

## Hybrid macroparticle algorithm for modeling space charge

Paul M. Jung<sup>1</sup>,\* Thomas Planche<sup>2</sup>, and Richard Baartman<sup>3</sup>

*TRIUMF, 4004 Wesbrook Mall, Vancouver, British Columbia V6T 2A3 Canada  
and University of Victoria, 3800 Finnerty Road, Victoria, British Columbia V8P 5C2 Canada*

 (Received 22 April 2021; revised 8 July 2022; accepted 15 August 2022; published 29 August 2022)

A space charge algorithm is constructed that is a hybrid between envelope and multiparticle models. The transverse dynamics is simplified by tracking the transverse envelope of each macroparticle. The equations of motion are derived from the Hamiltonian-fluid formulation of the Vlasov Poisson system. A novel electrostatic self-field solver is derived that solves directly for the self-consistent on-axis potential and linear defocusing force, including longitudinal and beam pipe boundary conditions. The implementation of the algorithm, HYPER3D, is presented. It is tested against the particle-in-cell code WARP, using various configurations of a periodic focusing structure with rf cavities. The required number of macroparticles is reduced substantially compared to standard particle-tracking codes. This model is well adapted to cases where the transverse dynamics is linear and where the details of the longitudinal dynamics are the principal point of interest; an example is where a dc beam is converted to a bunched beam.

DOI: [10.1103/PhysRevAccelBeams.25.084602](https://doi.org/10.1103/PhysRevAccelBeams.25.084602)

### I. INTRODUCTION

Many reduced dimension models exist to study space-charge effects in accelerators. Commonly, symmetries can be exploited to reduce the dimensionality of the beam distribution. Then, to discretize the algorithm, the particle distribution is either described by a set of macroparticles which are statistical samples of the distribution or alternatively, by a set of statistical moments.

The envelope model including space charge derived by Sacherer in Ref. [1] is unmatched in computational efficiency for modeling beams with space charge. This has many benefits including fast optimization and fast fitting making it ideal for online model-based tuning and iterative design, as implemented in TRANSOPTR [2–4]. However, the envelope model is limited to describing linear forces from optics and space charge. For applications outside this scope, significantly more complicated multiparticle simulations are required.

In contrast to moment methods, multiparticle codes model the beam using many computational particles. These computational particles can be considered as statistical samples of the beam distribution, introducing statistical noise into the calculation. Many multiparticle codes use reduced complexity descriptions of the beam by using lower dimensional particles or by exploiting a symmetry,

yielding performance advantages. The reduced model, SPUNCH [5], was designed to study longitudinal dynamics. It tracks uniform discs of charge, with a fixed radius in longitudinal phase space. However, this cannot self-consistently model the space charge effects of both longitudinal and transverse focusing; the transverse size needs to be prescribed. A reduced model for electron accelerators with axial symmetry, HOMDYN [6], tracks discs of charge with a varying radius in 4D radial and longitudinal phase space. The reduction made by HOMDYN is limited to cylindrically symmetric optics: it cannot model quadrupoles. The more standard approach is to use a 6D phase-space multiparticle simulation such as the fully featured particle-in-cell code WARP [7]. It tracks macroparticles in 6D phase space but also features reduced dimension macroparticles and field solvers.

This paper presents the derivation and implementation of a hybrid simulation that utilizes both a macroparticle and moment based discretization of the beam. The system of equations is derived using the Poisson bracket formalism for the Maxwell Vlasov system derived by Morrison [8–10]. This mathematical formalism was applied to the Vlasov system in the context of plasmas and accelerators by Evstatiev and Shadwick, [11,12]. Before then, Shadwick also proposed the use of this formalism to describe the envelope model for accelerators [13]. This method was later successfully applied to model a plasma-laser interaction in 1D using a grid-based approach [14]. Shadwick also proposed a “semi-discrete” model in the conclusion of Ref. [13], which was never implemented, until now.

This hybrid scheme describes the beam in 6D phase space while reducing the detail of the beam’s distribution transversely. This reduction is designed specifically for the

\*pjung@triumf.ca

*Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI.*

case where the transverse dynamics is well described by linear optics, but the longitudinal is not.

First, the Poisson bracket formalism is described and the discretization scheme is presented. The discrete system is reduced to a Poisson system of ordinary differential equations, which is later cast into canonical form for symplectic integration. The algorithm is implemented in the RUST programming language and features multithreading. The code, HYPER3D, is distributed by Ref. [15] with a GNU LGPLv3 license.

## II. HAMILTONIAN FLUIDS FOR PARTICLE DISTRIBUTIONS

Consider a single species of a particle with mass  $m$  and charge  $q$ . Any collection of these particles is completely described by the particle density function of the 6D phase-space coordinates:  $f(\mathbf{x}, \mathbf{p}, t)$ . The Hamiltonian of this distribution of particles is the single particle Hamiltonian, in this case, nonrelativistic, integrated over the distribution:

$$H = \int d^3\mathbf{x} d^3\mathbf{p} f(\mathbf{x}, \mathbf{p}, t) \left( \frac{1}{2m} [\mathbf{p} - q\mathbf{A}(\mathbf{x}, t)]^2 + q\phi(\mathbf{x}, t) \right), \quad (1)$$

where the single particle Hamiltonian depends on the electric potential  $\phi$  and magnetic vector potential  $\mathbf{A}$ .

Physical quantities of interest can be written as functionals of the particle distribution function. The Hamiltonian is one such functional, and as we will see, statistical moments are functionals as well. The Poisson bracket operates on functionals of  $f$ . For example, let  $F[f]$  and  $G[f]$  be arbitrary functionals of  $f$ , then the Poisson bracket between them is

$$\{F, G\} = \int d^3\mathbf{x} d^3\mathbf{p} f(\mathbf{x}, \mathbf{p}, t) \left[ \frac{\delta F}{\delta f}, \frac{\delta G}{\delta f} \right], \quad (2)$$

where “[ $\cdot, \cdot$ ]” is the canonical phase-space bracket with explicit form:

$$[a, b] = \nabla_{\mathbf{x}} a \cdot \nabla_{\mathbf{p}} b - \nabla_{\mathbf{p}} a \cdot \nabla_{\mathbf{x}} b, \quad (3)$$

for phase-space functions  $a(\mathbf{x}, \mathbf{p})$  and  $b(\mathbf{x}, \mathbf{p})$ . The variational derivative of a functional is given by

$$\frac{\delta}{\delta f} \int d^3\mathbf{x} d^3\mathbf{p} a(\mathbf{x}, \mathbf{p}) f(\mathbf{x}, \mathbf{p}, t) = a(\mathbf{x}, \mathbf{p}), \quad (4)$$

where  $a(\mathbf{x}, \mathbf{p})$  is a function of the phase-space variables; see Appendix for more details.

## III. HYBRID MOMENT MACROPARTICLES

The particle distribution  $f$  is divided into  $N_p$  independent distributions:

$$f(\mathbf{x}, \mathbf{p}, t) = \sum_{i=1}^{N_p} f_i(\mathbf{x}, \mathbf{p}, t). \quad (5)$$

Each of these groupings is a macroparticle which we assume has the form:

$$f_i(\mathbf{x}, \mathbf{p}, t) = w_i f_i^\perp(\mathbf{x}^\perp, \mathbf{p}^\perp, t) R(z - \langle z \rangle_i) \delta(p_z - \langle p_z \rangle_i), \quad (6)$$

where  $w_i$  is the weight of the  $i$ th macroparticle and  $f_i^\perp$  is the transverse distribution of the macroparticle, which remains general. Longitudinally, the macroparticle is described by its centroid:  $(\langle z \rangle_i, \langle p_z \rangle_i)$ . It has no longitudinal momentum spread but is distributed in  $z$  using a particle kernel function  $R$ , with a fixed width. The kernel function can be chosen to be a Dirac delta distribution, or an arbitrary-sized localized distribution, allowing for different amounts of smoothing [[16] Chap. 5]. The longitudinal macroparticle discretization is identical to the one presented in Ref. [11], whereas the moment component of the discretization is a simplified version of the discretization presented in Ref. [13].

The brackets,  $\langle \cdot \rangle_i$ , denote a quantity averaged over the distribution function  $f_i$ , as in

$$\langle a \rangle_i = \frac{1}{w_i} \int d^3\mathbf{x} d^3\mathbf{p} f_i(\mathbf{x}, \mathbf{p}) a(\mathbf{x}, \mathbf{p}), \quad (7)$$

where the  $1/w_i$  term normalizes by the number of particles in the macroparticle.

The envelope method tracks the set of second statistical moments of a beam distribution in phase space. Sacherer showed that the system of equations that tracked these second moments was self-consistent and closed if only linear forces were considered and that the linear force of the self-field only depended on the second moments [1,17]. Here, we treat the transverse distribution of each macroparticle using the general moment approach, a generalization of the envelope method derived in Ref. [13]. We assume that  $f_i^\perp(\mathbf{x}^\perp, \mathbf{p}^\perp, t)$  is centered on the axis so that the first moments are zero. Furthermore, for the purposes of this paper, we take the  $x$  and  $y$  phase spaces to be uncorrelated. Thus, the hybrid macroparticle, labeled  $i$ , is parameterized by the following coordinates:

$$X_i = (\langle x^2 \rangle_i, \langle x p_x \rangle_i, \langle p_x^2 \rangle_i, \langle y^2 \rangle_i, \langle y p_y \rangle_i, \langle p_y^2 \rangle_i, \langle z \rangle_i, \langle p_z \rangle_i). \quad (8)$$

### A. Reduced Poisson bracket

After using Eq. (5) to split the distribution function into many smaller distributions, the Poisson bracket becomes a summation:

$$\{F, G\} = \sum_{i=1}^{N_p} \int d^3\mathbf{x} d^3\mathbf{p} f_i(\mathbf{x}, \mathbf{p}, t) \left[ \frac{\delta F}{\delta f_i}, \frac{\delta G}{\delta f_i} \right]. \quad (9)$$

This Poisson bracket, with a given Hamiltonian, can be used to directly compute the equations of motion using functional derivatives without loss of generality. However, the bracket may be simplified to a reduced form in terms of discrete variables and partial derivatives, as follows. Let  $F[f]$  and  $G[f]$  be functionals of  $f$ , if they can be written as functions of coordinates, called  $\hat{F}(X_i)$  and  $\hat{G}(X_i)$ , respectively, we may compute the Poisson bracket using the following:

$$\{\hat{F}, \hat{G}\} = \sum_{i=1}^{N_p} (\nabla_{X_i} \hat{F})^\top \cdot \mathbf{B}_i(X_i) \cdot (\nabla_{X_i} \hat{G}), \quad (10)$$

where  $\mathbf{B}_i(X_i)$  is called the Poisson structure matrix, it is a generalization of the symplectic matrix and is computed by taking the Poisson bracket between each pair of coordinates. For a thorough mathematical treatment of the Poisson system, refer to [[18] Chap. VII Sec. 2].

The equations of motion follow from applying the Poisson bracket to the Hamiltonian and the coordinates in the usual way

$$\dot{X}_i = \{X_i, \hat{H}\} = \mathbf{B}_i(X_i) \cdot (\nabla_{X_i} \hat{H}), \quad (11)$$

where  $\hat{H}$  is the Hamiltonian written in terms of the discrete coordinates. For a set of coordinates that are canonically conjugate, this matrix will be the symplectic matrix. For our case, the matrix is partitioned into blocks:

$$\mathbf{B}_i = \begin{bmatrix} \mathbf{B}_i^x & 0 & 0 \\ 0 & \mathbf{B}_i^y & 0 \\ 0 & 0 & \mathbf{B}_i^z \end{bmatrix}, \quad (12)$$

where each of the block matrices is

$$\mathbf{B}_i^x = \frac{1}{w_i} \begin{bmatrix} 0 & 2\langle x^2 \rangle_i & 4\langle xp_x \rangle_i \\ -2\langle x^2 \rangle_i & 0 & 2\langle p_x^2 \rangle_i \\ -4\langle xp_x \rangle_i & -2\langle p_x^2 \rangle_i & 0 \end{bmatrix}, \quad (13)$$

$$\mathbf{B}_i^y = \frac{1}{w_i} \begin{bmatrix} 0 & 2\langle y^2 \rangle_i & 4\langle yp_y \rangle_i \\ -2\langle y^2 \rangle_i & 0 & 2\langle p_y^2 \rangle_i \\ -4\langle yp_y \rangle_i & -2\langle p_y^2 \rangle_i & 0 \end{bmatrix}, \quad (14)$$

$$\mathbf{B}_i^z = \frac{1}{w_i} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (15)$$

The structure matrix,  $\mathbf{B}_i$ , is antisymmetric and it can be used to verify that the reduced Poisson bracket satisfies the Jacobi identity.

## B. Discrete Hamiltonian

The Hamiltonian, Eq. (1), is a functional of  $f(\mathbf{x}, \mathbf{p}, t)$  in general. However, in order to use Eq. (11) to find the discrete equations of motion, we need to find  $\hat{H}$ , a function of the discrete coordinates. To make this reduction, the Hamiltonian must be Taylor expanded in the transverse directions, about the reference trajectory. It is then truncated to include only terms up to second order. This guarantees that the discrete system of equations relies on moments only up to second order. This procedure follows exactly the usual Hamiltonian approach to linear optics [19]. For a more general recipe, Shadwick derives a relativistic discrete Hamiltonian using a variational approach in Ref. [13].

For our case, we expand Eq. (1) in the electrostatic and nonrelativistic limits, yielding

$$H = \int d^3\mathbf{x} d^3\mathbf{p} f(\mathbf{x}, \mathbf{p}, t) \left( \frac{|\mathbf{p}|^2}{2m} + q\phi(z, t) + qx^2\varphi_{xx}(z, t) + qy^2\varphi_{yy}(z, t) \right), \quad (16)$$

where we have defined the on-axis potentials:

$$\phi(z, t) = \phi(\mathbf{x}, t)|_{\mathbf{x}^\perp=0} \quad (17)$$

$$\varphi_{xx}(z, t) = \frac{\partial^2 \phi(\mathbf{x}, t)}{\partial x^2} \Big|_{\mathbf{x}^\perp=0} \quad (18)$$

$$\varphi_{yy}(z, t) = \frac{\partial^2 \phi(\mathbf{x}, t)}{\partial y^2} \Big|_{\mathbf{x}^\perp=0} \quad (19)$$

neglecting all steering and  $xy$ -correlating terms, for simplicity. Finally, substituting the discrete form of the particle distribution and integrating give the discrete Hamiltonian:

$$\hat{H} = \sum_{i=1}^{N_p} \left[ \frac{w_i}{2m} (\langle p_x^2 \rangle_i + \langle p_y^2 \rangle_i + \langle p_z \rangle_i^2) + qw_i \int dz R(z - \langle z \rangle_i) (\phi + \langle x^2 \rangle_i \varphi_{xx} + \langle y^2 \rangle_i \varphi_{yy}) \right]. \quad (20)$$

## IV. ELECTROSTATIC SPACE CHARGE SOLVER

The electrostatic space charge solver is based on the uniform linear finite element discretization in the longitudinal direction. For a grid with  $N_g$  points and spacing  $h$ , the linear finite element corresponding to grid point  $\alpha$  is

$$\psi_\alpha(z) = \begin{cases} 1 - \frac{1}{h}|z - z_\alpha| & |z - z_\alpha| < h/2 \\ 0 & \text{else,} \end{cases} \quad (21)$$

where  $z_\alpha$  is the coordinate of the grid point. The self-potential is now discrete in the longitudinal direction and left to be general transversely

$$\phi(\mathbf{x}, t) = \sum_{\alpha=1}^{N_g} \varphi_\alpha(\mathbf{x}^\perp, t) \psi_\alpha(z). \quad (22)$$

Under this discretization, Poisson's equation becomes a system of partial differential algebraic equations for the set of transverse fields,  $\varphi_\alpha(\mathbf{x}^\perp, t)$ , as follows:

$$\sum_{\beta=1}^{N_g} (M_{\alpha\beta} \Delta_{\mathbf{x}^\perp} - D_{\alpha\beta}) \varphi_\beta(\mathbf{x}^\perp, t) = -\frac{1}{\epsilon_0} \rho_\alpha(\mathbf{x}^\perp, t), \quad (23)$$

for all  $\alpha = 1, \dots, N_g$  and where  $\Delta_{\mathbf{x}^\perp}$  denotes the 2D Laplacian with respect to  $\mathbf{x}^\perp$ . The finite element mass matrix  $\mathbf{M}$  and stiffness matrix  $\mathbf{D}$  appear in these equations. The components of these matrices are:

$$M_{\alpha\beta} = \int dz \psi_\alpha(z) \psi_\beta(z),$$

$$D_{\alpha\beta} = \int dz \frac{d\psi_\alpha(z)}{dz} \frac{d\psi_\beta(z)}{dz},$$

which are all constants that depend on the grid spacing  $h$ . The  $\mathbf{D}$  matrix denotes the discrete second derivative.

After factoring out  $h$ , the matrices  $\mathbf{M}$  and  $\mathbf{D}$  are simultaneously diagonalizable. This property applies to both metallic and periodic boundary conditions [20]. We call the orthogonal matrix of the eigenvectors of both matrices  $\mathbf{S}$ . The eigenvalues of  $\mathbf{M}$  are called  $\lambda_\alpha^M$  and similarly for  $\mathbf{D}$ ,  $\lambda_\alpha^D$ , all of which are positive real numbers [20].

The general solution to Poisson's equation in this discretization scheme for an arbitrary charge distribution is

$$\phi(\mathbf{x}, t) = \sum_{\alpha,\beta,\gamma=1}^{N_g} S_{\alpha\beta} S_{\beta\gamma} \psi_\alpha(z) (G_\beta * \rho_\gamma)(\mathbf{x}^\perp), \quad (24)$$

where  $\rho_\gamma(\mathbf{x}^\perp, t)$  is the transverse charge density at the  $\gamma$ th grid point. Also,  $(G_\beta * \rho_\gamma)$  denotes the 2D convolution between the charge density and the Green's function

$$(G_\beta * \rho_\gamma)(\mathbf{x}^\perp) = \int d^2\xi G_\beta(\mathbf{x}^\perp - \xi) \rho_\gamma(\xi). \quad (25)$$

The Green's function is the potential of a Dirac delta point charge which vanishes at infinity:

$$G_\alpha(\mathbf{x}^\perp) = -\frac{1}{4\pi\epsilon_0 h \lambda_\alpha^M} K_0 \left( \sqrt{\frac{\lambda_\alpha^D}{h^2 \lambda_\alpha^M}} |\mathbf{x}^\perp| \right), \quad (26)$$

where  $K_0$  is the zeroth order modified Bessel function of the second kind. Additionally, the image charge term for

the beam pipe may be computed by reflecting the charge distribution about the 2D circular domain in the usual way.

The full 3D potential is given by the above formulas, but only the on-axis potential and defocusing force need to be calculated for the transverse linear optics. So, the on-axis potential is

$$\varphi(z, t) = \sum_{\alpha,\beta,\gamma=1}^{N_g} S_{\alpha\beta} S_{\beta\gamma} \psi_\alpha(z) (G_\beta * \rho_\gamma)(\mathbf{x}^\perp = \mathbf{0}). \quad (27)$$

Similarly, the on-axis linear defocusing force is given explicitly by taking partial derivatives of the convolution and then evaluating on axis, yielding

$$\varphi_{xx}(z, t) = \sum_{\alpha,\beta,\gamma=1}^{N_g} S_{\alpha\beta} S_{\beta\gamma} \psi_\alpha(z) \frac{\partial^2 (G_\beta * \rho_\gamma)}{\partial x^2}(\mathbf{x}^\perp = \mathbf{0}), \quad (28)$$

$$\varphi_{yy}(z, t) = \sum_{\alpha,\beta,\gamma=1}^{N_g} S_{\alpha\beta} S_{\beta\gamma} \psi_\alpha(z) \frac{\partial^2 (G_\beta * \rho_\gamma)}{\partial y^2}(\mathbf{x}^\perp = \mathbf{0}). \quad (29)$$

These sums are computed assuming that the beam is centered on the axis, and at any given slice in  $z$ , the transverse distribution of the beam is a uniform elliptical distribution. Given these assumptions, each 2D convolution can be simplified to a 1D numerical integral, which is evaluated using the trapezoid rule, in the implementation.

The algorithm for solving the self-field is as follows: First, the average beam size and charge are computed at each grid point taking  $\mathcal{O}(N_g + N_p)$  computational time. Then, the  $\mathcal{O}(N_g^2)$  convolution terms are computed. Finally, the summations with the  $\mathbf{S}$  matrices take  $\mathcal{O}(N_g^3)$  computation time.

## V. INTEGRATION

Similar to other PIC codes, this algorithm relies on using an explicit symplectic integrator, despite not being symplectic overall. As pointed out by Evstatiev [11], PIC algorithms neglect critical terms on the order of accuracy of the discretization, which leads to loss of energy conservation, even with arbitrarily small time steps. In contrast, energy-conserving methods are derived from variational principles in order to retain these critical terms [11]. The algorithm, derived in this work, makes the same kind of approximation as conventional PIC codes and is not symplectic.

To integrate the macroparticles using a typical symplectic integrator, they must be transformed into a set of canonical coordinates and conserved quantities. A Poisson system is a set of coordinates equipped with a Poisson bracket that can be used to find the equations of motion. For any Poisson system, there exists a coordinate transformation to a set of canonical coordinates and conserved quantities, this is the result of the Darboux-Lie theorem [18]. In our case, each macroparticle is

a Poisson system formed by its coordinates, the set of moments, Eq. (8), and the discrete Poisson bracket, Eq. (10). Thus, there exists a canonical coordinate system for the macroparticles. In general, finding the set of canonical coordinates is nontrivial.

First, recall that for linear optics, the rms emittance is a conserved quantity. Similarly, the conserved quantities are the  $x$  and  $y$  emittance of each macroparticle

$$\mathcal{E}_i^x(X_i) = \sqrt{\langle x^2 \rangle_i \langle p_x^2 \rangle_i - \langle xp_x \rangle_i^2}, \quad (30)$$

$$\mathcal{E}_i^y(X_i) = \sqrt{\langle y^2 \rangle_i \langle p_y^2 \rangle_i - \langle yp_y \rangle_i^2}. \quad (31)$$

Note that these conserved quantities,  $\mathcal{E}_i^x$  and  $\mathcal{E}_i^y$ , must be initialized for each macroparticle and then remain unchanged throughout the entire simulation. Then we can identify the canonical coordinates of each macroparticle as

$$\begin{aligned} \tilde{X}_i &= (Q_i^x, P_i^x, Q_i^y, P_i^y, Q_i^z, P_i^z) \\ &= \left( \sqrt{\langle x^2 \rangle_i}, w_i \frac{\langle xp_x \rangle_i}{\sqrt{\langle x^2 \rangle_i}}, \sqrt{\langle y^2 \rangle_i}, w_i \frac{\langle yp_y \rangle_i}{\sqrt{\langle y^2 \rangle_i}}, \right. \\ &\quad \left. \langle z \rangle_i, w_i \langle p_z \rangle_i \right), \end{aligned} \quad (32)$$

Since these are canonical pairs, the structure matrix of these coordinates,  $\tilde{\mathbf{B}}(\tilde{X}_i)$  is the symplectic matrix. The factors of  $w_i$  indicate that the canonical momenta are the total mechanical momentum of all the particles represented by the macroparticle.

The discrete Hamiltonian is also coordinate transformed to find the final Hamiltonian

$$\begin{aligned} \hat{H} &= \sum_{i=1}^{N_p} \left[ \frac{w_i}{2m} \left( (P_i^x)^2 + (P_i^y)^2 + (P_i^z)^2 + \left[ \frac{\mathcal{E}_i^x}{Q_i^x} \right]^2 + \left[ \frac{\mathcal{E}_i^y}{Q_i^y} \right]^2 \right) \right. \\ &\quad \left. + qw_i \int dz R(z - Q_i^z) (\varphi + (Q_i^x)^2 \varphi_{xx} + (Q_i^y)^2 \varphi_{yy}) \right]. \end{aligned} \quad (33)$$

Finally, the equations of motion can be obtained from Hamilton's equations. In the code HYPER3D, integration is done using the asynchronous leapfrog method [16] and also provides the symplectic Euler and Strömer-Verlet integration schemes for comparison.

## VI. VALIDATION

The primary motivation for this algorithm is to self-consistently model the process of bunching, the transition from a dc to a bunched beam. This process occurs in the injection line to the TRIUMF 500 MeV cyclotron. The dc beam of  $H^-$  ions is extracted at 300 keV and is transported

down an electrostatic quadrupole lattice; bunching is driven by 2 two-gap rf bunchers.

### A. Electrostatic quadrupole lattice

For testing, the electrostatic quadrupole lattice of the injection system is simplified: it is approximated by a focus-drift-defocus-drift (FODO) lattice. The electrostatic quadrupoles have a length of 12.1 cm and an aperture radius of 2.54 cm and they are separated by a drift of 25.0 cm. The electrodes have an applied potential of  $\pm 3.5$  kV giving an integrated gradient of 6.56 kV/m. The dc beam of  $H^-$  ions at 300 keV has a current of 0.5 mA and a 4 rms geometrical emittance of 6.5  $\mu\text{m}$  in each direction.

The hybrid model is tested against the envelope code TRANSOPTR [3,21] and the multiparticle code WARP [7]. TRANSOPTR is run to model a 4D dc beam with space charge. For both particle-based codes, the simulation parameters were adjusted to achieve around 1% relative errors in the envelope compared to TRANSOPTR while minimizing the total simulation time. The code WARP is run with a Kapchinskij-Vladimirskij transverse distribution.

Both HYPER3D and WARP are initialized with a longitudinal thermal distribution with an rms momentum spread of 0.054%, corresponding to the source temperature of 2000 °C. The transverse distribution is initialized using the built-in envelope solver in WARP. For both WARP and HYPER3D, the initial beam and self-field grid initially span one FODO cell at the start of the lattice. The self-field grid is initialized with periodic longitudinal boundary conditions. The beam is tracked through the lattice until it has completely crossed the final view screen.

The WARP simulation uses 100,000 particles and a grid resolution of  $64 \times 64 \times 64$  in  $x$ ,  $y$ , and  $z$  respectively. The hybrid model uses 50 hybrid macroparticles with a self-field grid of 64 longitudinal grid points. Both macroparticle codes used 50-time steps per FODO lattice period.

The 2 rms envelope over the first five FODO lattice periods is shown in Fig. 1. The beam size calculated in both WARP and the hybrid model agrees with TRANSOPTR, with the maximum relative error of WARP being 1.7% and the maximum relative error of HYPER3D is 0.46% with respect to TRANSOPTR. The hybrid discretization scheme of HYPER3D is able to more accurately resolve the transverse linear optics with  $\mathcal{O}(N_p^{1/3})$  the number of macroparticles as WARP, or approximately one macroparticle per grid point.

### B. Quadrupole lattice with bunching

The same quadrupole lattice is extended to 28 periods, approximating the bunching distance of the injection line to the TRIUMF cyclotron. This model includes two dual gap rf bunchers operating at the first and second harmonics of the TRIUMF cyclotron, 22.61 MHz and 45.21 MHz respectively. The first buncher thus has  $\beta\lambda = 33.53$  cm and the second  $\beta\lambda = 16.77$  cm. The first buncher is placed

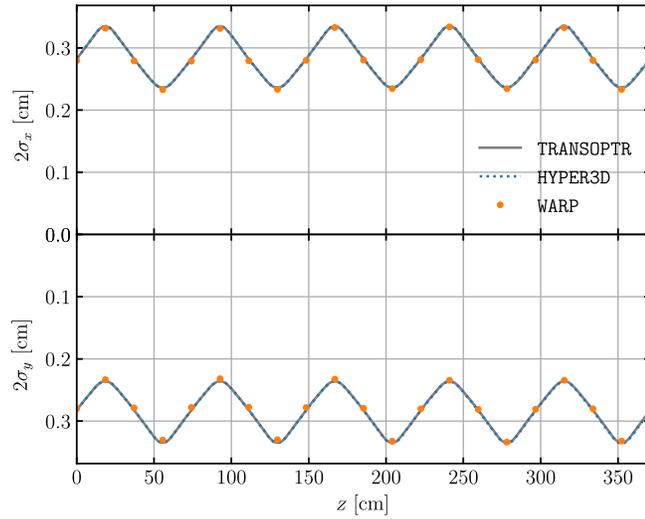


FIG. 1. Root mean square transverse beam size over the first five FODO lattice periods for different codes.

in the middle first drift of the third lattice period, the second buncher is placed 4.5 m downstream of the first.

Both bunchers are described by their longitudinal electric field on the axis. This is implemented into both HYPER3D and WARP as an axial multipole expansion of the electric field with a time-varying amplitude. The peak voltages of the bunchers are 4.704 and 2.450 kV, for the first and second buncher, respectively. These values were the result of an optimization of bunching done in Ref. [22].

The codes are initialized with the same parameters as the FODO test, with the following exceptions: both codes are run using 100,000 macroparticles and 128 longitudinal grid points. The 2 rms beam size is shown over the course of the bunching process in Fig. 2. The envelopes computed by

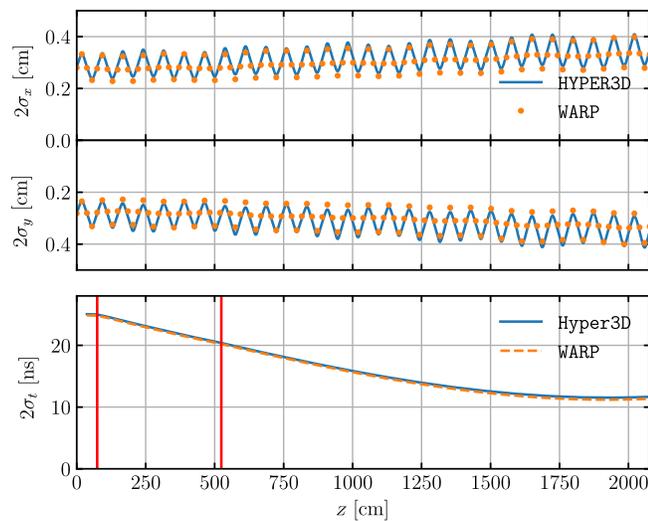


FIG. 2. The 2 rms beam envelope, over the bunching FODO lattice periods for different codes. The bunchers are centered at the locations indicated by the vertical lines.

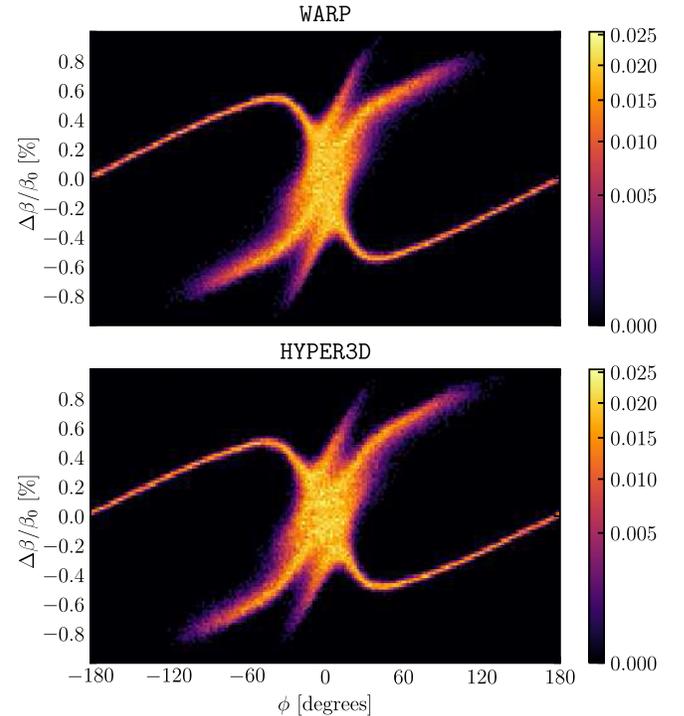


FIG. 3. The longitudinal phase space portrait measured at the final view screen for the WARP code (above) and HYPER3D (below) with 100,000 macroparticles. The color indicates the normalized particle density.

WARP and HYPER3D agree reasonably well, with a maximum relative difference of 7.9% transversely and 2.8% longitudinally. The phase space of the bunch at the final screen is shown in Fig. 3 for both codes and is nearly indistinguishable.

### C. Long lattice with longitudinal focusing

To test the long term stability of HYPER3D, the FODO lattice was extended to 1000 periods, a total length of 741 m. Additionally, to provide longitudinal focusing, a copy of the first-harmonic rf buncher was placed in the middle of every 10th FODO lattice. These rf bunchers were tuned to a frequency of 23.50 MHz so that each buncher would operate with the same phase and a voltage of 5.0 kV to provide adequate focusing.

The bunch is initialized with a  $200 \mu\text{m}$  longitudinal 4 rms emittance and contains 1.1 pC of charge. In HYPER3D, the bunched beam was initialized as a uniform 2D ellipse in longitudinal phase space with 500 macroparticles and 128 grid points. The transverse envelope of the beam was initialized using the output of TRANSOPTR with a quadratic taper in  $z$ , to represent the ellipsoidal shape. The total run time of this simulation was 221.3 s on the author's computer, using all four cores (see below). The emittance growth of the beam over the course of the simulation is shown in Fig. 4. The longitudinal emittance fluctuates about the initial emittance with a maximum deviation of no

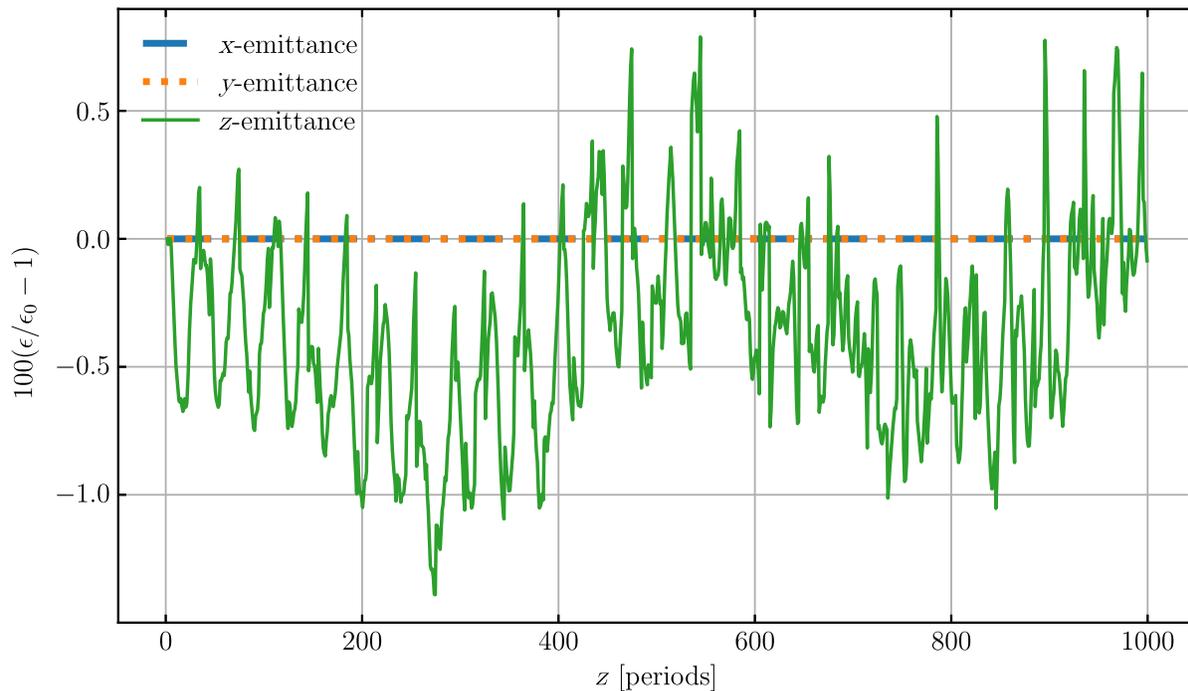


FIG. 4. The relative growth of the 4 rms emittance of a beam in a long focusing channel simulated in HYPER3D.

more than 1.5%. The transverse emittance is conserved exactly, as a consequence of the canonical coordinate transformation.

## VII. PERFORMANCE

The performance of HYPER3D was tested for algorithmic scaling, and the performance in the previous calculations was compared to the other codes. All computations were produced with a 3.5-GHz Intel quad-core desktop PC.

### A. Performance scaling

The single-threaded performance of HYPER3D was evaluated using the FODO lattice simulation as the benchmark to verify the algorithmic scaling. The average time to compute each time step for one run is calculated for many different discretization parameters and is shown in Fig. 5. The time to calculate one time-step scales linearly with the number of macroparticles and with grid points slightly more than quadratically: between  $\propto N_g^{2.1}$  and  $\propto N_g^{2.2}$ . This result is consistent with the self-field solver, showing that evaluating the  $N_g^2$  convolutions dominates the computation time of the self-field calculation.

### B. Quadrupole lattice with bunching

For the injection line simulation, both codes were run with equivalent discretization parameters for direct comparison purposes. The total run time for HYPER3D was 79.0 s and 94.4 s with WARP, both running multithreaded. However, note that the total run times are not directly

comparable since each simulation involves different post-processing computations. A more direct comparison is the average computation time per time step, which for WARP was 61.2 ms single-threaded, and 50.8 ms multithreaded, compared to HYPER3D running single-threaded 78.3 ms, and multithreaded 43.5 ms. For equivalent numbers of grid points and macroparticles, the codes HYPER3D and WARP have comparable performance.

### C. Electrostatic quadrupole lattice

To complete the FODO lattice calculation using the macroparticle codes, the average computation time per time step was 19.8 ms for WARP and 1.68 ms for HYPER3D, both were run multithreaded, with the same step size; taking 250-time steps for the beam to span the plotted region in Fig. 1. The disparity in computation time between WARP and HYPER3D, in this case, is due to the latter requiring 2000 times fewer macroparticles. In stark contrast, the total run time for TRANSOPTR was only 11 ms.

### D. Performance discussion

The FODO example demonstrates that the minimal number of macro-particles required for HYPER3D to resolve the transverse dynamics is approximately one per longitudinal grid point. This enables the code HYPER3D to run very fast coarse simulations with few macroparticles, limited only by the longitudinal sampling error. This trade-off is ideal for specific situations including design, optimization, training surrogate models such as in Ref. [23], and online models [24]. However, the hybrid discretization scheme

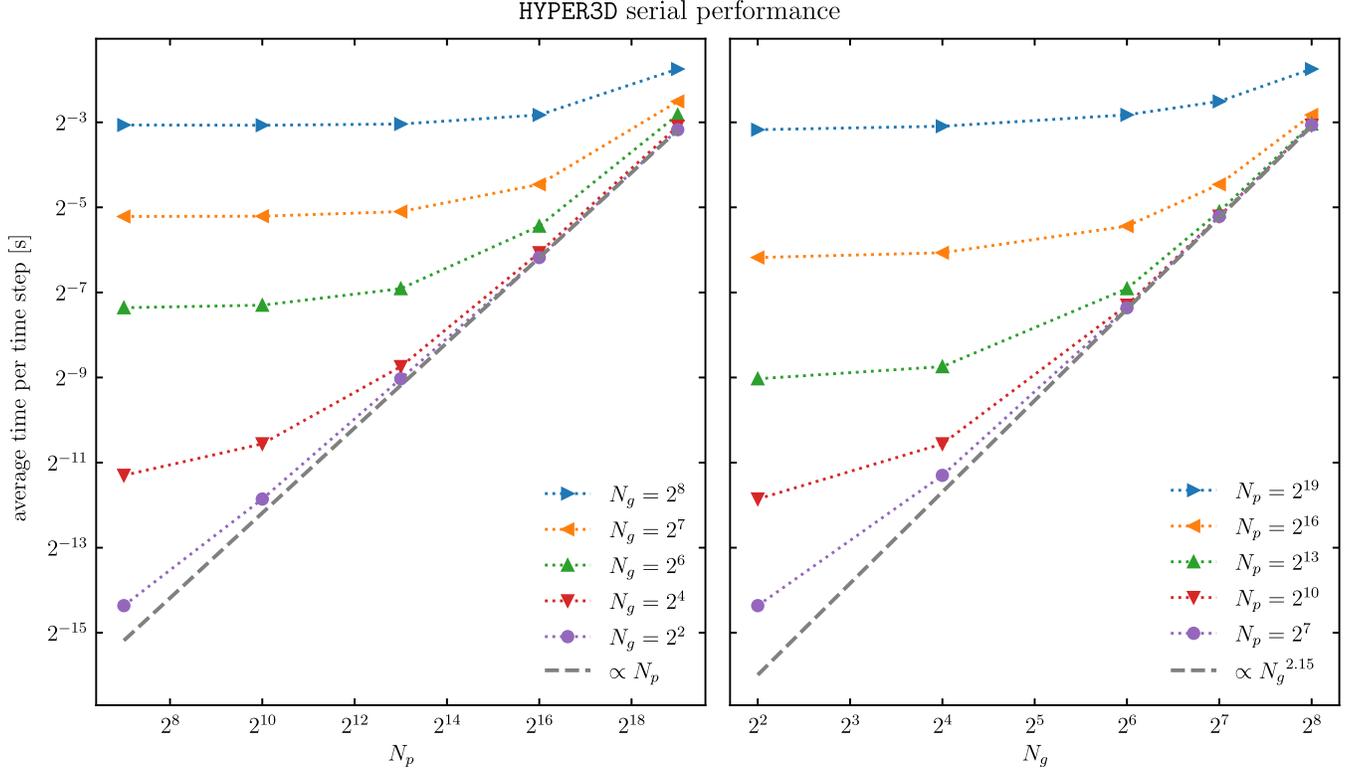


FIG. 5. The single-threaded performance of the HYPER3D code, measured as the average time in seconds per time step, for different numbers of grid points and macroparticles. The computation time scales linearly with the number of macroparticles and slightly more than quadratically with the number of grid points.

cannot be used to predict losses with high precision, similar to an envelope code. For large-scale simulations, the algorithmic scaling of HYPER3D is less advantageous compared to standard PIC codes.

### VIII. CONCLUSION

HYPER3D is the first implementation for modeling accelerators using a semireduced discretization: macroparticles with moments. It has a semianalytic self-field solver designed around transverse linear optics that solves for the on-axis potential and defocusing forces directly. The implementation was written in a low level language, RUST, with parallelism. It was tested on several examples including periodic focusing channels against the envelope code TRANSOPTR and the PIC code WARP. All examples showed excellent agreement, despite being fundamentally different approaches to describe the dynamics. The performance of the algorithm was quantified and found to be similar to the highly optimized code WARP for similar numbers of macroparticles and grid points. The performance scaling of HYPER3D was measured: the time stepping performance is linear in the number of macroparticles and slightly greater than quadratic in the number of longitudinal grid points.

The code was developed for online use in the control room to guide the tuning of the bunching system when

switching beam delivery modes in the TRIUMF 500 MeV. HYPER3D is able to run fast, coarse, simulations that are consistent with the transverse optics while yielding precise results about the injection efficiency, an ideal model for this kind of application.

### ACKNOWLEDGMENTS

TRIUMF is supported by a contribution through the NRC, Canada.

### APPENDIX: VARIATIONAL CALCULUS

For quantities averaged over particle distribution functions, variational differentiation is given by

$$\frac{\delta \langle a \rangle_i}{\delta f_j(\mathbf{x}', \mathbf{p}', t)} = \frac{a(\mathbf{x}', \mathbf{p}') - \langle a \rangle_i}{w_i} \delta_{ij}, \quad (\text{A1})$$

where  $\delta_{ij}$  is the Kronecker delta. For a general function of the coordinates  $\hat{F}(X_i)$ , the chain rule for functional derivatives can be used

$$\frac{\delta \hat{F}(X_i)}{\delta f_j(\mathbf{x}', \mathbf{p}', t)} = \nabla_{X_i} \hat{F}(X_i) \cdot \frac{\delta X_i}{\delta f_j(\mathbf{x}', \mathbf{p}', t)}, \quad (\text{A2})$$

where the remaining functional derivative can be computed by Eq. (A1).

- [1] F. Sacherer, RMS envelope equations with space charge, CERN, Technical Report No. CERN-SI-Int-DL-70-12, 1970.
- [2] R. Baartman, TRIUMF, Vancouver, Canada, Report No. TRI-BN-16-06, 2016.
- [3] E. Heighway and R. Hutcheon, *Nucl. Instrum. Methods Phys. Res.* **187**, 89 (1981).
- [4] E. Heighway and M. De Jong, TRANSOPTR—a beam transport design code with space charge, automatic internal optimization and general constraints, Atomic Energy of Canada Limited, Chalk River, Ontario, Technical Report No. AECL-6975 (rev.A), 1984.
- [5] R. Baartman, in *Proceedings of the Eleventh International Conference on Cyclotrons and their Applications* (1987).
- [6] M. Ferrario, J. Clendenin, D. Palmer, J. Rosenzweig, and L. Serafini, in *The Physics of High Brightness Beams* (World Scientific, Singapore, 2000), pp. 534–563.
- [7] A. Friedman, R. H. Cohen, D. P. Grote, S. M. Lund, W. M. Sharp, J. Vay, I. Haber, and R. A. Kishek, *IEEE Trans. Plasma Sci.* **42**, 1321 (2014).
- [8] P. J. Morrison, *Phys. Lett. A* **80**, 383 (1980).
- [9] A. Weinstein and P. J. Morrison, *Phys. Lett. A* **86**, 235 (1981).
- [10] P. J. Morrison, *AIP Conf. Proc.* **88**, 13 (1982).
- [11] E. G. Evstatiev and B. A. Shadwick, *J. Comput. Phys.* **245**, 376 (2013).
- [12] B. A. Shadwick, A. B. Stamm, and E. G. Evstatiev, *Phys. Plasmas* **21**, 055708 (2014).
- [13] B. Shadwick and J. Wurtele, in *Proceedings of the 18th Particle Accelerator Conference, New York, 1999*, (IEEE, New York, 1999), Vol. 4, pp. 2888–2890.
- [14] B. Shadwick, G. Tarkenton, E. Esarey, and F. M. Lee, *Commun. Nonlinear Sci. Numer. Simul.* **17**, 2153 (2012).
- [15] P. Jung, HYPER3D open source git repository, <https://gitlab.triumf.ca/pjung/hyper3d> (2022).
- [16] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (CRC Press, Boca Raton, Florida, 1988), 10.1201/9780367806934.
- [17] F. J. Sacherer, *IEEE Trans. Nucl. Sci.* **18**, 1105 (1971).
- [18] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations* (Springer, New York, 2006).
- [19] R. Baartman, in *Proceedings of the 28th Linear Accelerator Conference* (JACoW, Geneva, Switzerland, 2016), 10.18429/JACoW-LINAC2016-FR1A01.
- [20] P. M. Jung, Hybrid macro-particle moment accelerator tracking algorithm, Master’s thesis, University of Victoria, 2020.
- [21] R. Baartman, T. Planche, E. Heighway, and R. Hutcheon, TRIUMF TRANSOPTR open source git repository, <https://gitlab.triumf.ca/beamphys/transoptr> (2022) [accessed June 30, 2022].
- [22] R. Baartman and Y.-N. Rao, in *Proceedings of the 2003 Particle Accelerator Conference, Portland, OR* (IEEE, New York, 2003), Vol. 3, pp. 1578–1580.
- [23] A. Adelman, W. Hopkins, E. Kourlitis, M. Kagan, G. Kasieczka, C. Krause, D. Shih, V. Mikuni, B. Nachman, K. Pedro, and D. Winklehner, [arXiv:2203.08806](https://arxiv.org/abs/2203.08806).
- [24] O. Shelbaya, T. Angus, R. Baartman, P. M. Jung, O. Kester, S. Kiy, T. Planche, and S. D. Rädcl, *Phys. Rev. Accel. Beams* **24**, 124602 (2021).