Published by CERN in the Proceedings of the CAS-CERN Accelerator School: Plasma Wake Acceleration, Geneva, Switzerland, 23–29 November 2014, edited by B. Holzer, CERN-2016-001 (CERN, Geneva, 2016)

Particle-In-Cell Codes for Plasma-based Particle Acceleration

A. Pukhov
University of Düsseldorf, Düsseldorf, Germany

Abstract

In this report we discuss the basic principles of Particle-In-Cell (PIC) codes and their application to plasma-based acceleration. The *ab initio* full electromagnetic relativistic PIC codes provide the most reliable description of plasmas, and their properties are described in detail. However, while they represent the most fundamental model, full PIC codes are computationally expensive. Plasma-based acceleration is a multi-scale problem with very disparate scales. The smallest scale is the laser or plasma wavelength (on the order of one to a hundred microns), and the largest scale is the acceleration distance (which ranges from a few centimetres to metres or even kilometres). The Lorentz boost technique allows the scale disparity to be reduced, at the cost of complicating the simulations and causing unphysical numerical instabilities in the code. Another possibility is to use the quasi-static approximation, whereby the different scales are separated analytically.

Keywords

Particle-In-Cell method; plasma-based acceleration; bubble regime; AWAKE; high-performance computing.

1 Introduction

Plasma-based particle acceleration involves a nonlinear medium, the relativistic plasmas [1]. Modelling this type of medium requires proper numerical simulation tools. Over the past few decades, Particle-In-Cell (PIC) codes have proven to be a very reliable and successful method for kinetic plasma simulations [2–5]. The success of PIC codes relies to a large extent on their analogy with the actual plasma, as suggested by their name. The plasma is in reality an ensemble of many individual particles, electrons and ions, interacting with each other via self-consistently generated fields. PIC codes have a very similar set-up, with the difference being that the number of numerical particles, or macroparticles, that we follow in the code may be significantly smaller than the number of particles in an actual plasma. One could think of one numerical 'macroparticle' as representing a clump, or cloud, of many real particles, which occupy a finite volume in space and all move together with the same velocity. Thus we have a 'numerical plasma' consisting of heavy macroparticles that have the same charge-to-mass ratio as the real plasma electrons and ions, but where each macroparticle substitutes for many real particles.

Depending on the application, different approximations can be used. The most fundamental approximation is provided by the full electromagnetic PIC codes, which solve the Maxwell equations together with the relativistic equations of motion for the numerical particles. These *ab initio* simulations produce the most detailed results, but can be very expensive. In the case of long-scale acceleration, where the driver propagates distances many times greater than its own length, the quasi-static approximation can be exploited. In this case, it is assumed that the driver changes little as it propagates distances that are comparable with its own length. The quasi-static approximation enables separation of fast and slow variables and hence great acceleration of the simulation, but at the cost of omitting radiation: the laser pulse or any emitted radiation cannot be described directly by such codes; rather, an additional module for the laser pulse is required, usually in the envelope approximation.

Quasi-static : equations don't have time-der. Quantities evolve very slowly with time

In this report, we describe the basic principles of PIC methods—both the full electromagnetic codes and the quasi-static approximation—as well as their application to plasma-based acceleration.

http://dx.doi.org/10.5170/CERN-2016-001.181

2 The basic equations

First, let us formulate the problem we are going to solve. To conduct electromagnetic and kinetic simulations, we have to solve the full set of Maxwell equations [6]:

$$\frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi \mathbf{j}, \tag{1}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E}, \tag{2}$$

$$\nabla \cdot \mathbf{E} = 4\pi\rho, \tag{3}$$

centimetre–gram–second

$$\nabla \cdot \mathbf{B} = 0, \tag{4}$$

where we use metric cgs units and c is the speed of light in vacuum.

Let us pause for a moment at this very fundamental system of equations. The electric and magnetic fields, $\mathbf{E}$ and $\mathbf{B}$, evolve according to the time-dependent equations (1) and (2), where the source term is in the form of the current density $\mathbf{j}$. This current is produced by the self-consistent charge motion in our system of particles. It is well known from texts on electrodynamics (see, e.g., Ref. [6]) that the Gauss law (3) together with the curl-free part of Eq. (2) lead to the charge continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{j} = 0. \tag{5}$$

One can apply the operator $\nabla\cdot$ to the Faraday law (2) and use the Gauss equation (3) for $\nabla \cdot \mathbf{E}$ to obtain (5). The opposite is true as well: if the charge density always satisfies the continuity equation (5), then the Gauss equation (3) is satisfied automatically during the evolution of the system, if it was satisfied initially. By symmetry this is also valid for the magnetic field $\mathbf{B}$. As there is no magnetic charge, Eq. (4) remains valid always if it was valid initially.

This means that we may reduce our problem to solving the two evolution equations (1) and (2), while considering Eqs. (3) and (4) as initial conditions only. This is a very important and fruitful approach; PIC codes using it have a 'local' algorithm, i.e. at each time step information is exchanged between neighbouring grid cells only. No global information exchange is possible because the Maxwell equations have 'absolute future' and 'absolute past' [7]. This property makes the corresponding PIC codes perfectly suitable for parallelization, and in addition the influence of (always unphysical) boundary conditions is strongly reduced.

3 Kinetics and hydrodynamics

Now we define our source term $\mathbf{j}$. In general, to do this, we have to know the distribution function of the plasma particles,

$$F^N(\mathbf{x}_1, \mathbf{p}_1, \ldots, \mathbf{x}_N, \mathbf{p}_N), \tag{6}$$

which defines the probability that an N-particle system takes a particular configuration in the $6N$-dimensional phase space. Here $\mathbf{x}_n$ and $\mathbf{p}_n$ are the coordinates and momentum of the nth particle. The function (6) provides the *exhaustive* description of the system. However, as has been shown in statistical physics (see, e.g., Ref. [8]), knowing the single-particle distribution function for each species of particle may suffice to describe the full system. The sufficient condition is that the inter-particle correlations should be small and can be treated perturbatively. The equation governing the evolution of the single-particle distribution function $f(\mathbf{x}, \mathbf{p})$ is called the Boltzmann–Vlasov equation [9, 10]:

$$\frac{\partial f}{\partial t} + \frac{\mathbf{p}}{m\gamma}\nabla f + \frac{\mathbf{F}}{m}\nabla_p f = \mathrm{St}, \tag{7}$$

where m is the single-particle mass of the species, $\gamma = \sqrt{1 + (p/mc)^2}$ is the relativistic factor, $\mathbf{F}$ is the force, and St is the collisional term (inter-particle correlations).

The kinetic equation (7) for the single-particle distribution function is six-dimensional and still complicated. Solving it either analytically or numerically is a challenge.

However, under appropriate conditions one can make further simplifications. In statistics it has been shown that inter-particle collisions lead to the Maxwellian distribution function (H-theorem; see, e.g., Ref. [8]). In the non-relativistic case the Maxwellian distribution has the form

$$f(\mathbf{x},\mathbf{v}) = \frac{n(\mathbf{x})}{\sqrt{2\pi T}} \exp\left(-\frac{(\mathbf{v}-\mathbf{V})^2}{2mT}\right), \tag{8}$$

takes a characteristic time to get to Maxwell distribution

where $n(\mathbf{x})$ is the local particle density, T is the temperature, and $\mathbf{V}$ is the local streaming velocity. The characteristic time for establishment of the Maxwellian distribution is the inter-particle collision time. Therefore, if the effective collision time in the system is short in comparison with other characteristic times, the distribution function remains Maxwellian. It is sufficient, then, to write evolution equations for the *momenta* of the distribution function, such as the local density

if t_collision << other times, remains Maxwell

$$n(\mathbf{x}) = \int f(\mathbf{x},\mathbf{v})\,\mathrm{d}^3\mathbf{v}, \tag{9}$$

the hydrodynamic velocity

$$\mathbf{V}(\mathbf{x}) = \int \mathbf{v} f(\mathbf{x},\mathbf{v})\,\mathrm{d}\mathbf{v}, \tag{10}$$

basically just the expected value

and the temperature

$$T(\mathbf{x}) = \int \frac{(\mathbf{v}-\mathbf{V})^2}{2m} f(\mathbf{x},\mathbf{v})\,\mathrm{d}^3\mathbf{v}. \tag{11}$$

These quantities are said to be fluid-like, or of hydrodynamic type.

fluid-like. Youtube tutorials for more detail

4 Vlasov and PIC codes

If, however, the distribution function does deviate, or is expected to deviate, significantly from the Maxwellian distribution, then we have to solve the Boltzmann–Vlasov equation, (7). What would be the appropriate numerical approach here?

At the first glance, it might seem that the most straightforward approach is to solve the partial differential equation (7) using finite differences on the Eulerian grid in phase space. Indeed, this direction has been pursued by several groups [11], and has gained even more popularity with the rapid growth of available computing power. One potential advantage of these 'Vlasov' codes is the possibility of producing 'smooth' results. Indeed, Vlasov codes handle the distribution function, which outputs a smoothly varying real number that gives the probability of finding plasma particles at a particular point in phase space.

Vlasov codes, however, are very computationally expensive, and even one-dimensional problems may require the use of parallel supercomputers. The reason these codes need so much computational power can be seen from Fig. 1(a). It shows schematically a mesh that one would need for a 1d1v Vlasov code. The notation '1d1v' means that the code resolves one spatial coordinate and one coordinate in the momentum (velocity) space. The shaded area represents the region of phase space that is occupied by plasma particles, where the associated two-dimensional distribution function $f(x, p_x)$ is essentially non-zero. The unshaded region is void of particles, and nothing interesting happens there. Nevertheless, one has to maintain such empty regions as parts of the numerical arrays, and process them when solving Eq. (7) on the Eulerian grid. This processing of empty regions leads to an enormous waste of computational power. This drawback becomes even more severe with an increase in the dimensionality of the problem under consideration. The efficiency of Vlasov codes drops exponentially with the number of dimensions, and becomes minuscule in the real 3d3v case, where the computer has to retain in memory and process a six-dimensional mesh, most of it empty of particles.

shaded: region of space occupied by plasma particles
unshaded: void of particles but must be maintained as numerical arrays

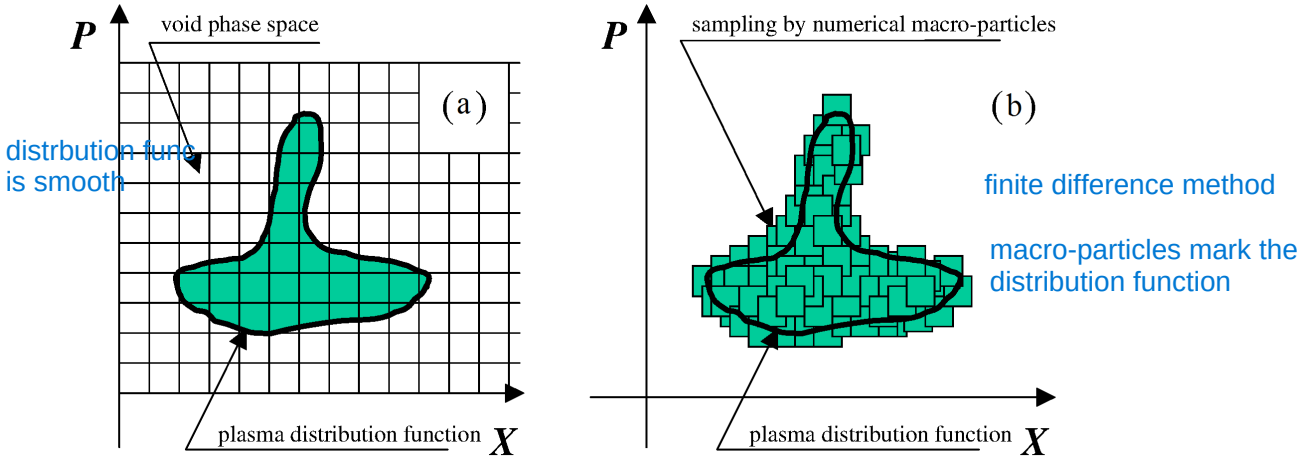

Fig. 1: Kinetic plasma simulations: (a) Vlasov method, using an Eulerian grid in the phase space; (b) PIC method, where numerical macroparticles mark the distribution function.

There is another, currently more computationally effective, method to solve the Boltzmann–Vlasov equation (7), namely the *finite element* method. The principle of this method is illustrated in Fig. 1(b). Again, imagine some distribution function in the phase space (the shaded region). Now, let us approximate, or sample, this distribution function by a set of Finite Phase-Fluid Elements (FPFEs):

$$f(\mathbf{x},\mathbf{p}) = \sum_n W_n^{\rm ph}\, S^{\rm ph}(\mathbf{x}-\mathbf{x}_n, \mathbf{p}-\mathbf{p}_n), \tag{12}$$

where $W_n^{\rm ph}$ is the 'weight' of the nth FPFE and $S^{\rm ph}(\mathbf{x},\mathbf{p})$ is the 'phase shape', or the support function in phase space. The centre of the nth FPFE is positioned at $(\mathbf{x}_n, \mathbf{p}_n)$. We are free to make a choice of the support function. For simplicity, we choose here a six-dimensional hypercube

$$S^{\rm ph}(\mathbf{x},\mathbf{p}) = 1, \quad |x_\alpha| < \frac{\Delta x_\alpha}{2}, \quad |p_\alpha| < \frac{\Delta p_\alpha}{2} \quad \text{for } \alpha = x, y, z,$$

where Δx_α is the FPFE size along the j-axis in configuration space, and Δp_α is the FPFE size along the p_α-axis in momentum space.

The 'phase fluid' transports the distribution function along the characteristics of the Boltzmann–Vlasov equation (see, e.g., Ref. [18]). So we have to advance the centres of the FPFEs along the characteristics:

$$\frac{{\rm d}\mathbf{x}_n}{{\rm d}t} = \frac{\mathbf{p}}{m\gamma}, \tag{13}$$

$$\frac{{\rm d}\mathbf{p}_n}{{\rm d}t} = \mathbf{F} + \mathbf{F}_{\rm St}, \tag{14}$$

where $\mathbf{F}_{\rm St}$ denotes the effective 'collisional' force due to the collision term in Eq. (7). The FPFEs follow the evolution of the distribution function in phase space. Of course, Eqs. (13) and (14) are just the relativistic equations of motion of particles! Thus, the FPFE method is equivalent to the PIC method.

A significant advantage of the finite element method over the Vlasov codes is that one does not need to maintain a grid in the full phase space. Instead, the FPFEs sample (or mark) only the interesting regions where particles are present and something important is going on. We still do maintain a grid in the *configuration* space to solve the field equations (1) and (2), but this grid has only three dimensions

Vlasov codes: need to maintain all regions
FPFEs: only the interesting regions where particles are present

grid space 3D instead of 6D in Vlasov

(and not six as in the Vlasov case). Thus, PIC codes may be viewed as 'packed' or 'Lagrangian' Vlasov codes. Moreover, the FPFE approach is even more fundamental than the Boltzmann–Vlasov equation itself, because it can easily be generalized to the case where one macroparticle corresponds to just one real particle, and where inter-particle correlations are not small. The corresponding codes are usually called P^3M (particle–particle–particle–mesh) codes [5].

As soon as we consider our macroparticles not simply as 'large clumps of real particles' but as finite elements in the phase space, we find that there is no fundamental obstacle to simulation of a cold plasma. Moreover, it is in this setting that the finite element approach really becomes effective computationally and superior to the Vlasov codes. The phase space of a cold plasma is degenerate: the particles occupy a mere three-dimensional hypersurface in the full six-dimensional phase space. Evidently, this hypersurface can be accurately sampled by even a relatively small number of macroparticles (FPFEs). As the system evolves, this surface deforms, stretches and contracts, but it remains degenerate and three-dimensional, unless any heating (i.e. diffusion in the phase space) is present. There is a full stock of interesting physical phenomena associated with relativistic laser–plasma interactions where the physical collisional heating is negligible. Unfortunately, the numerical heating that occurs in the 'standard' PIC codes [4] leads to an unphysical numerical diffusion in the phase space, which spoils the picture. Any code able to successfully simulate initially cold plasma must be energy-conserving.

5 Continuity equation

Historically, the first PIC codes were electrostatic [5], and they have to solve explicitly the Poisson equation

$$\nabla^2\phi = -4\pi\rho, \tag{15}$$

giving the static electric field

$$\mathbf{E}_{\parallel} = -\nabla\phi. \tag{16}$$

Generalization to the electromagnetic case seemed quite natural; one would simply add the vector potential $\mathbf{A}$ to get

$$\mathbf{E} = \mathbf{E}_{\parallel} + \mathbf{E}_{\perp} = -\nabla\phi - \frac{1}{c}\frac{\partial \mathbf{A}}{\partial t}. \tag{17}$$

Yet, there is another way of treating the electromagnetic fields. We established in the previous section that the simultaneous solution of Ampère's law (1) and the continuity equation (5) satisfies the Gauss law (3) automatically. Hence, one can work with the fields $\mathbf{E}$ and $\mathbf{B}$ directly, without introducing the electrostatic potential ϕ and without solving the Poisson equation (15).

It is very advantageous to avoid solving the Poisson equation (15), as it is *nonlocal*. This is an elliptic equation, and its solution essentially depends on the (usually unphysical) boundary conditions. A small perturbation or numerical error at the boundary may give rise to a global perturbation in the full simulation domain. In contrast, the Maxwell equations (1) and (2) are local. Any signal can propagate no faster than the vacuum speed of light, and we refer here to the Minkovski diagram, Fig. 2. As the central event (t_n, x_i, y_j, z_k) we choose some grid cell with indices (n, i, j, k) so that $t_n = n\tau$, $x_i = i\Delta x$, $y_j = j\Delta y$ and $z_k = k\Delta z$. Here $\tau, \Delta x, \Delta y$ and Δz denote the numerical steps along the time, X-, Y- and Z-axes. The light cone separates the full four-dimensional space into the regions of 'absolute past', 'absolute future', and 'absolutely distant' events. Only the events taking place in the 'absolute past' may stay in a casual connection with the central event. Thus, fields at the grid position (t_n, x_i, y_j, z_k) are influenced by the events happening at the instant t_{n-1} at the grid cells located within the circle $c\tau$ around the original cell. If we use an explicit numerical scheme, then the time step is limited through the Courant condition $c\tau < \min(\Delta x, \Delta y, \Delta z)$, and only the immediate neighbouring cells are involved. A numerical scheme that has this physical property is said to be *local*. In this sense, any numerical scheme that involves the solution of an elliptic equation, like the Poisson equation (15), is *nonlocal*.

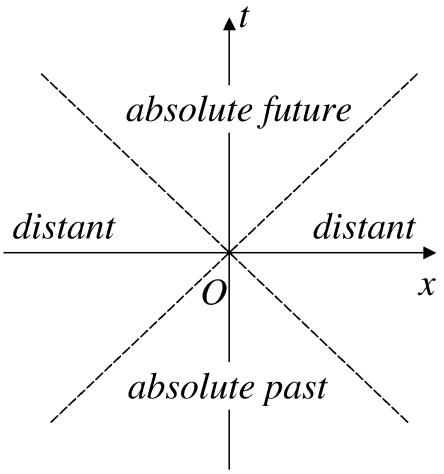

Fig. 2: Minkovski space–time diagram illustrating the causality in special relativity theory

A key issue in developing a local numerical scheme is the method of *current deposition* on the grid during particle motion. Let us consider a cubic FPFE (numerical particle) on a grid. We suppose that the particle and the grid elementary volume (i.e. cell volume) $V_c = \Delta x \Delta y \Delta z$ are *identical*, so that the particle length Δx_α is also the grid step along the α-axis, for $\alpha = x, y, z$. We mark the grid cells with the indices i, j and k along the x-, y- and z-axes

In discussing multi-dimensional PIC codes, we normally use a staggered or Yee lattice (grid), as illustrated in Fig. 3. We define the charge density on the grid at the centres of the cells, $\rho_{i+1/2,\, j+1/2,\, k+1/2}$, as

$$\rho_{i+1/2,\, j+1/2,\, k+1/2} = \sum_n W_n^\rho \, S^\rho(\mathbf{x}_{i+1/2,\, j+1/2,\, k+1/2} - \mathbf{x}_n). \tag{18}$$

The weight and form of the charge density interpolation for the particle are

$$\begin{aligned} S^\rho(\mathbf{x}) &= S_x^\rho(x) S_y^\rho(y) S_z^\rho(z), \\ S_j^\rho(\mathbf{x}_j) &= 1 - 2\frac{|x_j|}{\Delta_j}, \quad |x_j| < 0.5\Delta_j. \end{aligned} \tag{19}$$

The scheme (19) is the 'volume' (or 'area') weighting. It actually assigns the portion of the particle residing in a cell to the cell's centre.

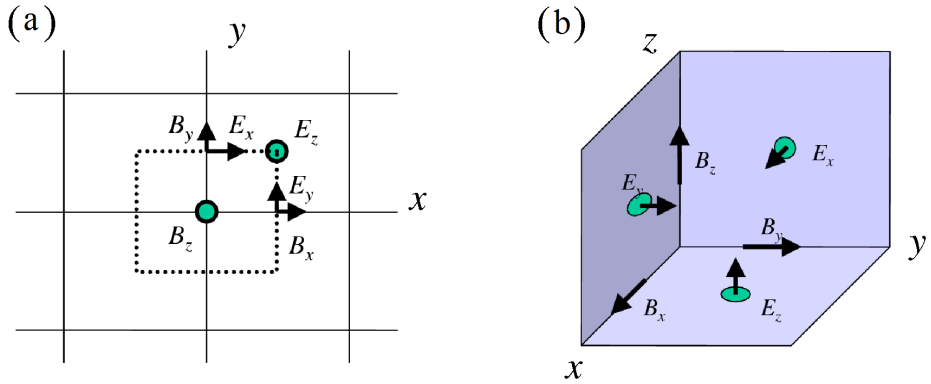

Fig. 3: Yee lattice in (a) 2D and (b) 3D

Now, if the particle moves, it generates current. How should one interpolate this current to the grid cells? One could try to use a straightforward interpolation, say $\mathbf{J} = \sum_n \mathbf{V}_n S_n^\rho$, or others like those discussed in Birdsall and Langdon's book [4]. Generally, such interpolations do not satisfy the continuity equation, i.e. defined in this way the current flux through a cell's boundaries does not represent the actual charge change in the cell. A further consequence is revealed when we integrate in time the Ampère law (1): the electric field obtained does not satisfy the Gauss law (3).

One possible way around this inconsistency is to *correct* the electric field obtained [4]. Suppose that we have advanced the electric field $\mathbf{E}'$ according to Eq. (1) and we run into difficulties with the Gauss law: $\nabla \cdot \mathbf{E}' \neq 4\pi\rho$. We could try to correct the electric field by introducing a potential $\delta\phi$ such that

$$\nabla^2 \delta\phi = -4\pi\rho + \mathbf{E}', \tag{20}$$

and then construct the corrected electric field as

$$\mathbf{E} = \mathbf{E} - \nabla\delta\phi, \tag{21}$$

which does satisfy the Gauss law:

$$\nabla\mathbf{E} = 4\pi\rho. \tag{22}$$

Unfortunately, this correction requires us to solve the nonlocal elliptic problem (20).

It turns out, however, that the currents can be defined in a self-consistent way [2]. To do so, one has to follow the particle trajectory in detail, and keep recording of how much charge has passed through each of the cell's boundaries. Fig. 4 illustrates the idea. Let us take a particle with centre located inside the grid elementary volume V_c centred at the grid vertex (i, j, k). The particle's initial position is (x_0, y_0, z_0), and after one time step the particle moves to the new position (x_1, y_1, z_1). To begin with, we suppose that the new position is still inside the elementary volume. We denote the particle displacement by $(\delta x, \delta y, \delta z)$. The particle generates the instantaneous current density $\mathbf{j} = \mathbf{V} W^\rho S^\rho$, and the current fluency $\mathbf{J}$, i.e. the charge that has crossed some surface Ω during the time step τ, is given by

$$\mathbf{J} = \int_\Omega \mathrm{d}\mathbf{\Omega} \int_0^\tau \mathbf{V} W^\rho S^\rho \,\mathrm{d}t = \int_\Omega \mathrm{d}\mathbf{\Omega} \int_{\mathbf{x_0}}^{\mathbf{x_1}} W^\rho S^\rho \,\mathrm{d}\mathbf{x}. \tag{23}$$

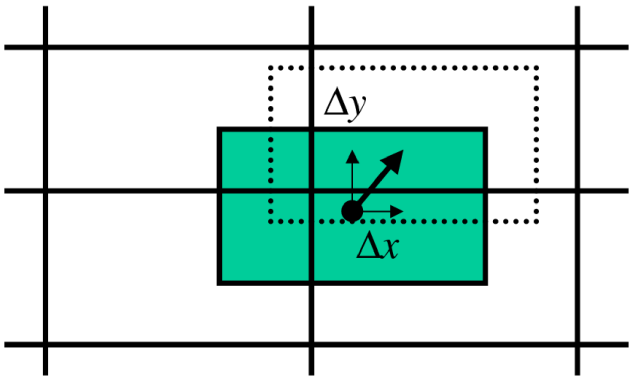

Fig. 4: Tracing of a particle's trajectory

Thus, we have to integrate along the particle trajectory. If, as usual [4], we are using the second-order finite difference scheme to advance the particle position, then the particle moves along a straight line during one time step. Assuming this, it is easy to calculate that if the particle remained within the elementary volume, it has induced the following currents on the grid:

$$J^x_{i,\,j+1/2,\,k+1/2} = \delta x\, W^\rho (a_y a_z + b_{yz}),$$

$$
\begin{aligned}
J^y_{i+1/2,\,j,\,k+1/2} &= \delta y\, W^\rho(a_z a_x + b_{zx}),\\
J^z_{i+1/2,\,j+1/2,\,k} &= \delta z\, W^\rho(a_x a_y + b_{xy}),\\
J^x_{i,\,j-1/2,\,k+1/2} &= \delta x\, W^\rho[(1-a_y)a_z - b_{yz}],\\
J^x_{i,\,j+1/2,\,k-1/2} &= \delta x\, W^\rho[a_y(1-a_z) - b_{yz}],\\
J^x_{i,\,j-1/2,\,k-1/2} &= \delta x\, W^\rho[(1-a_y)(1-a_z) + b_{yz}],\\
J^y_{i+1/2,\,j,\,k-1/2} &= \delta y\, W^\rho[(1-a_z)a_x - b_{zx}],\\
J^y_{i-1/2,\,j,\,k+1/2} &= \delta y\, W^\rho[a_z(1-a_x) - b_{zx}],\\
J^y_{i-1/2,\,j,\,k-1/2} &= \delta y\, W^\rho[(1-a_z)(1-a_x) + b_{zx}],\\
J^z_{i-1/2,\,j+1/2,\,k} &= \delta z\, W^\rho[(1-a_x)a_y - b_{xy}],\\
J^z_{i+1/2,\,j-1/2,\,k} &= \delta z\, W^\rho[a_x(1-a_y) - b_{xy}],\\
J^z_{i-1/2,\,j-1/2,\,k} &= \delta z\, W^\rho[(1-a_x)(1-a_y) + b_{xy}],
\end{aligned}
\tag{24}
$$

where

$$
\begin{aligned}
a_\alpha &= 1 - 2\frac{|x_\alpha + 0.5\Delta_\alpha|}{\Delta_\alpha},\\
b_{\alpha\beta} &= \frac{1}{12}\delta x_\alpha \delta x_\beta\\
&\text{for } \alpha,\beta \in \{x,y,z\}.
\end{aligned}
\tag{25}
$$

If the particle leaves the elementary volume where it was residing initially, then the full displacement must be split into several 'elementary' motions. During each elementary motion the particle must remain inside an elementary volume surrounding the corresponding vertex of the grid. This 'bookkeeping' of the particle motion does require some programming effort, but it appears to be very important to implement it.

The electric field is then advanced in time according to

$$
\mathbf{E}^{n+1} - \mathbf{E}^n = c\tau\hat{\nabla}\times\mathbf{B}^{n+1/2} - 4\pi\mathbf{J}^{n+1/2},
\tag{26}
$$

with the particular components of the electric field $\mathbf{E}$ defined at the same positions on the grid as the current $\mathbf{J}$; here $\hat{\nabla}\times$ denotes the finite difference version of the curl operator.

One may think of an alternative approach that is somewhat easier from the programming point of view: why don't we replace the actual, straight motion of the particle during one time step by the *average* of all possible *rectangular* paths along the grid axes which connect the initial and final positions of the particle? This approach was taken by Morse and Nielson [12]; however, this 'fake' integration led to an unacceptably rapid growth of *electromagnetic noise* in their code. The author has also found that even small deviations from the accurate current deposition (24) will immediately result in noise boosting, even if the deviated scheme is still charge-conserving.

Therefore, scheme (24) is *the* method to use to avoid solving elliptic equations and yet still satisfy the Gauss law numerically. In other words, we rigorously enforce the *detailed*—i.e. down to each grid cell—charge conservation and correct continuity equation.

6 Energy conservation

In the previous section we discussed how to develop an electromagnetic PIC code that is rigorously charge-conserving. Another important conservation law we would like to enforce is total energy conservation. Indeed, it is well known that one of the worst plagues of standard PIC codes is the effect of

numerical heating. The numerical 'temperature' (or rather the chaotic energy per numerical particle) is known to grow exponentially until the effective Debye length becomes comparable with the grid size; thereafter the exponential growth transitions to a more moderate linear heating. This is an effect of 'aliasing'—the inconsistent interpolation of the fields defined on the grid to the particle position.

As we hinted in the Introduction, there is no fundamental reason for numerical heating to occur if we adhere to the FPFE paradigm. Now we proceed to design an *energy-conserving electromagnetic code*. We start with the exact analytical equation for the full energy of the system,

$$H = \sum_n m_n c^2(\gamma - 1) \; + \; \frac{1}{8\pi} \int_V (E^2 + B^2) \, \mathrm{d}V, \tag{27}$$

where m_n is the particle's mass, $\gamma = \sqrt{1 + (p/m_n c)^2}$ is the relativistic γ-factor, and the integration is taken over the full volume V.

Next, we split the electric field into longitudinal $\mathbf{E}_\parallel$ and transverse $\mathbf{E}_\perp$ parts, so that

$$\begin{aligned} \nabla \cdot \mathbf{E}_\perp &= 0, \\ \nabla \times \mathbf{E}_\parallel &= 0. \end{aligned} \tag{28}$$

Then, we introduce a potential ϕ such that $\mathbf{E}_\parallel = -\nabla\phi$. One can show easily that

$$\int_V \mathbf{E}_\perp \mathbf{E}_\parallel \, \mathrm{d}V = -\int_\Omega \phi \mathbf{E}_\perp \, \mathrm{d}\boldsymbol{\Omega} = 0 \tag{29}$$

for an infinite or periodic volume. Here Ω is a surface surrounding the volume. As a consequence, we can write the energy of our system as

$$H = \sum_n m_n c^2(\gamma - 1) \; + \; \frac{1}{8\pi} \int_V (E_\parallel^2 + E_\perp^2 + B^2) \, \mathrm{d}V = H_{\mathrm{kin}} + H_{\mathrm{S}} + H_{\mathrm{EM}} + H_B, \tag{30}$$

where

$$H_{\mathrm{kin}} = \sum_p m_p c^2(\gamma - 1) \tag{31}$$

is the kinetic energy of the particles,

$$H_{\mathrm{S}} = \frac{1}{8\pi} \int_V E_\parallel^2 \, \mathrm{d}V \tag{32}$$

is the electrostatic part of the electric field energy,

$$H_{\mathrm{EM}} = \frac{1}{8\pi} \int_V E_\perp^2 \, \mathrm{d}V \tag{33}$$

is the electromagnetic part of the electric field energy, and

$$H_B = \frac{1}{8\pi} \int_V B^2 \, \mathrm{d}V \tag{34}$$

is the magnetic field energy.

The expression (30) for the energy is for continuous fields and individual particles; however, it is straightforward to write an analogue for a finite difference numerical scheme.

We are using the staggered grid (Yee lattice) and have fixed the current interpolation to the grid using the system (24). Now we have to define the *force interpolation* to the actual particle position in such a way that the resulting numerical scheme conserves the Hamiltonian (30).

Most dangerous in terms of numerical heating is the *electrostatic* part of the code. It is the electrostatic plasma waves that are responsible for the Debye shielding. Also, the $\mathbf{v} \times \mathbf{B}$ part of the Lorentz force acting on the particle conserves energy automatically, as does the $\mathbf{B}$-field advance according to the Faraday law (2). Hence, for the time being, we neglect the magnetic field and the magnetic energy part, and we enforce the conservation of $H_{\mathrm{E}} = H_{\mathrm{kin}} + H_{\mathrm{S}} + H_{\mathrm{EM}}$. The numerical scheme will conserve the energy if it is derived from equations in the canonical form

$$\frac{\mathrm{d}\mathbf{p}_p}{\mathrm{d}t} = -\partial_{\mathbf{x}_p} H_{\mathrm{E}}, \tag{35}$$

$$\frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} = \partial_{\mathbf{p}_p} H_{\mathrm{E}}, \tag{36}$$

where the index p runs through all particles.

We rewrite Eqs. (35) and (36) more explicitly as

$$\frac{\mathrm{d}\mathbf{p}_p}{\mathrm{d}t} = -\frac{1}{4\pi} \int_V \mathbf{E} \cdot \partial_{\mathbf{x}_p} \mathbf{E} \, \mathrm{d}V, \tag{37}$$

$$\frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} = \frac{\mathbf{p}_p}{\gamma_p} = \mathbf{V}_p. \tag{38}$$

To deal with Eq. (37), one has to refer to Eq. (26) for the advance in time of the electric field. To get the correct expression for $\partial_{\mathbf{x}_p} \mathbf{E}$, let us displace the particle p by a small distance $\delta\mathbf{x}$. This displacement generates a current $\delta\mathbf{J}$ on the adjacent grid positions according to (24). The resulting change $\delta\mathbf{E}$ in the electric field is

$$\delta\mathbf{E} = -4\pi\delta\mathbf{J} \tag{39}$$

at the same grid positions. Hence, we may rewrite the first canonical equation (37) as

$$\frac{\mathrm{d}\mathbf{p}_p}{\mathrm{d}t} = \int_V \mathbf{E} \cdot \partial_{\mathbf{x}_p} \delta\mathbf{J} \, \mathrm{d}V. \tag{40}$$

The expression (40) has a very simple and clear meaning: to make the PIC code energy-conserving, one has to employ the same scheme for the electric field interpolation to the particle position as for the current deposition. Thus, the energy-conserving interpolation scheme for the $\mathbf{E}$-field is

$$\begin{aligned}
E_x^p &= W^\rho \big[E^x_{i,\,j+1/2,\,k+1/2} a_y a_z + E^x_{i,\,j-1/2,\,k+1/2} (1-a_y) a_z \\
&\quad + E^x_{i,\,j+1/2,\,k-1/2} a_y (1-a_z) + E^x_{i,\,j-1/2,\,k-1/2} (1-a_y)(1-a_z) \big], \\
E_y^p &= W^\rho \big[E^y_{i+1/2,\,j,\,k+1/2} a_x a_z + E^y_{i-1/2,\,j,\,k+1/2} (1-a_x) a_z \\
&\quad + E^y_{i+1/2,\,j,\,k-1/2} a_x (1-a_z) + E^y_{i-1/2,\,j,\,k-1/2} (1-a_x)(1-a_z) \big], \\
E_z^p &= W^\rho \big[E^z_{i+1/2,\,j+1/2,\,k} a_y a_x + E^z_{i+1/2,\,j-1/2,\,k} (1-a_y) a_x \\
&\quad + E^z_{i-1/2,\,j+1/2,\,k} a_y (1-a_x) + E^z_{i-1/2,\,j-1/2,\,k} (1-a_y)(1-a_x) \big],
\end{aligned} \tag{41}$$

It is important that the electric field is taken *at the present particle position* and not averaged along the trajectory. Also, the higher-order corrections b_α which we introduced for the current depositions are absent here. This is because Eq. (40) gives the analytical expression for the infinitesimal particle displacements, i.e. it has to be considered in the limit $|\delta\mathbf{x}| \to 0$.

6.1 Particle push

For advancing the particle in time one could then use the Boris scheme [4], with the electric field $\mathbf{E}$ interpolated according to (41) and the magnetic field $\mathbf{B}$ interpolated using a different scheme, to be discussed later. The Boris scheme is

$$\frac{\mathbf{p}_1 - \mathbf{p}_0}{\tau} = e \left(\mathbf{E} + \frac{1}{c} \frac{\mathbf{p}_1 + \mathbf{p}_0}{2\gamma_{1/2}} \times \mathbf{B} \right), \tag{42}$$

where $\mathbf{p}_0$ and $\mathbf{p}_1$ are the initial and the final particle momenta and $\gamma_{1/2}$ is the γ-factor taken at the middle of the time step. This scheme is time-reversible and *semi*-implicit. It can be analytically resolved for the final momentum $\mathbf{p}_1$ (see Ref. [4]):

$$\mathbf{p}^{n+1/2} = \mathbf{p}^- - e\,\mathbf{E}\,\frac{\tau}{2}, \tag{43}$$

$$\mathbf{p}^{n-1/2} = \mathbf{p}^+ + e\,\mathbf{E}\,\frac{\tau}{2}, \tag{44}$$

$$\frac{\mathbf{p}^+ - \mathbf{p}^-}{\tau} = \frac{q}{2\gamma mc}\,\mathbf{p}^+ + \mathbf{p}^- \times \mathbf{B}, \tag{45}$$

$$\mathbf{p}' = \mathbf{p}^- + \frac{q\tau}{2\gamma}\,\mathbf{p}^- \times \mathbf{B}, \tag{46}$$

$$\mathbf{p}^+ = \mathbf{p}^- + \frac{2}{1+\left(\dfrac{q\tau B}{2\gamma}\right)^2}\,\mathbf{p}' \times \mathbf{B}. \tag{47}$$

The γ-factor should be calculated after step (43). However, as the magnetic field rotates the particle momentum, the Boris scheme is not exactly symmetric. An alternative scheme has been proposed recently; see Ref. [13]. We derive this alternative scheme below.

The equation of motion is discretized as

$$\frac{\mathbf{p} - \mathbf{p}_0}{\tau} = q\,\mathbf{E} + \frac{q}{2}\left(\frac{\mathbf{p}}{\gamma} + \frac{\mathbf{p}_0}{\gamma_0}\right) \times \mathbf{B}. \tag{48}$$

Here, $\mathbf{p_0}$ is the initial particle momentum and $\mathbf{p}$ is the particle momentum after the push with the corresponding γ-factors. We can rewrite this equation in the form

$$\mathbf{p} = \mathbf{a} + \frac{\mathbf{p}}{\gamma} \times \mathbf{b}, \tag{49}$$

where

$$\mathbf{a} = \mathbf{p}_0 + q\tau\mathbf{E} + \frac{q\tau}{2}\frac{\mathbf{p}_0}{\gamma_0} \times \mathbf{B} \tag{50}$$

and

$$\mathbf{b} = \frac{q\tau}{2}\mathbf{B}. \tag{51}$$

We rewrite (49) as

$$\gamma\mathbf{p} = \gamma\mathbf{a} + \mathbf{p} \times \mathbf{b}. \tag{52}$$

Taking the scalar product of (52) with $\mathbf{p}$ gives

$$p^2 = \mathbf{a} \cdot \mathbf{p}. \tag{53}$$

Taking the scalar product of (52) with $\mathbf{b}$ gives

$$\mathbf{b} \cdot \mathbf{p} = \mathbf{a} \cdot \mathbf{b}. \tag{54}$$

Taking the scalar product of (52) with $\mathbf{a}$ gives

$$\gamma\mathbf{a} \cdot \mathbf{p} - \gamma a^2 = \mathbf{a} \cdot (\mathbf{p} \times \mathbf{b}) = \mathbf{p} \cdot (\mathbf{b} \times \mathbf{a}) = \mathbf{b} \cdot (\mathbf{a} \times \mathbf{p})\,. \tag{55}$$

Taking the vector product of $\mathbf{a}$ and (52) gives

$$\gamma\mathbf{a} \times \mathbf{p} = \mathbf{p}\,(\mathbf{a} \cdot \mathbf{b}) - \mathbf{b}\,(\mathbf{a} \cdot \mathbf{p})\,. \tag{56}$$

Combining (53)–(56) gives

$$\gamma\left(p^2-a^2\right)=\frac{\mathbf{b}}{\gamma}\left[\mathbf{p}(\mathbf{a}\cdot\mathbf{b})-\mathbf{b}(\mathbf{a}\cdot\mathbf{p})\right]=\frac{\left[(\mathbf{a}\cdot\mathbf{b})^2-b^2p^2\right]}{\gamma} \tag{57}$$

or

$$\gamma^2\left(\gamma^2-1-a^2\right)=(\mathbf{a}\cdot\mathbf{b})^2-b^2\left(\gamma^2-1\right). \tag{58}$$

This leads to the quadratic equation

$$\gamma^4+\gamma^2\left(b^2-1-a^2\right)-b^2-(\mathbf{a}\cdot\mathbf{b})^2=0, \tag{59}$$

which has the solution

$$\gamma^2=\frac{1+a^2-b^2}{2}+\sqrt{\left(\frac{1+a^2-b^2}{2}\right)^2+b^2+(\mathbf{a}\cdot b)^2}. \tag{60}$$

Now we have to find the particle momentum after the push $\mathbf{p}$. To do this, we take the vector product of $\mathbf{b}$ and (52) to get

$$\gamma\left(\mathbf{b}\times\mathbf{p}-\mathbf{b}\times\mathbf{a}\right)=\mathbf{p}\cdot\mathbf{b}-\mathbf{b}\left(\mathbf{p}\cdot\mathbf{b}\right)=\mathbf{p}\,b^2-\mathbf{b}\left(\mathbf{a}\cdot\mathbf{b}\right). \tag{61}$$

Using (52), we find that

$$\gamma^2\mathbf{a}-\gamma^2\mathbf{p}-\gamma\mathbf{b}\times\mathbf{a}=\mathbf{p}b^2-\mathbf{b}\left(\mathbf{a}\cdot\mathbf{b}\right). \tag{62}$$

Solving for $\mathbf{p}$, we obtain

$$\mathbf{p}=\frac{\gamma^2\mathbf{a}+\gamma\mathbf{a}\times\mathbf{b}+\mathbf{b}(\mathbf{a}\cdot\mathbf{b})}{\gamma^2+b^2}. \tag{63}$$

This pusher is fully implicit and does not require splitting of the Lorentz operator into the electric field push and the magnetic field rotation.

6.2 Energy conservation tests

The interpolation scheme consisting of (24) and (41) conserves the energy *exactly* for time steps which are small enough that the particle does not leave the original grid cell. If, however, motion of the particle becomes highly relativistic, the system will exhibit a slow energy growth. Notwithstanding this small drawback, we have solved one of the major problems with PIC codes. We may now simulate *cold* plasma. As 'cold' usually means non-relativistic 'temperatures', the energy is conserved and numerical heating is absent. If we do have a hot plasma, with temperatures close to relativistic ones, the method of *stochastic sampling* of the phase space becomes valid. Fortunately, the Debye length of such plasmas is many grid cells anyway, and numerical heating is not an issue.

The scheme (41) for electric field interpolation to the particle position is identical to the energy-conserving scheme used in electrostatic codes with the charge–potential (ρ–ϕ) formalism [4,5]. However, as we shall see later, there is a significant difference between these electrostatic codes and the field–current (**E**–**J**) formalism used in our electromagnetic simulations.

Figure 5 shows the evolution of total energy in an isolated system of particles for an energy-conserving (EC) algorithm (solid lines) and a 'momentum-conserving' (MC) [4] algorithm (dashed lines) in the following two cases: (a) warm plasma, with Debye length $D=0.5\Delta x$; (b) cold plasma, with $D=5\times10^{-3}\Delta x$. Although energy conservation for the EC algorithm is not exact, it is much better than in the MC case. The actual energy change is only 2% over 100 plasma oscillations for the EC algorithm; this property makes it possible to simulate a cold plasma.

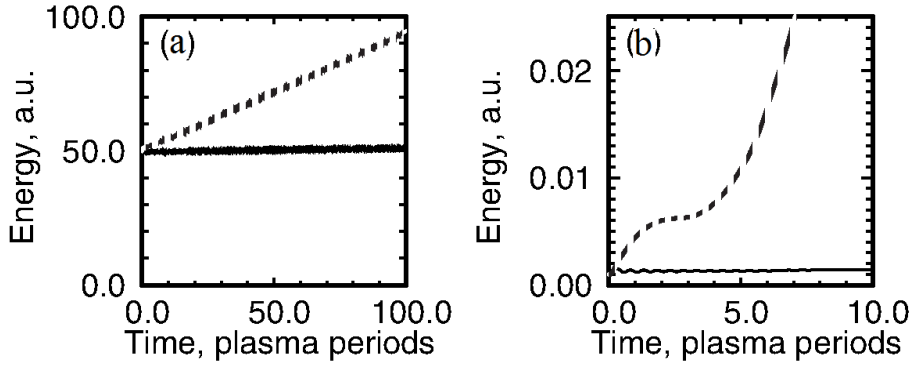

Fig. 5: Energy conservation in Virtual Laser Plasma Lab (VLPL) code based on an energy-conserving (EC) algorithm (solid lines) and numerical heating in a standard momentum-conserving (MC) algorithm (dashed lines), for the cases of (a) warm plasma, with Debye length $D = 0.5\Delta x$, and (b) cold plasma, with $D = 5 \cdot 10^{-3}\Delta x$. The energy change in the VLPL code is within 2% after 100 plasma periods.

7 Momentum and current conservation

It is known that the energy-conserving electrostatic PIC codes do not conserve momentum [4]. Indeed, it is easy to show that, strictly speaking, electric field interpolation to the particle position in the form of (41) does not conserve the total momentum:

$$\frac{\mathrm{d}\mathbf{P}_{\mathrm{total}}}{\mathrm{d}t} = \sum_p q_p E_p \neq 0, \tag{64}$$

if the particles cross cell boundaries during their motion.

Now, the question is: how detrimental is this lack of momentum conservation to the PIC code?

Momentum non-conservation in electrostatic PIC codes using the ρ–ϕ formalism leads to some notable consequences [4]. If one starts the simulation with electrons drifting with respect to the resting ions, the electrons will experience an average *drag force* from the grid. As a result, after a few plasma periods, an initially regular electron drift becomes chaotic, and the simulation ends up with disordered hot electrons without any net motion with respect to the ions. Although the final energy of the system is preserved, and remains the same as the initial kinetic energy of the drifting electrons, the failure of momentum conservation is spectacular.

However, this spectacular example of momentum non-conservation in the one-dimensional electrostatic PIC code is rather an artefact of the ρ–ϕ formalism. Moreover, even the inital 'equilibrium' of electrons drifting with respect to ions is an artefact itself. Indeed, the original Maxwell equations (1) and (2) simply *do not allow for freely drifting electrons in the one-dimensional geometry*! This drift would correspond to a constant current, which results in a fast build-up of the longitudinal electric field. Consequently, electrons must oscillate around their initial positions at the local plasma frequency. This contradiction with the Maxwell equations has apparently remained unmentioned in regard to the ρ–ϕ formulation of the electrostatic code.

In the more realistic **E**–**J** formalism of electromagnetic codes, the ions have to drift *together* with the electrons, unless the forward electron current is compensated for by some artificial 'return' current, such as the longitudinal part of $\nabla \times \mathbf{B}$, which evidently does not exist in the one-dimensional geometry. A code using the **E**–**J** formalism conserves the net current:

$$\langle \mathbf{J} \rangle = \mathbf{J}_0 = \mathrm{constant}, \tag{65}$$

where the averaging is done in time over the local plasma frequency. Indeed, the only possible deviations in the current are those due to the longitudinal part of the electric field, i.e. the *charge displacement* current.

The current conservation law (65) is a very important property, which may *compensate* for the absence of a detailed momentum conservation. As an example, let us consider the total electron current flowing in the simulation,

$$\mathbf{J}_{\mathrm{e}} = \sum_p eW_p \mathbf{V}_p, \tag{66}$$

where e is the electron charge and W_p is the 'weight' of the numerical particle p. When averaged over the plasma period, the current (66) is conserved by our code. Now we may write the electron momentum in the non-relativistic case as

$$\mathbf{P}_{\mathrm{e}} = \sum_p m_{\mathrm{e}} W_p \mathbf{V}_p = \frac{e}{m_{\mathrm{e}}} \mathbf{J}_{\mathrm{e}}, \tag{67}$$

where m_{e} is the electron mass.

It follows from (67) that the total momentum is simply proportional to the total current, and so it is conserved on average. This is good news for the energy-conserving PIC code we have designed here: the code does conserve momentum on average in the non-relativistic case. When the particles are moving with relativistic energies, however, the identity (67) breaks down, and the momentum conservation is no longer ideal. Fortunately, the current conservation (66) still imposes a strong enough symmetry to prevent bad consequences such as those discussed in Ref. [4].

8 Maxwell solver: numerical dispersion-free scheme

Earlier, we discussed how to push particles and collect currents on the grid. Now we discuss the finite difference solver for the time-dependent Maxwell equations (1) and (2).

The standard way to propagate the fields on the Yee lattice (Fig. 3) is to use the centred conservative scheme [4]. Let us first consider the two-dimensional geometry for simplicity. In the 2D X–Y geometry, one may distinguish two kinds of polarization: s-polarizations with E_z, B_x and B_y fields, and p-polarizations with E_x, E_y and B_z fields. It can be shown that if the initial condition contains p-polarized fields and J_x and J_y currents only, then the s-polarized fields are not excited at all [4]. Thus, we may take the p-polarization as an example. The standard 2D scheme for the p-polarization is

$$B_{z\,i,j}^{\,n+1/2} - B_{z\,i,j}^{\,n-1/2} = \frac{c\tau}{\Delta y}(E_{x\,i,j+1/2}^{\,n} - E_{x\,i,j-1/2}^{\,n}) - \frac{c\tau}{\Delta x}(E_{y\,i+1/2,j}^{\,n} - E_{y\,i-1/2,j}^{\,n}), \tag{68}$$

$$E_{x\,i,j+1/2}^{\,n+1} - E_{x\,i,j+1/2}^{\,n} = \frac{c\tau}{\Delta y}(B_{z\,i,j+1}^{\,n+1/2} - B_{z\,i,j}^{\,n+1/2}) - 4\pi\tau j_{x\,i,j+1/2}^{\,n+1/2}, \tag{69}$$

$$E_{y\,i+1/2,j}^{\,n+1} - E_{y\,i+1/2,j}^{\,n} = \frac{c\tau}{\Delta x}(B_{z\,i+1,j}^{\,n+1/2} - B_{z\,i,j}^{\,n+1/2}) - 4\pi\tau j_{y\,i+1/2,j}^{\,n+1/2}. \tag{70}$$

The scheme (68)–(70) uses centred expressions for the finite difference $\nabla\times$ operators, such as

$$(\nabla \times \mathbf{B}_z)_x = \frac{1}{\Delta y}(B_{z\,i,j+1}^{\,n+1/2} - B_{z\,i,j}^{\,n+1/2}). \tag{71}$$

The Maxwell equations (1) and (2) and the corresponding scheme (68)–(70) are essentially linear partial differential equations where the only nonlinear source terms are in the form of the currents $\mathbf{j}$. In this case, we decide on the quality of the finite difference scheme (68)–(70) by comparing its dispersion properties with those of the Maxwell equations themselves.

According to the Maxwell equations, all electromagnetic waves in vacuum travel at the speed of light c. There is no dispersion in vacuum; not so for the finite differences. We Fourier-analyse the scheme

(68)–(70) by decomposing the fields in plane waves:

$$\mathbf{E} = \sum_{\mathbf{k}} \mathbf{E}_{\mathbf{k}} \exp(-i\omega_{\mathbf{k}} t + i\mathbf{k}\cdot\mathbf{x}),$$
$$\mathbf{B} = \sum_{\mathbf{k}} \mathbf{B}_{\mathbf{k}} \exp(-i\omega_{\mathbf{k}} t + i\mathbf{k}\cdot\mathbf{x}), \tag{72}$$

where $\mathbf{k}$ is the wavevector, $\omega_{\mathbf{k}}$ is the corresponding frequency, and $\mathbf{E}_{\mathbf{k}}$ and $\mathbf{B}_{\mathbf{k}}$ are the amplitudes of the Fourier harmonics. For the continuum Maxwell equations we have the simple dispersion relation

$$\omega_{\mathbf{k}} = c|\mathbf{k}|, \tag{73}$$

while the discretization in the finite difference scheme (68)–(70) introduces the numerical dispersion

$$\frac{1}{c^2\tau^2}\sin^2\frac{\omega_{\mathbf{k}}\tau}{2} = \frac{1}{\Delta x^2}\sin^2\frac{k_x\Delta x}{2} + \frac{1}{\Delta y^2}\sin^2\frac{k_y\Delta y}{2}, \tag{74}$$

where Δx and Δy are the spatial grid steps and τ is the time step. The time step is limited by the Courant stability condition $c^2\tau^2 \le \Delta x^2\Delta y^2/(\Delta x^2 + \Delta y^2)$. If we run the code close to this limit of stability, then only the waves propagating along the grid diagonals are dispersion-free. The largest numerical dispersion is experienced by the waves running along the grid axes. This is illustrated in Fig. 6(a), where we plot the phase velocity of the numerical modes, $V_{\rm ph} = \omega_k/k$, for this standard scheme. We mention that we have chosen $\Delta y = 2\Delta x$ here, and this explains the apparent asymmetry of the plot.

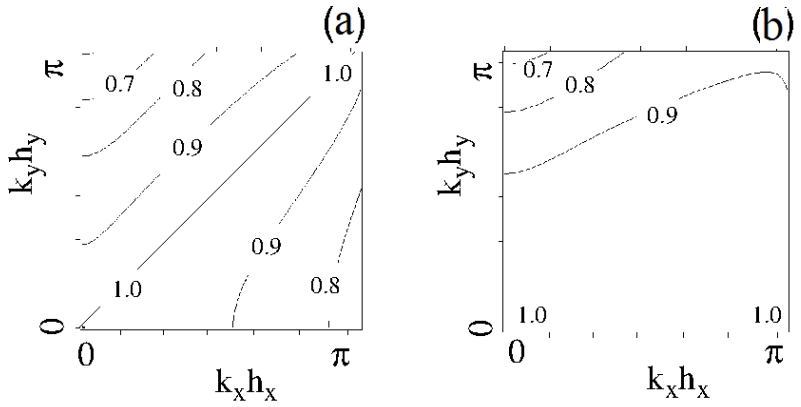

Fig. 6: Plots of the numerical phase velocity $v_{\rm ph}/c$ corresponding to (a) the standard scheme (74) and (b) the NDF scheme (80) used in the VLPL code. The grid cell aspect ratio is $\Delta x/\Delta y = 0.5$. The scheme (74) is dispersionless along the mesh diagonals $k_x = k_y$, while the NDF scheme (80) is dispersionless for waves travelling in the X-direction when $k_y = 0$.

Although numerical dispersion of the standard scheme may not be an issue when one is simulating dense, nearly critical plasma, it can cause trouble in simulations of very underdense plasmas, as it is important for particle acceleration [17]. In the dense plasma case, the plasma dispersion is usually stronger than the numerical dispersion, and so the problem is masked. In low-density plasma, however, the plasma dispersion is small; yet it has to be resolved accurately, as it influences the phase velocity of the laser pulse. As a consequence, one needs to use many grid cells per laser wavelength to obtain physically correct results. Of course, one could send the laser along one of the grid diagonals, but this is extremely inconvenient from the programming point of view.

We now aim at designing a superior numerical scheme which does not have numerical dispersion at all, or removes it to a large extent. Let us return to the finite difference expression for the curl operator,

(71). This centred operator can be written in a different way, using averages from the adjacent cells:

$$(\nabla \times \mathbf{B}_z)_x \to \frac{1}{2\Delta y}\big(B_{z\,i+1,\,j+1}^{\,n+1/2} + B_{z\,i-1,\,j+1}^{\,n+1/2} - B_{z\,i+1,\,j}^{\,n+1/2} - B_{z\,i-1,\,j}^{\,n+1/2}\big). \tag{75}$$

At first sight it is unclear what we have gained from averaging. It seems evident that the averaged scheme (75) may have only worse dispersion than the simpler scheme (71), but this is only partially true.

It turns out that one can choose a *linear combination* of the two schemes (71) and (75) in such a way that the dispersions of the two schemes *compensate for each other*! The resulting scheme for the p-polarization in the 2D geometry is

$$\begin{aligned} B_{z\,i,j}^{\,n+1/2} - B_{z\,i,j}^{\,n-1/2} = {} & \frac{c\tau}{\Delta y}\Big[b_x\big(E_{x\,i,\,j+1/2}^{\,n} - E_{x\,i,\,j-1/2}^{\,n}\big) \\ & + a_x\big(E_{x\,i+1,\,j+1/2}^{\,n} - E_{x\,i+1,\,j-1/2}^{\,n} + E_{x\,i-1,\,j+1/2}^{\,n} - E_{x\,i-1,\,j-1/2}^{\,n}\big)\Big] \\ & - \frac{c\tau}{\Delta x}\Big[b_y\big(E_{y\,i+1/2,\,j}^{\,n} - E_{y\,i-1/2,\,j}^{\,n}\big) \\ & + a_y\big(E_{y\,i+1/2,\,j+1}^{\,n} - E_{y\,i-1/2,\,j+1}^{\,n} + E_{y\,i-1/2,\,j-1}^{\,n} - E_{y\,i-1/2,\,j-1}^{\,n}\big)\Big], \end{aligned} \tag{76}$$

$$E_{x\,i,\,j+1/2}^{\,n+1} - E_{x\,i,\,j+1/2}^{\,n} = \frac{c\tau}{\Delta y}\big(B_{z\,i,\,j+1}^{\,n+1/2} - B_{z\,i,\,j}^{\,n+1/2}\big) - 4\pi\tau j_{x\,i,\,j+1/2}^{\,n+1/2}, \tag{77}$$

$$E_{y\,i+1/2,\,j}^{\,n+1} - E_{y\,i+1/2,\,j}^{\,n} = \frac{c\tau}{\Delta x}\big(B_{z\,i+1,\,j}^{\,n+1/2} - B_{z\,i,\,j}^{\,n+1/2}\big) - 4\pi\tau j_{y\,i+1/2,\,j}^{\,n+1/2}, \tag{78}$$

where the coefficients of the linear combination of the two schemes are

$$\begin{aligned} a_x &= a_y = 0.125\frac{\Delta x}{\Delta y}, \\ b_x &= 1 - 2a_x, \\ b_y &= 1 - 2a_y, \end{aligned} \tag{79}$$

and we have assumed that $\Delta x \le \Delta y$.

The dispersion relation for numerical scheme (76)–(78) is immediately found to be

$$\begin{aligned} \frac{1}{c^2\tau^2}\sin^2\frac{\omega\tau}{2} = {} & \frac{1}{\Delta x^2}\sin^2\frac{k_x\Delta x}{2}(b_y + 2a_y\cos k_y\Delta y) \\ & + \frac{1}{\Delta y^2}\sin^2\frac{k_y\Delta y}{2}(b_x + 2a_x\cos k_x\Delta x). \end{aligned} \tag{80}$$

It follows from (80) that the scheme is stable even at $c\tau = \Delta x$. This is quite a unique property for an explicit multi-dimensional finite difference scheme. In addition, the scheme (80) goes over to the usual Yee scheme in the limit $\Delta_x/\Delta_y \to 0$.

When used close to the stability limit, the scheme completely removes numerical dispersion along the X-axis (the laser propagation direction). For this reason we call this the NDF scheme, which stands for 'Numerical Dispersion- Free' [14]. The phase velocities of the numerical modes for the NDF scheme are plotted in Fig. 6(b). We mention that the region in which the numerical phase velocities are close to c becomes much wider than for the standard scheme; cf. Fig. 6(a).

The presence of the plasma changes the stability condition slightly, and the maximum τ is limited by the condition

$$1 - \frac{\tau}{\Delta x} > \frac{\omega_{\rm p}^2\tau^2}{4}, \tag{81}$$

where $\omega_{\rm p} = \sqrt{4\pi n_{\rm e}e^2/m_{\rm e}}$ is the maximum plasma frequency in the simulation domain. For an underdense plasma, however, this is an insignificant change.

The scheme (76)–(78) was written for the p-polarization in the 2D planar geometry. It must be slightly modified before it can be used in the full 3D space. The final version of the 3D NDF scheme is

$$\begin{aligned} B_{x_{i+1/2,\,j,\,k}}^{\,n+1/2} - B_{x_{i+1/2,\,j,\,k}}^{\,n-1/2} = &-\frac{c\tau}{\Delta y}\Big[b_z\big(E_{z_{i+1/2,\,j+1/2,\,k}}^{\,n} - E_{z_{i+1/2,\,j-1/2,\,k}}^{\,n}\big) \\ &+ a_z\big(E_{z_{i+1/2,\,j+1/2,\,k+1}}^{\,n} - E_{z_{i+1/2,\,j-1/2,\,k+1}}^{\,n} \\ &+ E_{z_{i+1/2,\,j+1/2,\,k-1}}^{\,n} - E_{z_{i+1/2,\,j-1/2,\,k-1}}^{\,n}\big)\Big] \\ &+ \frac{c\tau}{\Delta z}\Big[b_y\big(E_{y_{i+1/2,\,j,\,k+1/2}}^{\,n} - E_{y_{i+1/2,\,j,\,k-1/2}}^{\,n}\big) \\ &+ a_y\big(E_{y_{i+1/2,\,j+1,\,k+1/2}}^{\,n} - E_{y_{i+1/2,\,j+1,\,k-1/2}}^{\,n} \\ &+ E_{y_{i+1/2,\,j-1,\,k+1/2}}^{\,n} - E_{y_{i+1/2,\,j-1,\,k-1/2}}^{\,n}\big)\Big], \end{aligned} \tag{82}$$

$$\begin{aligned} B_{y_{i,\,j+1/2,\,k}}^{\,n+1/2} - B_{x_{i+1/2,\,j,\,k}}^{\,n-1/2} = &\frac{c\tau}{\Delta x}\Big[b_z\big(E_{z_{i+1/2,\,j+1/2,\,k}}^{\,n} - E_{z_{i-1/2,\,j+1/2,\,k}}^{\,n}\big) \\ &+ a_z\big(E_{z_{i+1/2,\,j+1/2,\,k+1}}^{\,n} - E_{z_{i-1/2,\,j+1/2,\,k+1}}^{\,n} \\ &+ E_{z_{i+1/2,\,j+1/2,\,k-1}}^{\,n} - E_{z_{i-1/2,\,j+1/2,\,k-1}}^{\,n}\big)\Big] \\ &- \frac{c\tau}{\Delta z}\Big[b_x\big(E_{x_{i,\,j+1/2,\,k+1/2}}^{\,n} - E_{x_{i,\,j+1/2,\,k-1/2}}^{\,n}\big) \\ &+ a_x\big(E_{x_{i+1,\,j+1/2,\,k+1/2}}^{\,n} - E_{x_{i+1,\,j+1/2,\,k-1/2}}^{\,n} \\ &+ E_{x_{i-1,\,j+1/2,\,k+1/2}}^{\,n} - E_{x_{i-1,\,j+1/2,\,k-1/2}}^{\,n}\big)\Big], \end{aligned} \tag{83}$$

$$\begin{aligned} B_{z_{i,\,j,\,k+1/2}}^{\,n+1/2} - B_{z_{i,\,j,\,k+1/2}}^{\,n-1/2} = &\frac{c\tau}{\Delta y}\Big[b_x\big(E_{x_{i,\,j+1/2,\,k+1/2}}^{\,n} - E_{x_{i,\,j-1/2,\,k+1/2}}^{\,n}\big) \\ &+ a_x\big(E_{x_{i+1,\,j+1/2,\,k+1/2}}^{\,n} - E_{x_{i+1,\,j-1/2,\,k+1/2}}^{\,n} \\ &+ E_{x_{i-1,\,j+1/2,\,k+1/2}}^{\,n} - E_{x_{i-1,\,j-1/2,\,k+1/2}}^{\,n}\big)\Big] \\ &- \frac{c\tau}{\Delta x}\Big[b_y\big(E_{y_{i+1/2,\,j,\,k+1/2}}^{\,n} - E_{y_{i-1/2,\,j,\,k+1/2}}^{\,n}\big) \\ &+ a_y\big(E_{y_{i+1/2,\,j+1,\,k+1/2}}^{\,n} - E_{y_{i-1/2,\,j+1,\,k+1/2}}^{\,n} \\ &+ E_{y_{i-1/2,\,j-1,\,k+1/2}}^{\,n} - E_{y_{i-1/2,\,j-1,\,k+1/2}}^{\,n}\big)\Big], \end{aligned} \tag{84}$$

$$\begin{aligned} E_{x_{i,\,j+1/2,\,k+1/2}}^{\,n+1} - E_{x_{i,\,j+1/2,\,k+1/2}}^{\,n} = &\frac{c\tau}{\Delta y}\Big[b_z\big(B_{z_{i,\,j+1,\,k+1/2}}^{\,n+1/2} - B_{z_{i,\,j,\,k+1/2}}^{\,n+1/2}\big) \\ &+ a_z\big(B_{z_{i,\,j+1,\,k+3/2}}^{\,n+1/2} - B_{z_{i,\,j,\,k+3/2}}^{\,n+1/2} \\ &+ B_{z_{i,\,j+1,\,k-1/2}}^{\,n+1/2} - B_{z_{i,\,j,\,k-1/2}}^{\,n+1/2}\big)\Big] \\ &- \frac{c\tau}{\Delta z}\Big[b_y\big(B_{y_{i,\,j+1/2,\,k+1}}^{\,n+1/2} - B_{y_{i,\,j+1/2,\,k}}^{\,n+1/2}\big) \\ &+ a_y\big(B_{y_{i,\,j+3/2,\,k+1}}^{\,n+1/2} - B_{y_{i,\,j+3/2,\,k}}^{\,n+1/2} \\ &+ B_{y_{i,\,j-1/2,\,k+1}}^{\,n+1/2} - B_{y_{i,\,j-1/2,\,k}}^{\,n+1/2}\big)\Big] \\ &- 4\pi\tau j_{x_{i,\,j+1/2,\,k+1/2}}^{\,n+1/2}, \end{aligned} \tag{85}$$

$$\begin{aligned} E_{y_{i+1/2,\,j,\,k+1/2}}^{\,n+1} - E_{y_{i+1/2,\,j,\,k+1/2}}^{\,n} = &-\frac{c\tau}{\Delta x}\Big[b_z\big(B_{z_{i+1,\,j,\,k+1/2}}^{\,n+1/2} - B_{z_{i,\,j,\,k+1/2}}^{\,n+1/2}\big) \\ &+ a_z\big(B_{z_{i+1,\,j,\,k+3/2}}^{\,n+1/2} - B_{z_{i,\,j,\,k+3/2}}^{\,n+1/2} \\ &+ B_{z_{i+1,\,j,\,k-1/2}}^{\,n+1/2} - B_{z_{i,\,j,\,k-1/2}}^{\,n+1/2}\big)\Big] \end{aligned}$$

$$
\begin{aligned}
&+ \frac{c\tau}{\Delta z}\Big[b_x\big(B_{x\,i+1/2,\,j,\,k+1}^{\,n+1/2} - B_{x\,i+1/2,\,j,\,k}^{\,n+1/2}\big) \\
&+ a_x\big(B_{x\,i+3/2,\,j,\,k+1}^{\,n+1/2} - B_{x\,i+3/2,\,j,\,k}^{\,n+1/2} \\
&+ B_{x\,i-1/2,\,j,\,k+1}^{\,n+1/2} - B_{x\,i-1/2,\,j,\,k}^{\,n+1/2}\big)\Big] \\
&- 4\pi\tau j_{y\,i+1/2,\,j,\,k+1/2}^{\,n+1/2}, \qquad (86)
\end{aligned}
$$

$$
\begin{aligned}
E_{z\,i+1/2,\,j+1/2,\,k}^{\,n+1} - E_{z\,i+1/2,\,j+1/2,\,k}^{\,n} = &\frac{c\tau}{\Delta x}\Big[b_y\big(B_{y\,i+1,\,j+1/2,\,k}^{\,n+1/2} - B_{y\,i,\,j+1/2,\,k}^{\,n+1/2}\big) \\
&+ a_y\big(B_{y\,i+1,\,j+3/2,\,k}^{\,n+1/2} - B_{y\,i,\,j+3/2,\,k}^{\,n+1/2} \\
&+ B_{y\,i+1,\,j-1/2,\,k}^{\,n+1/2} - B_{y\,i,\,j-1/2,\,k}^{\,n+1/2}\big)\Big] \\
&- \frac{c\tau}{\Delta y}\Big[b_x\big(B_{x\,i+1/2,\,j+1,\,k}^{\,n+1/2} - B_{x\,i+1/2,\,j,\,k}^{\,n+1/2}\big) \\
&+ a_x\big(B_{x\,i+3/2,\,j+1,\,k}^{\,n+1/2} - B_{x\,i+3/2,\,j,\,k}^{\,n+1/2} \\
&+ B_{x\,i-1/2,\,j+1,\,k}^{\,n+1/2} - B_{x\,i-1/2,\,j,\,k}^{\,n+1/2}\big)\Big] \\
&- 4\pi\tau j_{z\,i+1/2,\,j+1/2,\,k}^{\,n+1/2}, \qquad (87)
\end{aligned}
$$

where we are using the following expressions for the free parameters a_α and b_α:

$$
\begin{aligned}
a_x &= a_y + a_z, \\
a_y &= 0.125\frac{\Delta x}{\Delta y}, \\
a_z &= 0.125\frac{\Delta x}{\Delta z}, \qquad (88) \\
b_x &= 1 - 2a_x, \\
b_x &= 1 - 2a_y, \\
b_x &= 1 - 2a_z.
\end{aligned}
$$

Here we have chosen the coefficients (88) in such a way that the scheme is stable provided $\Delta x \le \Delta y, \Delta z$, and numerical dispersion is removed for waves running along the X-axis.

9 Lorentz boost

Plasma-based particle acceleration is a multi-scale problem, and the scales are very disparate; see Fig. 7. The smallest scale is the laser wavelength λ in the case of laser-driven acceleration, or the plasma wavelength $\lambda_{\rm p}$ for beam-driven plasma wakefields. The laser wavelength is on the order of micrometres, while the plasma wavelength can range from tens of micrometres to millimetres. The medium scale is the driver length; it can be comparable to the plasma wavelength in the bubble [16] and blow-out [15] regimes, or be much greater when we are relying on self-modulation in the plasma [19–22]. The largest scale is the acceleration length, which can range from centimetres to hundreds of metres or even kilometres [17]. It is the discrepancy between the driver scale and the acceleration distance that makes the simulations rather expensive.

One possible way to bring the scales together is to change the reference frame from the laboratory frame to a frame that is co-moving with the driver. This is called the Lorentz boost technique [23]. Let us assume that we transform from the laboratory frame L into a frame R moving in the propagation direction of the driver. The relative velocity of the R-frame is $V = \beta c$, and its relativistic factor is $\gamma = 1/\sqrt{1-\beta^2}$. Then the driver is Lorentz-stretched in the R-frame with factor $\gamma(1+\beta)$, and the propagation length is compressed by the same factor; see Fig. 7. Thus, potentially, the Lorentz transformation allows us to

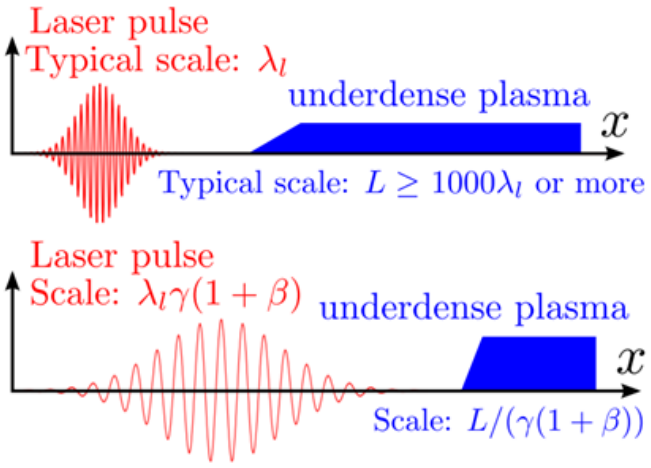

Fig. 7: Scale discrepancy in plasma-based acceleration; transformation to a co-moving frame reduces the discrepancy in scales.

increase the longitudinal grid step and time step—provided it is the grid step that limits the time step—and we have a smaller distance to propagate. The overall enhancement in performance could be huge.

Unfortunately, the background plasma becomes streaming in the R-frame with the same transformation velocity $-V$. The plasma density is higher in the R-frame than in the laboratory frame by a factor of γ. This leads to a source of free energy that can be converted into numerical plasma heating as the plasma particles interact with the numerical spatial grid. The associated numerical instability can have a considerable effect on the simulation quality [24]. The numerical noise generated by the unstable modes can completely mask the regular wake structure, as shown in the example in Fig. 8.

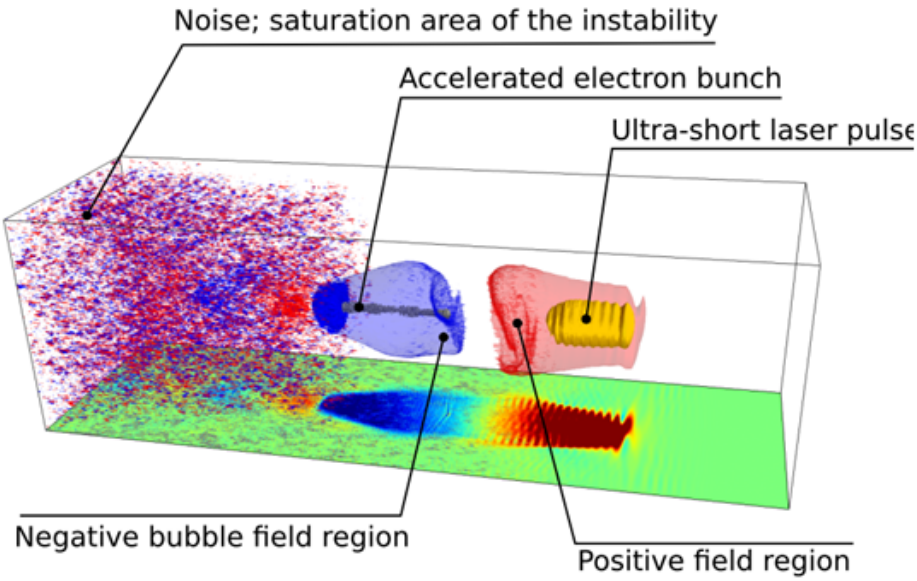

Fig. 8: Three-dimensional view of a numerical instability caused by plasma streaming in the co-moving frame

The main reason for the numerical instability is the Cerenkov resonance between the streaming plasma particles and the numerical electromagnetic modes on the grid. The numerical electromagnetic modes have subluminal phase velocities and can be in particle–wave resonance with the macroparticles

of the background plasma that stream through the grid with the relativistic factor γ. The mechanism of the numerical instability is illustrated in Fig. 9. Plasma fluctuations deflect particles from the straight-line trajectory; this leads to transverse currents. The transverse currents give rise to electromagnetic fields, and some of these electromagnetic modes propagate at exactly the particle velocity and can resonantly exchange energy with the particles. This is the Cerenkov mechanism.

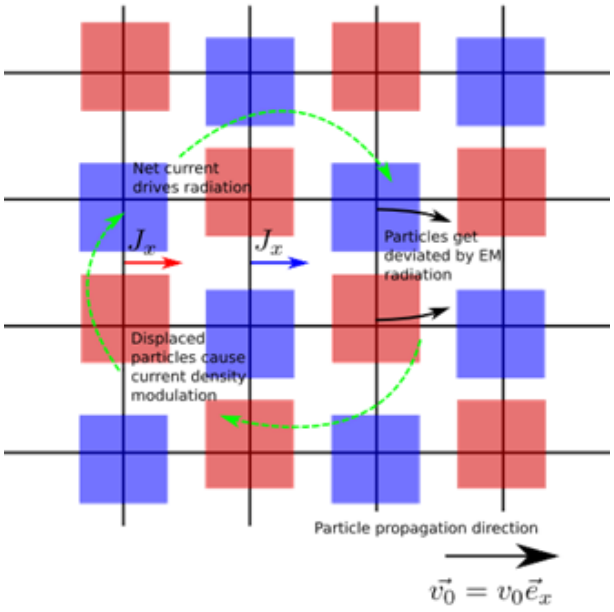

Fig. 9: The Cerenkov resonance mechanism underlying the numerical instability

One can solve the wave–particle dispersion relation for the standard Yee electromagnetic solver and calculate the growth rate of the instability analytically. A comparison of the observed electromagnetic modes in a PIC simulation and the analytical prediction is shown in Fig. 10. This tells us that, indeed, the reason behind the numerical instability is the Cerenkov resonance between relativistically moving plasma particles and numerical electromagnetic modes that have subluminal phase velocities on the Yee grid.

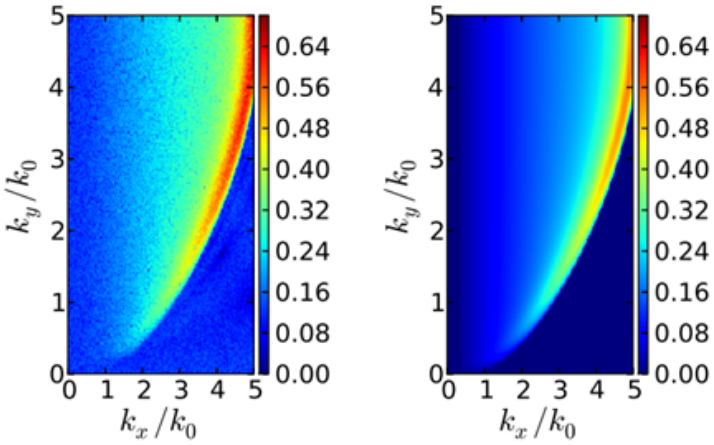

Fig. 10: Comparison of observed growing modes in a PIC simulation (left) and the analytically calculated growth rates of Cerenkov unstable modes (right).

One might hope to remove the instability by choosing a different Maxwell solver. For example, one could use a dispersion-free solver based on Fourier transformation. Indeed, a dispersion-free solver will reduce the instability growth rate, but unfortunately it is not able to eliminate the instability completely. The reason is the spatial and temporal aliasing on the grid; the relativistic particle starts to interact with the numerical modes from the other Brillouin zones. The non-resolved aliased frequencies of the numerical grid can be written as

$$\omega_{\text{eff}} = \pm \left(\sqrt{k_x^2 + k_y^2} - \frac{1}{\Delta t} \right), \tag{89}$$

where Δt is the time step. The resonance condition is then satisfied for electromagnetic waves with wavenumbers

$$k_y(k_x) = \frac{1}{h} \sqrt{h^2 k_x^2 \left(v_0^2 - 1\right) - 2hk_x v_0^2 + v_0^2}, \tag{90}$$

where h is the grid step. The analytical resonance curve and the growing modes observed in a numerical experiment are shown in Fig. 11.

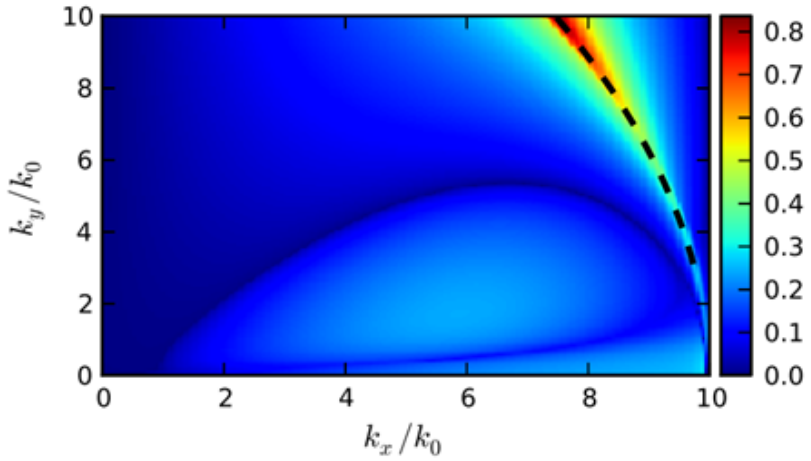

Fig. 11: Observed growing modes in a PIC simulation (colour scale) and the analytical resonance condition due to grid aliasing (dashed line).

The only viable way of taming the numerical instability in Lorentz-boosted simulations is to apply low-pass filters to the deposited currents before the electromagnetic fields are updated at each time step. The filtering reduces the instability to acceptable levels; yet a heavy filtering of the currents can in turn influence the dispersion of numerical modes, so one has to be very careful with this approach [25].

10 Quasi-static codes

Another way to bridge the gap between disparate scales in plasma-based acceleration is by using the quasi-static approximation. It separates explicitly the fast scale of the driver and the slow scale of acceleration [26]. To do so, we introduce new variables

$$\tau = t, \tag{91}$$

$$\zeta = z - ct. \tag{92}$$

We assume that the driver changes slowly as it passes a distance of its own length. Hence, as we are calculating plasma response to the driver, we neglect all derivatives with respect the slow time τ and advance from the front of the driver to the tail to calculate the wakefield configuration at a particular

time τ. After obtaining the fields and the plasma particle distribution, we can advance the driver with a large time step in τ. This procedure enhances the code's performance by many orders of magnitude. Simulations of large-scale plasma-based acceleration that require huge massively parallel computers when using the explicit PIC can be done on a desktop workstation in the quasi-static approximation. Of course, the quasi-static approximation is limited in that it does not describe radiation, only the static electromagnetic fields.

The first quasi-static PIC code, WAKE, was written by Mora and Antonsen [26]. It is a 2D code in cylindrical geometry and uses equations written in terms of the wake potential and the magnetic field. Later, a full 3D code, Quick-PIC, was developed that used the same equations [27]. Another 2D code in cylindrical geometry, called LCODE, has been developed by Lotov and uses equations on fields directly [28]. Here we describe the formalism used by the quasi-static version of the VLPL code. It is a full 3D code in Cartesian geometry. Like LCODE, it uses equations for the fields. Below we derive the quasi-static field equations.

We start again with the Maxwell equations (1)–(4) and write them in terms of the new variables (91) and (92), neglecting derivatives with respect to the slow time τ:

$$c\nabla \times \mathbf{B} = -c\frac{\partial \mathbf{E}}{\partial \zeta} + 4\pi \mathbf{j}, \tag{93}$$

$$c\frac{\partial \mathbf{B}}{\partial \zeta} = c\nabla \times \mathbf{E}, \tag{94}$$

$$\nabla \cdot \mathbf{E} = 4\pi\rho, \tag{95}$$

$$\nabla \cdot \mathbf{B} = 0. \tag{96}$$

First, we take the curl of the Ampère law (93) and the ζ-derivative of the Faraday law (94). Upon combining these two equations, we arrive at the first quasi-static equation on the magnetic field,

$$\nabla_\perp^2 \mathbf{B} = -\frac{4\pi}{c}\nabla \times \mathbf{j}, \tag{97}$$

where the $\nabla_\perp$ operator acts on coordinates transverse to the propagation direction.

Next, we take the gradient of the Poisson law (95). We use a well-known identity from vector analysis, $\nabla(\nabla \cdot \mathbf{E}) = \nabla^2 \mathbf{E} + \nabla \times \nabla \times \mathbf{E}$, and obtain for the transverse components of the electric field the equation

$$\nabla_\perp^2 \mathbf{E}_\perp = 4\pi \left(\nabla_\perp \rho - \frac{1}{c}\partial_\zeta \mathbf{j}_\perp \right). \tag{98}$$

For the longitudinal electric field component we obtain

$$\nabla_\perp^2 E_\parallel = 4\pi \frac{\partial}{\partial \zeta}\left(\rho - \frac{1}{c}\, j_\parallel \right) = \frac{4\pi}{c}\nabla_\perp \cdot \mathbf{j}_\perp, \tag{99}$$

where we have also used the continuity equation

$$\frac{\partial}{\partial \zeta}\rho = \frac{\partial}{\partial \zeta} j_\parallel + \nabla_\perp \cdot \mathbf{j}_\perp. \tag{100}$$

The continuity equation (100) can be used to remove the charge density ρ from the quasi-static equations and so work with the currents only. This can help to reduce the noise in PIC codes.

A typical quasi-static PIC code works as follows (see illustration in Fig. 12). First, the charge density and currents generated by the driver on the numerical grid are gathered. These are the sources that contribute to the basic equations (97)–(99). Then, a layer of numerical macroparticles is seeded at the front boundary (the head of the driver) of the simulation box. These numerical particles advance in the negative ζ direction (towards the tail of the driver) according to Eqs. (97)–(99). As the plasma

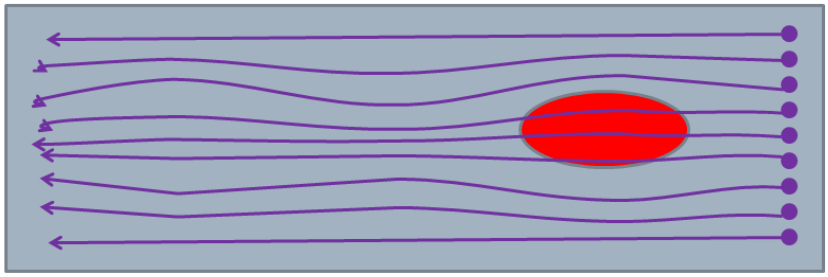

Fig. 12: Illustration of how the quasi-static PIC code is implemented

particles pass the whole simulation domain, the fields and density are defined on the grid and can be used to advance the driver in time τ.

If the driver is a charged particle beam, then we solve the equations of motion for beam particles in the calculated plasma fields. If the driver is a laser pulse, we have to solve an envelope equation on the laser pulse amplitude; this is required because the quasi-static equations do not describe radiation. Thus, an independent analytical model is required for the laser pulse. The laser pulse vector potential is represented as $\mathbf{A}(\tau,\zeta,\mathbf{r}) = \mathrm{Re}[\mathbf{a}(\tau,\zeta,\mathbf{r})\exp(ik\zeta)]$. The envelope equation for the complex amplitude $\mathbf{a}(\tau,\zeta,\mathbf{r})$ reads [26]

$$\left[\frac{2}{c}\frac{\partial}{\partial\tau}\left(ik_0+\frac{\partial}{\partial\zeta}\right)+\nabla_\perp^2\right]\mathbf{a}=\chi(\zeta,\mathbf{r}), \tag{101}$$

where $\chi(\zeta,\mathbf{r}) = \langle 4\pi q^2 n/(\gamma mc^2)\rangle$ is the plasma refraction averaged over all the particles in the cell with charge q, mass m and relativistic factor γ.

The laser pulse acts on the plasma particles via its ponderomotive force

$$F_\mathrm{p} = -\frac{q^2}{\gamma mc^2}\nabla a^2. \tag{102}$$

The ponderomotive force (102) is added to the standard Lorentz force in the particle pusher.

A typical time step of a quasi-static code is shown in Fig. 13. First, there is a cycle that proceeds along the fast variable ζ for the low-frequency (LF) fields. Then, the plasma refraction is calculated and the envelope equation for the high-frequency (HF) fields is updated.

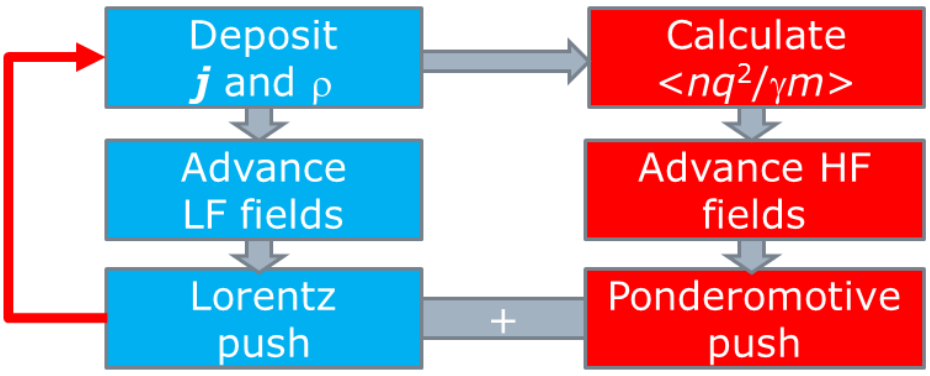

Fig. 13: Typical time step of a quasi-static code; the cycle for low-frequency (LF) plasma fields is followed by the cycle for the envelope equation of the high-frequency (HF) laser driver.

Fig. 14 shows a comparison of the quasi-static code (upper half of the simulation frame) and the full PIC code VLPL3D (lower half). We have simulated a blow-out generated by an overdense electron

bunch. The electron bunch density had the profile $n_{\rm b}(z,\mathbf{r}) = n_{\rm b0}\exp(-z^2/2\sigma_z^2)\exp(-r^2/2\sigma_r^2)$. The maximum bunch density was two times higher than the background plasma density: $n_{\rm b0} = 2n_{\rm p}$. The bunch was spherical with $k_{\rm p}\sigma_z = k_{\rm p}\sigma_r = 1$.

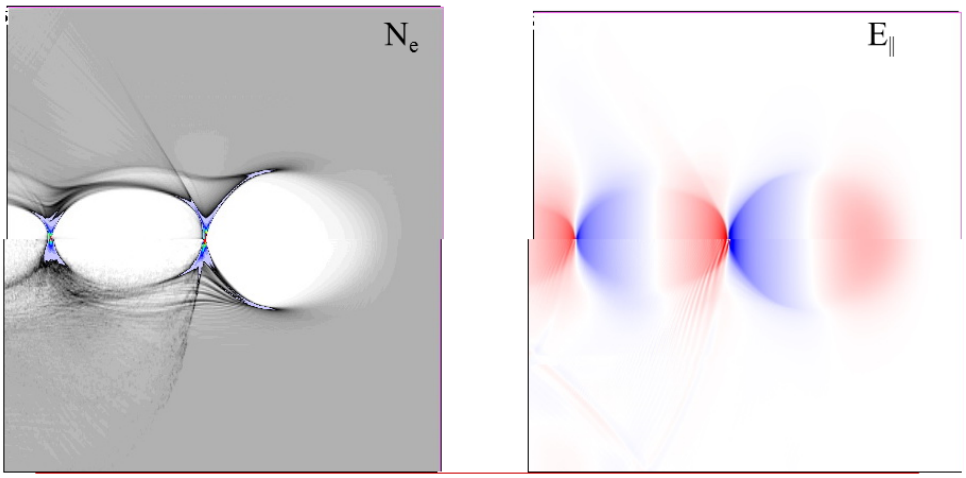

Fig. 14: Blow-out wakefield generated by an overdense electron bunch. The upper half of the frame is taken from a quasi-static simulation, and the lower part from a simulation using a full PIC code, VLPL3D. The left panel shows the plasma density, and the right panel shows the longitudinal electric field component of the wakefield. Cerenkov radiation emitted from the wavebreaking point is seen in the full 3D PIC simulation; this radiation is absent from the quasi-static code.

The two simulations are nearly identical. The only significant difference is the presence of Cerenkov radiation in the full 3D PIC simulation, emitted from the wavebreaking point at the tip of the tail of the first bubble. In the full simulation, we initialized the electron bunch vacuum in front of the plasma layer, so the plasma layer had to have a density ramp. The length of the bubble depends on the plasma density: the higher the density, the shorter the bubble. Consequently, the wavebreaking point moved with a superluminal velocity in the density ramp region and could emit Cerenkov radiation. In the quasi-static code, the Cerenkov radiation cannot be simulated. In addition, no density ramp is needed there: the field distribution is defined only by the local plasma density and by the instantaneous shape of the driver.

11 Computational costs of different codes

It is useful to estimate the number of operations required (called the computational cost) to simulate a particular plasma-based acceleration problem with different codes. The most general code, the full 3D PIC code, uses a 3D spatial grid of size $N_\parallel \times N_\perp^2$ cells and N_t time steps. The longitudinal step is limited by the laser wavelength, $h_\parallel \ll \lambda_0$, and the corresponding time step must be such that $\tau \ll \lambda_0/c$. The transverse grid steps are usually limited by the plasma wavelength, $h_\perp \ll \lambda_{\rm p}$. The number of time steps is $N_t = L_{\rm acc}/c\tau$, where $L_{\rm acc}$ is the acceleration distance. Hence, the number of operations required by the explicit PIC code scales as

$$N_{\rm op}^{\rm PIC} \propto \frac{L_{\rm acc} l_d}{\lambda_0^2} N_\perp^2. \tag{103}$$

Here we have assumed that the 'moving window' technique is used, so that the longitudinal size l_d of the simulation box scales with the plasma wavelength $\lambda_{\rm p}$.

If one uses the Lorentz boost technique with the transformation relativistic factor $\gamma_{\rm boost}$, the number of required operations reduces by a factor of $\gamma_{\rm boost}^2$ ideally:

$$N_{\rm op}^{\rm PIC\text{-}LB} \propto \gamma_{\rm boost}^{-2} \frac{L_{\rm acc} l_d}{\lambda_0^2} N_\perp^2 = \gamma_{\rm boost}^{-2} N_{\rm op}^{\rm PIC}. \tag{104}$$

The quasi-static approximation relaxes the time step restriction via the Courant condition. In addition, the grid step is no longer limited by the laser wavelength λ_0, but rather by the plasma wavelength λ_p. The time step must resolve the betatron oscillation of the beam particles, $\tau\omega_\beta \ll 1$, where the betatron frequency for a beam particle with mass M and relativistic factor γ_b is $\omega_\beta = \omega_\mathrm{p}\sqrt{m_\mathrm{e}/2\gamma_\mathrm{b}M}$. If the driver is a laser pulse, then the time step must resolve the diffraction length, $c\tau/Z_R \ll 1$, where the characteristic diffraction length of a laser pulse with focal spot R is defined by the Rayleigh length $Z_R = \pi R^2/\lambda_0$. The overall number of operations required by the quasi-static code scales as

$$N_\mathrm{op}^\mathrm{QS} \propto \frac{L_\mathrm{acc}\omega_\beta}{c}\frac{l_d}{\lambda_\mathrm{p}}N_\perp^2 = \frac{\lambda_0^2}{\lambda_\mathrm{p}\lambda_\beta}N_\mathrm{op}^\mathrm{PIC}, \tag{105}$$

where $\lambda_\beta = 2\pi c/\omega_\beta$ is the betatron wavelength. The performance gain, $\lambda_0^2/\lambda_p\lambda_\beta$, of the quasi-static code over the fully explicit PIC code can be huge and easily reach six orders of magnitude.

12 The future of PIC codes

Electromagnetic PIC codes provide a fundamental model for the dynamics of ideal plasma. Particularly in the relativistic regime of short-pulse laser–plasma interactions, these codes are unique in their predictive capabilities. In this regime, the binary collisions of plasma particles are either negligible or can be considered a small perturbation, and thus the electromagnetic PIC codes are the most appropriate tools.

However, the explicit PIC codes do have their limits. As soon as one tries to simulate laser interactions with highly overdense plasmas, these PIC codes become extremely expensive. Indeed, because the scheme is explicit, the code must resolve the plasma frequency and the skin depth. Even for uncompressed solid targets of high-Z materials, the plasma frequency can easily be 30 times higher than the laser frequency. The time and grid steps must be chosen accordingly. For a 3D code, simultaneous refinement of the grid and time steps in all dimensions by a factor of α leads to an increase in computational effort by a factor of α^4. For this reason, the simulation of a highly overdense plasma still poses a challenge for the explicit PIC codes.

A way around this difficulty might be to use implicit PIC codes, such the code LSP [29], or hybrid codes, such as a combination of a hydrodynamic description of the high-density background plasma and a PIC module for the hot electrons and ions [30]. Such codes alleviate the time step limitation, because they suppose the background plasma to be quasi-neutral and thus eliminate the fastest plasma oscillations at the Langmuir frequency. Very large plasma regions of high density can easily be simulated using such codes. Yet, these codes sacrifice a lot of the physics, and for any particular problem it must be checked whether the omitted physics is important or not. One possible way to perform this check is to benchmark the results of implicit codes against the direct PIC simulation on model problems which can be handled by both types of code.

Acknowledgements

This work was supported by the EU FP7 project EUCARD-2 and by BMBF, Germany.

References

[1] A. Pukhov, *Rep. Prog. Phys.* **66** (2001) 47. http://dx.doi.org/10.1088/0034-4885/66/1/202

[2] J. Villasenor and O. Buneman, *Comput. Phys. Commun.* **69** (1992) 306. http://dx.doi.org/10.1016/0010-4655(92)90169-Y

[3] J. Dawson, *Rev. Mod. Phys.* **55** (1983) 403. http://dx.doi.org/10.1103/RevModPhys.55.403

[4] C.K. Birdsall and A.B. Langdon, *Plasma Physics via Computer Simulations* (Adam Hilger, New York, 1991). http://dx.doi.org/10.1887/0750301171

[5] R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles* (McGraw-Hill, London, 1981).

[6] J.D. Jackson, *Classical Electrodynamics* (Wiley, New York, 1975).

[7] L. Landau and E. Lifshitz, *The Classical Theory of Fields*, 2nd ed. (Addison-Wesley, Reading, MA, 1962).

[8] N.A. Krall and A.W. Trivelpiece, *Principles of Plasma Physics* (McGraw-Hill, New York, 1973).

[9] S.I. Braginskii, Transport properties in a plasma, in *Reviews of Plasma Physics*, Ed. M.A. Leontovich (Consultants Bureau, New York, 1965).

[10] A.A. Vlasov, *Many-Particle Theory and Its Application to Plasma* (Gordon and Breach, New York, 1961).

[11] H. Ruhl and P. Mulser, *Phys. Lett.* **A205** (1995) 388; P. Bertrand, A. Ghizzo, T.W. Johnston, M. Shoucri, E. Fijalkov and M. R. Feix, *Phys. Fluids* **B5** (1990) 1028.

[12] R.C. Morse and C.W. Nielsen, *Phys. Fluids* **14** (1971) 830. http://dx.doi.org/10.1063/1.1693518

[13] R.H. Cohen, A. Friedman, D.P. Grote and J.L. Vay, *Nucl. Instrum. Methods Phys. Res.* **A606** (2008) 53. http://dx.doi.org/10.1016/j.nima.2009.03.083

[14] A. Pukhov, *J. Plasma Phys.* **61** (1999) 425. http://dx.doi.org/10.1017/S0022377899007515

[15] K. V. Lotov, *Phys. Rev.* E **69** (2004) 046405. http://dx.doi.org/10.1103/PhysRevE.69.046405

[16] A. Pukhov and J. Meyer-ter-Vehn, *Appl. Phys.* **B74** (2001) 355. http://dx.doi.org/10.1007/s003400200795

[17] E. Esarey, C.B. Schroeder and W.P. Leemans, *Rev. Mod. Phys.* **81** (2009) 1229. http://dx.doi.org/10.1103/RevModPhys.81.1229

[18] F.F. Chen, *Introduction to Plasma Physics and Controlled Fusion* (Plenum Press, New York, 1984). http://dx.doi.org/10.1007/978-1-4757-5595-4

[19] C. Joshi, T. Tajima, J.M. Dawson, H.A. Baldis and N.A. Ebrahim, *Phys. Rev. Lett.* **47** (1981) 1285. http://dx.doi.org/10.1103/PhysRevLett.47.1285

[20] N.E. Andreev, L.M. Gorbunov, V.I. Kirsanov, A.A. Pogosova and R.R. Ramazashvili, *Pis'ma Zh. Ehksp. Teor. Fiz.* **55** (1992) 551.

[21] T.M. Antonsen Jr. and P. Mora, *Phys. Rev. Lett.* **69** (1992) 2204. http://dx.doi.org/10.1103/PhysRevLett.69.2204

[22] A. Pukhov, N. Kumar, T. Tückmantel, A. Upadhyay, K. Lotov and P. Muggli, *Phys. Rev. Lett.* **107** (2011) 145003. http://dx.doi.org/10.1103/PhysRevLett.107.145003

[23] J.-L. Vay, *Phys. Rev. Lett.* **98** (2007) 130405. http://dx.doi.org/10.1103/PhysRevLett.98.130405

[24] J.-L. Vay, C.G.R. Geddes, E. Cormier-Michel and D.P. Grote, *J. Comput. Phys.* **230** (2011) 5908. http://dx.doi.org/10.1016/j.jcp.2011.04.003

[25] B.B. Godfrey and J.L. Vay, arXiv:1502.01387 (2015).

[26] P. Mora and T.M. Antonsen, *Phys. Plasmas* **4** (1997) 217. http://dx.doi.org/10.1063/1.872134

[27] C. Huang, V.K. Decyk, C. Ren, M. Zhou, W. Lu, W.B. Mori, J.H. Cooley, T.M. Antonsen Jr. and T. Katsouleas, *J. Comput. Phys.* **217** (2006) 658. http://dx.doi.org/10.1016/j.jcp.2006.01.039

[28] K.V. Lotov, *Phys. Rev. ST Accel. Beams* **6** (2003) 061301. http://dx.doi.org/10.1103/PhysRevSTAB.6.061301

[29] D.R. Welch, D.V. Rose, B.V. Oliver and R.E. Clark, *Nucl. Instrum. Methods Phys. Res.* **A464** (2001) 134. http://dx.doi.org/10.1016/S0168-9002(01)00024-9

[30] T. Tückmantel and A. Pukhov, *J. Comput. Phys.* **269** (2014) 168. http://dx.doi.org/10.1016/j.jcp.2014.03.019