^2} \\
 & + \frac{ie^{-\frac{1}{2}k_p(k_p\sigma\xi^2+2i\xi)}n_{b_0}\sqrt{\frac{\pi}{2}}\sigma\xi\delta(r-\sqrt{2}\sigma r)\operatorname{erfc}\left(\frac{\xi-ik_p\sigma\xi^2}{\sqrt{2}\sigma\xi}\right)}{k_p^2\sigma r^2} \\
 & - \frac{1}{2}ie^{-\frac{1}{2}k_p(k_p\sigma\xi^2+2i\xi)}n_{b_0}\sqrt{\frac{\pi}{2}}\sigma\xi\delta(r-\sqrt{2}\sigma r)\operatorname{erfc}\left(\frac{\xi-ik_p\sigma\xi^2}{\sqrt{2}\sigma\xi}\right) \\
 & + ie^{-\frac{1}{2}k_p(k_p\sigma\xi^2+2i\xi)}n_{b_0}\sqrt{\frac{\pi}{2}}\sigma\xi I_2(\sqrt{2}k_p\sigma r)K_0(k_p r)\delta(r-\sqrt{2}\sigma r) \\
 & \operatorname{erfc}\left(\frac{\xi-ik_p\sigma\xi^2}{\sqrt{2}\sigma\xi}\right) - ie^{-\frac{1}{2}k_p(k_p\sigma\xi^2+2i\xi)}n_{b_0}\sqrt{\frac{\pi}{2}}\sigma\xi I_0(k_p r)K_2(\sqrt{2}k_p\sigma r)\delta(r-\sqrt{2}\sigma r)\operatorname{erfc}\left(\frac{\xi-ik_p\sigma\xi^2}{\sqrt{2}\sigma\xi}\right) \\
 & - \frac{ie^{ik_p\xi-\frac{k_p^2\sigma\xi^2}{2}}n_{b_0}\sqrt{\frac{\pi}{2}}\sigma\xi\delta(r-\sqrt{2}\sigma r)\operatorname{erfc}\left(\frac{ik_p\sigma\xi^2+\xi}{\sqrt{2}\sigma\xi}\right)}{k_p^2\sigma r^2} + \frac{1}{2}ie^{ik_p\xi-\frac{k_p^2\sigma\xi^2}{2}}n_{b_0}\sqrt{\frac{\pi}{2}}\sigma\xi\delta(r-\sqrt{2}\sigma r) \\
 & \operatorname{erfc}\left(\frac{ik_p\sigma\xi^2+\xi}{\sqrt{2}\sigma\xi}\right) - ie^{ik_p\xi-\frac{k_p^2\sigma\xi^2}{2}}n_{b_0}\sqrt{\frac{\pi}{2}}\sigma\xi I_2(\sqrt{2}k_p\sigma r)K_0(k_p r)\delta(r-\sqrt{2}\sigma r)\operatorname{erfc}\left(\frac{ik_p\sigma\xi^2+\xi}{\sqrt{2}\sigma\xi}\right) \\
 & + ie^{ik_p\xi-\frac{k_p^2\sigma\xi^2}{2}}n_{b_0}\sqrt{\frac{\pi}{2}}\sigma\xi I_0(k_p r)K_2(\sqrt{2}k_p\sigma r)\delta(r-\sqrt{2}\sigma r)\operatorname{erfc}\left(\frac{ik_p\sigma\xi^2+\xi}{\sqrt{2}\sigma\xi}\right) \\
 & + ie^{-\frac{1}{2}k_p(k_p\sigma\xi^2+2i\xi)}n_{b_0}\sqrt{\frac{\pi}{2}}\sigma\xi\left(-\operatorname{erf}\left(\frac{\xi-ik_p\sigma\xi^2}{\sqrt{2}\sigma\xi}\right) - e^{2ik_p\xi} + e^{2ik_p\xi}\operatorname{erf}\left(\frac{ik_p\sigma\xi^2+\xi}{\sqrt{2}\sigma\xi}\right) + 1\right) \\
 & \frac{\left(\left(2k_p I_2(\sqrt{2}k_p\sigma r)K_1(k_p r)\sigma r^2 + 2k_p I_1(k_p r)K_2(\sqrt{2}k_p\sigma r)\sigma r^2 - r\right)\theta(\sqrt{2}\sigma r - r)\right)}{2\sigma r^2} \\
 & \frac{\left(-2k_p\sigma r^2 I_2(\sqrt{2}k_p\sigma r)K_1(k_p r)\right)}{2\sigma r^2}
 \end{aligned} \tag{3.29}$$

Radial integral fully simplified

$$\begin{aligned}
 & \frac{2K_0(k_p r)I_2(\sqrt{2}k_p\sigma r)}{k_p^2} \\
 & - \frac{\theta(\sqrt{2}\sigma r - r)\left(k_p^2(r^2 - 2\sigma r^2) + 4k_p^2\sigma r^2\left(K_0(k_p r)I_2(\sqrt{2}k_p\sigma r) - I_0(k_p r)K_2(\sqrt{2}k_p\sigma r)\right) + 4\right)}{2k_p^4\sigma r^2}
 \end{aligned} \tag{3.30}$$

# 4

## Ongoing work and concluding remarks

#### 4. Ongoing work and concluding remarks

# Bibliography