Space charge models for Placet

Reidar Lunde Lillestøl

CERN / Uni. Oslo

04. September 2013

Contents

Motivation

PETS power production:

$$P = \frac{1}{4}(R'/Q)\frac{\omega_{\text{rf}}}{v_g}L^2 I^2 F^2(\lambda)\eta_\Omega^2 \qquad (1)$$

where, assuming perfect bunch phase, the single-bunch form factor for Gaussian bunches is

$$F(\lambda) = F_b(\sigma_z) = \exp\left[-\frac{1}{2}(2\pi f_b \sigma_z/c)^2\right]. \qquad (2)$$

For maximum 1 % luminosity loss in the main beams, we need an energy spread of $\frac{\Delta_E}{E} < 7 \times 10^{-4}$ in CLIC. This translates to the following bunch length tolerances:

- 1.1 % coherent bunch length tolerance
- 3.3 % incoherent bunch length tolerance

Even though the CLIC drive beams are highly relativistic ($\gamma \in [470, 4700]$), there has been some concern that longitudinal space charge may violate these tolerances.

In addition, space charge may be relevant in the Beam Delivery System because the strong squeezing may cause a bunch lengthening there.

The Particle-in-Cell (PIC) approach

Particle-in-Cell (PIC) is a numerical method for solving partial differential equations.

A 1D, 2D or 3D grid is set up, and a basic procedure is iterated:

1. Interpolation of charge (and current) source terms to grid points
2. Computation of the fields at the grid points
    - Finite Difference methods
    - Finite Elements methods
    - Spectral methods
3. Interpolation of fields to the particle locations
4. Particle pushing based on the fields

Model 1 – 2D PIC

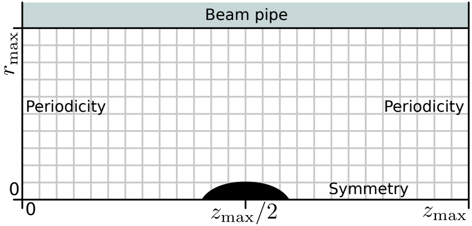
A space charge code has so far been developed in Octave.

Chosen model: 2D PIC (r, z) that assumes axisymmetry

- Full 3D very computationally intensive
- 1D has long-range Coulomb fields, and all particles get the same kick regardless of radial position
- 2D (r, z) is a compromise, but assumes axisymmetry and is most valid for round beams (like the drive beam). Calculations are done in the beam frame.

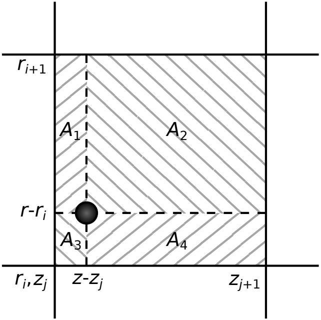
Boundary conditions:

- $r = 0$: Symmetry $(\phi_{-1,z} = \phi_{1,z})$
- $r = R$: Fixed $(\phi_{R,z} = 0)$
- $z = 0$: Periodic $(\phi_{r,-1} = \phi_{r,Z-1})$
- $z = Z$: Periodic $(\phi_{r,Z+1} = \phi_{r,1})$



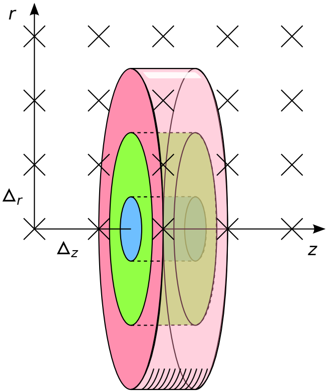
Linear charge interpolation from particles to grid points.

- Lower right grid point gets fractional charge based on A_1
- Lower left grid point gets fractional charge based on A_2
- Upper right grid point gets fractional charge based on A_3
- Upper left grid point gets fractional charge based on A_4



Model 1 – 2D PIC: Charge density calculation

Each grid point represents a volume in 3 dimensional space.



Charge density for a free space grid point is

$$\rho_{i,j} = \frac{q_{i,j}}{V_{i,j}} = \frac{q_{i,j}}{2\pi r_i \Delta_r \Delta_z}$$

On axis, we can make three approaches:

- $\rho_{0,j} = \frac{q_{i,j}}{\pi\left(\frac{\Delta_r}{2}\right)^2 \Delta_z} = \frac{q_{i,j}}{\pi \Delta_r^2 \Delta_z / 4}$ (intuitive)
- $\rho_{0,j} = \frac{q_{i,j}}{\pi \Delta_r^2 \Delta_z / 3}$ (from Verboncoeur, *should* be correct)
- $\rho_{0,j} = \frac{q_{i,j}}{\pi \Delta_r^2 \Delta_z / 2}$ (gives correct answers)

Model 1 – 2D PIC: Calculating potential

We solve Poisson's equation

$$\nabla^2\phi = \frac{1}{r}\frac{\partial \phi}{\partial r} + \frac{\partial^2 \phi}{\partial r^2} + \frac{\partial^2 \phi}{\partial z^2} = -\frac{\rho}{\epsilon_0} \tag{3}$$

on the grid using the Finite Difference method. Each grid point depends on 5 surrounding points in each dimension, to ensure an accurate solution.

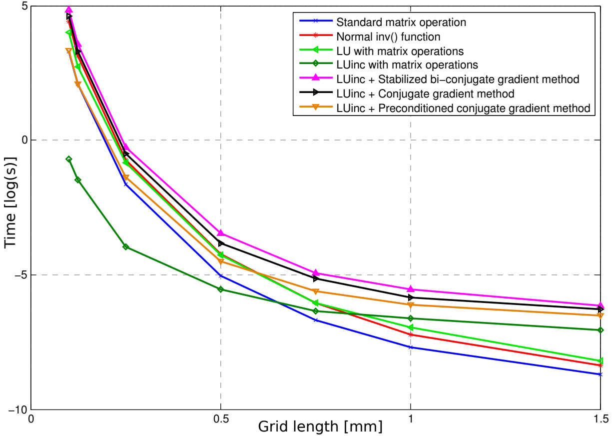
$$\begin{aligned}\nabla^2\phi\big|_{i,j} \approx \frac{1}{r_i}&\frac{\phi_{i-2,j} - 8\phi_{i-1,j} + 8\phi_{i+1,j} - \phi_{i+2,j}}{12\Delta_r}\\ &+ \frac{-\phi_{i-2,j} + 16\phi_{i-1,j} - 30\phi_{i,j} + 16\phi_{i+1,j} - \phi_{i+2,j}}{12\Delta_r^2}\\ &+ \frac{-\phi_{i,j-2} + 16\phi_{i,j-1} - 30\phi_{i,j} + 16\phi_{i,j+1} - \phi_{i,j+2}}{12\Delta_z^2}\\ &= -\frac{\rho_{i,j}}{\epsilon_0}.\end{aligned} \tag{4}$$

We get one equation per grid point, and can write all of them on matrix form as $\mathbf{A}_\phi \vec{\phi} = -\frac{1}{\epsilon_0}\vec{\rho}$

A is a huge (and sparse) $N_r N_z \times N_r N_z$ matrix, which must be inverted to solve the equation.

Model 1 – 2D PIC: Solving for potential in Octave

The code has so far been developed in Octave. Several methods to solve $\mathbf{A}_\phi \vec{\phi} = -\frac{1}{\epsilon_0}\vec{\rho}$ were looked into, and the fastest for small grid lengths (many points) was incomplete LU decomposition.



Model 1 – 2D PIC: Calculating E-fields

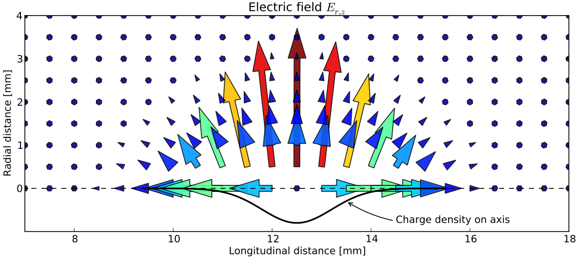
The electric field is found from the gradient of the potential,

$$E = -\nabla\phi. \tag{5}$$

Using the Finite Difference method with 5 points and matrix equations, we have

$$E_r|_{i,j} = \frac{-\phi_{i-2,j} + 8\phi_{i-1,j} - 8\phi_{i+1,j} + \phi_{i+2,j}}{12\Delta_r} \quad \Rightarrow \quad \vec{E_r} = \mathbf{A}_r\vec{\phi}, \tag{6}$$

$$E_z|_{i,j} = \frac{-\phi_{i,j-2} + 8\phi_{i,j-1} - 8\phi_{i,j+1} + \phi_{i,j+2}}{12\Delta_z} \quad \Rightarrow \quad \vec{E_z} = \mathbf{A}_z\vec{\phi}. \tag{7}$$



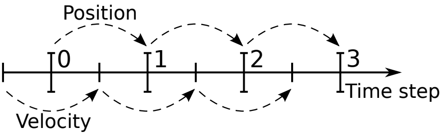
Above: Electric field from a “Gaussian wire bunch”.

Model 1 – 2D PIC: Particle pushing

The force on the particles is calculated using the Lorentz force. We only use the E fields since we solve in the beam frame,

$$\vec{F} = q(\vec{E} + \vec{v} \times \cancelto{0}{\vec{B}}) = q\vec{E}. \tag{8}$$

A leapfrog algorithm is used for symplecticity.



This originates from Newton's 2^{nd} law and the Lorentz force:

$$F_{r,z} = qE_{r,z} = ma_{r,z} \tag{9}$$

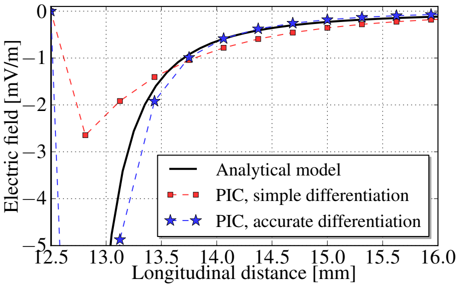
Using central differentiation, we get

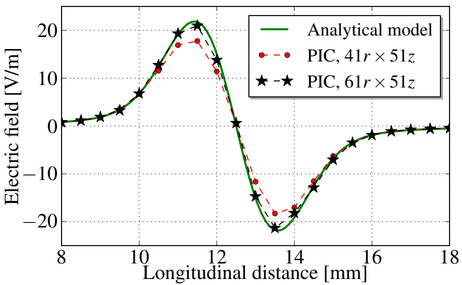
$$a_{r,z}(t_i) = \frac{v_{r,z}(t_i + \delta_t/2) - v_{r,z}(t_i - \delta_t/2)}{\delta_t} = \frac{q}{m} E_{r,z}(t_i) \tag{10}$$

$$\Rightarrow \qquad v_{r,z}(t_i + \delta_t/2) = \frac{q}{m}\delta_t E_{r,z}(t_i) + v_{r,z}(t_i - \delta_t/2) \tag{11}$$

A similar calculation can be used to find updated positions.

Model 1 – 2D PIC: Benchmarking with analytical models





Longitudinal field from a point charge.

$$E_z(z) = \frac{q}{4\pi\epsilon_0} \frac{(z - z_0)}{|z - z_0|^3}$$

Here we see the need for the more accurate differentiation in eq. (4) to ensure convergence.

Longitudinal field from a Gaussian bunch (infinitely thin).

$$E_z(z) = \frac{1}{4\pi\epsilon_0} \int \frac{\rho(s)(\bar{z} - s)}{|\bar{z} - s|^3} d\bar{z}$$

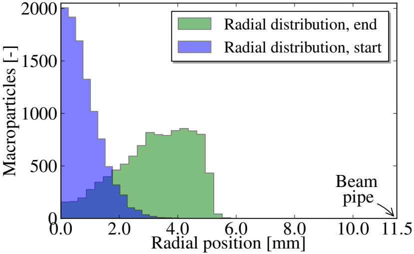
Here we see the need for a good radial grid resolution to ensure convergence.

Model 1 – 2D PIC: Simple simulations

The developed Octave code only simulates space charge. Some simple simulations have been performed, but without quadrupole focusing etc.

A simulation of the CLIC decelerator gave a bunch length increase of a factor $(1 + 6 \times 10^{-7})$. The transverse size increased more significantly.

When quadrupoles are included, we expect them to correct for transverse effects and make longitudinal effects larger.



A simulation of the decelerator Test Beam Line (TBL) in the CTF3 gave a bunch length increase of a factor $(1 + 5 \times 10^{-5})$.

Model 2 – Differentiation of charge distribution

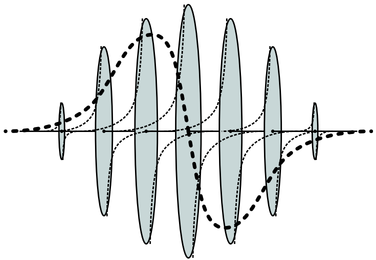
PIC requires some computation time (a few hours for the whole decelerator). In addition, 2D PIC is best suited to round beams like the CLIC drive beam.

We also consider a more simple model that should be considerably faster and more applicable to the main beam.

This model is based on integration of the longitudinal charge distribution.

$$F_z(z) = qE_z(z) = \frac{q}{4\pi\epsilon_0} \sum_i \frac{\rho(z_i)(z - z_i)}{|z - z_i|^3}$$

- For sliced beams, we can sum the fields from each slice (as point particles).
- For particle beams, we can group particles in bins or use a Gaussian fit.



Implementation in Placet

- If PIC is used, we need a certain number of timesteps per lattice elements based on the stability criterion

$$\delta_t \leq \frac{1}{c} \frac{1}{\sqrt{\frac{1}{\Delta_r^2} + \frac{1}{\Delta_z^2}}}. \tag{12}$$

- A new step function has been made (locally) that also includes space charge.
- We open for one space charge class per model, with its own kick function etc.
- Space charge can therefore be a property of each element. By default it will be off.
  Example: To use model 2 in a quadrupole we use

```
Quadrupole -spacecharge 2
```

- This also allows space charge calculation only in relevant areas
- Large elements like the field solver matrices will be stored and reused every iteration.

Summary

- Two models have so far been considered for implementation in Placet.
- A 2D PIC code has been developed in Octave and benchmarked with simple analytical models.
- A simulation of the decelerator without focusing showed a negligible bunch lengthening.
- We consider separate space charge classes in Placet and the possibility to change models in different lattice elements.

Thanks to Kyrre Sjøbæk, Yngve Levinsen, Erik Adli and Andrea Latina
for useful tips and fruitful discussions!

R. L. Lillestol *et al.*, in *Proceedings of the International Particle Accelerator Conference, Shanghai, 2013.*

J. P. Verboncoeur, J. Comput. Phys., **174** 421-7 (2001)

Backup slides

Calculating charge density (PIC)

Charge density is calculated as $\rho_{i,j} = \frac{q_{i,j}}{V_{r_i} V_{z_j}}$, where in free space we have

$$V_{r_i} = \frac{2\pi}{\Delta_r} \left(\int_{r_{i-1}}^{r_i} r(r - r_{i-1}) dr + \int_{r_i}^{r_{i+1}} r(r_{i+1} - r) dr \right) = \quad \dots \quad = 2\pi r_i \Delta_r, \tag{13}$$

$$V_{z_j} = \frac{1}{\Delta_z} \left(\int_{z_{j-1}}^{z_j} (z - z_{j-1}) dz + \int_{z_j}^{z_{j+1}} (z_{j+1} - z) dz \right) = \quad \dots \quad = \Delta_z. \tag{14}$$

Radial volume on axis:

$$V_{r_0} = \frac{2\pi}{\Delta_r} \int_0^{r_1} r(r_1 - r) dr = \frac{2\pi}{\Delta_r} \left[\frac{r^2}{2} r_1 - \frac{r^3}{3} \right]_0^{r_1} = \frac{2\pi}{\Delta_r} \left(\frac{r_1^3}{2} - \frac{r_1^3}{3} \right) = \frac{2\pi}{\Delta_r} \frac{\Delta_r^3}{6} = \frac{\pi \Delta_r^2}{3} \tag{15}$$