A PARTICLE SIMULATION CODE FOR LONGITUDINAL DYNAMICS OF HEAVY ION BEAMS

https://escholarship.org/uc/item/8sr2q3cb

Sternlieb, A.

1981

Peer reviewed
A PARTICLE SIMULATION CODE FOR LONGITUDINAL DYNAMICS OF HEAVY ION BEAMS

A. Sterulieb, L. Smith, L.J. Laslett, J. Bisognano, and I. Haber

January 1981
A PARTICLE SIMULATION CODE FOR LONGITUDINAL DYNAMICS OF HEAVY ION BEAMS*

A. Sternlieb, L. Smith, L.J. Laslett, J. Bisognano
Accelerator and Fusion Research Division
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

I. Haber
Naval Research Laboratory
Washington, D.C. 20375

January 1981

*This work was supported by the Director, Office of Energy Research, Office of Inertial Fusion, Research Division of the U.S. Department of Energy under Contract No. W-7405-ENG-48.
A Particle Simulation Code for Longitudinal Dynamics of Heavy Ion Beams

A. Sternlieb, L. Smith, L.J. Laslett, J. Bisognano, I. Haber**

Abstract

A particle computer simulation code is presented, designed to describe the longitudinal motion of charged particles in a nonneutralized beam. The code is 1-d, electrostatic, and is capable of simulating coasting or bunched beams, cold or warm, with arbitrary wall impedance. External focusing forces are introduced in the bunched beam case. A variety of distribution functions can be represented. The code is self-consistent and it can follow both the linear and nonlinear evolution of a beam.

**Naval Research Laboratory, Washington, D.C. 20375
I. Introduction

In the last few years, heavy ion beams have been suggested as possible drivers in an inertial confinement fusion scheme. Because of the relatively high nonneutralized currents involved, space-charge effects are important during the acceleration and transport of the beam through the highly evacuated focusing system. The transverse stability of such beams has been investigated to a certain extent, using both theoretical analysis and computer simulation. The agreement found is encouraging.

Recently, an attempt has been made to formulate analytically the problem of longitudinal stability of a bunched beam. Certain stationary distribution functions \( f(x, v_x) \) have been proposed and linearly analyzed for stability against perturbations. A linear analysis for a cold bunch and arbitrary density distribution has also been performed. In the above stability analyses, the transverse and longitudinal motions of the particles in the bunch are assumed to be completely decoupled. In addition, a resistive wall term was introduced in the motion equations. The general conclusion until now is that, with a resistive wall as a perturbation a bunched beam is linearly stable.

However, it is clear that the analytical approach to the longitudinal stability problem has serious shortcomings, especially in using various disputable approximations and assumptions. Therefore, the need arises for a simulation program, which can study in a self-consistent way both the linear and nonlinear longitudinal stability and evolution of a bunched beam. In the following we describe a particle code which has been designed to study those phenomena not amenable to theoretical analysis.

II. The Simulation Code

(a) General Remarks

The simulation code is one-dimensional, electrostatic, and it follows the motion of several thousands of particles in the self-consistent electric field generated by space-charge (plus possible externally imposed forces). The simulation system is chosen in a reference frame moving with the beam. The time evolution is performed according to a first-order time-centered leap-frog scheme, as follows:
The upper indices are fractions of the basic time-step $\Delta t$. The positions and fields are defined at integral numbers of time steps, and the velocities (and currents) are defined at half time-steps; $t_0$ is the starting point of the simulation. The system is assumed to be periodic in $x$ for particles and fields and no loss of particles is allowed.

A basic one time-step loop in the simulation process works as follows:

1. Find electric field from charge distribution:
   a. Find Fourier components of $\rho(x)$ ($\rho(x)$ is the charge density).
   b. Find field in Fourier space by using: $E_x = \sum Z_k \rho_k e^{ikx}$, where $E_x$ is the electric field, and $Z_k$ is a weighting function of $k$.

2. Push particles to new locations and find new charge distribution.

An external focusing force is applied in the bunched beam case. In the simplest case, it is proportional to the distance from the center of the bunch, when the space-charge force is linear. In this case, the strength constant of the focusing force is automatically determined by the code such as to balance the emittance and space-charge defocusing effects. The focusing force is turned off when studying the coasting beam case. A resistive force can also be introduced. As a special case, it may be taken to be proportional to the linear charge density $\rho(x)$.

In the following section we describe in more detail the structure of the simulation code.

**(b) Technical Details of the Simulation Code**

The flow-chart of the code is presented in Figure 1. The system has $NX$ real cells and two dummy cells at both ends, to adjust boundary conditions. Initially, the electric field and charge density arrays are zeroed, to make place for the new values.

For the bunched beam case, the length of the bunch is defined as:
$L_B = 2 \cdot \text{DELX}$,

where \text{DELX} is in cell units (a cell \(\Delta x\) is one unit length in the code).

![Diagram](image)

Then, a "phase-velocity" \(\omega_p\) is defined by:

$$\omega_p^2 = 4\pi N_{\text{TOT}} q^2/(m \cdot L_B) \quad (\omega_p^2 \equiv 4\pi \nu_p^2; \nu_p \text{ is defined in Section III}),$$

where \(N_{\text{TOT}}\) is the total number of simulation particles, \(q\) is the charge of one particle and \(m\) is the mass of one particle (taken to be equal to one). As a special case, when we investigate a parabolic density and velocity distribution\(^{(4)}\), the requirement of a stationary distribution function implies a linear focusing force:

$$F = -k \cdot (x - x_{\text{initial center of bunch}}),$$

where \(k\) is given by:

$$k = \nu_0^2 = \nu^2 + \omega_0^2$$

Here:

$$\nu = \frac{v_{x, \text{th}}}{\text{DELX}} \quad (v_{x, \text{th}} \text{ is the maximum } x\text{-velocity}),$$

where \(\nu\) is the emittance defocusing constant, and

$$\omega_0 = \sqrt{\frac{1.5 \cdot \omega_p}{2\pi \cdot \text{DELX}}}$$

where \(\omega_0\) is the space-charge defocusing constant. Both defocusing forces are linear.

The initial distribution function is given by\(^{(4)}\):

$$f(x, v_x) = \frac{3N_{\text{TOT}}}{2\pi v_{x, \text{th}} \cdot \text{DELX}} \cdot \sqrt{1 - \frac{x^2}{\text{DELX}^2} - \frac{v_x^2}{v_{x, \text{th}}^2}}$$

(1)
For the bunched beam case, initial particle positions and velocities are specified according to (1), which gives a parabolic distribution for both density distribution \( p(x) \) and velocity distribution \( f(v_x) \).

During the first time-step, the velocities are moved backwards half a time-step, for leap-frog centering purpose.

A variety of spatial and velocity distributions, such as gaussian or uniformly random distribution can also be provided by the code.

The initial charge distribution is found by proportional area-weighting of all particle contributions at the cell centers. The width of a particle is equal to the cell width. After initialization, periodicity is applied to the ends of the system:

\[
\rho(1) \rightarrow \rho(1) + \rho(NX+1) \\
\text{first (dummy) cell} \rightarrow \text{last real cell}
\]

Now, the main time-integrating loop is started. First, one calculates the new field by the following steps:

(a) The average charge density is set equal to zero:

\[
\rho(x) = \rho^{\text{old}}(x) - <\rho^{\text{old}}(x)>_x
\]

Therefore, \( \rho(x) \) may have both positive and negative values.

(b) Calculate the Fourier transform of the charge density, using a sin-cos expansion:

\[
\rho(x) = \sum_n b_n \cos k_n x + \sum_n c_n \sin k_n x , \text{ where:}
\]

\[
k_n = \frac{2\pi n}{NX} , (n = 1, 2, \ldots)
\]
(c) Convert \( p(x) \) to a complex exponential form:

\[
\rho(x) = \sum_n \rho_n e^{ik_n x} + \sum_n \rho_n^* e^{-ik_n x},
\]

where

\[
\rho_n = \frac{1}{NX} \int_{-NX/2}^{+NX/2} p(x) \exp(-ik_n x) dx
\]

and one obtains the relationships:

\[
b_n = \rho_n + \rho_n^* = \frac{2}{NX} \int_{-NX/2}^{+NX/2} p(x) \cos(k_n x) dx
\]

\[
c_n = i(\rho_n - \rho_n^*) = \frac{2}{NX} \int_{-NX/2}^{+NX/2} p(x) \sin(k_n x) dx
\]

Therefore:

\[
\rho_n = \frac{1}{2} \left( b_n + \frac{c_n}{i} \right)
\]

and \( \rho_n \) is usually complex:

\[
\rho_n = \rho_n^{\text{real}} + i \rho_n^{\text{imag.}}
\]

(d) Finally, one calculates the electric field from the usual long-wavelength formula:

\[
E_x = -i \sum_k Z_k \rho_k e^{ikx}
\]

The average electric field is set to zero. Then, we introduce gaussian shapes for the simulation particles:

\[
\rho^1(x) \text{ of one particle } \sim \frac{e^{-x^2/2(a_x^2)}}{\sqrt{2\pi} a_x} \quad (0 < a_x < 1)
\]

Here \( x \) is measured from the center of each particle: \(-1/2 < x < 1/2\). We take \( a_x = 1 \) ("fat" particles). Also \( \int_{-1/2}^{1/2} \rho^1(x) dx = q \). In Fourier space, this
corresponds to the multiplication of any Fourier component of the density $\rho$ by a "smoothing" factor of $e^{-k^2/n^2}$, which cuts off the highest $k$'s.

From (2), we have in Fourier space, including the smoothing factor:

$$E_k = -i Z_k \rho_k e^{-k^2} \quad (a_x = 1, k_n \equiv k, n > k)$$

A resistive force can be introduced of the form:

$$F_{\text{Res}} = -qR_p(x)$$

For example, if the space-charge force is of the form:

$$F_{\text{sp. ch.}} \sim -g \frac{\partial \rho}{\partial x},$$

then with resistivity, (and $g = 1$), the electric field in its Fourier form becomes:

$$E_k = (ik\rho_k + R\rho_k) (-)e^{-k^2}$$

Therefore:

$$E_k^{\text{imag.}} = \left( k \cdot \rho_k^{\text{real}} + R\rho_k^{\text{imag.}} \right)(-)e^{-k^2}$$

$$E_k^{\text{real}} = \left( -k \cdot \rho_k^{\text{imag.}} + R\rho_k^{\text{real}} \right)(-)e^{-k^2}$$

(e) The electric field is converted to a sin-cos form in Fourier space, necessary for Fourier synthesis.

(f) The reverse Fourier transform is performed, giving the electric field in real space, $E_x(x)$.

Boundary conditions for the field are specified as follows:

$$E_x(NX + 1) = E_x(1)$$

External fields may be added at this point:

$$E_x \rightarrow E_x + E_x^{\text{ext.}}$$

Finally, the source array is zeroed:

$$\rho(x) = 0$$
Particle Pushing

If this is the first time-step, velocities are moved backwards by half a time-step, for time-centering purposes:

\[ v_x^{-1/2} = v_x^0 - \frac{\Delta t}{2} \cdot qE_x^w / m \]  \hspace{1cm} (m = 1 in the code)

\( E_x^w \) is the area-weighted field at the location of each particle.

For the bunched beam case, when a linear focusing force is applied, one obtains:

\[ v_x = v_x^0 - \frac{\Delta t}{m} K \left( x - x_{\text{initial}} \right) \]

\[ k = v_0^2 = v^2 + \omega_0^2 \] (defined previously)

\( m = 1 \), in our code.

At this point, various diagnostics can be performed on the particles, as follows:

- Kinetic, thermal and drift energies are calculated and printed.
- Phase-space graphs are printed on-line.
- Velocity and density distributions are calculated.

Next, the actual particle pushing is done in a few steps:

(a) Area-weighted fields \( E_x^w \) are calculated at each particle position (center of particle).

(b) The velocities are advanced:

\[ \Delta v_x = \frac{q\Delta t}{m} \cdot E_x^w \]

\[ v_x' = v_x + \Delta v_x \]

(c) The positions are advanced:

\[ x' = x + v_x' \cdot \Delta t \]

(d) Check new positions:

Particles exiting the simulation system are pushed back, as following:

\[ x' = x - NX, \text{ for particles exiting to the right.} \]
\[ x' = x + NX, \text{ for particles exiting to the left.} \]

Therefore, in principle loss of particles should never happen, unless particles are pushed too far to the right or to the left.
The new charge distribution $\rho(x)$ is calculated at cell center by area-weighting of particles in their new positions. The boundary conditions are established:

$$\rho(l) \rightarrow \rho(l) + \rho(NX + 1)$$

Finally, the field diagnostics are performed, calculating and printing the Fourier components and the field energy. The time-integration loop is repeated for as many time steps as needed.

Because of the peculiarity of this 1-d code, which uses $E_x = \sum Z_k \rho_k e^{ikx}$ for the space-charge electric field, instead of the usual Poisson equation: $dE/dx = \rho(x)$, the electric field energy is also modified, as follows:

$$\text{Field Energy} = \frac{1}{2} \sum_k \rho_k^* \phi_k$$
$$= \frac{1}{2} \sum_k \rho_k^* \frac{E_k}{\lambda_k}$$

where $E_k$ is the electric field component, and $\rho_k$ is the complex coefficient in the expansion:

$$\rho(x) = \sum_k \rho_k e^{ikx}$$

Let:

$$E_k = Z_k \rho_k$$

Then:

$$\text{Field Energy} = \frac{1}{2} \sum_k \frac{\rho_k^* Z_k \rho_k}{\lambda_k}$$
$$= \frac{1}{2} \sum_k \left| \rho_k \right|^2 \frac{Z_k}{\lambda_k}$$

The reality of $E_x(x)$ implies:

$$Z_k^* = Z_{-k}$$
Therefore:

\[
\text{Field Energy} = \frac{1}{2} \sum_{k > 0} |p_k|^2 \left[ \frac{Z_k}{1k} - \frac{\bar{Z}_k}{1k} \right]
\]

Finally

\[
\text{Field Energy} = \sum_{k > 0} |p_k|^2 \text{Im} \frac{Z_k}{k}
\]

In particular, if \( \text{Im} Z_k = k \), then:

\[
\text{Field Energy} = \sum_{k > 0} |p_k|^2
\]

The resistive force does work on the system at a rate of:

\[
\frac{dE_{\text{Res}}}{dt} = 2 \text{Re} \sum_{k > 0} p_k^* \left( \frac{R}{1k} \right) \dot{p}_k
\]

The total energy conservation is expressed as:

\[
\text{Kinetic Energy} + \text{Field Energy} + \frac{1}{2} \int_0^X v_{\text{ext}} \rho dx + \int_0^t dt \frac{dE_{\text{Res}}}{dt} = \text{constant (at time t)}
\]

### III. Conversion from Code Units to Physical Units

As an example, let us consider a case with a space-charge force of the form \(-g \frac{\partial p}{\partial x}\). Let us define a phase velocity \(v_p\) in code units:

\[
v_p = \sqrt{\frac{q^2 n}{m}} \quad \text{(all quantities in code units)}
\]

(The relation between this \(v_p\) and \(\omega_p\) defined in Section II-(b) is: \(\omega_p^2 = 4\pi v_p^2\))

\(n\) is the number of simulation particles per cell, \(q\) is the charge per particle and \(m\) is 1. \((\rho \equiv qn)\)

Then, the equation of motion used by the computer is (all quantities are expressed in computer units):

\[
m \frac{d^2 x}{dt^2} = -q^2 (g \frac{dn}{dx} + R \cdot n) \quad \text{(code units; } g = 1)\]
\[ R = R' \cdot v_B, \] where \( R' \) is the resistance per unit length, and \( v_B \) is the beam velocity. This equation is also true in CGS units.

In MKS-C units, the phase velocity becomes:

\[ v_p = \sqrt{\frac{\epsilon_0}{m}} \cdot v, \]

and the equation of motion becomes:

\[ m \frac{d^2\bar{x}}{dt^2} = -\frac{q^2}{4\pi \epsilon_0} \left( g \frac{dn}{dx} + 4\pi \epsilon_0 R \cdot \bar{n} \right), \quad \left( \bar{R} \equiv R' \cdot \bar{v}_B \right) \]

All barred quantities are in MKS-C units.

Let: \( \bar{x} = r_L \cdot x \)

\( \bar{t} = r_T \cdot t \)

Then: \( \bar{v}_p = r_L \cdot r_T^{-1} v_p \)

\[ \bar{R} = r_L^{-1} \cdot \frac{R}{4\pi \epsilon_0} \]

\[ \bar{R} = r_L^{-1} \cdot \bar{R} \]

Therefore, this is the simplest possible correspondence between code units and MKS-C units, which gives:

\[ \frac{R}{k g} = \frac{4\pi \epsilon_0 \bar{R}}{k g} \]

The ratio between resistive and space-charge forces is kept the same, both in code and MKS-units.

Therefore:

\[ \bar{v} = r_L \cdot r_T^{-1} v \quad \text{(for any velocity)} \]

and \( \bar{\omega} = r_T^{-1} \omega \), where \( \omega = \omega_k + i \gamma_k \)

\( \gamma_k \) is the growth rate and \( \omega_k \) is the real frequency for mode \( k \) (unbarred quantities are in code units).
In order to interpret a simulation, one may first find $r_L$ from:

$$ R = r_L^{-1} k $$

For finite physical and computer systems (of lengths $l$, $L$ respectively), we have:

$$ r_{\min} = r_L^{-1} k_{\min}, \text{ or } r_L = \frac{l}{L} k_{\min}, $$

where $k_{\min} = \frac{2\pi}{L}$, $r_{\min} = \frac{2\pi}{l}$, and this $r_L$ value is valid for the entire $k$-spectrum.

Once $r_L$ is known, $R$ (to be used in the simulation) is found from:

$$ R = r_L \cdot 4\pi\varepsilon_0 \cdot \bar{R} $$

Then:

$$ r_T = r_L (v_p/\bar{v}_p), $$

where $v_p$ and $\bar{v}_p$ are given quantities. Therefore $\bar{v}$, $\bar{\omega}_k$ or $\bar{\gamma}_k$ can be found, once the corresponding simulation values are known, by using (A).

One can change the parameter $q$ in the simulation, in order to avoid problems caused by initial noise or nonlinear effects, for certain ranges of $R$. This will change $r_T$, if $r_L$ remains the same. If we want to change $r_L$ by requiring a certain value for $R$, then (for given $R$), the simulation length (number of simulation cells) will have to be changed accordingly, to preserve the ratio between the resistive and space-charge forces. However, the simulation length (and also the total number of particles in the system) cannot be changed too much (at most by an order of magnitude).

Also, $q$ or $\Delta t$ cannot be increased too much, if we require (to avoid numerical instabilities):

$$ \Delta v_x \leq v_x $$

and

$$ v_x \Delta t \leq \Delta x, \text{ where } \Delta x \text{ is the length of a simulation cell}. $$

Therefore, in the case where $q$ or $\Delta t$ or the simulation length cannot be changed too much, it might become necessary to perform lengthy simulations for
weak instabilities, which usually require either a large number of particles or a sophisticated method to produce quiet starts with relatively modest number of simulation particles.

References


(2) L. Smith et al., LBL Reports HIFAN 13, 14 and 15 (1977).


(4a) A. Hofmann, CERN 77-13, p. 159 (July 19, 1977).


(5) P. J. Channell et al., LBL-11367, July 1980.

Acknowledgement

This work was supported by the Director, Office of Energy Research, Office of Inertial Fusion, Research Division of the U.S. Department of Energy under Contract No. W-7405-ENG-48.
Fig. 1: Flow Chart of the Simulation Code

**Initialization:** $f(x, v_x)$

- **Find $\rho(x)$**

Start Main Time-Loop

- **Find Electric Field:**
  \[ E \sim -\partial\rho/\partial x \]
- Add in Resistive Force
- Add in External Fields

**Focus**

**Particle Diagnostics**

**Push**

- **Find $\rho(x)$**

**Field Diagnostics**

$t > t + \Delta t$