



A particle-in-Fourier method with semi-discrete energy conservation for non-periodic boundary conditions

Changxiao Nigel Shen^a, Antoine Cerfon^b, Sriramkrishnan Muralikrishnan^{c,*}

^a Massachusetts Institute of Technology, USA

^b Type One Energy Group - Canada Inc., Canada

^c Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Germany

ARTICLE INFO

Keywords:

Particle-in-Fourier

Particle-in-cell

Spectral method

Energy conservation

Free-space boundary conditions

ABSTRACT

We introduce a novel particle-in-Fourier (PIF) scheme based on [1,2] that extends its applicability to non-periodic boundary conditions. Our method handles free space boundary conditions by replacing the Fourier Laplacian operator in PIF with a mollified Green's function as first introduced by Vico-Greengard-Ferrando [3]. This modification yields highly accurate free space solutions to the Vlasov-Poisson system, while still maintaining energy conservation up to an error bounded by the time step size. We also explain how to extend our scheme to arbitrary Dirichlet boundary conditions via standard potential theory, which we illustrate in detail for Dirichlet boundary conditions on a circular boundary. We support our approach with proof-of-concept numerical results from two-dimensional plasma test cases to demonstrate the accuracy, efficiency, and conservation properties of the scheme. By avoiding grid heating and finite grid instability we are able to show an order of magnitude speedup compared to the standard PIC scheme for a long time integration cyclotron simulation.

1. Introduction

The particle-in-cell (PIC) method has become a common approach to numerically solve kinetic equations in plasma physics [4–8]. It is physically intuitive, comparatively easy to parallelize, and robust for a wide range of physical scenarios. Traditional PIC schemes, however, are known to be unable to preserve energy [9,10], leading to a non-physical phenomenon known as “grid-heating” and numerical instability for long-time simulations [11]. Earlier attempts to improve the conservation properties of PIC fall into several categories: some are based on the Lagrangian formulation in [12], such as [13,14]; others take advantages of implicit integrators, such as [15,16]. Structure-preserving geometric PIC schemes are also being developed based on a variational formulation [17,18] and discretization of the underlying Hamiltonian structure [19,20]. Although many of these schemes have excellent long time stability and conservation properties, they fail to conserve one of either charge or momentum. They also differ significantly from the standard PIC framework and hence may not be as intuitive or easy to transition to from an application and implementation point of view.

Among recent developments for structure-preserving PIC, the particle-in-Fourier (PIF) scheme developed in [1,2] retains the intuitive nature of standard PIC schemes as well as their relative simplicity, and at the same time has excellent conservation properties. By exploiting the non-uniform fast Fourier transform [21–23], PIF directly evaluates the Fourier modes of the field efficiently, without any spatial charge spreading and interpolation processes. Both charge and momentum are conserved in this scheme, and energy is

* Corresponding author.

E-mail address: s.muralikrishnan@fz-juelich.de (S. Muralikrishnan).

conserved in the continuous time limit. The finite-grid instability and mode-coupling effects are also eliminated [24]. In [1] the authors show that in finite dimensions PIF is the only scheme which can conserve both energy and momentum simultaneously before time discretization. Even though the exact energy conservation is lost after time discretization with explicit time integrators, the scheme still possesses excellent stability and convergence properties for long time integration simulations as shown in Section 4.2.1. PIF scheme combined with implicit time integration has not been explored yet and could be an interesting research direction to explore in the future. The use of the non-uniform fast Fourier transform reduces the complexity of the algorithm from $\mathcal{O}(N_p N_m^d)$ to $\mathcal{O}(N_p + N_m^d \log N_m^d)$, where N_p is the total number of particles and N_m is the number of Fourier modes used in the simulation. Such a Fourier approach has also proved to be successful in other numerical contexts, such as in its application for the immersed boundary method [25].

On the other hand, the choice of a Fourier basis is naturally incompatible with non-periodic boundary conditions, such as free space and Dirichlet boundary conditions. Although particle methods are frequently applied to periodic problems [26,27], most problems of physics and engineering interest cannot be approximated accurately with periodic boundary conditions [26,28,29]. One of the strengths of particle methods have been their adaptability to complex geometries and boundary conditions [30,31,28]. Limiting PIF to periodic problems would severely limit its range of applicability and its desirability. To handle nonperiodic boundary conditions, one could consider other basis functions, such as Legendre polynomials. Unfortunately, no basis other than the Fourier basis has been found that preserves the spatial translational invariance of the Lagrangian describing the kinetic plasma of interest [1,32].

In this paper, we propose a modified version of PIF scheme enabling the method to adequately handle free space boundary conditions and Dirichlet boundary conditions. We replace the periodic Fourier Laplacian operator $1/k^2$ used in PIF with a spatially truncated free space Green's function $\hat{g}^L(\mathbf{k})$ proposed in [3], with its truncation radius L determined based on our computational domain and the shape function that we choose. It has been shown in [3,33] that the Fourier kernel \hat{g}^L gives us a spectrally accurate solution to the free space Poisson problem for any smooth source $\rho(\mathbf{x})$ on a Cartesian grid. We show that this kernel yields a spectrally accurate solution even if the source comes from Lagrangian particles, which are not aligned with grid points. Therefore, by applying \hat{g}^L to our PIF scheme, we are able to obtain free space solutions to the Vlasov-Poisson system with high accuracy and exact charge, momentum and semi-discrete energy conservation. Additionally, by combining the free space PIF algorithm with a Laplace solver based on boundary integral method, we can obtain the solution to any Dirichlet problem.

The remainder of this paper is structured as follows. In Section 2, we revisit the numerical details of the particle-in-cell and particle-in-Fourier algorithms; then we give a brief overview of the free space Poisson solver proposed by Vico, Greengard and Ferrando [3], which plays a key role in our new numerical scheme. In Section 3, we present our strategy of combining the free space Poisson solver and the Laplace solver with the PIF algorithm, followed by an error analysis for the conservation of energy and momentum. In Section 4, we report our numerical results using two-dimensional test cases. In Section 5, we summarize our work and suggest directions for future work.

2. Prerequisites

Among its various applications, PIC is very often used to solve the kinetic equations that arise in plasma physics [34,9]. In this work, we will focus on the kinetic equation describing the evolution of a single species, collisionless, electrostatic plasma immersed in a constant, uniform magnetic field \mathbf{B}_0 . The evolution of the distribution function $f(\mathbf{x}, \mathbf{v}, t)$ of that single-species plasma is described by the following modified Vlasov-Poisson system (here we set the vacuum permittivity ϵ_0 to 1 for simplicity):

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}_0) \cdot \nabla_{\mathbf{v}} f = 0, \quad (1)$$

$$\mathbf{E} = -\nabla \varphi, \quad -\Delta \varphi = q \int_{\mathbb{R}^3} f d\mathbf{v} \equiv \rho. \quad (2)$$

We note that in equation (2), a background charge density ρ_0 is often subtracted in the context of periodic boundary conditions. In contrast, in this manuscript, we are interested in situations in which the charge density ρ is the physical charge density, which may be the charge density of an isolated non-neutral beam. The only requirement on ρ in this work is that it must have a compact support.

2.1. Particle-in-cell method

In order to solve the system of equations (1)–(2), PIC represents the distribution function as particles in a continuous phase space, each particle having a unique position \mathbf{X}_j and velocity \mathbf{V}_j . The motion of the particles is described by Newton's law:

$$\dot{\mathbf{X}}_j = \mathbf{V}_j, \quad \dot{\mathbf{V}}_j = \frac{q}{m} (\mathbf{E}_j + \mathbf{V}_j \times \mathbf{B}_0), \quad (3)$$

where \mathbf{E}_j indicates the self-consistent electric field \mathbf{E} evaluated at the particle position \mathbf{X}_j .

In the PIC method, \mathbf{E} is first computed on a grid, before it is evaluated at the particle positions via interpolation in order to advance the particles via Newton's law. Specifically, the charge density carried by the particles is spread onto grid points according to

$$\rho(\mathbf{x}) = \sum_{j=1}^{N_p} q S(\mathbf{X}_j - \mathbf{x}) \quad (4)$$

where S is the particle shape function. S is usually chosen as a b-spline function of a certain order, which ensures exact charge conservation and retains positivity of the interpolation at any spatial location. Once the charge density ρ is known at the grid points, one can compute \mathbf{E} and φ by solving Poisson's equation on that grid using standard solvers. For example, for periodic boundary conditions and a uniform Cartesian grid, the electric field may be efficiently computed with Fast Fourier Transforms (FFTs):

$$\hat{\rho}(\mathbf{k}) = \sum_{\mathbf{x}} \rho(\mathbf{x}) \exp(-i\mathbf{k} \cdot \mathbf{x}), \quad (5)$$

$$\hat{\varphi}(\mathbf{k}) = \frac{1}{k^2} \hat{\rho}(\mathbf{k}), \quad \varphi(\mathbf{x}) = \sum_{\mathbf{k}} \hat{\varphi}(\mathbf{k}) \exp(i\mathbf{k} \cdot \mathbf{x}), \quad (6)$$

$$\hat{\mathbf{E}}(\mathbf{k}) = \frac{\mathbf{k}}{ik^2} \hat{\rho}(\mathbf{k}), \quad \mathbf{E}(\mathbf{x}) = \sum_{\mathbf{k}} \hat{\mathbf{E}}(\mathbf{k}) \exp(i\mathbf{k} \cdot \mathbf{x}). \quad (7)$$

Here, we use the notation $\sum_{\mathbf{k}}$ to denote the summation taken over a finite number of modes \mathbf{k} . Specifically, if we let $\mathbf{k} = [k_1, k_2]$, then

$$\sum_{\mathbf{k}} f(\mathbf{k}) = \sum_{k_1=-N_m/2}^{N_m/2-1} \sum_{k_2=-N_m/2}^{N_m/2-1} f([k_1, k_2]) \quad (8)$$

for an arbitrary function $f(\mathbf{k})$, where N_m is the number of modes in each dimension. This convention is followed throughout the paper. Once the electric field is known at all grid points, it is evaluated at the particle positions by interpolation with the same shape function S :

$$\mathbf{E}(\mathbf{X}_j) = \sum_{\mathbf{x}} \mathbf{E}(\mathbf{x}) S(\mathbf{x} - \mathbf{X}_j). \quad (9)$$

2.2. Particle-in-Fourier scheme

In order to improve energy conservation in the PIC method, a new scheme called particle-in-Fourier (PIF) has been proposed [1,2]. In contrast to the standard PIC approach, the PIF scheme avoids grid heating effects by directly depositing the charge in Fourier space, as well as interpolating the electric field from the Fourier space to the particle locations. Let us assume our problem domain to be $\Omega = \bigotimes_d \left[-\frac{1}{2}, \frac{1}{2}\right]$, where d is the number of dimensions of the problem of interest, and denote N_m^1 as the number of Fourier modes in each dimension. If we perform a Fourier transform to (4), we get

$$\hat{\rho}(\mathbf{k}) = \int_{\Omega} q \sum_{j=1}^{N_p} S(\mathbf{X}_j - \mathbf{x}) \exp(-i\mathbf{k} \cdot \mathbf{x}) d\mathbf{x}, \quad (10)$$

$$= q \hat{S}(\mathbf{k}) \sum_{j=1}^{N_p} \exp(-i\mathbf{k} \cdot \mathbf{X}_j), \quad (11)$$

where \mathbf{k} is of the form $2\pi \left(\sum_d n_d \hat{\mathbf{d}} \right)$ for integers $n_d \in \left[-\frac{N_m}{2}, \frac{N_m}{2} - 1\right]$, $\hat{\mathbf{d}}$ being the unit vector in d -th dimension. Here, the Fourier coefficients of the shape function $\hat{S}(\mathbf{k})$ can be evaluated analytically. Note that $\hat{\rho}(\mathbf{k})$ in equation (11) is different from the one in (5). This is because the b-spline functions are not band-limited in Fourier space, and in (11) we have to truncate its analytical Fourier spectrum down to N_m modes. We then solve for the electric field using equation (7). Once we obtain $\hat{\mathbf{E}}(\mathbf{k})$ and $\hat{\varphi}(\mathbf{k})$ using the standard spectral method (equations (6) and (7)), we evaluate the acceleration as

$$\mathbf{a}_j = \frac{q}{m} \left(\mathbf{v}_j \times \mathbf{B}_0 + \sum_{\mathbf{k}} \hat{\mathbf{E}}(\mathbf{k}) \hat{S}(\mathbf{k}) \exp(i\mathbf{k} \cdot \mathbf{X}_j) \right). \quad (12)$$

Note that both (11) and (12) require discrete Fourier transforms of data between equi-spaced points in Fourier space and non-uniformly distributed particle positions. This can be evaluated efficiently using the so-called non-uniform fast Fourier transform (NUFFT) [22,21]. Type 1 NUFFT evaluates the summation

$$\hat{f}(\mathbf{k}) = \sum_{j=1}^{N_p} f(\mathbf{X}_j) \exp(-i\mathbf{X}_j \cdot \mathbf{k}), \quad (13)$$

and its dual, Type 2 NUFFT, evaluates

$$f(\mathbf{X}_j) = \sum_{\mathbf{k}} \hat{f}(\mathbf{k}) \exp(i\mathbf{X}_j \cdot \mathbf{k}). \quad (14)$$

¹ For simplicity let us assume N_m to be even.

Both Type 1 and 2 NUFFT have a complexity of $\mathcal{O}(N_p + N_m^d \log N_m^d)$, which is much faster than a naive direct sum with complexity $\mathcal{O}(N_p N_m^d)$. The NUFFT thus makes the PIF scheme affordable to run long-time particle simulations with high numbers of particles and Fourier modes.

2.3. Vico-Greengard-Ferrando free space Poisson solver

Consider Poisson's equation

$$\Delta\varphi = -\rho, \quad (15)$$

in \mathbb{R}^d with open (free space) boundary conditions, and where ρ is a source distribution with a compact support Ω . Without loss of generality, let us suppose that $\Omega = \bigotimes_d \left[-\frac{1}{2}, \frac{1}{2}\right]$ is a d -dimensional unit box. The solution to Poisson's equation can then be written as

$$\varphi(\mathbf{x}) = \int_{\Omega} g(\mathbf{x} - \mathbf{y}) \rho(\mathbf{y}) d\mathbf{y}, \quad (16)$$

where $g(\mathbf{r}) = \frac{1}{4\pi r}$ for $d = 3$ and $g(\mathbf{r}) = \frac{-1}{2\pi} \log(r)$ for $d = 2$.

The slow decay of the Green's functions g requires the application of fast algorithms for the efficient computation of the convolution in (16). For uniform Cartesian grids, this can be done with the convolution theorem: the Fourier transform of the convolution of the two functions in configuration space is the point-wise product of their Fourier transforms. Equation (16) can therefore be evaluated efficiently by computing the Fourier transforms of g and ρ via FFTs, multiplying their Fourier transforms, and computing the inverse Fourier transform of the result, again with an FFT. This is what the Hockney-Eastwood scheme [34,35], commonly used in PIC codes to deal with free space boundary conditions, does, by zero padding the charge distribution in each spatial direction on an interval of the same length as the original domain, and defining a periodic Green's function \tilde{g} on that extended domain. However, the scheme is only second-order accurate [33], because of the second difficulty arising in (16), namely the fact that the Green's functions are singular at the origin. Hockney and Eastwood propose regularizing the Green's functions according to $\tilde{g}(\mathbf{0}) = 1$. The lack of smoothness of \tilde{g} associated with this choice limits the convergence to second order [36,37]. Other elementary regularization procedures have been proposed [38,39], but they do not generally increase the order of convergence because they do not improve the smoothness of the regularized Green's function [36,37].

The Vico-Greengard-Ferrando scheme also relies on a uniform grid, the convolution theorem and FFTs, but significantly improves the accuracy of the computation of the convolution in (16) as compared to Hockney-Eastwood, by addressing the singularity of the Green's functions at the origin in a different manner.

First, note that the Fourier transforms of the Poisson Green's functions are in fact known analytically. However, a difficulty arises from the fact that in Fourier space, this transform diverges like $1/|\mathbf{k}|^2$ at the origin. Vico, Greengard and Ferrando address that difficulty as follows. The key observation is that since we are only interested in the values of φ and \mathbf{E} inside Ω , one can substitute the Green's functions g with truncated Green's functions g^L without affecting the results in Ω . Specifically, let

$$g^L(\mathbf{r}) = \begin{cases} \frac{1}{4\pi r} \text{rect}\left(\frac{r}{2L}\right) & \text{if } d = 3 \\ \frac{-1}{2\pi} \log(r) \text{rect}\left(\frac{r}{2L}\right) & \text{if } d = 2 \end{cases}, \quad (17)$$

where $\text{rect}(x)$ is the characteristic function on the unit interval:

$$\text{rect}(x) = \begin{cases} 1 & \text{for } |x| < 1/2 \\ 0 & \text{otherwise} \end{cases}. \quad (18)$$

If we set $L > \sqrt{d}$, then the solution φ in Ω is exactly

$$\varphi(\mathbf{x}) = \int_{\Omega} g^L(\mathbf{x} - \mathbf{y}) \rho(\mathbf{y}) d\mathbf{y}. \quad (19)$$

The advantage of truncating the Green's functions in physical space in this manner is that it removes the singularity of their Fourier transforms. Indeed, the Fourier transforms of the truncated Green's functions can also be computed analytically, and are the following smooth functions:

$$\mathcal{F}(g^L) = \hat{g}^L = \begin{cases} 2 \left(\frac{\sin(Ls/2)}{s} \right)^2 & \text{if } d = 3 \\ \frac{1 - J_0(Ls)}{s^2} - \frac{L \log(L) J_1(Ls)}{s} & \text{if } d = 2 \end{cases}. \quad (20)$$

Because of the oscillatory nature of the truncated Green's functions \hat{g}^L , Vico, Greengard and Ferrando explain that their scheme in principle requires zero-padding to an extended domain $\tilde{\Omega}$ with a grid of size $(4N_g)^d$ in order to compute the convolution (16) without aliasing errors. That would make their scheme more expensive than the Hockney-Eastwood method. However, they explain that the slightly more unfavorable scaling of the scheme may just be applied to a precomputation step to compute the numerical mollified Green's function T on a grid of size $(2N_g)^d$. Once computed, this numerical kernel T can be used just like the periodic \tilde{g} in the Hockney-Eastwood scheme, to compute φ with the same computational complexity as in the Hockney-Eastwood scheme, via the convolution theorem and FFTs, and with much higher accuracy. Recently a modification taking advantage of the symmetry has been proposed in [40], which improves the memory storage costs of the Vico-Greengard-Ferrando scheme similar to that of the Hockney-Eastwood solver.

3. Combining PIF with a spectral free space Poisson solver

In this section, we present a novel way to combine PIF with the Poisson solver developed by Vico et al., [3] so that we retain the advantages of both, and can solve problems with free-space boundary conditions. Specifically, we can calculate the electric forces on particles efficiently and accurately for plasmas in free space, while achieving good conservation of charge, momentum, and energy.

3.1. A gridless free space Poisson solver

Since Vico et al. solve Poisson's equation by convoluting the source with a mollified Green's function via the Fourier convolution theorem, we can view their solver as a spectral solver which is inherently compatible with PIF. Indeed, as shown in (11), in PIF we naturally evaluate $\hat{\rho} = \mathcal{F}(\rho)$ required for the Vico-Greengard-Ferrando scheme with the NUFFT. Note that here, the charge density is again just

$$\rho = qS(\mathbf{x}) * \left(\sum_j \delta(\mathbf{X}_j - \mathbf{x}) \right) \quad (21)$$

where $(*)$ denotes the convolution operation. Since we now impose free space boundary condition for the Poisson solver, it is more natural to use radially symmetric shape functions. Hence, here $S(\mathbf{x})$ is a radially symmetric function, as opposed to tensor product B-spline kernels commonly used in PIC [9]. This is particularly convenient for the PIF scheme because one can analytically determine the Fourier coefficients of radially symmetric shape functions. The use of radially symmetric shape functions has two main advantages, namely 1) they preserve rotational invariance, as it should be in the case of free space boundary conditions, and 2) the convolution between the mollified Green's function and the shape function is still radially symmetric, as we will see later in this section. Some sample radially symmetric shape functions that we used in this work are provided in Appendix A.

For the solution to Poisson's equation, we may therefore write:

$$\varphi(\mathbf{x}) = g^L(\mathbf{x}) * \rho(\mathbf{x}) = q(g^L * S)(\mathbf{x}) * \left(\sum_j \delta(\mathbf{X}_j - \mathbf{x}) \right), \quad (22)$$

$$\mathbf{E}(\mathbf{x}) = -\nabla\varphi(\mathbf{x}). \quad (23)$$

Now, in order to determine the acceleration, we need to perform one more convolution between the fields and the shape function S . We may view the results as potential and electric force fields mollified by the shape function. For the sake of conciseness, let us denote these mollified fields as

$$\psi(\mathbf{x}) = (\varphi * S)(\mathbf{x}), \quad (24)$$

$$\mathcal{E}(\mathbf{x}) = (\mathbf{E} * S)(\mathbf{x}). \quad (25)$$

The accelerations are expressed as

$$\mathbf{a}_s = \frac{q}{m} (\mathcal{E}(\mathbf{X}_s) + \mathbf{V}_s \times \mathbf{B}_0), \quad (26)$$

$$= \frac{q}{m} (-(\nabla\varphi * S)(\mathbf{X}_s) + \mathbf{V}_s \times \mathbf{B}_0), \quad (27)$$

$$= \frac{q}{m} \left[-q \nabla (g^L * S * S) * \left(\sum_j \delta(\mathbf{X}_j - \mathbf{x}) \right) * \delta(\mathbf{x} - \mathbf{X}_s) + \mathbf{V}_s \times \mathbf{B}_0 \right]. \quad (28)$$

For our scheme, we view $g^L * S$ and $\nabla(g^L * S * S)$ as our new convolution kernels in replacement of g^L , and apply the same Vico-Greengard-Ferrando scheme to compute the acceleration:

$$\mathbf{a}_s = \frac{q}{m} \left[-iq \sum_{\mathbf{k}} \left(\mathbf{k} \hat{g}^L(\mathbf{k}) \hat{S}^2(\mathbf{k}) \sum_j \exp(-i\mathbf{k} \cdot \mathbf{X}_j) \right) \exp(i\mathbf{k} \cdot \mathbf{X}_s) + \mathbf{V}_s \times \mathbf{B}_0 \right], \quad (29)$$

This can be performed efficiently using Type 1 & 2 NUFFT.

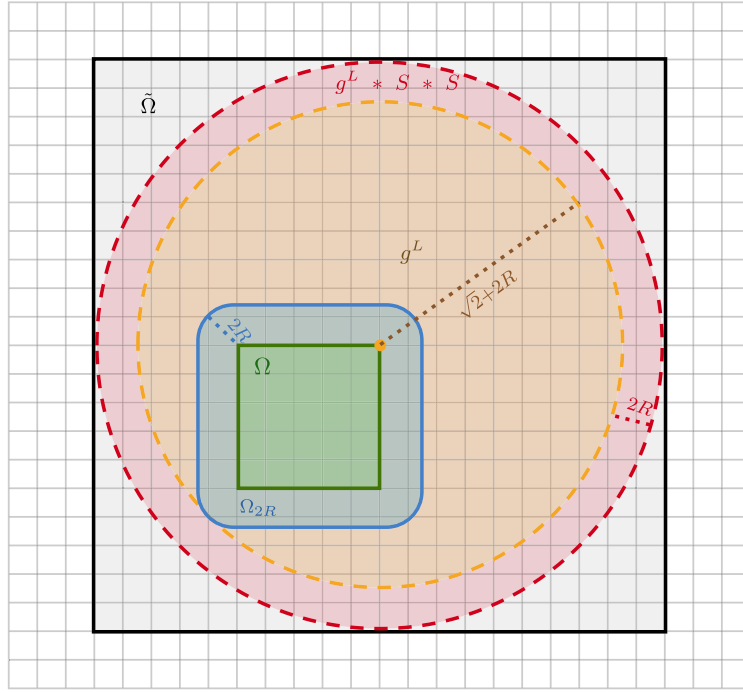


Fig. 1. Our simulation domain based on the shape function and mollified Green's function chosen. Particles are within the domain Ω (green), and $S * S * \sum_j \delta(\mathbf{x} - \mathbf{X}_j)$ has a support Ω_{2R} (blue). The radius of the Green's function is chosen such that g^L is able to cover the entire blue region Ω_{2R} whenever the center of g^L is located inside the green region Ω . To determine the smallest possible radius, we place the center of g^L at the upper right corner of Ω . After a simple calculation, we find the smallest radius to be $\sqrt{2} + 2R$ for g^L . Finally, the extended domain is $\tilde{\Omega}$ (gray). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

We observe at this point that the explicit computations of the electric and potential fields are not required at any given time step of the PIF simulation, since Eq. (29) is sufficient to push the particles at any time step. The potential and electric force fields could therefore in principle only be computed as a post-processing step if desired, when analyzing the results of the simulations. For example, the potential field φ can be computed using the following expression:

$$\hat{\varphi}(\mathbf{k}) = q \hat{g}^L(\mathbf{k}) \hat{S}(\mathbf{k}) \sum_j \exp(-i\mathbf{k} \cdot \mathbf{X}_j). \quad (30)$$

The simplicity of the formulation described above partially conceals two minor difficulties, in that we need to be careful in choosing the size of the Fourier spectrum for the convolutions, and the radius of the mollified Green's function g^L .

Let us start with the second point. Vico et al. choose $L = 1.5 > \sqrt{d}$ in two dimensions so that the convolution between ρ and g^L gives the correct potential inside Ω (here we assume $d = 2$ but the analysis applies to any dimension). In our solver, instead of g^L , the kernel function is $\nabla(g^L * S * S)$: our goal here is to correctly evaluate ψ and \mathcal{E} at any target position inside Ω . Since the radius of the shape function is predetermined and the particle positions are given at each iteration, let us view $S * S * \sum_j \delta(\mathbf{x} - \mathbf{X}_j)$ as the source of the field. Its compact support Ω_{2R} is larger than the original domain Ω by an amount $2R$, as shown in Fig. 1. Following this observation, we choose the size of our mollified Green's function based on the following criterion:

$$\forall \mathbf{x}_0 \in \Omega, \quad \Omega_{2R} \subset D_L(\mathbf{x}_0) \quad (31)$$

where $D_L(\mathbf{x}_0)$ represents the disk of radius L centered on \mathbf{x}_0 . In other words, when we translate the center of the mollified Green's function to any possible particle position inside Ω , seen as a target, the distance from that target to any source in Ω_{2R} should be less than L , so that the mollified Green's function gives the same result as the true Green's function, and ψ and \mathcal{E} can be correctly evaluated at particle positions. It is easy to see that this is satisfied if $L \geq \sqrt{d} + 2R$, as shown in Fig. 1. When this condition is set for L , we note that the radius of $g^L * S$ is larger than $\sqrt{d} + 3R$ and the radius of $g^L * S * S$ is larger than $\sqrt{d} + 4R$. In practice, the radius of the shape function is small compared to \sqrt{d} , so that the modification to the minimal radius associated with R is not significant. In our two-dimensional examples below, we chose $L = 1.5$ as for the standard Poisson solver in two dimensions by Vico, Greengard and Ferrando.

Now, let us address the question of the required Fourier resolution. As discussed in Section 2.3, Vico et al. explain that a zero-padding region extending to $\tilde{\Omega} = \otimes_d [-2, 2]$, which is four times larger than the original domain Ω , is necessary so that the aperiodic convolutions do not lead to aliasing, and so that the oscillatory Fourier transforms of the convolution kernels are properly sampled. Our scheme has the same sampling requirements, unless we implement a precomputation step as suggested in [3] and discussed in the

next section. To accurately compute our convolutions in the absence of the precomputation step, we thus upscale the resolution of the Fourier grid by a factor of 4. Although all our operations are done in Fourier space when we do not implement the precomputation step, in the remainder of this article we will sometimes refer to this upsampling in Fourier space as zero-padding of the domain Ω to the extended domain $\tilde{\Omega} = \bigotimes_d [-2, 2]$, thereby using the same language as Vico et al.

3.2. On precomputation

As discussed in Section 2.3, in [3] Vico et al. explain that in the common case in which the simulation domain and its grid remain fixed during the entire simulation, the extended computational grid with $(4N_g)^d$ grid points may only be used once, at the start of the simulation. This is to compute the real space representation T on the computational grid of the convolutional kernel for the solution to Poisson's equation, which can be interpreted as a mollified Green's function associated with g^L . Once T is computed, the electrostatic potential is calculated for any grid point in the domain Ω with the following convolution (for a two-dimensional setting)

$$\varphi(i, j) = \sum_{i', j'} T(i - i', j - j') \rho(i, j), \quad (32)$$

where i, j are indices of the grid points. The advantage of precomputing T in this manner is that Eq. (32) may then be evaluated using standard aperiodic convolution, which only requires zero-padding by a factor of 2. As long as the simulation domain and the discretization remain the same, T can be reused throughout the simulation, and only $(2N_g)^d$ grid points are needed for the calculation of the electrostatic potential.

Our PIF scheme does not lend itself naturally to the same precomputation method. Indeed, in our algorithm, the locations of the targets at which the potential must be evaluated are not known in advance, and change from one time step to the next. One therefore has two choices. The first choice is to rely on a quadruply extended grid in each direction as discussed above throughout the simulation, guaranteeing the absence of aliasing errors and spectral accuracy for our Poisson solver at each time step. The second choice is to precompute effective convolution operators T_1 and T_2 associated with the two kernels $g^L * S$ and $\nabla g^L * S * S$ via inverse FFTs for a quadruply refined Fourier grid in each direction, as in Vico et al., and then use T_1 and T_2 for standard aperiodic convolutions on a doubly extended grid in each direction for the entire simulation. This second approach is less expensive computationally. However, since T_1 and T_2 only lead to spectrally accurate quadrature rules for target locations matching those of the precomputation, this second approach introduces aliasing errors at the arbitrary target locations of our particles, as with standard FFT-based Poisson solvers [34]. Accuracy is therefore reduced in the same way as for these solvers, with second order convergence to be expected. Both choices have merits, depending on the application, and on the number of Fourier modes and of particles for the simulation. In our numerical tests below, we will show results obtained for each of these two options. In most situations, the numerical error of particle based schemes such as ours is dominated by the noise associated with the Monte Carlo nature of the algorithm, and a second-order convergent Poisson solver provides sufficient accuracy. In such cases, relying on T_1 and T_2 and smaller padding regions does not lead to noticeable loss of accuracy, and is the most computationally efficient choice.

We note that the Hockney-Eastwood algorithm [34] also has second order convergence, and for a standard PIC scheme has the advantage that an analytical expression for the Green's function is available in the physical space. One may then think that it could be considered as a simpler alternative to the Vico-Greengard solver with precomputation. However, for our PIF scheme, this advantage is irrelevant, as we have to convolve the Green's function with the shape function in Fourier space. In order to perform the convolution correctly, the usual extension of the Green's function to a $2L$ domain as in the standard Hockney-Eastwood solver will not be sufficient and the Green's function has to be zero padded adequately depending on the radius of the shape function before the convolution. This partially defeats the simplicity of the Hockney-Eastwood algorithm. On the other hand, the Vico-Greengard solver is a natural option for PIF schemes as the mollified Green's function is directly given as an analytical expression in Fourier space.

3.3. Dirichlet boundary conditions

Following standard potential theory [41], our free space particle-in-Fourier (FSPIF) scheme can be combined with Laplace solvers in order to simulate systems with Dirichlet boundary conditions. Consider the standard Poisson problem with Dirichlet boundary conditions:

$$\begin{cases} \Delta \varphi = -\rho, & \text{in } \Omega_\rho \\ \varphi = f & \text{on } \partial\Omega_\rho \end{cases}, \quad (33)$$

where $\partial\Omega_\rho$ is the boundary of Ω_ρ . Let φ^P be the free-space solution to the Poisson equation $\Delta \varphi^P = -\rho$ discussed previously, and let $\varphi := \varphi^P + \varphi^H$, where φ^H is a harmonic function satisfying the following Laplace problem

$$\begin{cases} \Delta \varphi^H = 0, & \text{in } \Omega_\rho \\ \varphi^H = f - \varphi^P & \text{on } \partial\Omega_\rho \end{cases}. \quad (34)$$

Then φ solves the desired Poisson problem with Dirichlet boundary conditions, equation (33).

Here, we illustrate this generalization with a simple example, corresponding to two-dimensional plasmas confined within a disk on whose boundary one may apply a potential f , which may depend on the position along the disk boundary. For this particular situation, it is well-known that explicit solutions to equation (34) may be constructed via elementary tools of complex analysis. Let us remind the reader of this construction. Suppose that φ satisfies

$$\begin{cases} \Delta\varphi = -\rho, & \text{in } D_1(\mathbf{0}) \\ \varphi = f & \text{on } C_1(\mathbf{0}) \end{cases}. \quad (35)$$

As explained above, we split the solution $\varphi = \varphi^H + \varphi^P$, where φ^P is the free space solution which we numerically solve according to [3]. The other component, φ^H , satisfies Laplace's equation on the unit disk:

$$\begin{cases} \Delta\varphi^H = 0, & \text{in } D_1(\mathbf{0}) \\ \varphi^H = f - \varphi^P & \text{on } C_1(\mathbf{0}) \end{cases}. \quad (36)$$

Poisson's formula then gives an explicit expression for φ^H [42]:

$$\varphi^H(\mathbf{x}) = \frac{1}{2\pi} \int_{|\mathbf{z}|=1} \frac{1-|\mathbf{x}|^2}{|\mathbf{z}-\mathbf{x}|^2} (f - \varphi^P)(\mathbf{z}) d\theta. \quad (37)$$

Writing $\mathbf{z} = [u, v]$, $\mathbf{x} = [x, y]$, the components of the electric field at the particle position corresponding to φ^H are

$$E_x^H = \frac{1}{\pi} \int_{|\mathbf{z}|=1} \frac{x[(u-x)^2 + (v-y)^2] - [1 - (x^2 + y^2)](u-x)}{[(u-x)^2 + (v-y)^2]^2} (f(\mathbf{z}) - \varphi^P(\mathbf{z})) d\theta, \quad (38)$$

$$E_y^H = \frac{1}{\pi} \int_{|\mathbf{z}|=1} \frac{y[(u-x)^2 + (v-y)^2] - [1 - (x^2 + y^2)](v-y)}{[(u-x)^2 + (v-y)^2]^2} (f(\mathbf{z}) - \varphi^P(\mathbf{z})) d\theta. \quad (39)$$

The integrals involved in the computation of the electrostatic potential and the electric field can easily be approximated numerically with the trapezoidal quadrature rule, which has spectral accuracy for analytic functions [43]. Let us assume that we put N_B number of equi-spaced points \mathbf{z}_j on the boundary, we can then approximate φ^H using the following equation:

$$\varphi_{\text{num}}^H(\mathbf{x}) = \frac{1}{N_B} \sum_{j=1}^{N_B} \frac{1-|\mathbf{x}|^2}{|\mathbf{z}_j - \mathbf{x}|^2} (f - \varphi^P)(\mathbf{z}_j), \quad (40)$$

$$E_{x,\text{num}}^H = \frac{2}{N_B} \sum_{j=1}^{N_B} \frac{x[(u_j - x)^2 + (v_j - y)^2] - [1 - (x^2 + y^2)](u_j - x)}{[(u_j - x)^2 + (v_j - y)^2]^2} (f(\mathbf{z}_j) - \varphi^P(\mathbf{z}_j)), \quad (41)$$

$$E_{y,\text{num}}^H = \frac{2}{N_B} \sum_{j=1}^{N_B} \frac{y[(u_j - x)^2 + (v_j - y)^2] - [1 - (x^2 + y^2)](v_j - y)}{[(u_j - x)^2 + (v_j - y)^2]^2} (f(\mathbf{z}_j) - \varphi^P(\mathbf{z}_j)). \quad (42)$$

As long as the applied potential f is smooth enough and the charge density ρ is smooth enough for the free-space electrostatic potential φ^P to be smooth, and as long as we do not evaluate the electrostatic potential too close to the boundary of the disk [44], we get high order convergence for φ^H and \mathbf{E} with a simple application of the trapezoidal rule. For situations in which these circumstances are not satisfied, specialized quadrature rules could be implemented to maintain high order accuracy [44].

However, in order to determine the acceleration, we need to accurately evaluate $\psi^H = \varphi^H * S$ at particle positions. In our PIF framework, this is not trivial, because in order to compute convolutions in Fourier space, we need $\hat{\varphi}^H$, which is not readily available given the boundary integral form of φ^H . Here, we show that this convolution through Fourier space can be avoided if some additional conditions are met. To do so, we make the following assumptions: the shape function S is spherically symmetric, and φ , which is originally defined on the unit disk, is smooth enough to be analytically extended to $D_{1+R}(\mathbf{0})$ where R is the radius of the shape function. Note that we do not have to evaluate this analytical extension, but we do need to assume its existence for the following derivation. Recall that $\psi = \varphi * S$, so that ψ satisfies

$$\Delta\psi(\mathbf{x}) = -(\rho * S)(\mathbf{x}), \quad \mathbf{x} \in D_{1+R}(\mathbf{0}). \quad (43)$$

In general, it is a complicated matter to determine the boundary conditions that ψ satisfy, and therefore solve Eq. (43). However, if we further assume that ρ has a slightly smaller compact support $D_{1-R}(\mathbf{0})$, which is often the case in applications [45,46], then we have

$$\Delta\varphi = 0 \quad \text{in} \quad \bigcup_{\mathbf{x} \in C_1(\mathbf{0})} D_R(\mathbf{x}). \quad (44)$$

We can therefore use the mean value property of harmonic functions for φ and the spherical symmetry of the shape function to obtain the following equalities: $\forall \mathbf{x} \in C_1(\mathbf{0})$,

$$\psi(\mathbf{x}) = \int_{D_R(\mathbf{x})} S(\mathbf{y} - \mathbf{x}) \varphi(\mathbf{x}) d\mathbf{y}, \quad (45)$$

$$= \int_0^R S(r) \int_{C_r(\mathbf{x})} \varphi(\mathbf{y}) d\mathbf{y} dr, \quad (46)$$

$$= \int_0^R 2\pi r \cdot S(r) \varphi(\mathbf{x}) dr, \quad (47)$$

$$= \varphi(\mathbf{x}) = f. \quad (48)$$

This identity provides the required boundary condition for $\psi(\mathbf{x})$. We thus conclude that $\psi(\mathbf{x})$ satisfies the following Poisson problem:

$$\begin{cases} \Delta \psi(\mathbf{x}) = -(\rho * S)(\mathbf{x}), & \mathbf{x} \in D_1(\mathbf{0}) \\ \psi = f & \text{on } C_1(\mathbf{0}) \end{cases}. \quad (49)$$

Let us define $\psi^P = \varphi^P * S$. Then we have

$$\Delta \psi^P = -\rho * S \quad (50)$$

with free space boundary conditions. If we call ψ^H the solution to the problem

$$\begin{cases} \Delta \psi^H(\mathbf{x}) = 0, & \mathbf{x} \in D_1(\mathbf{0}) \\ \psi^H = f - \psi^P & \text{on } C_1(\mathbf{0}) \end{cases}, \quad (51)$$

then $\psi = \psi^P + \psi^H$ satisfies Eq. (49) as desired. Similarly, $\mathcal{E} = \mathbf{E} * S$ can be found using equation (38) and equation (39) by replacing $f - \varphi^P$ with $f - \psi^P$. The potential energy due to φ^H can also be simply determined numerically:

$$U_E^H = \frac{1}{2} \int_{\Omega} \rho(\mathbf{x}) \varphi^H(\mathbf{x}) d\mathbf{x}, \quad (52)$$

$$= \frac{q}{2} \sum_{\mathbf{x}_j} \int_{\Omega} S(\mathbf{X}_j - \mathbf{x}) \varphi^H(\mathbf{x}) d\mathbf{x}, \quad (53)$$

$$= \frac{q}{2} \sum_{\mathbf{x}_j} (\varphi^H * S)(\mathbf{X}_j), \quad (54)$$

$$= \frac{q}{2} \sum_{\mathbf{x}_j} \psi^H(\mathbf{X}_j). \quad (55)$$

The simple methods of potential theory we present above do not only apply to the unit disk but to any geometry. For some two-dimensional domains, our approach can be combined with conformal mapping [42,47] for explicit or semi-explicit formulas. For other domains and for three-dimensional problems, standard numerical Laplace solvers may be used.

3.4. Algorithm outline

We combine our gridless free-space Poisson solver with the standard PIF scheme to arrive at the following algorithm, outlined below

1. Before the start of simulation, construct convolutional kernels $g^L * S$ and $\nabla(g^L * S * S)$ in Fourier space, and perform the precomputation step if desired;
2. Initialize the particle positions, velocities and charges;
3. Use Type 1 NUFFT to transform particle positions into an upscaled Fourier grid, in correspondence with the size of the extended domain $\tilde{\Omega}$. Let us relabel $\mathbf{k} \rightarrow \mathbf{k}/\alpha$ for the upscaled Fourier grid, then:

$$\hat{X}(\mathbf{k}) = \sum_{j=1}^{N_p} \exp(-i\mathbf{k} \cdot \mathbf{X}_j); \quad (56)$$

4. Find the corresponding Fourier modes of the free space potential φ and acceleration \mathbf{a} :

$$\hat{\varphi}(\mathbf{k}) = q \hat{g}^L(\mathbf{k}) \hat{S}(\mathbf{k}) \hat{X}(\mathbf{k}), \quad (57)$$

$$\hat{\mathbf{a}}(\mathbf{k}) = -\frac{ik}{m} q^2 \hat{g}^L(\mathbf{k}) \hat{S}^2(\mathbf{k}) \hat{X}(\mathbf{k}); \quad (58)$$

5. Find the free space acceleration of each particle using Type 2 NUFFT:

$$\mathbf{a}_j^P = \sum_{\mathbf{k}} \hat{\mathbf{a}}(\mathbf{k}) \exp(i\mathbf{k} \cdot \mathbf{X}_j) + \frac{q}{m} \mathbf{V}_j \times \mathbf{B}_0. \quad (59)$$

If free space boundary conditions apply to the electric field, then the true acceleration is $\mathbf{a}_j = \mathbf{a}_j^P$. Otherwise, if a Dirichlet boundary condition is imposed for the electric field, we need to additionally determine \mathbf{a}_j^H using a Laplace solver, and add \mathbf{a}_j^H to get the true acceleration $\mathbf{a}_j = \mathbf{a}_j^P + \mathbf{a}_j^H$;

6. Push the particles exactly as in standard PIC / PIF using the leapfrog algorithm if no magnetic field is applied, or using the Boris algorithm if there is a constant \mathbf{B}_0 .

We make the following observations regarding the outline of the algorithm presented above. First, strictly speaking, the computation of $\hat{\varphi}$ in step 4. is not required to push the particles in step 6. The computation of φ and $\hat{\varphi}$ may therefore only be included at certain time steps if desired, or not at all, and the electrostatic potential computed as a post-processing step if needed to interpret simulations. Second, the leapfrog and Boris algorithms are suggested as suitable algorithms for particle pushing, but our scheme could be combined with other, possibly higher order, algorithms with suitable momentum and energy conservation properties.

3.5. Energy and momentum conservation analysis

In this section, we provide energy and momentum conservation analyses for our free space particle-in-Fourier scheme, before and after time discretization.

3.5.1. Energy conservation

In [2] the authors provided a proof for the energy conservation of the PIF scheme under the continuous time limit with periodic boundary conditions. Here we generalize this proof to free space boundary conditions. Without loss of generality, we let $q = m = 1$. We define the potential energy, U_E , and the kinetic energy U_K , at time t in terms of the Fourier modes of the relevant physical quantities as follows:

$$U_E = \frac{1}{2} \sum_{\mathbf{k}} \overline{\hat{\rho}(\mathbf{k})} \hat{\varphi}^P(\mathbf{k}), \quad (60)$$

$$U_K = \frac{1}{2} \sum_j ||\mathbf{V}_j||^2. \quad (61)$$

Taking time derivative of the first quantity U_E :

$$\frac{d}{dt} U_E = \frac{d}{dt} \left(\frac{1}{2} \sum_{\mathbf{k}} \overline{\hat{\rho}(\mathbf{k})} \hat{\varphi}^P(\mathbf{k}) \right), \quad (62)$$

$$= \frac{1}{2} \sum_{\mathbf{k}} G(\mathbf{k}) \frac{d}{dt} \left(\hat{\rho}(\mathbf{k}) \overline{\hat{\rho}(\mathbf{k})} \right), \quad (63)$$

where G is the Green's function we choose depending on the boundary condition: $G = \frac{1}{k^2}$ for periodic boundary conditions, and $G = \hat{g}^L$ for free space boundary conditions. Notice that since ρ is a real function, $\hat{\rho}(\mathbf{k}) = -\hat{\rho}(-\mathbf{k})$; Also, since the Green's function g is symmetric by construction, $G(\mathbf{k}) = G(-\mathbf{k})$. Therefore, we have the following derivation:

$$\frac{d}{dt} U_E = \frac{1}{2} \sum_{\mathbf{k}} G(\mathbf{k}) \left(\hat{\rho}(\mathbf{k}) \frac{d}{dt} \overline{\hat{\rho}(\mathbf{k})} + \overline{\hat{\rho}(\mathbf{k})} \frac{d}{dt} \hat{\rho}(\mathbf{k}) \right), \quad (64)$$

$$= \frac{1}{2} \sum_{\mathbf{k}} \left(G(\mathbf{k}) \hat{\rho}(\mathbf{k}) \frac{d}{dt} \overline{\hat{\rho}(\mathbf{k})} + G(-\mathbf{k}) \overline{\hat{\rho}(-\mathbf{k})} \frac{d}{dt} \hat{\rho}(-\mathbf{k}) \right), \quad (65)$$

$$= \frac{1}{2} \sum_{\mathbf{k}} \left(G(\mathbf{k}) \hat{\rho}(\mathbf{k}) \frac{d}{dt} \overline{\hat{\rho}(\mathbf{k})} + G(\mathbf{k}) \hat{\rho}(\mathbf{k}) \frac{d}{dt} \overline{\hat{\rho}(\mathbf{k})} \right), \quad (66)$$

$$= \sum_{\mathbf{k}} G(\mathbf{k}) \hat{\rho}(\mathbf{k}) \frac{d}{dt} \left(\overline{\hat{\rho}(\mathbf{k})} \right), \quad (67)$$

$$= \sum_{\mathbf{k}} G(\mathbf{k}) \hat{\rho}(\mathbf{k}) \frac{d}{dt} \left(\sum_j \overline{\hat{S}(\mathbf{k}) \exp(-i\mathbf{k} \cdot \mathbf{X}_j)} \right), \quad (68)$$

$$= i \sum_{\mathbf{k}} \sum_j \mathbf{k} \cdot \mathbf{V}_j G(\mathbf{k}) \overline{\hat{S}(\mathbf{k})} \hat{\rho}(\mathbf{k}) \exp(i\mathbf{k} \cdot \mathbf{X}_j), \quad (69)$$

$$= i \sum_{\mathbf{k}} \sum_j \mathbf{k} \cdot \mathbf{V}_j G(\mathbf{k}) \hat{S}(\mathbf{k}) \hat{\rho}(\mathbf{k}) \exp(i\mathbf{k} \cdot \mathbf{X}_j), \quad (70)$$

where the last line comes from the fact that $\hat{S}(\mathbf{k})$ is real. Finally, we take the time derivative of the kinetic energy:

$$\frac{d}{dt} U_K = \frac{1}{2} \sum_j \frac{d}{dt} \|\mathbf{V}_j\|^2, \quad (71)$$

$$= \sum_j \mathbf{V}_j \cdot \frac{d\mathbf{V}_j}{dt}, \quad (72)$$

$$= \sum_j \mathbf{V}_j \cdot (-S * \nabla \varphi(\mathbf{X}_j)), \quad (73)$$

$$= \sum_j \mathbf{V}_j \cdot \left(- \sum_{\mathbf{k}} \hat{S}(\mathbf{k}) (i\mathbf{k} \hat{\varphi}(\mathbf{k})) \exp(i\mathbf{k} \cdot \mathbf{X}_j) \right), \quad (74)$$

$$= - \sum_j \sum_{\mathbf{k}} i\mathbf{k} \cdot \mathbf{V}_j \hat{S}(\mathbf{k}) G(\mathbf{k}) \hat{\rho}(\mathbf{k}) \exp(i\mathbf{k} \cdot \mathbf{X}_j). \quad (75)$$

One can easily check the identity

$$\frac{d}{dt} (U_E + U_K) = 0, \quad (76)$$

from equations (70) and (75). Furthermore, we show that with the standard leapfrog time integrator and in the sole presence of electric forces, the energy conservation error has second order convergence for both periodic and free space boundary conditions. We begin by noticing that Eq. (21) and Eq. (22) take the following form in Fourier space:

$$\hat{\rho}^n(\mathbf{k}) = \hat{S}(\mathbf{k}) \left(\sum_j \exp(-i\mathbf{k} \cdot \mathbf{X}_j^n) \right), \quad \hat{\varphi}^n(\mathbf{k}) = G(\mathbf{k}) \hat{\rho}^n(\mathbf{k}), \quad (77)$$

where the superscript n denotes the quantities at time step n . We therefore immediately find the potential energy to be:

$$U_E^n = \frac{1}{2} \sum_{\mathbf{k}} G(\mathbf{k}) \left| \hat{S}(\mathbf{k}) \right|^2 \left| \sum_j \exp(-i\mathbf{k} \cdot \mathbf{X}_j^n) \right|^2. \quad (78)$$

Since the leapfrog scheme uses the particle velocities at half time steps, we write:

$$\mathbf{V}_s^{n+1/2} = \mathbf{V}_s^{n-1/2} + \Delta t \cdot \mathbf{a}_s^n, \quad (79)$$

$$= \mathbf{V}_s^{n-1/2} + \Delta t \sum_{\mathbf{k}} \left(-i\mathbf{k} G(\mathbf{k}) \left| \hat{S}(\mathbf{k}) \right|^2 \exp(i\mathbf{k} \cdot \mathbf{X}_s^n) \sum_j \exp(-i\mathbf{k} \cdot \mathbf{X}_j^n) \right), \quad (80)$$

where the expression for \mathbf{a}_s^n was introduced in equation (29). If we use the midpoint rule to define \mathbf{V}^n , then we have the following relationship:

$$\mathbf{V}_s^{n+1/2} = \mathbf{V}_s^n + \frac{\Delta t}{2} \sum_{\mathbf{k}} \left(-i\mathbf{k} G(\mathbf{k}) \left| \hat{S}(\mathbf{k}) \right|^2 \exp(i\mathbf{k} \cdot \mathbf{X}_s^n) \sum_j \exp(-i\mathbf{k} \cdot \mathbf{X}_j^n) \right). \quad (81)$$

We can then compute particle positions at time $n+1$ in terms of positions and velocities at time n :

$$\mathbf{X}_s^{n+1} = \mathbf{X}_s^n + \Delta t \mathbf{V}_s^{n+1/2}, \quad (82)$$

$$= \mathbf{X}_s^n + \Delta t \mathbf{V}_s^n + \frac{\Delta t^2}{2} \sum_{\mathbf{k}} \left(-i\mathbf{k} G(\mathbf{k}) \left| \hat{S}(\mathbf{k}) \right|^2 \exp(i\mathbf{k} \cdot \mathbf{X}_s^n) \sum_j \exp(-i\mathbf{k} \cdot \mathbf{X}_j^n) \right), \quad (83)$$

$$= \mathbf{X}_s^n + \Delta t \mathbf{V}_s^n + \mathcal{O}(\Delta t^2). \quad (84)$$

We can plug this expression into the equation (78) for the potential energy:

$$U_E^{n+1} = \frac{1}{2} \sum_{\mathbf{k}} G(\mathbf{k}) \left| \hat{S}(\mathbf{k}) \right|^2 \left| \sum_s \exp(-i\mathbf{k} \cdot (\mathbf{X}_s^n + \Delta t \mathbf{V}_s^n + \mathcal{O}(\Delta t^2))) \right|^2. \quad (85)$$

Now, for Δt small,

$$\exp(-i\mathbf{k} \cdot (\mathbf{X}_s^n + \Delta t \mathbf{V}_s^n + \mathcal{O}(\Delta t^2))) = \exp(-i\mathbf{k} \cdot \mathbf{X}_s^n) (1 - i\Delta t \mathbf{k} \cdot \mathbf{V}_s^n + \mathcal{O}(\Delta t^2)). \quad (86)$$

Substituting this equality in equation (85) gives

$$U_E^{n+1} = \frac{1}{2} \sum_{\mathbf{k}} G(\mathbf{k}) |\hat{S}(\mathbf{k})|^2 \left| \sum_s \exp(-i\mathbf{k} \cdot \mathbf{X}_s^n) (1 - i\Delta t \mathbf{k} \cdot \mathbf{V}_s^n + \mathcal{O}(\Delta t^2)) \right|^2, \quad (87)$$

$$= U_E^n - \Delta t \sum_{\mathbf{k}} G(\mathbf{k}) |\hat{S}(\mathbf{k})|^2 \left(\sum_s i\mathbf{k} \cdot \mathbf{V}_s^n \exp(-i\mathbf{k} \cdot \mathbf{X}_s^n) \right) \left(\sum_j \exp(-i\mathbf{k} \cdot \mathbf{X}_j^n) \right) + \mathcal{O}(\Delta t^2), \quad (88)$$

$$= U_E^n - \sum_s \mathbf{V}_s^n \cdot \Delta t \sum_{\mathbf{k}} \left(-i\mathbf{k} G(\mathbf{k}) |\hat{S}(\mathbf{k})|^2 \sum_j \exp(i\mathbf{k} \cdot (\mathbf{X}_s^n - \mathbf{X}_j^n)) \right) + \mathcal{O}(\Delta t^2). \quad (89)$$

The velocities at the next time step, \mathbf{V}_s^{n+1} , are calculated as:

$$\mathbf{V}_s^{n+1} = \frac{1}{2} (\mathbf{V}_s^{n+1/2} + \mathbf{V}_s^{n+3/2}), \quad (90)$$

$$= \frac{\mathbf{V}_s^{n+1/2}}{2} + \frac{1}{2} (\mathbf{V}_s^{n+1/2} + \Delta t \cdot \mathbf{a}_s^{n+1}), \quad (91)$$

$$= \mathbf{V}_s^{n+1/2} + \frac{\Delta t}{2} \mathbf{a}_s^{n+1}, \quad (92)$$

$$= \mathbf{V}_s^{n+1/2} + \frac{\Delta t}{2} \sum_{\mathbf{k}} \left(-i\mathbf{k} G(\mathbf{k}) |\hat{S}(\mathbf{k})|^2 \exp(i\mathbf{k} \cdot \mathbf{X}_s^{n+1}) \sum_j \exp(i\mathbf{k} \cdot -\mathbf{X}_j^{n+1}) \right), \quad (93)$$

$$= \mathbf{V}_s^{n+1/2} + \frac{\Delta t}{2} \sum_{\mathbf{k}} \left(-i\mathbf{k} G(\mathbf{k}) |\hat{S}(\mathbf{k})|^2 \sum_j \exp(i\mathbf{k} \cdot (\mathbf{X}_s^n - \mathbf{X}_j^n) + i\Delta t \mathbf{k} \cdot (\mathbf{V}_s^{n+1/2} - \mathbf{V}_j^{n+1/2})) \right), \quad (94)$$

$$= \mathbf{V}_s^{n+1/2} + \frac{\Delta t}{2} \sum_{\mathbf{k}} \left(-i\mathbf{k} G(\mathbf{k}) |\hat{S}(\mathbf{k})|^2 \sum_j \exp(i\mathbf{k} \cdot (\mathbf{X}_s^n - \mathbf{X}_j^n)) \right) + \mathcal{O}(\Delta t^2), \quad (95)$$

$$= \mathbf{V}_s^n + \Delta t \sum_{\mathbf{k}} \left(-i\mathbf{k} G(\mathbf{k}) |\hat{S}(\mathbf{k})|^2 \sum_j \exp(i\mathbf{k} \cdot (\mathbf{X}_s^n - \mathbf{X}_j^n)) \right) + \mathcal{O}(\Delta t^2). \quad (96)$$

Note that we again used the midpoint rule for the last equality. This equality gives us the total kinetic energy at time $n+1$ to second order in Δt :

$$U_K^{n+1} = \frac{1}{2} \sum_s \|\mathbf{V}_s^{n+1}\|^2, \quad (97)$$

$$= U_K^n + \Delta t \sum_s \mathbf{V}_s^n \cdot \sum_{\mathbf{k}} \left(-i\mathbf{k} G(\mathbf{k}) |\hat{S}(\mathbf{k})|^2 \sum_j \exp(i\mathbf{k} \cdot (\mathbf{X}_s^n - \mathbf{X}_j^n)) \right) + \mathcal{O}(\Delta t^2). \quad (98)$$

Adding equations (98) and (89) we conclude that we have second order convergence for energy conservation with respect to Δt :

$$U_K^{n+1} + U_E^{n+1} = U_K^n + U_E^n + \mathcal{O}(\Delta t^2). \quad (99)$$

This result, showing second order convergence for energy conservation independently of the number of particles and modes, is not shared by classical particle-in-cell scheme. In a standard explicit particle-in-cell scheme, the grid heating effect and finite grid instability will cause energy to increase without bound over time if the spatial grid does not resolve the Debye length. We note that for our numerical tests, we will consider physical situations that involve other fields, such as an external magnetic field \mathbf{B}_0 or a homogeneous potential field φ^H derived from Dirichlet boundary conditions. By choosing suitable numerical algorithms, we can preserve the second order convergence of energy conservation in these situations. In the presence of a magnetic field \mathbf{B}_0 , one may choose the Boris algorithm to update the particle positions and velocities, which achieves exact energy conservation if $\mathbf{E} = \mathbf{0}$ and at least second order otherwise [48]. In the presence of a homogeneous potential field φ^H to satisfy Dirichlet boundary conditions, one can use the boundary integral method mentioned in Section 3.3 to compute the full electric field. The mathematical proof of energy conservation for these situations is left as future work; the numerical convergence of energy conservation in these cases will however be confirmed in the next section.

3.5.2. Momentum conservation

In this section, we analyze the momentum conservation of the free space particle-in-Fourier scheme. Without loss of generality we assume the mass of a particle is $m = 1$. We define the total momentum of the system at time $t^{n-1/2}$ as

$$\mathbf{p}^{n-1/2} = \sum_{i=1}^{N_p} \mathbf{V}_i^{n-1/2}. \quad (100)$$

By equation (80), we are able to determine the momentum at the next time step:

$$\mathbf{p}^{n+1/2} = \mathbf{p}^{n-1/2} + \Delta t \sum_{j=1}^{N_p} \exp(-i\mathbf{k} \cdot \mathbf{X}_j^n) \left(\sum_{\mathbf{k}} \left(-i\mathbf{k}G(\mathbf{k})|\hat{S}(\mathbf{k})|^2 \sum_{i=1}^{N_p} \exp(i\mathbf{k} \cdot \mathbf{X}_i^n) \right) \right). \quad (101)$$

Therefore, the difference between the momentums at two adjacent time steps is

$$\mathbf{p}^{n+1/2} - \mathbf{p}^{n-1/2} = \Delta t \sum_{j=1}^{N_p} \exp(-i\mathbf{k} \cdot \mathbf{X}_j^n) \left(\sum_{\mathbf{k}} \left(-i\mathbf{k}G(\mathbf{k})|\hat{S}(\mathbf{k})|^2 \sum_{i=1}^{N_p} \exp(i\mathbf{k} \cdot \mathbf{X}_i^n) \right) \right), \quad (102)$$

$$= -i\Delta t \sum_{\mathbf{k}} \mathbf{k}G(\mathbf{k})|\hat{S}(\mathbf{k})|^2 \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \exp(i\mathbf{k} \cdot (\mathbf{X}_i^n - \mathbf{X}_j^n)). \quad (103)$$

Finally, we make note that $G(\mathbf{k}) = G(-\mathbf{k})$, $\hat{S}(\mathbf{k}) = \hat{S}(-\mathbf{k})$, and

$$B(\mathbf{k}) \equiv \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \exp(i\mathbf{k} \cdot (\mathbf{X}_i^n - \mathbf{X}_j^n)), \quad (104)$$

$$= \sum_{j=1}^{N_p} \sum_{i=1}^{N_p} \exp(i\mathbf{k} \cdot (\mathbf{X}_i^n - \mathbf{X}_j^n)), \quad (105)$$

$$= \sum_{j=1}^{N_p} \sum_{i=1}^{N_p} \exp(-i\mathbf{k} \cdot (\mathbf{X}_j^n - \mathbf{X}_i^n)), \quad (106)$$

$$= \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \exp(-i\mathbf{k} \cdot (\mathbf{X}_i^n - \mathbf{X}_j^n)), \quad (107)$$

$$= B(-\mathbf{k}), \quad (108)$$

are all even functions of \mathbf{k} , whereas $\mathbf{k} = -(-\mathbf{k})$ is an odd function of \mathbf{k} . We therefore conclude that the summation in equation (103) equals zero, and thus the total momentum of the system is conserved over time.

3.6. Comparison between PIC and PIF schemes

The computational cost of PIF schemes scales as $\mathcal{O}((|\log \varepsilon| + 1)^d N_p + N_m^d \log N_m^d)$, where ε is the tolerance used for the NUFFT. In contrast, the computational cost of a standard PIC scheme with cloud-in-cell shape function and an FFT-based field solver scales as $\mathcal{O}(2^d N_p + N_m^d \log N_m^d)$. Thus the standard PIC scheme has a computational complexity similar to the coarsest NUFFT PIF scheme with the tolerance of $\varepsilon = 10^{-1}$. In other words, for the same total number of particles and number of modes, PIF schemes are costlier than the standard PIC scheme. However, the PIF scheme has advantages in the following respects:

1. As mentioned in [2], we can use pre-computed shape functions of arbitrary order in PIF schemes without any increase in their computational cost, whereas for PIC schemes the computational cost increases with the order of the shape function.
2. We can choose the tolerance ε of the NUFFT in a PIF scheme depending on the desired levels of conservation in energy and charge for the problem of interest. In practice, even with the same number of particles and grid points/modes, PIF schemes can only be slightly costlier than the PIC schemes as shown in Section 4.2.1.
3. The mesh sizes in explicit PIC schemes have to be on the order of the Debye length in order to avoid grid heating. This typically requires a much higher number of grid points than the number of modes considered in PIF schemes. Moreover, in order to have the same number of particles per cell, the total number of particles has to be increased as well. As a result of these practical constraints, one ends up having a much higher number of particles and grid points in an explicit PIC scheme compared to the PIF scheme, which in turn leads to increased time to solution. As an example we have shown in Section 4.2.1 that for comparable accuracies in the quantities of interest, the PIF scheme leads to more than an order of magnitude speedup compared to the explicit PIC scheme we implemented for comparison.
4. PIF schemes are spectral particle methods, which belong to the general class of geometric, structure-preserving PIC schemes [1]. There are other geometric PIC schemes [17–20] which also offer excellent long time conservation and stability properties. Compared to them, PIF schemes retain the intuitive and ease of implementation nature of standard explicit PIC schemes. A more detailed comparison between PIF schemes and other geometric PIC schemes in terms of accuracy and performance is beyond the scope of current work and will be carried out elsewhere in the future.

In terms of parallelization strategies, usually domain decomposition is used for PIC schemes where both the particles and grid points are divided between the cores. PIF schemes are relatively more global in nature than PIC schemes. Hence, as mentioned in [2] they are more suited for particle decomposition parallelization strategy where only the particles are divided between cores and each core carries all the Fourier modes. However, this strategy has a bottleneck in both memory and computing costs when the simulation

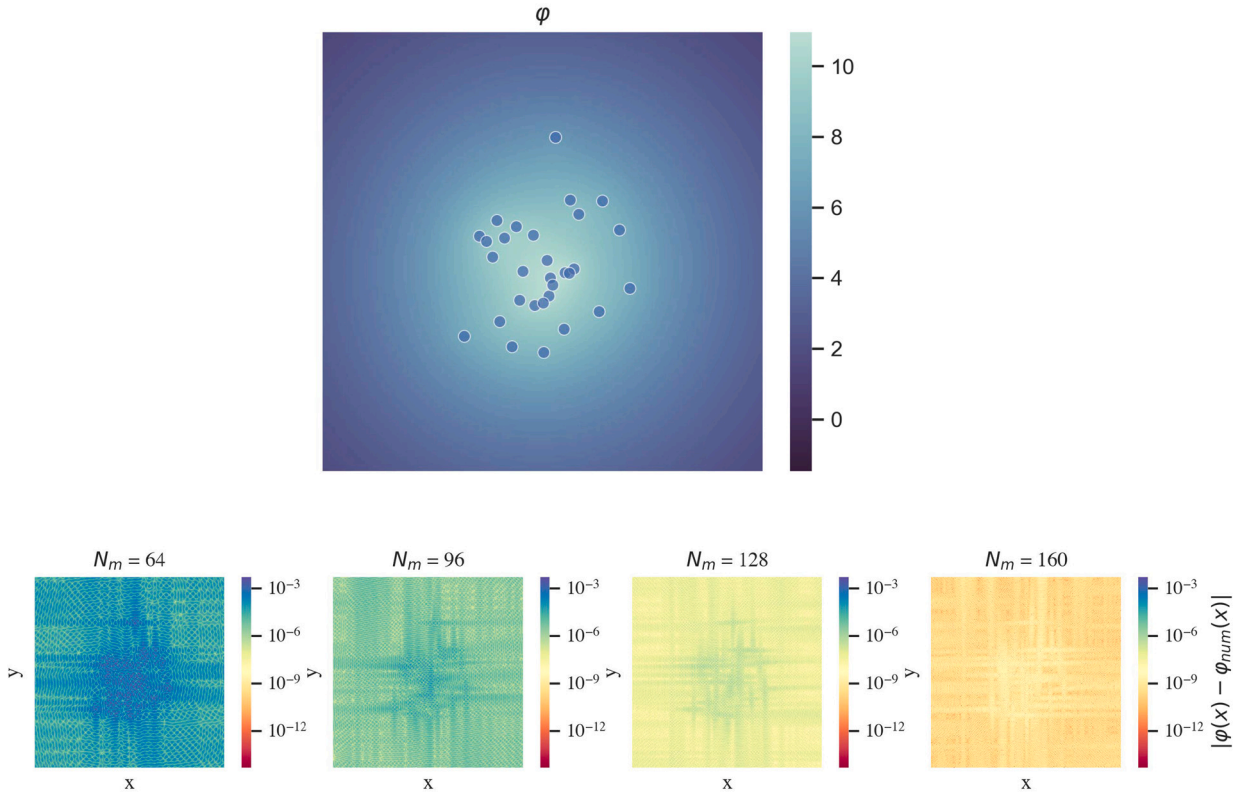


Fig. 2. Potential ϕ from the analytic solution given by equation (110) with particles represented as blue circles (top) and numerical error of our free space Poisson solver using different numbers of modes (bottom). We choose the radius of the Green's function to be $L = 1.75$, and the solver uses a $(4N_m)^2$ box.

requires a lot of Fourier modes (as they are duplicated and not decomposed). In order to alleviate this issue recently a space-time parallelization strategy has been proposed in [49] which shows promise for large scale 3D-3V PIF simulations on extreme-scale computing architectures.

4. Numerical tests

In this section, we first test the numerical accuracy of our gridless free space Poisson solver with the method of manufactured solutions. We then verify our free space PIF implementation in a two-dimensional beam dynamics problem which has been extensively studied analytically [50,51]. Finally, the accuracy of our PIF scheme for Dirichlet boundary conditions based on FSPIF and the boundary integral method is also examined. For NUFFT's needed for these examples, we rely on `f1NUFFT`, an efficient library presented in [52]. The tolerance of NUFFT's are chosen as $\epsilon = 10^{-14}$ in all the tests, except in sections 4.2.1 and 4.3 where we used a tolerance of $\epsilon = 10^{-4}$. The source codes for all our numerical results can be found at <https://github.com/Nigel-Shen/Free-Space-Particle-In-Fourier>.

4.1. Free space Poisson problem

In this test case, we place 30 random particles in the unit box, and solve the free space Poisson problem using our gridless solver. We choose the shape function to be a truncated Gaussian with standard deviation $\sigma = 1/100$ and truncation radius $R = 1/8$ (see Appendix A), so that ρ is almost C^∞ within the domain Ω . We choose the radius of the mollified Green's function to be $L = 1.75$, and let the extended box, which includes the zero padded regions, be $\tilde{\Omega} = [-2, 2] \times [-2, 2]$.

Let us denote the particle positions as \mathbf{X}_j . Both ρ and ϕ can be determined analytically:

$$\rho(\mathbf{x}) = q \sum_j S(\mathbf{X}_j - \mathbf{x}) = \frac{q}{2\pi\sigma^2} \sum_j \exp\left(-\frac{||\mathbf{X}_j - \mathbf{x}||^2}{2\sigma^2}\right), \quad (109)$$

$$\phi(\mathbf{x}) = \frac{q}{4\pi} \sum_j -\left(\text{Ei}\left(\frac{||\mathbf{X}_j - \mathbf{x}||^2}{2\sigma^2}\right) - \log(||\mathbf{X}_j - \mathbf{x}||^2)\right) \quad (110)$$

where Ei is the exponential integral function defined as

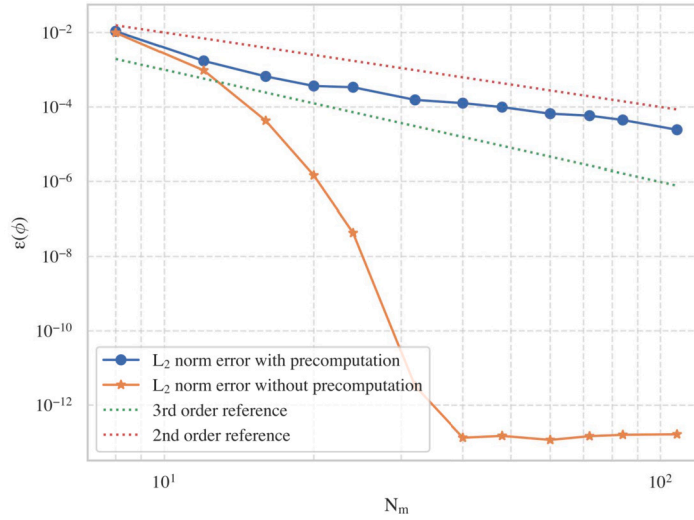


Fig. 3. Global absolute errors in L_2 norms versus the number of modes in each dimension N_m , calculated in an upscaled physical grid of size 512×512 using Fourier interpolation. Notice that without performing the precomputation step, the solver achieves spectral accuracy, whereas with the precomputation step, it has second order convergence.

$$\text{Ei}(x) = - \int_{-x}^{\infty} \frac{e^{-t}}{t} dt. \quad (111)$$

We vary the number of modes N_m and evaluate the numerical solution $\varphi(\mathbf{x})$ on a refined grid of resolution 128×128 using Fourier interpolation (*i.e.*, padding zeros in the spectral space before inverse FFT), and compare it with the analytical solution.

We begin by plotting the analytical solution φ and compare it with the numerical solutions obtained when $N_m = 16, 24, 32$ and 40 in Fig. 2. We then plot the absolute error $\|\varepsilon\|_2 = \|\varphi - \varphi_{\text{num}}\|_2$ as a function of the number of modes N_m on a log-log scale, shown in Fig. 3. Spectral accuracy can be clearly observed as the L_2 norm decays faster than any polynomial order in the case without precomputation step. If we do a precomputation step before solving the system, the solver now is second order if evaluated at arbitrary positions inside the domain, for the reason explained in Section 3.2.

4.2. Infinitely long beam confined by a guiding magnetic field, in free space

In this example, we consider an infinitely long charged particle beam in the z direction, in free space, and confined by a strong magnetic field aligned with the z axis. This situation has been considered as an insightful model for understanding the space charge dynamics in high intensity cyclotrons, and in particular the process of beam axisymmetrization [53,50,51]. We let our domain be $\Omega = [-1/2, 1/2] \times [-1/2, 1/2]$. The initial extended box $\tilde{\Omega}$ including the zero-padded regions is set to be $[-2, 2] \times [-2, 2]$. A precomputation step is optionally performed, which then enables us to only consider $[-1, 1] \times [-1, 1]$ as the computational box including the zero padded regions. We normalize the spatial length unit by the Debye length $\lambda_D = \sqrt{\varepsilon_0 T / q \rho}$, and the temporal unit by the inverse plasma frequency $\omega_p^{-1} = \sqrt{\varepsilon_0 m / q \rho}$, where the particle mass m and charge q are each normalized to one. In order to confine our charged particles, we apply an external constant magnetic field $\mathbf{B}_0 = 300\mathbf{z}$. The magnitude of \mathbf{B}_0 has been chosen in such a way that the force due to the magnetic field is approximately 50 to 100 times larger in magnitude compared to the force due to the electric field. The initial particle distribution is an elongated normal distribution in configuration space, and Maxwellian in velocity space:

$$f(x, y, v_x, v_y, 0) = \frac{n(x, y)}{2\pi} \exp\left(-\frac{v_x^2 + v_y^2}{2}\right) \quad (112)$$

where

$$n(x, y) = \frac{1}{\pi \sigma_x \sigma_y} \exp\left(-\frac{x^2}{\sigma_x^2}\right) \exp\left(-\frac{y^2}{\sigma_y^2}\right) \text{ with } \sigma_x = \frac{1}{30}, \sigma_y = \frac{1}{10}. \quad (113)$$

We fix the number of particles to be 40,000, and use a second-order radially symmetric b-spline function for the shape function (see Appendix A). The radius of the shape function is set to be $R = 1/N_m$ where N_m is the Fourier resolution. We first perform the simulation with $N_m = 128$ in FSPIF and compare it with the result obtained using the standard PIC method. For FSPIC, the charges are spread onto the grid points using the second order tensor-product b-spline function, the standard Vico solver [3] is applied for spectrally accurate field solutions at the grid points, and the force is interpolated back to the particles using the same shape function as the one for the spreading. As shown in Fig. 4, beam spiraling and axisymmetrization are observed in both our FSPIC and FSPIF

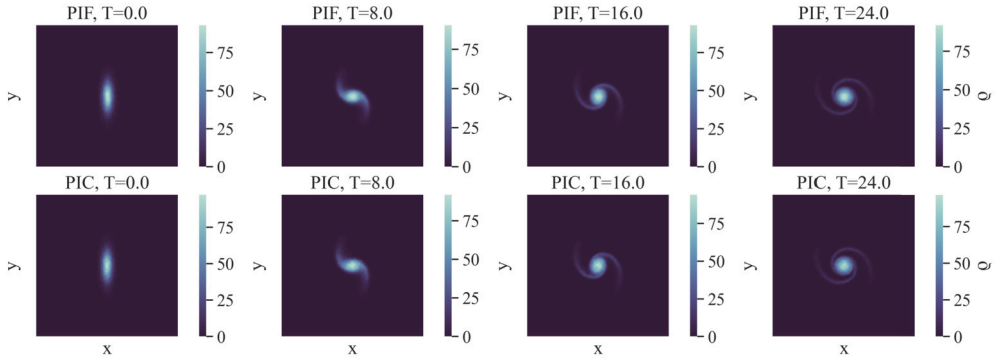


Fig. 4. Charge density ρ at different times of an infinitely long non-neutral beam confined by a strong constant and uniform magnetic field, simulated using 40,000 particles and $N_m = N_g = 128$ using free space PIC and PIF methods. For the FSPIF scheme, the precomputation step is not performed.

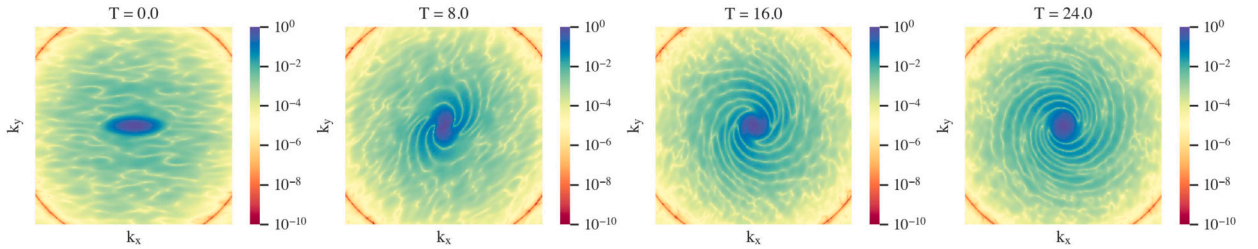


Fig. 5. Absolute values of Fourier modes when $N_m = 128$ for the same nonneutral beam simulation as in Fig. 4. These figures indicate that most high-frequency modes are not very significant, and there is no obvious mode-coupling or aliasing effects.

numerical simulations, as expected, in agreement with the results in [50,51]. We use negatively charged particles in all our simulations and hence Fig. 4 shows the negative of charge density. The same is true for all the rest of the figures where we show charge density from the test cases. In order to demonstrate that the number of modes in Fig. 4 is sufficient to resolve the spatial scales of this problem, we present in Fig. 5 the absolute values of the Fourier modes for $N_m = 128$, and it is clear that this Fourier resolution is sufficient to capture the part of the Fourier spectrum that has a significant impact on the evolution of the beam.

We measure the total energy E over the full duration of the simulation for different choices of Δt . Fig. 6 shows the energy fluctuation over time for $\Delta t = 0.001, 0.0005, 0.00025$. Notice that the fluctuation is smooth and has a period equal to the cyclotron period, which suggests that the error is mainly caused by the strong magnetic field. Since the Boris algorithm is energy-conserving if $\mathbf{E} = \mathbf{0}$ and at least second order convergent in the presence of an electric field, the global energy error has a second order of convergence, i.e., $\epsilon(E) \sim \Delta t^2$, as can be seen in Fig. 7. Moreover, from Fig. 7 we observe that even with the precomputation step, the energy still converges quadratically, and in fact the error is smaller than in the case without precomputation by a constant factor.

We also compare FSPIF and FSPIC with the same number of modes for FSPIF as the number of grid points for FSPIC in each dimension ($N_m = N_g = 32$), the same number of particles $N_p = 40,000$, and the same time step size $\Delta t = 0.0005$, and show the numerical results in Fig. 8. Clearly, the use of a grid in FSPIC causes the energy to increase over time, whereas it is very stable with FSPIF. This will be more pronounced for certain long time integration simulations as we show in the next section.

4.2.1. Grid heating in a long time integration simulation

In this section, we consider a long time integration simulation, and compare FSPIC and FSPIF schemes in terms of the accuracies of the physical quantities of interest, as well as time to solution. We consider a setting slightly different from the one in Section 4.2, with the domain $\Omega = [-1, 1] \times [-1, 1]$, external magnetic field $\mathbf{B}_0 = 40\mathbf{z}$, total charge $Q = -20$ and the standard deviations in equation (113) scaled corresponding to the domain as $\sigma_x = 1/15$ and $\sigma_y = 1/5$. We compare an FSPIF simulation with $N_m = 64$ modes with two FSPIC simulations, one with $N_m = 64$ grid points in each direction, and one with $N_m = 512$ grid points in each direction. We take $\Delta t = 0.01$, total number of particles 40,000 and integrate until final time $T = 1000$ for FSPIC and FSPIF with $N_m = 64$, whereas for FSPIC with $N_m = 512$ grid points in each direction, we only integrate until $T = 160$ to reduce computing time requirements. FSPIF with precomputing is used for the simulations.

We compare the energy conservation between FSPIC and FSPIF schemes in Fig. 9. While the energy error is very stable for FSPIF with $N_m = 64$ in Fig. 9(b), it is two orders of magnitude larger and increases continuously for FSPIC scheme with $N_m = 64$ in Fig. 9(a) due to aliasing and grid heating. For FSPIC with $N_m = 512$ the slope of the error increase is much smaller in Fig. 9(c) compared to Fig. 9(a) but the values are still an order of magnitude higher than for the FSPIF scheme in Fig. 9(b).

The grid heating and lack of energy conservation in FSPIC scheme are reflected physically in the charge density and phase-space plots in Figs. 10 and 11. We can clearly see that the beam radius for the FSPIC scheme with $N_m = 64$ is larger in Fig. 10(a) as compared to the beam radius in Fig. 10(b) for the FSPIF scheme with $N_m = 64$. The beam radius for FSPIC with $N_m = 512$ in Fig. 10(c)

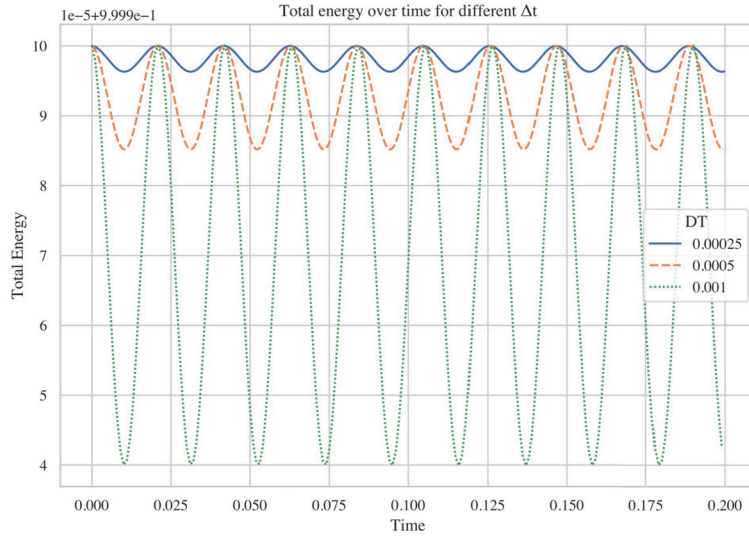


Fig. 6. Total energy versus time for the full duration of the simulation and for different time step sizes $\Delta t = 0.001, 0.0005$, and 0.00025 . The periodic variation of the energy has a period equal to the cyclotron period, indicating that the energy fluctuation is mostly caused by the strong magnetic field.

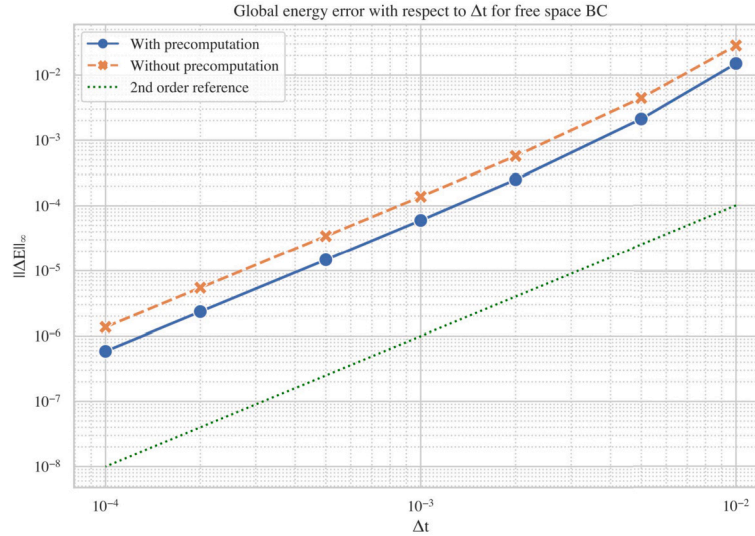


Fig. 7. Convergence of the global relative energy error (measured by the L_∞ norm) for different time step sizes Δt for our free space particle-in-Fourier simulation using the Boris algorithm. The Fourier resolution is set to $N_m = 32$. Second order convergence is observed whether or not we perform a precomputation step. Interestingly, the precomputation step reduces the energy error by a constant compared to the errors without precomputation.

is comparable to the FSPIF scheme, however, since the total number of particles is not increased, it has much more statistical noise. We observe similar results in the x -phase space plot in Fig. 11(c) where the x -emittance is comparable for FSPIF with $N_m = 64$ and FSPIC with $N_m = 512$, whereas for FSPIC with $N_m = 64$ the semi-minor axis is twice larger. The effects are much more pronounced at $T = 1000$ (this corresponds to 10^5 time steps, which is typical for many production-level cyclotron simulations), where the semi-minor axis is now approximately three times larger in Fig. 11(d) and the corresponding beam radius is also larger as shown in Fig. 11(a).

Even though in the earlier tests we used a strict tolerance of 10^{-14} close to machine precision for the NUFFT in the FSPIF scheme, it can be made lenient depending on the time step size in the explicit time integrator so that the energy conservation is not affected. For the chosen time step size of $\Delta t = 0.01$ we found from numerical experiments that the NUFFT tolerance of 10^{-4} produces indistinguishable results from that of 10^{-14} . We note here that while the momentum conservation is unaffected by the tolerance of NUFFT, charge conservation is proportional to it and for this test a charge conservation error of $\mathcal{O}(10^{-4})$ did not change the results in any significant way.

Now the times to solution for the FSPIC and FSPIF schemes are compared in Table 1. The computations are performed in one node of CPU partition of JURECA DC supercomputer at the Jülich Supercomputing Centre. Each node of JURECA DC has 128 cores of AMD EPYC 7742, 2.25 GHz processors with 512 GB RAM. The computations are neither parallelized nor optimized explicitly

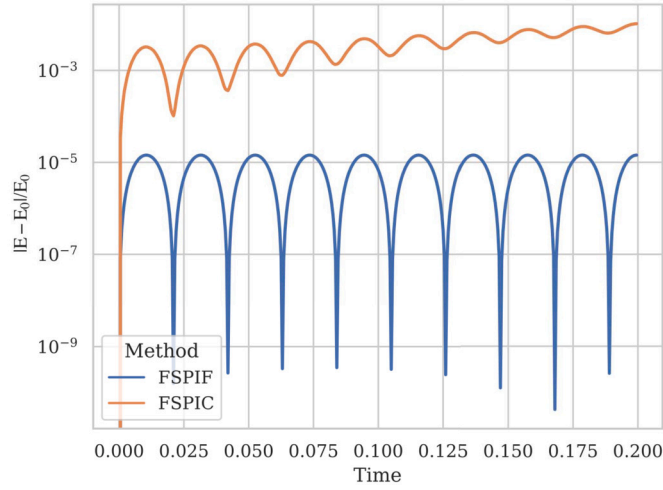


Fig. 8. Comparison of the total energy fluctuation as a function of time between our PIC and PIF implementations. Our PIC scheme uses the Vico solver to compute the free space Poisson electrostatic potential on the collocated grid. The time step size is $\Delta t = 0.0005$, and the number of modes in PIF and grid points in PIC is set to be $N_g = N_m = 32$.

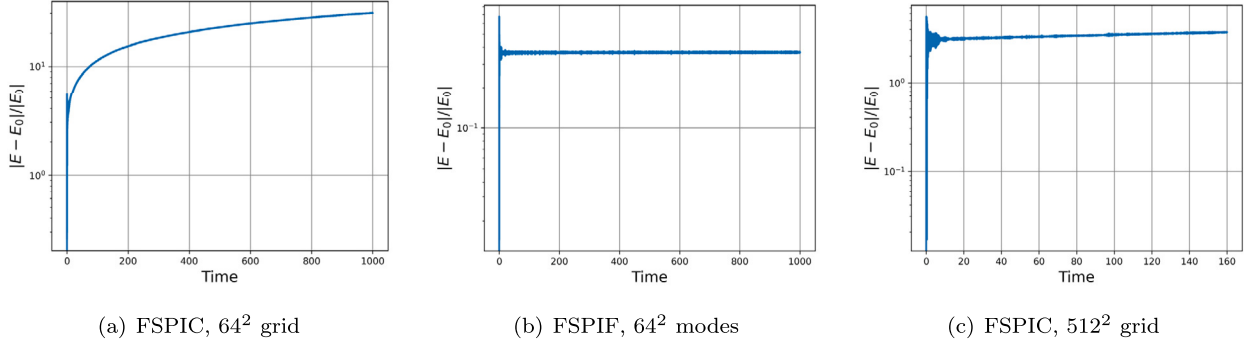


Fig. 9. Comparison of the total energy error as a function of time between FSPIC and FSPIF schemes for the simulations presented in Section 4.2.1.

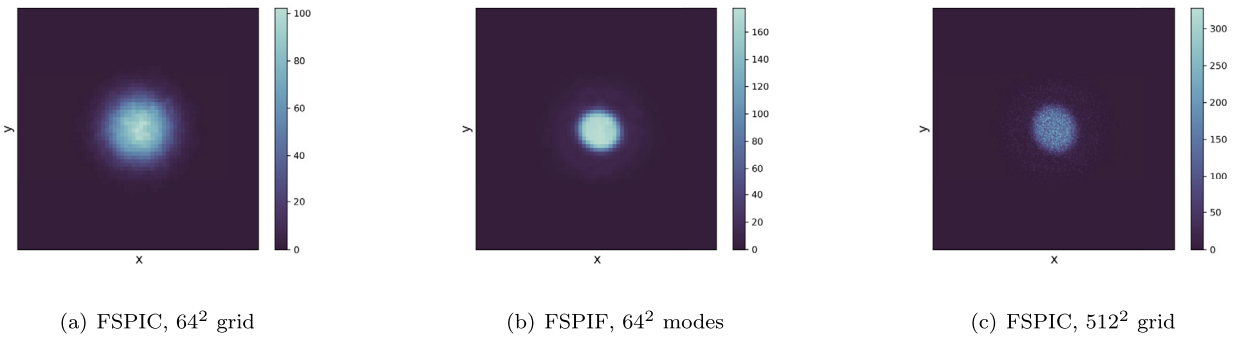


Fig. 10. Charge density comparison at $T = 160$ between FSPIC and FSPIF schemes for the simulations presented in Section 4.2.1.

and the default settings of `finUFFT` as well as other libraries such as `numpy` and `scipy` are used. While the FSPIF scheme with $N_m = 64$ and NUFFT tolerance of 10^{-4} is slightly more expensive than the FSPIC scheme with $N_m = 64$ (approximately 1.1 times), it is approximately 12 times faster than the FSPIC scheme with $N_m = 512$ which is needed for comparable accuracies in physical quantities of interest such as beam radius and emittance. Thus even in 2D-2V, FSPIF gives significant speedup compared to the FSPIC scheme with the fine resolutions necessary to avoid grid heating. We expect much larger benefits in 3D-3V simulations due to the increased number of grid points and particles.

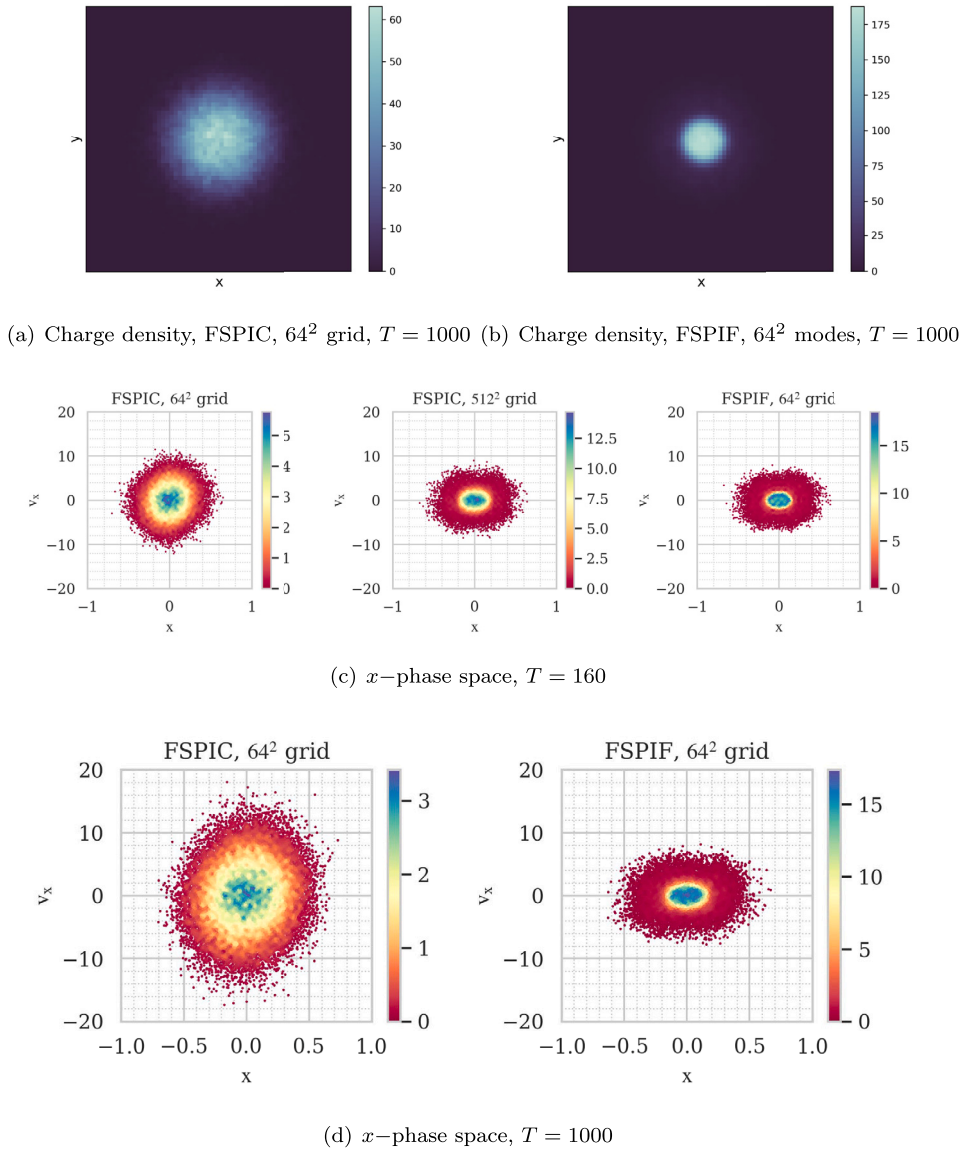


Fig. 11. Phase space and charge density comparisons at $T = 160$ and $T = 1000$ for FSPIC and FSPIF schemes for the simulations presented in Section 4.2.1.

Table 1

Time to solution for FSPIC and FSPIF schemes with 40,000 particles, $\Delta t = 0.01$ integrated up to $T = 160$. FSPIF scheme with 64^2 modes and FSPIC scheme with 512^2 grid points have comparable accuracies in quantities of interest however FSPIF is approximately 12 times faster than FSPIC.

Scheme	No. of grid points/modes	CPU time (s)
FSPIC	64^2	272.6
FSPIF	64^2	298.6
FSPIC	512^2	3557.5

4.3. Expanding plasma in free space

We consider an expanding plasma in free space in this section, and verify the momentum conservation property of the PIF scheme numerically. The parameters for this test case are as follows: domain $\Omega = [-5, 5] \times [-5, 5]$, no external magnetic field, total charge $Q = -1$ and the standard deviations in equation (113) scaled corresponding to the domain as $\sigma_x = 1/3$ and $\sigma_y = 1$. We consider a larger domain and a smaller total charge than the previous section so that the expanding plasma does not leave the domain. Similar

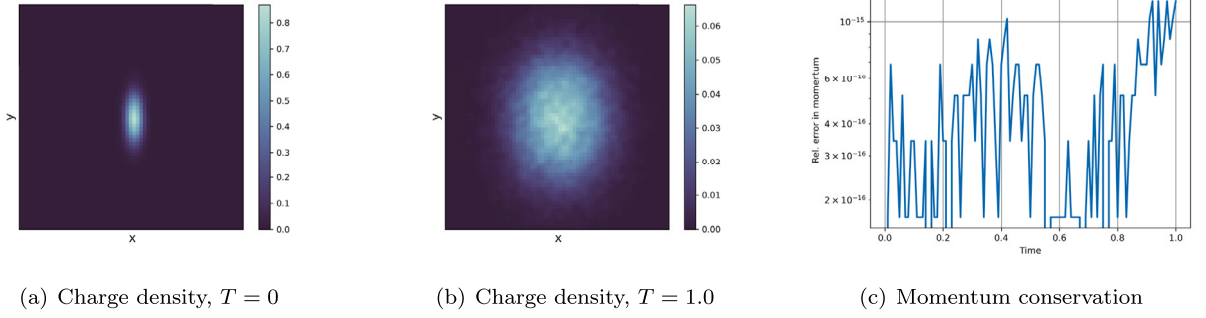


Fig. 12. Charge density at initial time (left), $T = 1.0$ (center) and relative error in total momentum conservation (right) for an expanding plasma in free space with the FSPIF scheme.

to the previous section, we take $\Delta t = 0.01$, total number of particles 40,000, $N_m = 64$ and integrate until $T = 1.0$. The tolerances for NUFFT are taken as 10^{-4} and precomputation is used for the FSPIF scheme.

In Fig. 12 we show the initial and final charge densities as well as the relative error in total momentum over time. We get momentum conservation close to double precision accuracy as seen from Fig. 12(c) which confirms our theoretical result in Section 3.5.2.

4.4. Infinitely long beam confined by a guiding magnetic field and a perfectly conducting cylinder

In this section, we consider the same infinitely long beam confined by a guiding magnetic field aligned with it as in Section 4.2, except that the beam is now also surrounded by a perfectly conducting cylinder. The examples we study serve to verify our implementation of the Dirichlet boundary conditions for our PIF solver, which follows the scheme discussed in Section 3.3. Physically, they are motivated by experiments by Noah Hurst and collaborators [46].

We begin by first checking the accuracy of our boundary integral solver using the method of manufactured solutions. We choose our analytic solution to be a harmonic function

$$u(x, y) = x^2 - y^2 + 2. \quad (114)$$

Let the coordinate of a boundary point be described by a single variable θ where $\sin(\theta) = x$ and $\cos(\theta) = y$, then the boundary data for our Laplace equation is:

$$f(\theta) = \sin^2(\theta) - \cos^2(\theta) + 2. \quad (115)$$

We vary the number of boundary points and use the trapezoidal rule to evaluate equation (40) on a 256×256 equi-spaced grid that covers the entire domain $D_1(\mathbf{0})$. Due to the singular integrand, the numerical value u_{num} does not converge in the vicinity of the boundary. This behavior is expected, and advanced quadrature schemes have been developed to address this issue [44]. However, the implementation of such schemes is not required if the beam remains sufficiently far from the conductor during its evolution, and is beyond the scope of this work. We therefore only measure the error $\epsilon(\mathbf{x}) = |u_{\text{num}} - u|(\mathbf{x})$ within a slightly shrunk domain $D_{0.9}(\mathbf{0})$, and plot them in Fig. 13. Overall, the solver provides spectral accuracy inside $D_{0.9}(\mathbf{0})$ away from the boundary, which matches the spectral convergence of the trapezoidal rule for periodic analytic functions [54,43].

Now we combine our boundary integral solver with the free space PIF algorithm as described in Section 3.3, and test its numerical properties. We consider the same initial charge distribution as in Section 4.2. In order to keep our computation in the unit box, in this test we use $D_{0.5}(\mathbf{0})$ as our domain, instead of the unit disk as in Section 3.3. When solving for the fields and accelerations, we add the harmonic function from the boundary integral method in order to satisfy Dirichlet boundary conditions at the boundary, $C_{0.5}(\mathbf{0})$. We perform tests for two Dirichlet problems, described as

$$f_1(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in C_{0.5}(\mathbf{0}), \quad (116)$$

$$f_2(\mathbf{x}) = y \quad \forall \mathbf{x} \in C_{0.5}(\mathbf{0}). \quad (117)$$

The evolution of the system under these two different Dirichlet boundary conditions over time is plotted in Figs. 14 and 15 with Fourier resolution $N_m = 128$, where the blue scatter dots are the 128 boundary points where we evaluate the boundary condition. For the first Dirichlet BC problem $f = f_1$, we expect the solution to be similar to the free-space solution; for the second Dirichlet BC problem $f = f_2$, we expect the beam to receive an extra force along the y axis, so the non-neutral beam will drift upward in the y direction.

Our simulations, shown in Figs. 14 and 15, confirm our intuition. The global relative energy error measured with the L_∞ norm is plotted as a function of the time step size Δt in Fig. 16. As in the free space case, the energy error converges in second order with respect to Δt .

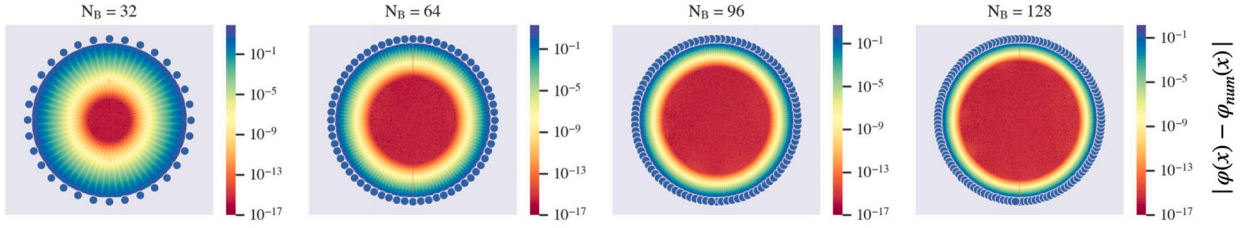


Fig. 13. Convergence of the boundary integral method as a function of the number of boundary points N_B in the domain $D_1(0)$. The method is spectrally accurate inside the domain except for the regions in the vicinity of the boundary.

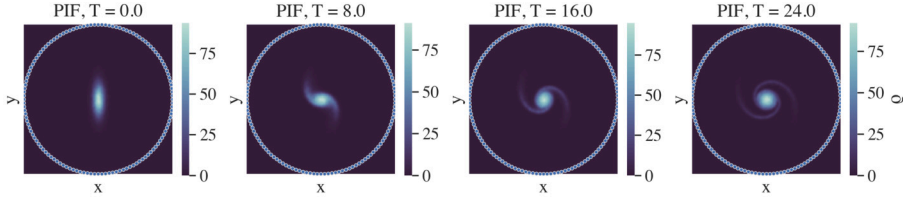


Fig. 14. Charge density ρ at different times of an infinitely long non-neutral beam confined by a strong constant and uniform magnetic field and a grounded cylinder. We choose 128 boundary points to evaluate ϕ^H and E^H using the method described in Section 3.3. The number of modes is $N_m = 128$, and the number of particles is 40,000. The solution is similar to the vacuum solution in Fig. 4, as expected.

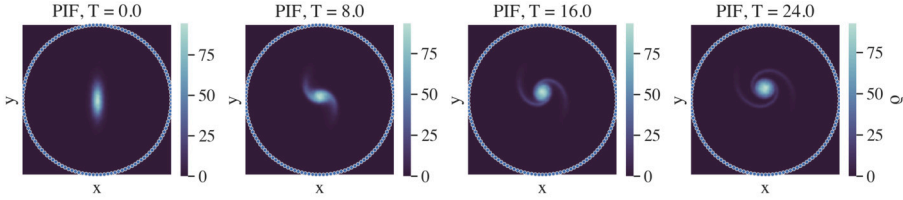


Fig. 15. Charge density ρ at different times of an infinitely long non-neutral beam confined by a strong constant and uniform magnetic field and a cylinder with biasing potential $f_2(x) = y$. We choose 128 boundary points to evaluate ϕ^H and E^H using the method described in Section 3.3. We choose the number of modes to be $N_m = 128$ and the number of particles to be 40,000. The beam drifts upwards in the y direction over time, as expected.

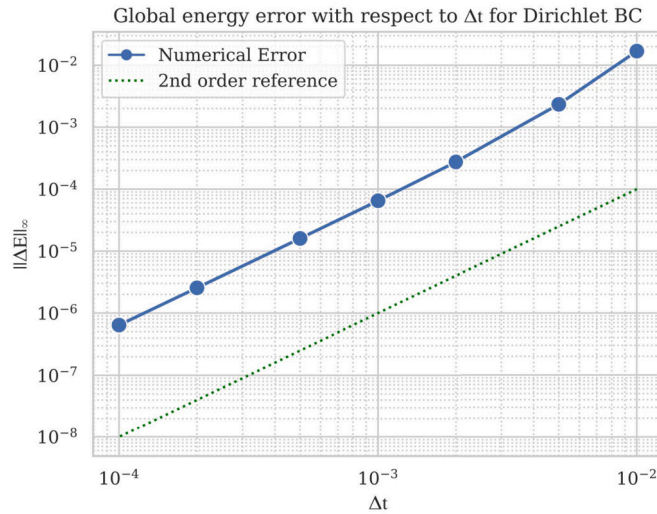


Fig. 16. Convergence of the global relative energy error for different time step sizes Δt for our particle-in-Fourier simulation with Dirichlet boundary conditions $f_1(x)$, corresponding to a grounded, perfectly conducting cylinder. Here, the time integrator is the Boris algorithm. The Fourier resolution is set to $N_m = 32$. As for the free space version of our scheme, second order convergence is observed.

5. Conclusion

We present a novel Particle-In-Fourier (PIF) scheme which can be applied to simulations with free space boundary conditions while maintaining semi-discrete energy conservation. We show that, by using a mollified Green's function kernel, we can obtain highly accurate solutions to the free space Poisson equation, even for non-equispaced data. Combining this Poisson solver with the PIF scheme allows us to simulate the evolution of plasmas described by the Vlasov equation with free space boundary conditions. We prove the energy and momentum conservation of the scheme in semi-discrete and fully discrete settings and validate it with numerical results.

Our approach can be extended to general Dirichlet boundary conditions on general domains, via standard potential theory. As an illustration of this, we presented an algorithm combining the free space PIF scheme with the boundary integral method on a unit disk for general Dirichlet boundary conditions. We show similar accuracy and energy conservation properties as those of our free space PIF scheme. Numerical implementation and validation of three-dimensional examples in arbitrary geometries are left as future work.

The high accuracy of the field solver and conservation of energy observed in our numerical tests show potential and advantages for long-time three-dimensional plasma simulations based on particle codes. It is however important to remember that in most cases, the error in a particle code is dominated by the noise of particle sampling rather than numerical error due to the field solver. Therefore, in order to further increase the accuracy and performance of our scheme, noise reduction strategies are of particular interest. Noise reduction is typically achieved by running simulations with a larger number of computational particles, taking advantage of scalable massively parallel computing. The design of efficient parallelization strategies for our PIF schemes therefore represents a natural direction for future work. In a complementary way, sparse grids have been used to enhance the performance of PIC by drastically reducing particle sampling error for a given number of computational particles [55,56]. The development of counterparts to sparse grid PIC schemes in the PIF setting appears to be another promising research direction.

CRediT authorship contribution statement

Changxiao Nigel Shen: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Antoine Cerfon:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Sriramkrishnan Muralikrishnan:** Writing – review & editing, Supervision, Project administration, Methodology, Software, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

C.N.S. was partially supported by National Science Foundation (award DMS-1646339) via the Applied Math Summer Undergraduate Research Experience at NYU Courant. A.J.C. was partially supported by the United States National Science Foundation under Grant No. PHY-1820852. We would like to acknowledge the computing time in the JURECA DC supercomputer provided under the project CSTMA.

Appendix A. Radially symmetric shape functions

In this work we use radially symmetric shape functions in our PIF code to conduct numerical experiments. These shape functions preserve exact rotational invariance as compared to the tensor-product shape functions used in PIC. The zeroth order radially symmetric b-spline function in 2D can be defined as

$$S_0(\mathbf{x}) = \begin{cases} \frac{4}{\pi} & \text{if } |\mathbf{x}| < 1/2, \mathbf{x} \in \mathbb{R}^2 \\ 0 & \text{otherwise.} \end{cases} \quad (118)$$

The l -th order radially symmetric b-spline function is defined as

$$S_l(\mathbf{x}) = \int_{\mathbb{R}^2} S_0(\mathbf{x} - \mathbf{x}') S_{l-1}(\mathbf{x}') d\mathbf{x}'. \quad (119)$$

The Fourier modes of the l -th order radially symmetric b-spline function in 2D is calculated as:

$$\hat{S}_l(\mathbf{k}) = \left(\frac{J_1(k)}{k} \right)^l \quad (120)$$

where $k = |\mathbf{k}|$.

In addition, in the free space Poisson solver test (Section 4.1), we use a spatially truncated Gaussian function as our shape function:

$$S^R(\mathbf{x}; \sigma) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{||\mathbf{x}||^2}{2\sigma^2}\right) & \text{if } ||\mathbf{x}|| < R \\ 0 & \text{otherwise.} \end{cases} \quad (121)$$

The Fourier transform of the above shape function is:

$$\hat{S}^R(\mathbf{k}; \sigma) = \exp\left(-\frac{\sigma^2 k^2}{2}\right) \Re\left(\text{erf}\left(\frac{R}{\sqrt{2}\sigma} + \frac{i\mathbf{k}\sigma}{\sqrt{2}}\right)\right). \quad (122)$$

References

- [1] E. Evstatiev, B. Shadwick, Variational formulation of particle algorithms for kinetic plasma simulations, *J. Comput. Phys.* 245 (2013) 376–398.
- [2] M.S. Mitchell, M.T. Mićnikowski, G. Beylkin, S.E. Parker, Efficient Fourier basis particle simulation, *J. Comput. Phys.* 396 (2019) 837–847.
- [3] F. Vico, L. Greengard, M. Ferrando, Fast convolution with free-space Green's functions, *J. Comput. Phys.* 323 (2016) 191–203.
- [4] C.K. Birdsall, D. Fuss, Clouds-in-clouds, clouds-in-cells physics for many-body plasma simulation, *J. Comput. Phys.* 3 (4) (1969) 494–511.
- [5] O. Buneman, G. Kooyers, Computer simulation of the electron mixing mechanism in ion propulsion, *AIAA J.* 1 (11) (1963) 2525–2528.
- [6] J. Dawson, One-dimensional plasma model, *Phys. Fluids* 5 (4) (1962) 445–459.
- [7] A.B. Langdon, C.K. Birdsall, Theory of plasma simulation using finite-size particles, *Phys. Fluids* 13 (8) (1970) 2115–2122.
- [8] R. Morse, Multidimensional plasma simulation by the particle-in-cell method, Tech. rep., Los Alamos Scientific Lab., N. Mex., 1970.
- [9] C.K. Birdsall, A.B. Langdon, *Plasma Physics via Computer Simulation*, CRC Press, 2018.
- [10] A.B. Langdon, Effects of the spatial grid in simulation plasmas, *J. Comput. Phys.* 6 (2) (1970) 247–267.
- [11] H. Okuda, Nonphysical noises and instabilities in plasma simulation due to a spatial grid, *J. Comput. Phys.* 10 (3) (1972) 475–486.
- [12] F. Low, A Lagrangian formulation of the Boltzmann-Vlasov equation for plasmas, *Proc. R. Soc. Lond. Ser. A, Math. Phys. Sci.* 248 (1253) (1958) 282–287.
- [13] J.W. Eastwood, The virtual particle electromagnetic particle-mesh method, *Comput. Phys. Commun.* 64 (2) (1991) 252–266.
- [14] H.R. Lewis, Energy-conserving numerical approximations for Vlasov plasmas, *J. Comput. Phys.* 6 (1) (1970) 136–141.
- [15] G. Chen, L. Chacón, D. Barnes, An energy- and charge-conserving, implicit, electrostatic particle-in-cell algorithm, *J. Comput. Phys.* 230 (18) (2011) 7018–7036.
- [16] S. Markidis, G. Lapenta, The energy conserving particle-in-cell method, *J. Comput. Phys.* 230 (18) (2011) 7037–7052.
- [17] M. Campos Pinto, K. Kormann, E. Sonnendrücker, Variational framework for structure-preserving electromagnetic particle-in-cell methods, *J. Sci. Comput.* 91 (2) (2022) 46.
- [18] J. Squire, H. Qin, W.M. Tang, Geometric integration of the Vlasov-Maxwell system with a variational particle-in-cell scheme, *Phys. Plasmas* 19 (8) (2012).
- [19] M. Kraus, K. Kormann, P.J. Morrison, E. Sonnendrücker, GEMPIC: geometric electromagnetic particle-in-cell methods, *J. Plasma Phys.* 83 (4) (2017) 905830401.
- [20] Y. He, Y. Sun, H. Qin, J. Liu, Hamiltonian particle-in-cell methods for Vlasov-Maxwell equations, *Phys. Plasmas* 23 (9) (2016).
- [21] A. Dutt, V. Rokhlin, Fast Fourier transforms for nonequispaced data, *SIAM J. Sci. Comput.* 14 (6) (1993) 1368–1393.
- [22] G. Beylkin, On the fast Fourier transform of functions with singularities, *Appl. Comput. Harmon. Anal.* 2 (1995) 363–381.
- [23] W.H. Press, G.B. Rybicki, Fast algorithm for spectral analysis of unevenly sampled data, *Astrophys. J., Part 1* (ISSN 0004-637X) 338 (March 1, 1989) 277–280.
- [24] 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, *Comput. Phys. Commun.* 207 (2016) 123–135.
- [25] Z. Chen, C.S. Peskin, A Fourier spectral immersed boundary method with exact translation invariance, improved boundary resolution, and a divergence-free velocity field, *arXiv preprint*, arXiv:2302.08694, 2023.
- [26] H. Ji, W. Doughton, J. Jara-Almonte, A. Le, A. Stanier, J. Yoo, Magnetic reconnection in the era of exascale computing and multiscale experiments, *Nat. Rev. Phys.* 4 (4) (2022) 263–282.
- [27] H. Klion, R. Jambunathan, M.E. Rowan, E. Yang, D. Willcox, J.-L. Vay, R. Lehe, A. Myers, A. Huebl, W. Zhang, Particle-in-cell simulations of relativistic magnetic reconnection with advanced Maxwell solver algorithms, *Astrophys. J.* 952 (1) (2023) 8.
- [28] G. Le Bars, J.-P. Hogge, J. Loizu, S. Alberti, F. Romano, A. Cerfon, Self-consistent formation and steady-state characterization of trapped high-energy electron clouds in the presence of a neutral gas background, *Phys. Plasmas* 29 (8) (2022) 082105.
- [29] F. Taccogna, F. Cichocki, D. Eremin, G. Fubiani, L. Garrigues, Plasma propulsion modeling with particle-based algorithms, *J. Appl. Phys.* 134 (15) (2023) 150901.
- [30] S. Ku, C.S. Chang, R. Hager, R.M. Churchill, G.R. Tynan, I. Cziegler, M. Greenwald, J. Hughes, S.E. Parker, M.F. Adams, E. D'Azevedo, P. Worley, A fast low-to-high confinement mode bifurcation dynamics in the boundary-plasma gyrokinetic code XGC1, *Phys. Plasmas* 25 (5) (2018) 056107.
- [31] M.D.J. Cole, R. Hager, T. Moritaka, J. Dominski, R. Kleiber, S. Ku, S. Lazerson, J. Riemann, C.S. Chang, Verification of the global gyrokinetic stellarator code XGC-S for linear ion temperature gradient driven modes, *Phys. Plasmas* 26 (8) (2019) 082501.
- [32] J. Ameres, Stochastic and spectral particle methods for plasma physics, Ph.D. thesis, Technische Universität München, 2018.
- [33] J. Zou, E. Kim, A.J. Cerfon, FFT-based free space Poisson solvers: why Vico-Greengard-Ferrando should replace Hockney-Eastwood, *arXiv preprint*, arXiv:2103.08531, 2021.
- [34] R. Hockney, J. Eastwood, *Computer Simulation Using Particles*, Taylor & Francis, Inc., 1988.
- [35] A. Adelmann, P. Calvo, M. Frey, A. Gsell, U. Locans, C. Metzger-Kraus, N. Neveu, C. Rogers, S. Russell, S. Sheehy, et al., OPAL a versatile tool for charged particle accelerator simulations, *arXiv preprint*, arXiv:1905.06654, 2019.
- [36] J.T. Rasmussen, Particle methods in bluff body aerodynamics, Ph.D. thesis, Technical University of Denmark, 2011.
- [37] M.M. Hejlesen, G. Windckelms, J.H. Walther, Non-singular Green's functions for the unbounded Poisson equation in one, two and three dimensions, *Appl. Math. Lett.* 89 (2019) 28–34.
- [38] R.D. Budiardja, C.Y. Cardall, Parallel FFT-based Poisson solver for isolated three-dimensional systems, *Comput. Phys. Commun.* 182 (10) (2011) 2265–2275.
- [39] P. Chatelain, P. Koumoutsakos, A Fourier-based elliptic solver for vortical flows with periodic and unbounded directions, *J. Comput. Phys.* 229 (7) (2010) 2425–2431.
- [40] S. Mayani, V. Montanaro, A. Cerfon, M. Frey, S. Muralikrishnan, A. Adelmann, A massively parallel performance portable free-space spectral Poisson solver, *arXiv preprint*, arXiv:2405.02603, 2024.
- [41] T. Askham, A.J. Cerfon, An adaptive fast multipole accelerated Poisson solver for complex geometries, *J. Comput. Phys.* 344 (2017) 1–22.

- [42] L. Ahlfors, *Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable*, McGraw-Hill Education, 1979.
- [43] L.N. Trefethen, J. Weideman, The exponentially convergent trapezoidal rule, *SIAM Rev.* 56 (3) (2014) 385–458.
- [44] A. Klockner, A. Barnett, L. Greengard, M. O’Neil, Quadrature by expansion: a new method for the evaluation of layer potentials, *J. Comput. Phys.* 252 (2013) 332–349.
- [45] R. Davidson, *Physics of Nonneutral Plasmas*, World Scientific Publishing Company, 2001.
- [46] N.C. Hurst, J.R. Danielson, D.H.E. Dubin, C.M. Surko, Evolution of a vortex in a strain flow, *Phys. Rev. Lett.* 117 (2016) 235001.
- [47] T.A. Driscoll, L.N. Trefethen, *Schwarz-Christoffel Mapping*, vol. 8, Cambridge University Press, 2002.
- [48] E. Hairer, C. Lubich, Energy behaviour of the Boris method for charged-particle dynamics, *BIT Numer. Math.* 58 (2018) 969–979.
- [49] S. Muralikrishnan, R. Speck, ParaPIF: a parareal approach for parallel-in-time integration of particle-in-Fourier schemes, *arXiv preprint*, arXiv:2407.00485, 2024.
- [50] A. Cerfon, J. Freidberg, F. Parra, T. Antaya, Analytic fluid theory of beam spiraling in high-intensity cyclotrons, *Phys. Rev. Accel. Beams* 16 (2) (2013) 024202.
- [51] A.J. Cerfon, Vortex dynamics and shear-layer instability in high-intensity cyclotrons, *Phys. Rev. Lett.* 116 (17) (apr 2016).
- [52] A.H. Barnett, J. Magland, L. af Klinteberg, A parallel nonuniform fast Fourier transform library based on an “exponential of semicircle” kernel, *SIAM J. Sci. Comput.* 41 (5) (2019) C479–C504.
- [53] J. Yang, A. Adelman, M. Humbel, M. Seidel, T. Zhang, et al., Beam dynamics in high intensity cyclotrons including neighboring bunch effects: model, implementation, and application, *Phys. Rev. Spec. Top., Accel. Beams* 13 (6) (2010) 064201.
- [54] L.N. Trefethen, *Spectral Methods in MATLAB*, SIAM, 2000.
- [55] L.F. Ricketson, A.J. Cerfon, Sparse grid techniques for particle-in-cell schemes, *Plasma Phys. Control. Fusion* 59 (2) (2016) 024002.
- [56] S. Muralikrishnan, A.J. Cerfon, M. Frey, L.F. Ricketson, A. Adelman, Sparse grid-based adaptive noise reduction strategy for particle-in-cell schemes, *J. Comput. Phys.* X 11 (2021) 100094.