Efficient Fourier Basis Particle Simulation

Matthew S. Mitchell\textsuperscript{a}, Matthew T. Miecnikowski\textsuperscript{a}, Gregory Beylkin\textsuperscript{b}, Scott E. Parker\textsuperscript{a}

\textsuperscript{a}Department of Physics, University of Colorado, Boulder, CO 80303 USA
\textsuperscript{b}Department of Applied Mathematics, University of Colorado, Boulder, CO 80303 USA

Abstract

The standard particle-in-cell algorithm suffers from finite grid errors which break energy conservation, cause numerical dispersion, and create numerical instabilities. There exists a gridless alternative which bypasses the deposition step and calculates each Fourier mode of the charge density directly from the particle positions. We show that a gridless method can be computed efficiently through the use of an Unequally Spaced Fast Fourier Transform (USFFT) algorithm. After a spectral field solve, the forces on the particles are calculated via the inverse USFFT (a rapid solution of an approximate linear system) \cite{1, 2}. We provide a one dimensional implementation of this algorithm with an asymptotic runtime of $O(N_p + N_m \log N_m)$ for each iteration, identical to the standard PIC algorithm (where $N_p$ is the number of particles and $N_m$ is the number of Fourier modes). Higher dimensional formulations would scale as $O(N_p + N_m^D \log N_m^D)$, where $D$ is the spatial dimensionality of the problem. We demonstrate superior energy conservation and reduced noise, as well as convergence of the energy conservation at small time steps.

Keywords: Numerical, Plasma, Particle-in-cell, Energy conserving, Momentum conserving, Fourier transform

1. Introduction

A common approach to numerically solving the Vlasov-Poisson system is to represent the distribution function using particles, with fields solved on a grid and interpolated at the particle positions \cite{3, 4}. This scheme, known as the particle-in-cell (PIC) method, has been enormously successful for simulating plasmas and is used in a wide variety of applications, but does not conserve total energy \cite{5}. Energy-conserving schemes based on variational formulations have been proposed \cite{6, 7}, but generally do not conserve momentum because of a lack of translational invariance \cite{5}, though the momentum error can be kept small in many cases due to the choice of integrator \cite{9}.

In addition to the lack of energy conservation, PIC also suffers from a finite grid instability, in which sufficiently high Fourier modes experience exponential growth \cite{10, 11} due to coupling with lower modes. This phenomenon causes
numerical heating, which saturates when the Debye length is on the order of the grid spacing \[10, 8\]. The finite grid instability is of particular relevance to problems involving cold plasmas or multiple length scales \[10, 12\]. Various approaches have been proposed to reduce this instability, such as the use of smoother particle shapes \[13\], grid jiggling \[12\], filtering \[14\], and high-order Galerkin methods \[13\], but many of these schemes suffer from issues such as a lack of charge conservation and high computational costs.

Several energy-conserving particle algorithms \[6, 7\] based on the Lagrangian formulated by Low \[15\] have been suggested as alternatives to PIC. Charge-conserving approaches \[9, 16\] using implicit time integration have also been proposed. Energy-conserving algorithms have the benefit of eliminating numerical heating, but suffer from several drawbacks. They generally do not conserve momentum \[17\], sometimes suffer from increased noise, and heavily restrict the choice of particle shape \[8\]. However, these methods have seen widespread use due to their efficiency and simplicity.

It has been demonstrated that exact energy and momentum conservation in a particle code can only be achieved by depositing charge on a truncated Fourier basis \[8\]. This method has also been shown to eliminate the finite grid instability and reduce coupling between modes \[18\]. However, due to the poor scaling and high computational cost of this approach (\(O(N_m N_p)\), where \(N_m\) is the number of Fourier modes and \(N_p\) is the number of particles), it has not been seriously considered in practice.

We present a similar algorithm to the one proposed in \[8\], in which we model the charge density as a sum of shape functions in continuous space and perform the field solve with a truncated Fourier series. However, we reduce the computation time to \(O(N_p + N_m \log N_m)\), which is equivalent to that of conventional PIC with a spectral field solve, by making use of an Unequally Spaced FFT (USFFT) \[1, 2\].

This paper is organized as follows: we begin by reviewing the standard PIC method in Section 2. We then propose a gridless algorithm based on \[8\] and demonstrate that it can be made efficient via the USFFT. In Section 3 we analyze the results of several numerical experiments, comparing our code to an implementation of conventional PIC and providing experimental confirmation of its energy conservation. Formal proofs of energy conservation in the continuous-time limit and momentum conservation are given in Section 4. In Section 5 we present an analysis of the algorithm’s asymptotic time complexity and a possible parallelization scheme along with the results of several scaling experiments. Section 6 contains a summary of our results and a discussion of possible generalizations.

2. Numerical Methods

The behavior of a collisionless plasma is given by the Vlasov-Poisson system:

\[
\frac{\partial f}{\partial \tau} + \mathbf{v} \cdot \nabla f + \frac{q \mathbf{E}}{m} \cdot \nabla f = 0, \tag{1}
\]
∇^2 φ = \frac{ρ}{\varepsilon_0} \tag{2}

2.1. Particle-in-Cell

This system is commonly solved via the particle-in-cell (PIC) algorithm \[3, 4]\, which tracks particles in continuous phase space, while fields are tracked on a spatial grid with \(N_g\) points. Each timestep begins by calculating the charge density \(ρ(x)\) at the grid points from particle positions, via deposition of charge on the grid, such that

\[ρ_x = \sum_n qS(X_n - x)\] \tag{3}

where \(S\) is some shape function corresponding to the weighting scheme. \(E\) and \(φ\) are then calculated at the grid points from \(ρ\), usually via spectral methods, as follows:

\[\tilde{ρ}_k = \sum_x ρ_x e^{-ikx}\] \tag{4}
\[\tilde{E}_k = \frac{1}{ik\varepsilon_0} \tilde{ρ}_k\] \tag{5}
\[\tilde{φ}_k = \frac{1}{k^2\varepsilon_0} \tilde{ρ}_k\] \tag{6}
\[E_x = \sum_k \tilde{E}_k e^{ikx}\] \tag{7}

After the field solve, the forces on the particles are calculated by interpolating the E-field from the grid to the particle positions as

\[F_n = \sum_x qE_x S(X_n - x).\] \tag{8}

In this paper, we assume without loss of generality that the same weighting scheme is used for both deposition and interpolation, but this need not be the case.

2.2. Particle-in-Fourier

Our algorithm, termed Particle-in-Fourier (or PIF), begins by following conventional Fourier basis algorithms \[5\]. Instead of depositing the charge on a grid, we treat \(ρ(x)\) as a sum of shape functions (usually \(S(x) = δ(x)\)) in continuous space and “deposit” on a truncated Fourier basis. This can be accomplished by calculating each mode of \(\tilde{ρ}\) directly from the particle positions, as
\[
\tilde{\rho}_k = \int_0^L dx \, e^{-ikx} \rho(x),
\]

(9)

\[
= \int_0^L dx \, e^{-ikx} \sum_{n=1}^{N_p} qS(x - X_n),
\]

(10)

\[
n = q\tilde{S}_k \sum_{n=1}^{N_p} e^{-ikX_n},
\]

(11)

where \( k = 2\pi n / L \), for some integer \( n \) such that \(-N_m/2 \leq n \leq N_m/2\), where \( N_m \) is the number of modes (analogous to the number of grid cells in PIC). \( \tilde{S}_k \) denotes the Fourier transform of the shape function, given by

\[
\tilde{S}_k = \int_0^L dx \, e^{-ikx} S(x).
\]

(12)

A unique feature of the PIF method is the ability to use a physical particle shape given by \( S(x) \) as an alternative to filtering. In many ways, this is similar to PIC, except without a computational penalty for using higher order shape functions (e.g. Gaussian). Because convolution with the shape function is equivalent to multiplication in Fourier space and \( \tilde{S}_k \) only needs to be computed once, the step time is independent of the weighting scheme, permitting the use of arbitrary shape functions with no additional computational cost.

We now perform the field solve, in a manner identical to conventional PIC, as

\[
\tilde{E}_k = \frac{1}{ik\varepsilon_0} \tilde{\rho}_k,
\]

(13)

\[
\tilde{\phi}_k = \frac{1}{k^2 \varepsilon_0} \tilde{\rho}_k.
\]

(14)

Using a similar approach to the deposition, we can determine the forces on the particles by summing over all modes of \( E \):

\[
F_n = \int_0^L dx \, qE(x)S(x - X_n),
\]

(15)

\[
= \sum_k e^{ikX_n} q\tilde{E}_k \tilde{S}_k.
\]

(16)

### 2.3. Unequally Spaced Fast Fourier Transform (USFFT)

Our algorithm relies on rapid evaluation of two computationally expensive sums: \( \sum_n e^{-ikX_n} \) (done for every mode), and \( \sum_k \tilde{S}_k \tilde{E}_ke^{ikX_n} \) (done for every particle). Naively, computing these sums requires \( O(N_pN_m) \) operations for \( N_p \)
particles and $N_m$ modes. However, we can reduce the computational cost by using an Unequally Spaced FFT (USFFT) [1] [2].

The USFFT algorithm computes the sum

$$\hat{g}_n = \sum_{l=1}^{N_p} g_ne^{\pm 2\pi i x_n n}$$

(17)

for all $-N_m/2 \leq n \leq N_m/2$, where $x_l \in [-1/2, 1/2]$ and $g_l$ are arbitrary complex coefficients. The dual version of the USFFT rapidly evaluates the sum

$$g(x_l) = \sum_{n=-N_m/2}^{N_m/2-1} \tilde{g}_n e^{\pm 2\pi inx_l},$$

(18)

where $\tilde{g}_n$ are arbitrary complex coefficients. The computational cost is again $O(N_p + N_m \log N_m)$. It is important that the accuracy of the USFFT is user-controlled, and since the result is guaranteed by the algorithm, one can use the USFFT in a manner similar to the FFT (tight accuracy estimates can be found in [1]).

The USFFT algorithm uses the FFT as one of the steps, and for this reason, it is helpful to estimate the computational cost of the USFFT in terms of that of the FFT. The USFFT yielding double precision accuracy in one dimension is $\approx 2.5 - 3$ times slower than the FFT of the same size. In two dimensions, the factor is $\approx 10 - 20$ (in both cases for a random distribution of points).

The first step of the algorithm can be understood as a projection on a subspace of multiresolution analysis. The second step applies the FFT to the projected data, and the last step corrects the computed values by a predetermined factor. In [1] the projection step uses b-splines, whereas [2] uses Gaussians instead. Other choices are possible with a mild effect on the speed of the algorithm. The key to the speed is the splitting of operations between the spatial and the Fourier domains so that the combined effect is an application of a nearly ideal filter in the Fourier domain.

We use the USFFT by setting $g_l = 1$ and $x_l = X_l/L$ for all particles, so that we can efficiently compute the sum in Eq. (17). Similarly, by taking $\tilde{g}_n$ as the amplitude of each mode of the electric field, we can compute the sum in Eq. (16). As a result, PIF has the same computational complexity as standard PIC.

We analyze this further in Section 5.

3. Numerical Experiments

In this section, we test the behavior of the two methods on two classic kinetic problems in plasma physics for which PIC has been well verified with theory.

3.1. Two Stream Instability

An implementation of PIF was created, using a serial implementation of the USFFT [1]. The performance and energy conservation of the code were
compared to an implementation of conventional PIC with a linear particle shape, first order interpolation, and a spectral field solve. Shared-memory parallelism was implemented in both codes using OpenMP. Source code can be found at https://github.com/matt2718/ftpic.

The superior energy conservation of PIF becomes evident when we examine the two stream instability. PIF and the reference PIC implementation were used to simulate the mixing of two counter-moving electron beams against a neutralizing background (see panel (i) of Fig. 1). Both codes were run with 10000 particles and 16 modes/grid cells for 20000 time steps, with the effective grid spacing equal to approximately 0.88\(\lambda_D\). Fig. 2 plots the normalized field, kinetic, and total energy of each system. It is clear that the total energy is not conserved for conventional PIC. Fig. 3 illustrates the total energy over time for both simulations, demonstrating energy conservation of PIF, but not in PIC.

Aside from the difference in energy conservation, the two codes produced similar results. Fig. 1 illustrates the evolution of phase space over time in the PIF simulation. The two beams are represented by different colors, but all particles are identical.

3.2. Landau Damping

Our implementations of PIF and conventional PIC were used to simulate a Maxwellian plasma of 10000 particles with a sinusoidal density perturbation of the second Fourier mode given by \(n_1 = a \cos(4\pi x/L)\). This was performed for both 32 grid cells in PIC and 32 modes in PIF (\(\Delta x = 1.0\lambda_D\)) and for 128 grid cells in PIC and 128 modes in PIF (\(\Delta x = 0.25\lambda_D\)). The observed damping rate of the second mode of the E-field was compared to the theoretical rate, illustrated in Fig. 4 and Fig. 5.
Fig. 2: Kinetic, potential, and total energy for the PIC method applied to the two stream instability. Total energy is not conserved.

Fig. 3: Comparison of total energy over time in PIF and PIC simulations of the two stream instability, demonstrating the superior energy conservation of PIF.

At small grid spacings, we see that the results of the PIC simulation converge to those of PIF. PIF, however, accurately reproduces the theoretical damping rate regardless of the number of modes present.

4. Conservation Properties and Numerical Error

The standard PIC algorithm does not conserve energy. It is shown by Langdon [5] that, if the field energy is taken to be \( U_E = \frac{\Delta x^2}{\varepsilon_0} \sum_k \hat{\rho}_k \hat{\phi}_k \), then

\[
\frac{d}{dt} U_E = -\int_{-\infty}^{\infty} dk \frac{F(-k)J(k)}{2\pi q} \frac{k}{\kappa(k)},
\]

and the time derivative of kinetic energy is

\[
\frac{d}{dt} T = \int_{-\infty}^{\infty} dk \frac{F(-k)J(k)}{2\pi q},
\]
where $\kappa$ is defined by

$$\tilde{E}(k) = -i\kappa(k)\tilde{\phi}(k).$$

(21)

Therefore, total energy is conserved only when $\kappa(k) = k$ for all values of $k$. However, due to grid aliasing, this condition only holds in the first Brillouin zone ($-k_g/2 < k < k_g/2$, where $k_g$ is the grid wavenumber).

For two particles at $X_1$ and $X_2$ in a periodic system, the effective force on the second particle in PIF is given by

$$\tilde{\rho}_k = q\tilde{S}_ke^{-ikX_1},$$

$$\tilde{E}_k = \frac{1}{ik\epsilon_0}\tilde{\rho}_k = \frac{q}{ik\epsilon_0}\tilde{S}_ke^{-ikX_1},$$

$$F_{X_2} = \sum_k q\tilde{E}_k\tilde{S}_ke^{ikX_2} = \sum_k \frac{q^2}{ik\epsilon_0}\tilde{S}_k^2e^{ik(X_2-X_1)}.$$  

(22)

(23)

(24)

This is independent of displacement, so grid aliasing is not present in PIF, and $\kappa = k$ in all Brillouin zones. Therefore, energy is conserved in the continuous-time limit.

Because the sum runs from $-N/2$ to $N/2$ (ignoring 0) and contains a $1/k$ term, all cosine terms in the expansion will cancel out, so the effective force is odd and the total force on the system is 0. This can be generalized to a system with an arbitrary number of particles.

We have shown energy conservation in the continuous time limit, but an actual simulation involves discretization in the time domain. Experimentally,
we observe that the single-timestep error in the total energy of PIF converges as \( O(\Delta t^3) \), in agreement with [8] and [18]. This is illustrated in Fig. 6.

![Fig. 6: Single-timestep energy error with respect to timestep size. In agreement with [8], we see convergence of roughly \( O(\Delta t^3) \) for FT-PIC in the asymptotic regime.](image)

5. Performance and Scaling

The asymptotic runtime of the USFFT algorithm is \( O(N_p + N_m \log N_m) \) [1]. Because the field solve and the particle push take \( O(N_m) \) and \( O(N_p) \) time respectively, the total asymptotic runtime of the particle in Fourier algorithm is \( O(N_p + N_m \log N_m) \) for a single iteration. This is identical to the standard PIC algorithm, assuming the field solve is done spectrally. This scaling is demonstrated experimentally in Fig. 7.

Though we only provide a 1D implementation, the same scheme can be extended to higher dimensions, while still scaling well. Higher dimensional versions of the USFFT library exist, requiring \( O(N_p + N_m^D \log N_m^D) \) time (where \( D \) is the dimensionality of the system). As an ordinary \( D \)-dimensional FFT requires \( O(N_m^D \log N_m^D) \) time, the scaling is identical to that of PIC in any number of dimensions.

In addition, because convolution with an arbitrary shape function corresponds to multiplication in Fourier space, increasing the width of the shape function beyond a single grid cell requires no additional time. This can provide a huge advantage for PIF as higher order shape functions (even Gaussian) are essentially free, whereas in PIC, higher order shape functions involve many more calculations and difficult-to-parallelize scatter operations.

For reasonable parameters, the \( N_p \) term dominates over the \( N_m \log N_m \) term. This suggests the following parallelization scheme for the algorithm: We divide the particles between nodes, with each node performing the field solve for its own particles. Before the push step, the E-fields from every node are summed together.
In order to verify that this scaling holds, the two codes were run for 1000 time steps with 80000 particles and 64 grid cells/mode on a single 68-core Xeon Phi node of the Cori supercomputer at the National Energy Research Scientific Computing Center (NERSC). The number of OpenMP threads was varied from 1 to 64, with the problem size kept constant. Both algorithms exhibited similar strong scaling, as shown in Fig. 8. On average, PIF required 2.9 times as much time as PIC, and the two algorithms demonstrated comparable scaling.

Weak scaling was investigated by increasing the problem size, such that each run had 20000 particles per thread. The number of grid cells was kept constant between runs. Results are shown in Fig. 9. We see that PIF demonstrates better weak scaling than our implementation of PIC for large numbers of particles. We
suspect that this is because of the difficulty of parallelizing the deposition step due to the scatter operations involved.

Fig. 9: PIF has superior weak scaling to PIC as the number of threads is increased from 1 to 64, with 20000 particles/thread and 64 grid cells.

6. Discussion

We have demonstrated that the PIF gridless scheme is a feasible approach to plasma simulation, as it can be implemented with comparable performance and identical scaling to the conventional PIC method while conserving both energy and momentum in the continuous-time limit. We have provided a proof of these conservation properties and verified them through several numerical experiments. In addition, we have verified that the results of the PIF model agree with theory for standard test cases such as Landau damping and the two stream instability.

We did find that PIF was slower than a reference implementation of PIC by a factor of approximately 2.9 in the 1D case. However, PIF demonstrated better weak and strong scaling and avoids costly scatter operations during the deposition step. Furthermore, the computational cost of PIF is independent of the particle shape function, permitting the use of higher order or Gaussian weighting schemes without additional time.

We expect that the PIF algorithm can be generalized to higher dimensional energy conserving electrostatic and electromagnetic models, as the field solves can still be performed with spectral methods. This can be accomplished through the use of two and three dimensional USFFTs, still with a computational cost of $O(N_p + N_{Dm}^2 \log N_{Dm})$. In addition, we expect that our method could be generalized to $\delta f$ based codes, which are of particular interest to fusion plasmas.
Acknowledgments

Research supported by the United States Department of Energy under grants DE-FG02-08ER54954 and DE-SC000801.

References

[1] G. Beylkin, On the Fast Fourier Transform of Functions with Singularities, Applied and Computational Harmonic Analysis 2 (4) (1995) 363–381. doi:10.1006/acha.1995.1026
URL http://www.sciencedirect.com/science/article/pii/S1063520385710263

[2] A. Dutt, V. Rokhlin, Fast Fourier Transforms for Nonequispaced Data, SIAM Journal on Scientific Computing 14 (6) (1993) 1368–1393. doi:10.1137/0914081
URL https://epubs.siam.org/doi/abs/10.1137/0914081

[3] C. K. Birdsall, A. B. Langdon, Plasma physics via computer simulation, McGraw-Hill, New York, 1985.

[4] R. W. Hockney, J. W. Eastwood, Computer simulation using particles, special student ed Edition, A. Hilger, Bristol [England] ; Philadelphia, 1988.

[5] A. B. Langdon, Effects of the spatial grid in simulation plasmas, Journal of Computational Physics 6 (2) (1970) 247–267. doi:10.1016/0021-9991(70)90024-0
URL http://www.sciencedirect.com/science/article/pii/0021999170900240

[6] H. R. Lewis, Energy-conserving numerical approximations for Vlasov plasmas, Journal of Computational Physics 6 (1) (1970) 136–141. doi:10.1016/0021-9991(70)90012-4
URL http://www.sciencedirect.com/science/article/pii/0021999170900124

[7] J. W. Eastwood, The virtual particle electromagnetic particle-mesh method, Computer Physics Communications 64 (2) (1991) 252-266. doi:10.1016/0010-4655(91)90036-K
URL http://www.sciencedirect.com/science/article/pii/001046559190036K

[8] E. G. Evstatiev, B. A. Shadwick, Variational formulation of particle algorithms for kinetic plasma simulations, Journal of Computational Physics 245 (2013) 376–398. doi:10.1016/j.jcp.2013.03.006
URL http://www.sciencedirect.com/science/article/pii/S0021999113001800
[9] G. Chen, L. Chacn, D. C. Barnes, An energy- and charge-conserving, implicit, electrostatic particle-in-cell algorithm, Journal of Computational Physics 230 (18) (2011) 7018–7036. doi:10.1016/j.jcp.2011.05.031
URL http://www.sciencedirect.com/science/article/pii/S0021999111003421

[10] H. Okuda, Nonphysical noises and instabilities in plasma simulation due to a spatial grid, Journal of Computational Physics 10 (3) (1972) 475–486. doi:10.1016/0021-9991(72)90048-4
URL http://www.sciencedirect.com/science/article/pii/0021999172900484

[11] E. L. Lindman, Dispersion relation for computer-simulated plasmas, Journal of Computational Physics 5 (1) (1970) 13–22. doi:10.1016/0021-9991(70)90049-5
URL http://www.sciencedirect.com/science/article/pii/0021999170900495

[12] J. W. Brackbill, G. Lapenta, A Method to Suppress the Finite-Grid Instability in Plasma Simulations, Journal of Computational Physics 114 (1) (1994) 77–84. doi:10.1006/jcph.1994.1150
URL http://www.sciencedirect.com/science/article/pii/S0021999184711508

[13] G. B. Jacobs, J. S. Hesthaven, High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids, Journal of Computational Physics 214 (1) (2006) 96–121. doi:10.1016/j.jcp.2005.09.008
URL http://www.sciencedirect.com/science/article/pii/S0021999105004250

[14] R. W. Hockney, S. P. Goel, J. W. Eastwood, Quiet high-resolution computer models of a plasma, Journal of Computational Physics 14 (2) (1974) 148–158. doi:10.1016/0021-9991(74)90010-2
URL http://www.sciencedirect.com/science/article/pii/0021999174900102

[15] F. E. Low, A Lagrangian formulation of the Boltzmann-Vlasov equation for plasmas, Proc. R. Soc. Lond. A 248 (1253) (1958) 282–287. doi:10.1098/rspa.1958.0244
URL http://rspa.royalsocietypublishing.org/content/248/1253/282

[16] S. Markidis, G. Lapenta, The energy conserving particle-in-cell method, Journal of Computational Physics 230 (18) (2011) 7037–7052. doi:10.1016/j.jcp.2011.05.033
URL http://www.sciencedirect.com/science/article/pii/S0021999111003448
[17] A. B. Langdon, Energy-conserving plasma simulation algorithms, Journal of Computational Physics 12 (2) (1973) 247–268. doi:10.1016/S0021-9991(73)80014-2
URL http://www.sciencedirect.com/science/article/pii/S0021999173800142

[18] C. K. Huang, Y. Zeng, Y. Wang, M. D. Meyers, S. Yi, B. J. Albright, Finite grid instability and spectral fidelity of the electrostatic Particle-In-Cell algorithm, Computer Physics Communications 207 (2016) 123–135. doi:10.1016/j.cpc.2016.05.021.
URL http://www.sciencedirect.com/science/article/pii/S001046551630145X

[19] S. E. Parker, W. W. Lee, A fully nonlinear characteristic method for gyrokinetic simulation, Physics of Fluids B: Plasma Physics 5 (1) (1993) 77–86. doi:10.1063/1.860870
URL https://aip.scitation.org/doi/abs/10.1063/1.860870