University of Waterloo

Physics and Astronomy

Implementation of a Novel Symplectic Particle Accelerator Tracking Code

TRIUMF

Vancouver, BC

Paul M. Jung, ID: 20463518, Term: 4A

Sept. 2016

Abstract

The implementation of the accelerator tracking code we proposed in Ref. [5] was successful. Symplectic methods of integration were applied to equations of motion for both particles and electromagnetic fields, a novel approach to particle-in-cell simulations. The symplectic integration of particles was tested by calculating single-particle orbits in a classical AVF Cyclotron field. The field and multi-particle tracking was tested by simulating the expansion of a uniform sphere of charge. Further implementation ideas are discussed.

1 Introduction

The design and study of modern particle accelerators relies heavily on accurate simulations. One physical phenomena observed in accelerators is the creation of electromagnetic fields by the moving particles; which influence other particles. Computationally modeling this problem is significantly more difficult to solve than linear beam dynamics, hence the effects of this phenomena are often not considered.

Recent plasma physics research presented by H. Qin *et al.* [7] show a unique approach to plasma simulation. They created a discretized Hamiltonian system for a non-relativistic plasma using Hamiltonian field theory. In Ref. [5] we adapted this method to particle accelerators.

This report provides supplementary material on the implementation and testing of the symplectic particle-in-cell tracking code we derived in Ref. [5]. It considers the problem of solving a non-linear set of partial differential equations and approximations that can be made to improve performance.

2 Hamiltonian System

To summarize Ref. [5]: a modified version of the Lagrangian proposed by F.E. Low [4] yields the Hamiltonian for a collisionless relativistic plasma:

$$H(\mathbf{x}, \boldsymbol{\mathcal{P}}, \mathbf{A}, -\boldsymbol{\mathcal{E}}; t) = \iint \sqrt{\tilde{m}^2 + \left(\boldsymbol{\mathcal{P}} - \tilde{q}\mathbf{A}(\mathbf{x}, t)\right)^2} \mathrm{d}\mathbf{v}_1 \mathrm{d}\mathbf{x}_1 + \int \frac{\boldsymbol{\mathcal{E}}^2}{2} + \frac{\nabla \times \mathbf{A}^2}{2} \, \mathrm{d}\mathbf{x}_1 \,, \tag{1}$$

with:

$$\tilde{m} = f_1 m \,, \tag{2}$$

$$\tilde{q} = f_1 q \,, \tag{3}$$

where m and q are the mass and charge of the elementary particles and $f_1(\mathbf{x}_1, \mathbf{v}_1, t_1)$ is the local plasma density distribution at time t_1. The system of units has been chosen such that $c = \mu_0 = \epsilon_0 = 1$. Vector quantities are denoted in bold. This Hamiltonian is expressed in terms of canonically conjugate variables, namely using Poisson bracket notation:

$$\{\mathbf{x}_i, \boldsymbol{\mathcal{P}}_j\} = \delta_{ij} \,, \quad \{\mathbf{A}(\mathbf{x}), -\boldsymbol{\mathcal{E}}(\mathbf{y})\} = \delta(\mathbf{x} - \mathbf{y}) \,. \tag{4}$$

The first term in the Hamiltonian is the relativistic energy of fluid-particles. The second Hamiltonian term is the energy contained in the electric field $\boldsymbol{\mathcal{E}}$ and magnetic vector potential $\mathbf{A}$ produced by the plasma. $\boldsymbol{\mathcal{P}}$ is the canonical momentum density:

$$\boldsymbol{\mathcal{P}} = \frac{\tilde{m}\dot{\mathbf{x}}}{\sqrt{1 - \dot{\mathbf{x}}^2}} + \tilde{q}\mathbf{A} \,, \tag{5}$$

As for the fields, the temporal gauge (or Weyl gauge) condition (the electric scalar potential $\Phi = 0$) was chosen as L does not depend on $\dot{\Phi}$.

For convenience the definition of the gradient operator is generalized to

functional derivatives:

$$\nabla_{\mathbf{q}} = \begin{bmatrix} \dfrac{\delta}{\delta q_1} \\ \vdots \\ \dfrac{\delta}{\delta q_i} \end{bmatrix}, \quad \nabla_{\mathbf{p}} = \begin{bmatrix} \dfrac{\delta}{\delta p_1} \\ \vdots \\ \dfrac{\delta}{\delta p_i} \end{bmatrix}. \tag{6}$$

The equations of motion are obtained from:

$$\dot{\mathbf{q}} = \nabla_{\mathbf{p}} H(\mathbf{q}, \mathbf{p}), \tag{7}$$

$$\dot{\mathbf{p}} = -\nabla_{\mathbf{q}} H(\mathbf{q}, \mathbf{p}), \tag{8}$$

which leads to:

$$\dot{\mathbf{x}} = \frac{(\boldsymbol{\mathcal{P}} - \tilde{q}\mathbf{A})}{\sqrt{m^2 + (\boldsymbol{\mathcal{P}} - \tilde{q}\mathbf{A})^2}}, \tag{9}$$

$$\dot{\boldsymbol{\mathcal{P}}} = \tilde{q}\nabla(\dot{\mathbf{x}} \cdot \mathbf{A}), \tag{10}$$

$$\dot{\mathbf{A}} = -\boldsymbol{\mathcal{E}}, \tag{11}$$

$$\dot{\boldsymbol{\mathcal{E}}} = \nabla \times \nabla \times \mathbf{A} - \mathbf{j}, \tag{12}$$

where $\mathbf{j}$ is the local current density.

3 Symplectic Integration

Symplectic integration conserves geometric properties of the flow of Hamiltonian systems. One consequence of symplecticity as explained by Hairer *et*

al. [3, p. 410] is that: "For higher-dimensional systems, symplecticity means that the flow preserves the sum of the oriented areas of the projections of $\varphi_t(A)$ onto the (p_i, q_i)-coordinate planes, for any two-dimensional bounded manifold of initial values A" where $\varphi_t(A)$ is the flow and (p_i, q_i)-coordinate planes are two-dimensional Poincaré maps. This can be interpreted as the conservation of the sum of two-dimensional emittances of the beam:

$$\epsilon = \epsilon_1 + \cdots + \epsilon_n \,, \tag{13}$$

where:

$$\epsilon_i = |p_i \wedge q_i| \,. \tag{14}$$

Since the electromagnetic fields in eq. (1) are canonically conjuagate variables, symplectically integrating them tracks their evolution while conserving the two-form $\mathbf{A} \wedge -\boldsymbol{\mathcal{E}}$. This is in contrast to other particle-in-cell codes which recalculate the fields from the particle distribution at each step.

A defining characteristic of an integration method is its order of accuracy, n. It is related to the truncation error of the numerical solution:

$$\text{Error}(h) \sim \mathcal{O}(h^n) \,, \tag{15}$$

where h is the step size.

3.1 Euler Method

The symplectic Euler method is a first order integrator written for general Hamiltonian systems $H(\mathbf{q}, \mathbf{p})$ as:

$$\mathbf{p}_{n+1} = \mathbf{p}_n - h\nabla_{\mathbf{q}} H(\mathbf{q}_n, \mathbf{p}_{n+1})\,, \tag{16}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\nabla_{\mathbf{p}} H(\mathbf{q}_n, \mathbf{p}_{n+1})\,, \tag{17}$$

which leads to the implicit equation:

$$\mathbf{p}_{n+1} - \mathbf{p}_n + h\nabla_{\mathbf{q}} H(\mathbf{q}_n, \mathbf{p}_{n+1}) = 0\,. \tag{18}$$

Numerical methods for solving this equation are discussed in section 4.1. Since the Euler method requires that $\mathbf{p}_{n+1}$ is iterated before $\mathbf{q}_{n+1}$, the application to the equations of motion eqs. (9) to (11) is not trivial. This procedure requires that the momenta $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{E}}$ must be iterated before their respective coordinates $\mathbf{x}$ and $\mathbf{A}$.

A reversed symplectic Euler method, in which $\mathbf{q}_{n+1}$ is calculated before $\mathbf{p}_{n+1}$ was implemented as well, for the use in section 3.2.

3.2 Störmer-Verlet Method

As explained by Hairer *et al.*, [3] the Störmer-Verlet method is a time-symmetric second order integration method. It can be implemented using

the forward and reverse symplectic Euler methods:

$$\begin{aligned} \mathbf{p}_{n+\frac{1}{2}} &= \mathbf{p}_n - \frac{h}{2}\nabla_{\mathbf{q}} H(\mathbf{p}_{n+\frac{1}{2}}, \mathbf{q}_n)\,, \\ \mathbf{q}_{n+\frac{1}{2}} &= \mathbf{q}_n + \frac{h}{2}\nabla_{\mathbf{p}} H(\mathbf{p}_{n+\frac{1}{2}}, \mathbf{q}_n)\,, \end{aligned} \tag{19}$$

$$\begin{aligned} \mathbf{q}_{n+1} &= \mathbf{q}_{n+\frac{1}{2}} + \frac{h}{2}\nabla_{\mathbf{p}} H(\mathbf{p}_{n+\frac{1}{2}}, \mathbf{q}_{n+1})\,, \\ \mathbf{p}_{n+1} &= \mathbf{p}_{n+\frac{1}{2}} - \frac{h}{2}\nabla_{\mathbf{q}} H(\mathbf{p}_{n+\frac{1}{2}}, \mathbf{q}_{n+1})\,, \end{aligned} \tag{20}$$

noting the implicit equations for $\mathbf{p}_{n+\frac{1}{2}}$ and $\mathbf{q}_{n+1}$:

$$\begin{aligned} \mathbf{p}_{n+\frac{1}{2}} - \mathbf{p}_n + \frac{h}{2}\nabla_{\mathbf{q}} H(\mathbf{p}_{n+\frac{1}{2}}, \mathbf{q}_n) = 0\,, \\ \mathbf{q}_{n+1} - \mathbf{q}_{n+\frac{1}{2}} - \frac{h}{2}\nabla_{\mathbf{p}} H(\mathbf{p}_{n+\frac{1}{2}}, \mathbf{q}_{n+1}) = 0\,. \end{aligned} \tag{21}$$

Any second-order symplectic integrator, like this one, can be extended via the Yoshida method [8] to any desired $(2n+2)^{\text{th}}$-order.

4 Multidimensional Root-finding Algorithms

4.1 Multidimensional Newton-Raphson Method

The Newton-Raphson method is reccomended by Ref. [6] as a reliable method for root finding non-linear multidimensional systems. The derivation follows: let eq. (18) define a vector function, $\mathbf{f}$, where h, $\mathbf{p}_n$, and $\mathbf{q}_n$ are taken to be constant,

$$\mathbf{f}(\mathbf{p}_{n+1}) = 0. \tag{22}$$

Taking the Taylor expansion for a small perturbation $\boldsymbol{\delta}\mathbf{p}$,

$$\mathbf{f}(\mathbf{p}_{n+1} + \boldsymbol{\delta}\mathbf{p}) = \mathbf{f}(\mathbf{p}_{n+1}) + \mathbf{J}_{\mathbf{f}}(\mathbf{p}_{n+1}) \cdot \boldsymbol{\delta}\mathbf{p} + \mathcal{O}\big(\boldsymbol{\delta}\mathbf{p}^2\big) \ , \tag{23}$$

where $\mathbf{J}_{\mathbf{f}}(\mathbf{p}_{n+1})$ is the Jacobian matrix of the vector field $\mathbf{f}(\mathbf{p}_{n+1})$. We require $\mathbf{f}(\mathbf{p}_{n+1} + \boldsymbol{\delta}\mathbf{p}) = 0$, leading to the following equation for $\boldsymbol{\delta}\mathbf{p}$;

$$\boldsymbol{\delta}\mathbf{p} = -\mathbf{f}(\mathbf{p}_{n+1}) \cdot \mathbf{J}_{\mathbf{f}}^{-1}(\mathbf{p}_{n+1}) \, . \tag{24}$$

Which yields the final iterative formula:

$$\mathbf{p}_{n+1}^{m+1} = \mathbf{p}_{n+1}^{m} + h \left(\mathbf{p}_{n+1}^{m} - \mathbf{p}_n - h \nabla_{\mathbf{q}} H(\mathbf{p}_{n+1}^{m}, \mathbf{q}_n) \right) \cdot \left[\mathbb{I} + h \mathbf{J}_{\mathbf{p}}(\mathbf{p}_{n+1}^{m}, \mathbf{q}_n) \right]^{-1} , \tag{25}$$

where m denotes the step of the Newton-Raphson method and n denotes the step of the symplectic integrator, as before. $\mathbf{J}_{\mathbf{p}}(\mathbf{p}_{n+1}^{m}, \mathbf{q}_n)$ is the Jacobian matrix of the vector field $\left[-\nabla_{\mathbf{q}} H(\mathbf{p}_{n+1}^{m}, \mathbf{q}_n) \right]$ and $\mathbb{I}$ is the identity matrix. In Cartesian coordinates, $\mathbf{J}_{\mathbf{p}}(\mathbf{p}_{n+1}^{m}, \mathbf{q}_n)$ can be expressed as:

$$\mathbf{J}_{\mathbf{p}}(\mathbf{p}_{n+1}^{m}, \mathbf{q}_n)^{\mathrm{T}} = \begin{bmatrix} \partial_{p_x} \left(-\nabla_{\mathbf{q}} H(\mathbf{p}_{n+1}^{m}, \mathbf{q}_n) \right) \\ \partial_{p_y} \left(-\nabla_{\mathbf{q}} H(\mathbf{p}_{n+1}^{m}, \mathbf{q}_n) \right) \\ \partial_{p_z} \left(-\nabla_{\mathbf{q}} H(\mathbf{p}_{n+1}^{m}, \mathbf{q}_n) \right) \end{bmatrix} . \tag{26}$$

This matrix can also be expressed, by the symmetry of derivatives and the anti-symmetry of the equations of motion:

$$\mathbf{J_p}(\mathbf{p}, \mathbf{q}) = -\mathbf{J_q}(\mathbf{p}, \mathbf{q})\,. \tag{27}$$

So, the root finding methods for the implicit equations for $\mathbf{q}_{n+1}$ can be found with a straightforward substitution.

4.2 Quasi-Newton-Raphson Method

The Newton-Raphson method is computationally expensive since it requires the calculation of $\mathbf{J_f}^{-1}$ at each iteration. The Quasi Newton-Raphson method avoids this by taking the following approximation:

$$\mathbf{J_p}(\mathbf{p}^m_{n+1}, \mathbf{q}_n) \approx \mathbf{J_p}(\mathbf{p}^{m_0}_{n+1}, \mathbf{q}_n)\,. \tag{28}$$

The accuracy of this approximation depends on the accuracy of the initial guess. However since the Hamiltonian is reasonably smooth, the difference between the initial guess and the final root is generally small.

4.3 Newton-Raphson Convergance

Despite the Newton-Raphson method being one of the best for multidimensional systems of equations, there are specific cases where it can fail to find a root. Firstly, this method fails immediately if the Jacobian matrix $\mathbf{J_f}$ is

not invertible. Another source of error is the initial guess. The initial guess could send the algorithm into a cycle about the root. The stability of the algorithm depends on the root being well defined; a root that only exists in a limit will cause the algorithm to run indefinitely. There are advanced algorithms for backtracking and using more stable methods to avoid these errors [6]. However these problems have not been encountered during the development and testing of the code.

Lastly, if the Jacobian matrix is discontinuous at the root then convergence is not guaranteed. This implies that the interpolator chosen in section 6.2 must be continuous, at least to second order. Explicitly, if $\mathbf{A} \in C^2(\mathbb{R}^3, \mathbb{R}^3)$ then function $\mathbf{f}(\mathbf{p}_{n+1}) \in C^1(\mathbb{R}^3, \mathbb{R}^3)$.

4.4 Avoiding Root-Finding

Note that, in the canonical case of a seperable Hamiltonian, where

$$H(\mathbf{q}, \mathbf{p}) = T(\mathbf{p}) + V(\mathbf{q}) \,, \tag{29}$$

the implicit equations eq. (18) and eq. (21) become explicit:

$$\begin{aligned} \mathbf{p}_{n+1} &= \mathbf{p}_n + \Delta t \cdot \dot{\mathbf{p}}(\mathbf{q}) \,, \\ \mathbf{q}_{n+1} &= \mathbf{q}_n + \Delta t \cdot \dot{\mathbf{q}}(\mathbf{p}) \,, \end{aligned} \tag{30}$$

eliminating the need for root finding. This simplification applies to the equations of motion for $\mathbf{A}$ eq. (11) and $\boldsymbol{\mathcal{E}}$ eq. (12) but does not apply to $\mathbf{x}$ eq. (9)

and $\boldsymbol{\mathcal{P}}$ eq. (10).

Solving the implicit equations with root finding techniques is very costly compared to the explicit eq. (30). E. Forest [2] explains that many different accelerator physics tracking simulations are based on specific splits of the relativistic single particle Hamiltionian:

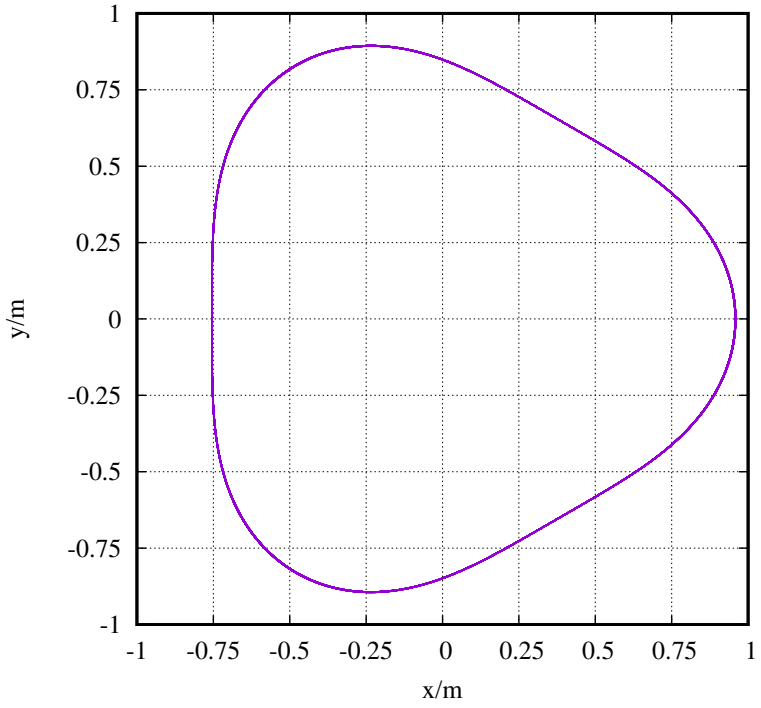
$$H(\mathbf{x}, \mathbf{p}) = \sqrt{m^2 + (\mathbf{p} - q\mathbf{A}(\mathbf{x}))^2} + q\Phi(\mathbf{x}) \tag{31}$$

Canonical transformations to Frenet-Serret coordinates allow the use of the paraxial approximation [1], which leads to a split Hamiltonian. Further work should be done to implement useful splits of eq. (1) for specific applications of the code.

5 Test of Integration Methods

The system chosen to test the integration methods consists of an Azimuthally Varying Field (AVF) Cyclotron with three sectors and a field modulation, or flutter, of 1.0. A single 299.792 MeV proton is tracked near the equilibrium orbit, shown in fig. 1, without tracking the $\mathbf{A}$ and $\boldsymbol{\mathcal{E}}$ fields. To compare integration methods, Poincaré maps are produced by perturbing the initial particle from the equilibrium orbit by $\epsilon = 0, 0.05, 0.075, 0.1$ where $x_0 = x_{\text{equib}}(1 + \epsilon)$ is the initial x position, the intial momentum remains unchanged; the particle is tracked for 500 turns. A view screen located on the

Figure 1: Reference orbit of the AVF system, viewed from above.



x-z plane, for positive x, observes the phase of the particle for each orbit. The Euler method was used to produce fig. 2 and the Störmer-Verlet method for fig. 3; both were integrated with a constant step size of 0.00573448s and their implicit equations of motion were solved using the Newton-Raphson method. The key feature of these plots is the behaviour of symplectic integration which is that the phase-space area is conserved in both figures however, the area and shape of fig. 2 is less accurate than fig. 3; $\epsilon = 0$ should correspond to the reference orbit, or a single point on the plot.

Figure 2: Poincaré map integrated with the Euler method.

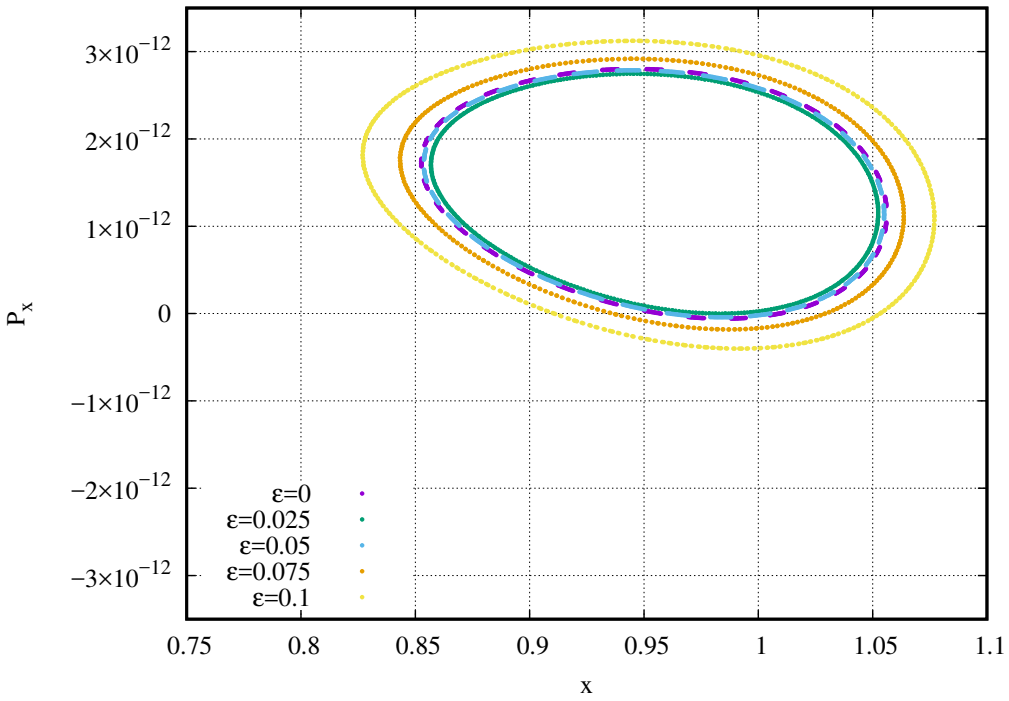
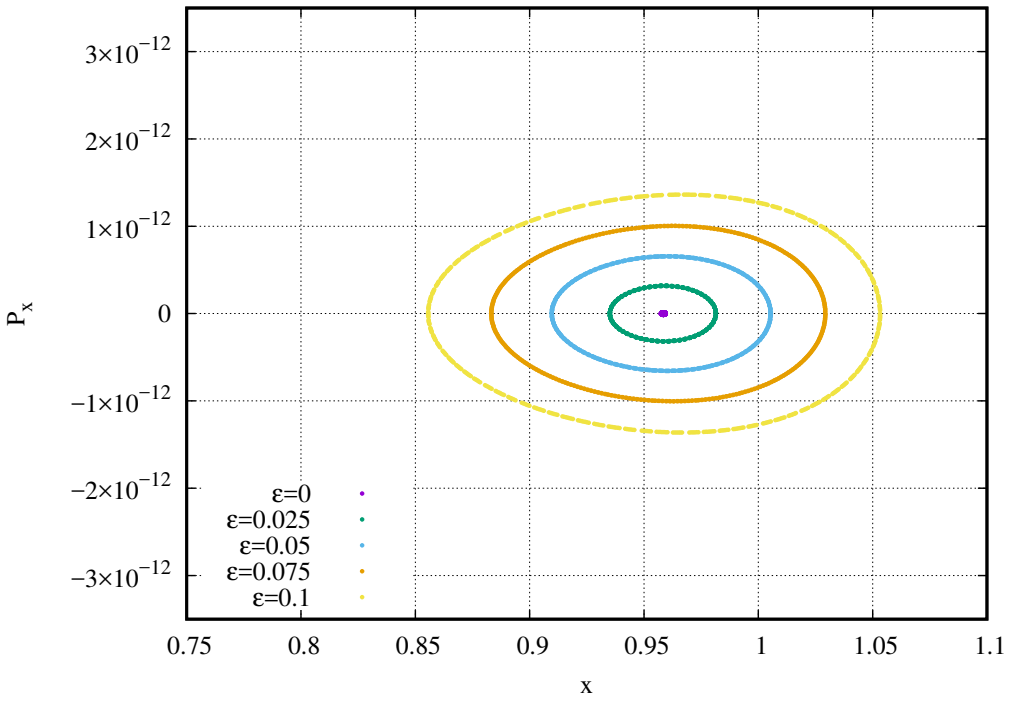


Figure 3: Poincaré map integrated with the Störmer-Verlet method.



6 Discretization

As of yet, the system is comprised of a continuous field and plasma distributions. Discretization of these distributions is necessary for computation.

6.1 Macro-Particles

The initial plasma distribution is discretized into a finite number of macro-particles. Each of which is given a dimensionless weight:

$$\mathrm{W} = \frac{f_1}{\Delta \mathbf{x}_1 \Delta \mathbf{v}_1} \tag{32}$$

which is time independent. Where $\Delta \mathbf{x}_1 \Delta \mathbf{v}_1$ is the initial phase-space volume represented by the macro-particle.

6.2 Vector Field

To discretize the field $\mathbf{A}$ a third-order continuous polynomial interpolator was created using the method from Ref. [7]. Using a discrete coordinate grid with regular spacing, we transform a point in real space into normalized grid space, in one dimension:

$$q_x = \frac{x - x_{\min}}{\Delta x} \,. \tag{33}$$

Where $x_{\min}$ is the minimal boundary of the grid and Δx is the grid spacing. The kernel is the following piecewise polynomial:

$$\mathcal{W}(q) = \begin{cases} q^8\frac{3}{512} + q^7\frac{3}{64} - q^6\frac{7}{32} + q^4\frac{77}{128} - q^2\frac{63}{64} + \frac{179}{256}, & 0 \leq |q| \leq 1\,, \\ -q^8\frac{3}{512} + q^7\frac{3}{64} - q^6\frac{7}{64} + q^4\frac{7}{32} - q + 1, & 1 \leq |q| \leq 2\,, \\ 0, & |q| > 2\,. \end{cases} \quad (34)$$

In Cartesian coordinates let (q_i, q_j, q_k) represent the normalized space coordinate of the neighbouring discrete value $\mathbf{A}_{ijk}$ then:

$$\mathbf{A}(\mathbf{x}) = \sum_i \sum_j \sum_k \mathbf{A}_{ijk} \mathcal{W}\left(q_i - q_x\right) \mathcal{W}\left(q_j - q_y\right) \mathcal{W}\left(q_k - q_z\right)\,. \quad (35)$$

An advantage to this interpolation scheme is the continuous derivative and second derivative can be computed using the interpolator directly:

$$\partial_x \mathbf{A}(\mathbf{x}) = \sum_i \sum_j \sum_k \mathbf{A}_{ijk} \left(\frac{-\mathcal{W}'\left(q_i - q_x\right)}{\Delta x}\right) \mathcal{W}\left(q_j - q_y\right) \mathcal{W}\left(q_k - q_z\right)\,, \quad (36)$$

$$\partial_x^2 \mathbf{A}(\mathbf{x}) = \sum_i \sum_j \sum_k \mathbf{A}_{ijk} \left(\frac{\mathcal{W}''\left(q_i - q_x\right)}{\Delta x^2}\right) \mathcal{W}\left(q_j - q_y\right) \mathcal{W}\left(q_k - q_z\right)\,, \quad (37)$$

where cross derivatives are computed similarly.

The polynomial kernel function samples four discretized points are sampled per dimension, so in total, 64 points are sampled per interpolation.

7 Test of Field Tracking

The test of field tracking was to simulate the expansion of a uniformly distributed sphere of charged particles which are initially at rest. The macro-particles were given a weight factor so that the sphere would double in radius within ten time steps, non-relativistically. The field $\mathbf{A}$ is zero initially and $\boldsymbol{\mathcal{E}}$ is initialized with the analytic solution for a uniformly charged sphere, given by Gauss's law. The following test uses a system of 10^4 particles and a grid size of 20^3 in Cartesian coordinates. The time evolution of the self-induced fields along the x-axis is shown in fig. 5 and fig. 6 where $t = 0$ corresponds to the initial conditions. Note the discretization noise in the $\boldsymbol{\mathcal{E}}$ field for $t = 0.0267s$ and $t = 0.0334s$. This noise does not appear in $\mathbf{A}$, which is the only field affecting the particles directly, in eq. (9) and eq. (10).

Figure 4: The time evolution sphere projected on the x-y plane.

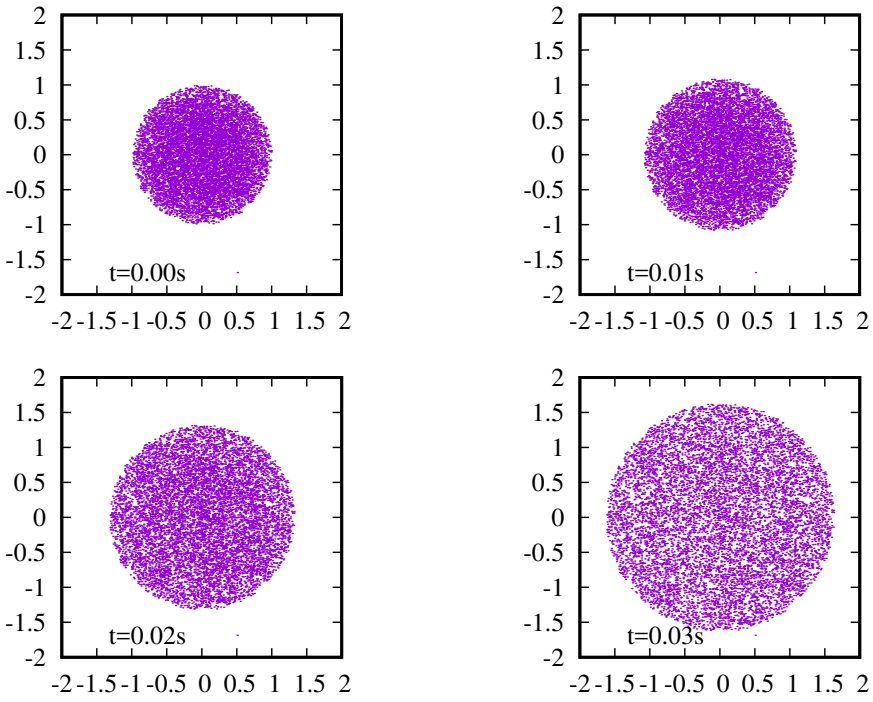


Figure 5: The time evolution of $\mathbf{A}_x$ along the x-axis.

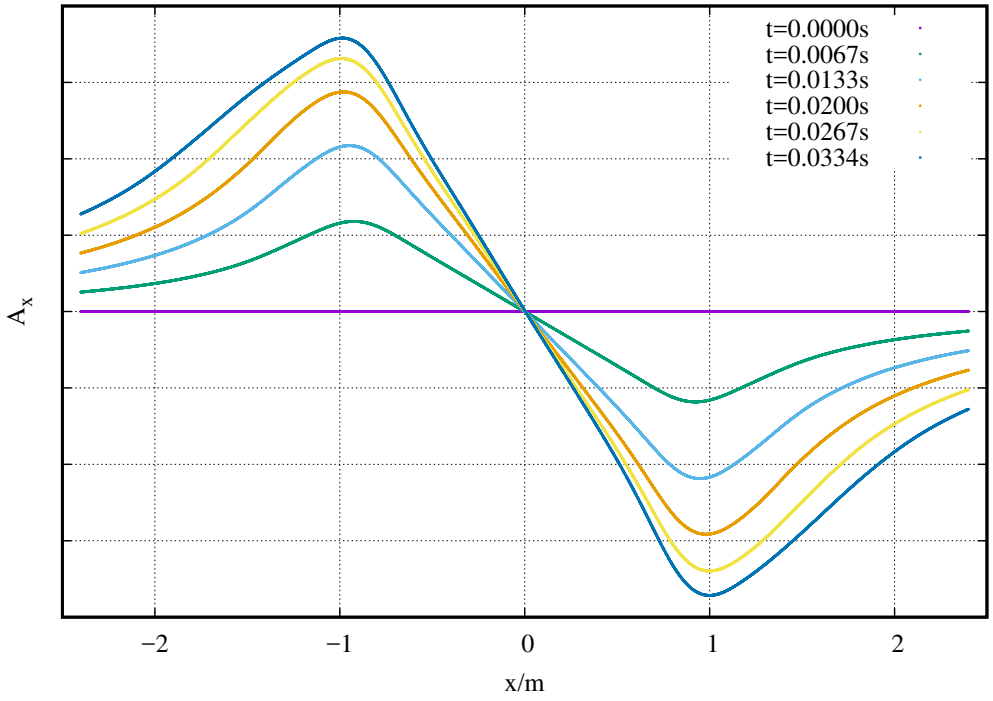
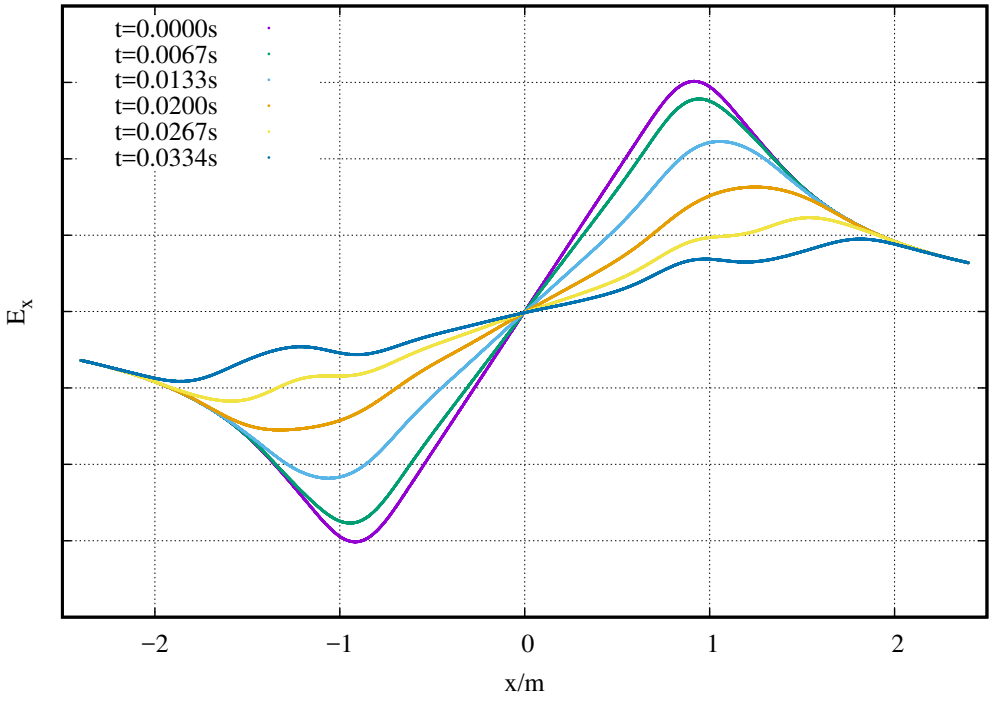


Figure 6: The time evolution of $\boldsymbol{\mathcal{E}}_x$ along the x-axis.



8 Conclusions

The implementation and testing of the code is successful.

Higher-order integration methods, should be used for more accurate and physical results, dispite the cost of additional computation time.

9 Recommendations

Splits of the Hamiltonian should be derived and the resulting equations of motion implemented.

More general symplectic integrators should be developed for experimenting with different systems and symmetries; the first of which should be the Yoshida method.

Explicit integration methods should also be developed for use with the split Hamiltonian.

A second order polynomial kernel interpolator, could be used instead of the current third order for additional speed.

The simulation of a physical system such as the TRIUMF cyclotron should be the next test case.

References

[1] Ernest D Courant and Hartland S Snyder. Theory of the alternating-gradient synchrotron. *Annals of physics*, 3(1):1–48, 1958.

[2] Etienne Forest. Geometric Integration for Particle Accelerators. *J. Phys.*, A39:5321–5378, 2006.

[3] Ernst Hairer, Christian Lubich, Gerhard Wanner, et al. Geometric numerical integration illustrated by the Stormer-Verlet method. *Acta numerica*, 12(12):399–450, 2003.

[4] F. E. Low. A Lagrangian Formulation of the Boltzmann-Vlasov Equation for Plasmas. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 248(1253):282–287, 1958.

[5] Thomas Planche and Paul Matthew Jung. Relativistic Lagrangian and Hamiltonian Description of a Beam with Space-Charge. *arXiv preprint arXiv:1603.02976*, 2016.

[6] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C (2Nd Ed.): The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.

[7] Hong Qin, Jian Liu, Jianyuan Xiao, Ruili Zhang, Yang He, Yulei Wang, Joshua W. Burby, Leland Ellison, and Yao Zhou. Canonical symplectic

particle-in-cell method for long-term large-scale simulations of the Vlasov-Maxwell system. *Nucl. Fusion*, 56:014001, 2016.

[8] Haruo Yoshida. Construction of higher order symplectic integrators. *Physics Letters A*, 150(5):262 – 268, 1990.