Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 217 (2006) 658–679
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

QUICKPIC: A highly efficient particle-in-cell code for modeling wakefield acceleration in plasmas

C. Huang [a,*], V.K. Decyk [a], C. Ren [a,1], M. Zhou [a], W. Lu [a], W.B. Mori [a], J.H. Cooley [b], T.M. Antonsen Jr. [b], T. Katsouleas [c]

[a] *Departments of Electrical Engineering and Physics and Astronomy, University of California, Los Angeles, CA 90095, USA*
[b] *Departments of Physics and Electrical Engineering, University of Maryland, College Park, MD 20742, USA*
[c] *Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA*

Received 22 April 2005; received in revised form 11 January 2006; accepted 12 January 2006
Available online 15 March 2006

Abstract

A highly efficient, fully parallelized, fully relativistic, three-dimensional particle-in-cell model for simulating plasma and laser wakefield acceleration is described. The model is based on the quasi-static or frozen field approximation, which reduces a fully three-dimensional electromagnetic field solve and particle push to a two-dimensional field solve and particle push. This is done by calculating the plasma wake assuming that the drive beam and/or laser does not evolve during the time it takes for it to pass a plasma particle. The complete electromagnetic fields of the plasma wake and its associated index of refraction are then used to evolve the drive beam and/or laser using very large time steps. This algorithm reduces the computational time by 2–3 orders of magnitude. Comparison between the new algorithm and conventional fully explicit models (OSIRIS) is presented. The agreement is excellent for problems of interest. Direction for future work is also presented.

Keywords: Plasma accelerator; Quasi-static; PIC; Beam plasma interaction; Laser plasma interaction

1. Introduction and motivation

The subject of intense laser and beam plasma interactions is rich in nonlinear, relativistic, and ultra-fast physics and is part of the rapidly emerging field of high-energy density science. Interestingly, the focused intensities (or energy densities) of the state-of-the-art particle beams and laser pulses are both on the order of 10^{21} W/cm^2. At these intensities the self-fields of the beam and the fields of the laser are large enough to rapidly accelerate plasma electrons to relativistic energies and are high enough to tunnel ionize gas atoms.

* Corresponding author. Tel.: +1 310 825 6099; fax: +1 310 825 4057.
E-mail address: huangck@ee.ucla.edu (C. Huang).
[1] Present address: Department of Mechanical Engineering, University of Rochester, Rochester, NY 14627, USA.

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.01.039

Understanding the interplay among the nonlinear physical processes in these contexts is greatly aided by extensive numerical simulations.

One area in which numerical simulation has played a crucial role is that of plasma-based accelerators. There are four basic concepts being studied in plasma-based acceleration. These are the Laser Wakefield Accelerator (LWFA) [1,2], the Self-Modulated Laser Wakefield Accelerator (SMLWFA) [1,3], the Plasma Beat Wave Accelerator (PBWA) [1,4], and the Plasma Wakefield Accelerator (PWFA) [1,5]. These ideas are summarized pictorially in Fig. 1. In each of these schemes a driver, either a laser (LWFA, SMLWFA, PBWA) or a particle beam (PWFA), traverses a plasma creating a plasma wave wake. The phase velocity of the wake is roughly equal to the velocity of the driver in complete analogy with the water wave wake left behind a motor boat in a lake. The plasma wave wake has a longitudinal electric field so it can efficiently accelerate a trailing bunch of charged particles. Since the drivers are usually moving at nearly the speed of light (if one wanted to accelerate protons for example then the driver might have a lower phase velocity) then the phase velocity of the wake moves nearly at the speed of light, making it ideal to accelerate particles to relativistic energies. Plasma-based accelerators are of interest because the accelerating fields in the plasma wave structures can in principle be many orders of magnitude above current RF technology. Acceleration gradients in excess of 100 GeV/m have already been demonstrated.

In the past two years there has been tremendous progress in the field of plasma based accelerators. At the recent advanced accelerator concepts workshop [6], Joshi reported that 3 GeV energy gain over only ~10 cm has been observed in PWFA experiments at SLAC. In a recent issue in Nature, three independent groups reported that they observed mono-energetic, self-injected, beams of electrons with central energies ~100 MeV in LWFA experiments [7–9]. However, there are still many critical issues needed to be addressed before this work can be extended to 50 GeV or even to 1 TeV acceleration stages.

To model the full scale of a plasma-based accelerator, one needs a code (or codes) that can model the evolution of the driver, the generation and evolution of the wake, and the acceleration of the trailing bunch of particles. It turns out, perhaps not surprisingly, that in most cases to do this properly one needs particle based models. That is, one needs to follow the trajectories of particles in their self-consistent fields. The reasons for this are that in many cases the wake excitation process is highly nonlinear and results in nonlaminar particle trajectories, and that any reasonable beam loading scenario will require very tight spot sizes. These situations cannot be modeled using fluid descriptions.

The most straightforward particle based model is the fully explicit PIC algorithm [10]. In this algorithm, particles are loaded onto a spatially gridded simulation domain. The charge and current densities at the grid points can then be calculated by assigning the charge and current of nearby particles to the grid. These charge and current densities are used to advance the fields (also defined on the grid) via Maxwell's equations

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

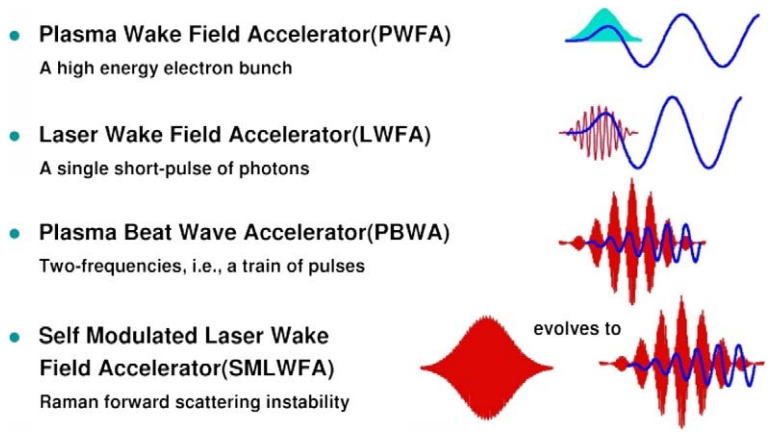

Fig. 1. Schematics of the four basic plasma-based accelerator schemes.

and

$$\nabla \times \mathbf{B} = \frac{4\pi}{c}\mathbf{J} + \frac{1}{c}\frac{\partial \mathbf{E}}{\partial t}.$$

The updated fields are used to advance the particles to new positions and velocities via the relativistic equation of motion

$$\frac{d\mathbf{P}}{dt} = q\left(\mathbf{E} + \frac{\mathbf{V} \times \mathbf{B}}{c}\right).$$

Although simple in concept, there are many subtle issues with solving these equations on a computer including the way in which charge and current are deposited on the grid and the way in which the equations of motion are integrated.

An incredible amount of progress has been made during the past 20 years using the full PIC algorithm [11]. However, because the algorithm makes few physics approximations it is also very CPU intensive. Using a full PIC code it takes $\sim 10^{13}$ particle pushes to model a single GeV PWFA stage (and $\sim 10^{14}$ to model a GeV LWFA stage). On today's fastest computers, such a simulation takes ~10,000 (100,000 for LWFA) CPU hours. Clearly, it is not possible to model 50 GeV or greater stages using the full PIC method. Furthermore, the use of the full PIC method is limited by the fact that the dispersion relation for an electromagnetic wave in a full PIC code is not ideal and therefore spurious Cerenkov radiations could develop in a simulation for a high γ beam moving almost at the speed of light.

Luckily, the physics of plasma-based acceleration allows one to make approximations that can in principle reduce the computational needs significantly and that are extremely accurate. In a plasma accelerator stage a short drive beam (either a laser pulse or particle beam) propagates through long regions of plasma and a trailing beam of electrons or positrons gets accelerated by the resulting wake. The driver and trailing beam evolve on a very different length scale than the plasma wake wavelength or the driver length. In a fully explicit code, one needs to choose a cell size that resolves the shortest length scale (either the laser wavelength or the plasma wavelength) and the time step is constrained by the Courant condition. For typical plasma accelerator parameters, the drive beam might not evolve for over 1000's of time steps. For example, for a PWFA the drive beam evolves on the scale of the betatron wavelength which is $(2\gamma)^{1/2}$ times longer than the plasma wavelength. For a 50 GeV beam this is a factor of ~500 times longer. For a LWFA the driver evolves on the Rayleigh length which is also orders of magnitude longer than the wavelength of the wake (for LWFA the shortest spatial scale is the laser wavelength so the potential CPU savings of reduced models over the fully explicit method can be considerably higher).

The quasi-static or frozen field approximation takes advantage of this disparity of scales and separates out the evolution of the driver from the plasma wake generation. Essentially, this approximation makes use of the fact that individual plasma electrons are passed over by the driver and its wake in a short time compared with the time over which the shape of the driver and wake evolve. Developing plasma based accelerator PIC codes using the quasi-static approximation was done independently by Mora and Antonsen [12] for laser drivers and by Whittum [13] for particle beam drivers. Mora and Antonsen's code, called WAKE, was confined to two dimensions and did not include the ability to model wake excitation from particle beams or to model beam loading. Whittum's code did not include the evolution of a laser field and made approximations to the wake field equations that are only appropriate to narrow driver beam bunches with moderate amounts of charge. Very recently, Lotov [14] reported on a 2D quasi-static code which is essentially identical to WAKE but for modeling PWFA.

In this paper, we describe in detail a new code, called QuickPIC, which makes the quasi-static approximation, but is fully three dimensional (in a sense that is to be defined later), is fully parallelized, puts no restrictions on the amount of beam charge, and can model both LWFA, PWFA and beam loading. We will also show that QuickPIC can completely reproduce the results from a full PIC code such as OSIRIS [15] with at least a savings of 100 in CPU time. (Furthermore, the quasi-static approximation does not suffer from unphysical Cerenkov radiation [10] that occurs in full PIC codes.) The development of QuickPIC is not a straightforward extension of the 2D algorithms of Mora and Antonsen and of Lotov or the approximate 3D model of Whittum. Complexities arise when the full quasi-static equations are parallelized and extended

to 3D. These include the need for an iterative solver with two transverse dimensions and the parallel routines for two types of distinct data structures, i.e., the driver (3D) and plasma particles (2D). The details of these issues and the solutions are presented in the following sections.

This paper is outlined as follows. In Section 2 we describe the quasi-static model. In particular, we describe the basic equations formulated using the Lorentz gauge. In Section 3, we describe how the algorithm was numerically implemented concentrating on a particle beam driver. In Section 4, we describe the numerical method used to advance the laser and to couple the laser to the plasma. In Section 5, the construction of the code is described including the details of the parallelization. In Section 6, we discuss the savings in computer needs as well as provide comparisons between QuickPIC and full PIC (OSIRIS) results for both PWFA and LWFA. In Section 7, we indicate where future improvements can be made and provide ideas for future applications of the code. Last, we summarize the advantages and limitations of QuickPIC and conclude in Section 8.

2. Quasi-static PIC equations in the Lorentz gauge

In this section, we describe the physical model that forms the basis of QuickPIC and the quasi-static approximation that QuickPIC employs. We begin by describing the model equations without a laser driver, then we will add the laser driver to the model. The implementation chosen for QuickPIC starts from the Maxwell equations in the Lorentz gauge (the code WAKE uses the transverse Coulomb gauge),

$$\left(\frac{1}{c^2}\frac{\partial^2}{\partial t^2}-\nabla^2\right)\phi(x,y,z,t)=4\pi\rho(x,y,z,t), \tag{1}$$

$$\left(\frac{1}{c^2}\frac{\partial^2}{\partial t^2}-\nabla^2\right)\mathbf{A}(x,y,z,t)=\frac{4\pi}{c}\mathbf{J}(x,y,z,t), \tag{2}$$

where ρ and $\mathbf{J}$ are the charge and current densities. Next we make a mathematical transform of the coordinates from (x,y,z,t) to (x,y,s,ξ), where $s=z$ (z is the direction in which the beam is moving), $\xi=ct-z$. Then $\partial_z=\partial_s-\partial_\xi$ and $c\partial_t=\partial_\xi$. Part of the quasi-static approximation amounts to assuming $\partial_s\ll\partial_\xi$ in Eqs. (1) and (2), therefore $\partial_z\approx-\partial_\xi$ and $c\partial_t=\partial_\xi$. Under these assumptions, the full set of quasi-static equations derived from Eqs. (1) and (2) for the wake fields can be written as

$$-\nabla_\perp^2\phi(x,y,s,\xi)=4\pi\rho(x,y,s,\xi), \tag{3}$$

and

$$-\nabla_\perp^2\mathbf{A}(x,y,s,\xi)=4\pi\mathbf{J}(x,y,s,\xi)/c. \tag{4}$$

Also, within the quasi-static approximation the Lorentz gauge condition can be written as

$$\nabla_\perp\cdot\mathbf{A}_\perp(x,y,s,\xi)=-\frac{\partial\psi(x,y,s,\xi)}{\partial\xi}, \tag{5}$$

where $\psi=\phi-A_z$ and A_z is the longitudinal component of vector potential.

Physically, the quasi-static assumption means that the driver does not change shape during the time it takes for it to pass by a plasma particle. While the variables ξ and s both have units of length, they correspond to fast and slow time scales. The variable ξ corresponds to the fast time scale associated with the duration of the driver and the time scale of the plasma wake that develops in response to the driver. The variable s measures the time scale over which the shape of the driver evolves. This is illustrated in Fig. 2. Eqs. (1) and (2) describe the excitation of both the wake fields and in the case of a laser pulse driver the laser pulse fields as well. The wake fields and laser fields require separate treatment, due to their different time dependences. The laser fields vary rapidly in time as it is assumed that the laser frequency is much greater than the plasma frequency. We will discuss the laser field after the discussion of the wake fields.

Separate equations of motion are written for particles traveling at the speed of the driver, which we refer to as beam particles, and for plasma particles, which are quickly passed over by the driver and wake. For beam particles which are moving at nearly the speed of light, $V_{bz}\cong c$, the variable s can be used to integrate the equations of motion and the resulting evolution equations for momentum become

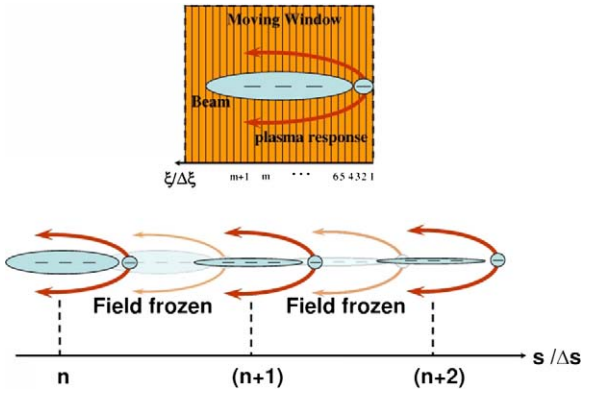

Fig. 2. The two variable s and ξ for the different time scales for drive beam and plasma evolution.

$$\frac{\mathrm{d}\mathbf{P}_{\mathrm{b}\perp}}{\mathrm{d}s} = -\frac{q_{\mathrm{b}}}{c}\nabla_{\perp}\psi, \tag{6a}$$

and

$$\frac{\mathrm{d}\mathbf{P}_{\mathrm{bz}}}{\mathrm{d}s} = \frac{q_{\mathrm{b}}}{c}\frac{\partial\psi}{\partial\xi}, \tag{6b}$$

where the subscript b denotes beam particles and q_{b} is the charge of a beam particle and $\mathbf{P}_{\mathrm{b}}$ is its momentum. In (6a) and (6b), the right-hand sides are derived assuming that beam particles are moving at nearly the speed of light, $V_{\mathrm{bz}} \cong c$ and hence $|V_{\mathrm{bz}}| \gg |\mathbf{V}_{b\perp}|$. The spatial trajectories of beam particles are given by

$$\frac{\mathrm{d}\mathbf{x}_{\mathrm{b}\perp}}{\mathrm{d}s} = \frac{\mathbf{P}_{\mathrm{b}\perp}}{\gamma m_{\mathrm{e}} c}, \tag{7a}$$

$$\frac{\mathrm{d}\xi_{\mathrm{b}}}{\mathrm{d}s} = 1 - \frac{\mathbf{P}_{\mathrm{bz}}}{\gamma m_{\mathrm{e}} c}. \tag{7b}$$

Beam charge and current density is then assigned to the three-dimensional grid in (x, y, ξ) in the same way as in a conventional PIC code.

For the plasma particles, for which s changes little, the evolution of the transverse momentum is followed in ξ rather than in s [12],

$$\frac{\mathrm{d}\mathbf{P}_{\mathrm{p}\perp}}{\mathrm{d}\xi} = \frac{q_{\mathrm{p}}}{c - V_{\mathrm{pz}}}\left[\mathbf{E}_{\perp} + \left(\frac{\mathbf{V}_{\mathrm{p}}}{c}\times\mathbf{B}\right)_{\perp}\right]. \tag{8}$$

Note that the right-hand side of Eq. (8) depends on s but that we will neglect this dependence. The transverse position is given by

$$\frac{\mathrm{d}\mathbf{x}_{\mathrm{p}\perp}}{\mathrm{d}\xi} = \frac{\mathbf{V}_{\mathrm{p}\perp}/c}{1 - V_{\mathrm{pz}}/c}, \tag{9}$$

where $\mathrm{d}\xi = (1 - V_{\mathrm{pz}}/c)\,\mathrm{d}t$ and the subscript p denotes plasma particles and $\mathbf{V}_{\mathrm{p}\perp} = \mathbf{P}_{\mathrm{p}\perp}/(\gamma_{\mathrm{p}} m_{\mathrm{e}})$, where

$$\gamma_{\mathrm{p}} = \left[1 + \frac{P_{\mathrm{p}}^2}{(m_{\mathrm{e}}c)^2}\right]^{1/2}$$

is the relativistic factor for a plasma particle. Note that Eq. (8) only gives the components of momentum transverse to the moving direction of the driver. The axial momentum of plasma particles can be obtained via the constant of the motion [12],

$$\gamma_{\mathrm{p}} - P_{\mathrm{pz}}/m_{\mathrm{e}}c = 1 - q_{\mathrm{p}}\psi/m_{\mathrm{e}}c^2, \tag{10}$$

giving

$$P_{pz}/(m_e c) = \frac{1 + P_{p\perp}^2/(m_e c)^2 - [1 - q_p\psi/(m_e c^2)]^2}{2[1 - q_p\psi/(m_e c^2)]}. \tag{11}$$

In principle, the plasma particle equations should also be augmented by a variable that measures the time of flight of a plasma particle as it is overtaken by the driver and wake,

$$\frac{ds_p}{d\xi} = \frac{1}{1 - V_{pz}/c}, \tag{12}$$

where s_p is the longitudinal position of the particular plasma particle.

This relation is used in computing charge and current densities; however, as it is assumed that the driver and wake depend weakly on this variable, this equation is not integrated.

As the driver sweeps over the plasma, plasma particles continuously follow the trajectories determined by Eqs. (8) and (9). Thus, to calculate the plasma contribution to the local charge and current density it is necessary to account for the amount of time a particle will spend in an interval of ξ. This time Δt is proportional to the factor $ds_p/d\xi$ just calculated above. The plasma particle charge and current density can then be accumulated on a grid by using the following deposition schemes for charge density and current,

$$\rho_p = \frac{1}{\text{Volume}} \sum_i \frac{q_{pi}}{1 - V_{pzi}/c}, \tag{13}$$

and

$$\mathbf{J}_p = \frac{1}{\text{Volume}} \sum_i \frac{q_{pi}\mathbf{V}_{pi}}{1 - V_{pzi}/c}. \tag{14}$$

Here the sum is over particles contributing to the charge and current densities at a given grid point. This sum must include weighting factors to distribute the particle charge to neighboring grid points.

The fact that a given simulation particle does not represent a fixed amount of charge can be viewed another way. The continuity equation for a collection of discrete particles each with charge Q_i is

$$0 = \frac{\partial}{\partial t} \sum_i Q_i \delta(\mathbf{x} - \mathbf{x}_i(t)) + \nabla \cdot \left[\sum_i Q_i V_i(t) \delta(\mathbf{x} - \mathbf{x}_i(t)) \right].$$

Under the quasi-static approximation where $\partial_s \ll \partial_\xi$ and $s_p(\xi)$ is the same for each particle, this reduces to

$$0 = \frac{\partial}{\partial \xi} \sum_i Q_i \left[1 - \frac{V_{zi}(\xi)}{c} \right] \delta(\mathbf{x}_\perp - \mathbf{x}_{\perp i}(\xi)) + \nabla_\perp \cdot \left[\sum_i Q_i \frac{V_{\perp i}(\xi)}{c} \delta(\mathbf{x}_\perp - \mathbf{x}_{\perp i}(\xi)) \right].$$

Therefore, at any value of ξ we can integrate along x and y to obtain

$$0 = \frac{d}{d\xi} \sum_i Q_i [1 - V_{zi}(\xi)/c].$$

So, when advancing plasma electrons forward in ξ, the quantity $Q_i(1 - \frac{V_{zi}}{c}) \equiv q_i$ (not Q_i) of each particle is constant. From which it follows that the charge on each particle is $Q_i \equiv \frac{q_i}{1 - V_{zi}/c}$, where q_i is a constant.

The above set of equations does not describe the evolution of laser pulse or its effect on particle motion. To include a laser pulse, the following modifications need to be made. First, an additional vector potential that generates the electric and magnetic fields of the laser pulse is added to Eq. (2). The laser pulse is described in terms of an envelope and a rapidly varying phase,

$$\frac{e\mathbf{A}}{m_e c^2} = Re[\hat{\mathbf{a}}(x, y, \xi, s) \exp(-ik_0\xi)], \tag{15}$$

where $k_0 = \omega_0/c$ is the central wave number of the laser pulse corresponding to a frequency ω_0 and the hat denotes a normalized quantity. The envelope satisfies an augmented paraxial wave equation that is derived from Eq. (2)

$$2\frac{\partial}{\partial s}\left(-ik_0+\frac{\partial}{\partial\xi}\right)\hat{\mathbf{a}}-\nabla_\perp^2\hat{\mathbf{a}}=\frac{4\pi e}{m_e c^3}\hat{\mathbf{J}}=k_0^2\chi_p\hat{\mathbf{a}}. \tag{16}$$

The mixed derivative term is needed to describe group velocity reduction due to the plasma, the direct forward Raman scattering and self-modulational instabilities [16]. The high frequency current is expressed in terms of a product of a susceptibility χ_p and the laser vector potential. The relative dielectric constant is thus, $\varepsilon=1+\chi_p$. The plasma contribution to the susceptibility is calculated using the following deposition scheme,

$$\chi_p=-\left\langle\frac{\omega_p^2}{\omega_0^2\gamma_p}\right\rangle=-\frac{4\pi e}{m_e\omega_0^2}\frac{1}{\text{Volume}}\sum\frac{q_p}{\bar{\gamma}_p(1-V_{pz}/c)}. \tag{17}$$

As shown by Mora and Antonsen [12] the laser period-averaged relativistic gamma factor is modified when the laser's vector potential is included,

$$\bar{\gamma}_p=[1+P_p^2/(m_e c)^2+|\hat{\mathbf{a}}|^2/2]^{1/2}.$$

The plasma particles are now influenced by the ponderomotive force of the laser, which is included in the equation of motion as

$$\frac{d\mathbf{P}_{p\perp}}{d\xi}=\frac{q_p}{c-V_{pz}}\left[\mathbf{E}_\perp+\left(\frac{\mathbf{V}_p}{c}\times\mathbf{B}\right)_\perp-\frac{m_e c^2}{\bar{\gamma}_p}\nabla_\perp\frac{|\hat{\mathbf{a}}|^2}{4}\right]. \tag{18}$$

We can now summarize the quasi-static approximation as follows. Physically, it is based on the fact that for short particle or laser beams the driver does not evolve (it is static) during the time it takes for it to pass by a plasma particle. Mathematically, we make this approximation by transforming from the variables from (x,y,z,t) to $(x,y,s\equiv z,\xi\equiv ct-z)$. In the field equations we assume $\partial_s\ll\partial_\xi$ which leads to $\frac{1}{c^2}\frac{\partial^2}{\partial t^2}-\frac{\partial^2}{\partial z^2}\cong\frac{\partial^2}{\partial\xi^2}$. This is reasonable so long as $k_\beta\ll\omega_p/c$, i.e., $\sqrt{2\gamma_b}\gg1$. For parameters of interest, $\gamma_b>2000$, this is trivially satisfied. We also assume that for each plasma particle $s_p(\xi)$ is the same. Therefore, we do not integrate Eq. (12) and on the right-hand side of Eq. (8) we assume that each plasma particle at a value of (x,y,ξ) feels the same fields. This can be quantified as $\frac{\partial}{\partial s}\mathbf{F}_\perp\times\int\frac{d\xi}{1-V_{pz}/c}\ll\mathbf{F}_\perp$.

This is reasonable so long as a particle is not trapped in the wake, i.e., V_{pz} does not approach $V_b\cong c(1-\frac{1}{2\gamma_b^2})$ or $V_g\cong c(1-\frac{\omega_p^2}{2\omega_0^2})$ for much of a particle's trajectory. Stated another way, the axial displacement of a particle should be much less than the bunch length during the transit time through the pulse. When including a laser driver, we also make the ponderomotive guiding center approximation in which we evolve the laser using the envelope approximation and we assume the forces of the plasma electrons are due to the time averaged ponderomotive force. As described in Appendix A in Ref. [12], the derivation of the constant of motion requires that $1-V_{pz}/c\gg\omega_p/\omega_0$. This condition comes from the requirement that a particle should not move much in the radial direction during the time it takes for one wavelength of the laser to pass by it. Since a single particle in vacuum is pushed forward with a velocity $\frac{V_{pz}}{c}=\frac{\hat{\mathbf{a}}^2/2}{(1+\hat{\mathbf{a}}^2/2)}$, this condition can be estimated as $|\hat{\mathbf{a}}|\ll\sqrt{2\frac{\omega_0}{\omega_p}}$ for large $|\hat{\mathbf{a}}|$.

However, developing an efficient code based on the above equations and approximations is not straightforward. While the system described is fully three dimensional, only the two transverse coordinates are truly space-like. The coordinates s and ξ can be thought of alternately as being time-like or space-like depending on the equation in which they appear. For example, if one considers the plasma particle equations of motion (Eqs. (8), (9) and (18)) and their associated wake fields (Eqs. (3) and (4)), ξ is a time-like variable and s is a parameter. The field equations (Eqs. (3) and (4)) are local in ξ; that is they can be solved in 2D space (x,y) for each value of ξ. The resulting system has the character of a 2D electrostatic model (albeit with both electrostatic and magnetostatic like interactions) with ξ playing the role of time. The difficulty is that both the field equation (Eq. (4)) and the equation of motion (Eq. (8)) imply a relation between $\mathbf{J}_\perp$ and $\mathbf{A}_\perp$ that must be satisfied for each ξ. For the field equation (Eq. (4)) the relation is clearly apparent, while for the particle equations one needs to consider that a portion of $\mathbf{E}_\perp$ is proportional to $\partial\mathbf{A}_\perp/\partial\xi$. If one were to sum over particles in a box, the equation of motion would relate $\partial\mathbf{J}_\perp/\partial\xi$ to $\partial\mathbf{A}_\perp/\partial\xi$. Thus, these two equations must be solved self-consistently. In the numerical implementation this is done by iteration. This situation does not arise in normal PIC codes where the particle equations and field equations can be advanced sequentially because they are time

centered. The problem can be traced to the cancellation of time and space derivatives in the field equations that occur in the drive beam frame, which makes the electromagnetic interaction local in ξ. This problem also does not occur in the equations of Whittum [13] in which among other things $\mathbf{A}_\perp$ is ignored. However, as the benchmarks to be presented in Section 6 will show, the assumptions of Whittum (neglect of $\mathbf{J}_\perp$ as well as J_z and V_z and relativistic mass effects) are not valid for parameters of interest. For the beam particles, ξ plays the role of a space-like variable and s plays the role of time. The beam electrons are loaded into a 3D grid in the coordinates (x, y, ξ) and are updated in time every time step Δs.

3. Implementation of particle and wake equations

In this section, we describe how the full set of quasi-static equations (Eqs. (3)–(14)) are solved numerically. For simplicity, we begin by assuming there is only a particle beam driver and leave a discussion of how to include the laser's ponderomotive force until the end.

We use the particle-in-cell (PIC) technique [10]. At a given value of s the beam particles are distributed throughout a three-dimensional (x, y, ξ) grid. The charge and current densities of the beam are deposited using standard area weighting (or higher order spline) methods. At each value of s, we then initialize a collection of plasma particles at an initial value of ξ sufficiently ahead of the driver. The x and y coordinates of each plasma particle are advanced forward in ξ (backward through the beam) using Eqs. (8) and (9) with electric and magnetic fields arising from the charges and currents from both the plasma and the beam. The key assumption is that the variable s_p is assumed to be the same for every plasma particle during the advance in ξ, i.e., we do not integrate s_p in ξ.

At a given ξ the fields are calculated using Eqs. (3)–(5) where ρ and $\mathbf{J}$ from the plasma are calculated using Eqs. (13) and (14) and area weighting (or higher order splines) methods, and where ρ and $\mathbf{J}$ from the beam have already been calculated (we do not use the transverse current $\mathbf{J}_\perp$ from the beam since it is small compared with the longitudinal beam current and it is not needed to satisfy the continuity equation under the quasi-static approximation). The value of V_z for each plasma particle is determined from Eqs. (10) and (11). Once the trajectories of the particles $(x(\xi), y(\xi))$ have been advanced forward in ξ a desired amount such that beam has passed them, then the beam particles are advanced in s. This cycle can be repeated a desired number of "time" steps in s. This flow is illustrated in Fig. 3. The structure of the algorithm therefore is that of a two-dimensional (x, y) PIC code with the ξ being a time-like variable, embedded in a three-dimensional (x, y, ξ) PIC code, with s being the time-like variable.

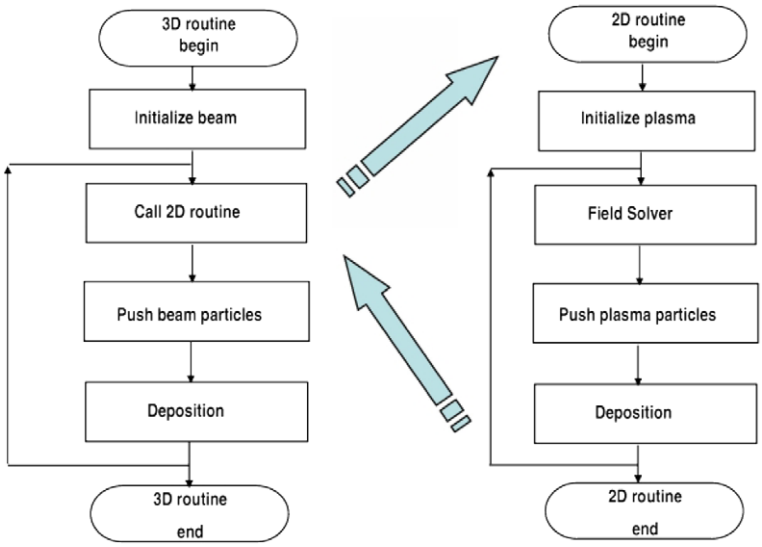

Fig. 3. Flow chart of the QuickPIC quasi-static algorithm showing a 2D routine embedded in a 3D routine.

We now describe the details of the numerics in the two-dimensional part. At a point in the two-dimensional loop where the electric and magnetic fields are known, plasma particles are updated with the Lorentz force. Although the equations of motion are fully relativistic, they can be cast into the form of a non-relativistic Boris pusher [17] with a modified electric field and an effective charge q_{eff} to simplify calculation.

$$\frac{\mathrm{d}\mathbf{u}_{\mathrm{p}\perp}}{\mathrm{d}\xi} = \frac{q_{\text{eff}}}{m_{\mathrm{e}}}\left[\gamma_{\mathrm{p}}\mathbf{E}_{\perp} + \left(\frac{\mathbf{u}_{\mathrm{p}}}{c}\times\mathbf{B}\right)_{\perp}\right] \tag{19}$$

and

$$\frac{\mathrm{d}\mathbf{x}_{\mathrm{p}\perp}}{\mathrm{d}\xi} = \frac{\mathbf{u}_{\mathrm{p}\perp}}{1 - q_{\mathrm{p}}\psi/m_{\mathrm{e}}c^2}, \tag{20}$$

where the definition of the effective charge is

$$q_{\text{eff}} \equiv \frac{q_{\mathrm{p}}}{1 - q_{\mathrm{p}}\psi/m_{\mathrm{e}}c^2}.$$

In deriving Eqs. (19) and (20), Eq. (10) is used to express the relativistic factor and we use the proper velocity $\mathbf{u} = \gamma\mathbf{V}$. Once the positions and velocities are updated, the charge and current densities can be deposited,

$$\rho_{\mathrm{p}} = \frac{1}{\text{Volume}}\sum_i \frac{q_{\mathrm{p}i}}{1 - V_{\mathrm{p}zi}/c} = \frac{1}{\text{Volume}}\sum_i \gamma_{\mathrm{p}i} q_{\text{eff}}, \tag{21}$$

$$\mathbf{J}_{\mathrm{p}} = \frac{1}{\text{Volume}}\sum_i \frac{q_{\mathrm{p}i}}{1 - V_{\mathrm{p}zi}/c}\mathbf{V}_{\mathrm{p}i} = \frac{1}{\text{Volume}}\sum_i q_{\text{eff}}\mathbf{u}_{\mathrm{p}i}, \tag{22}$$

where

$$\gamma_{\mathrm{p}i} = \frac{1 + \mathbf{u}_{\mathrm{p}i\perp}^2/c^2 + (1 - q_{\mathrm{p}i}\psi/m_{\mathrm{e}}c^2)^2}{2(1 - q_{\mathrm{p}i}\psi/m_{\mathrm{e}}c^2)}, \tag{23}$$

$$u_{\mathrm{p}iz} = \frac{1 + \mathbf{u}_{\mathrm{p}i\perp}^2/c^2 - (1 - q_{\mathrm{p}i}\psi/m_{\mathrm{e}}c^2)^2}{2(1 - q_{\mathrm{p}i}\psi/m_{\mathrm{e}}c^2)}. \tag{24}$$

The equation of motion (Eq. (19)) requires the evaluation of the wake electric and magnetic fields. These are given in terms of the scalar and vector potentials. The axial vector potential is determined by taking the difference between φ and ψ,

$$A_z = \varphi - \psi. \tag{25}$$

The electric field and magnetic field can then be found using the following equations:

$$\mathbf{E}_{\perp} = -\nabla_{\perp}\varphi - \mathbf{A}_{\perp\xi}, \tag{26}$$

$$E_z = \frac{\partial}{\partial\xi}\psi, \tag{27}$$

$$\mathbf{B}_{\perp} = (\mathbf{A}_{\perp\xi} + \nabla_{\perp}A_z)\times\hat{z}, \tag{28}$$

$$B_z = [\nabla_{\perp}\cdot(\mathbf{A}_{\perp}\times\hat{z})]\hat{z}, \tag{29}$$

where we define $\mathbf{A}_{\perp\xi}$ as

$$\mathbf{A}_{\perp\xi} \equiv \frac{\partial\mathbf{A}_{\perp}}{\partial\xi}.$$

The quantity $\mathbf{A}_{\perp\xi}$ satisfies a 2D Poisson equation with a source term $\mathbf{J}_{\perp\xi} \equiv \frac{\partial\mathbf{J}_{\perp}}{\partial\xi}$,

$$-\nabla_{\perp}^2\mathbf{A}_{\perp\xi} = 4\pi\mathbf{J}_{\perp\xi}/c. \tag{30}$$

As described in the previous section, this system contains two equations, Eqs. (19) and (30), relating $\mathbf{A}_{\perp\xi}$ and $\mathbf{J}_{\perp\xi}$ that need to be solved consistently. Furthermore, this set of equations is not time centered. For both of these reasons, a predictor–corrector is required when these equations are discretized in ξ and the predictor–corrector needs to be iterated for accuracy.

For the discrete version of Eqs. (19)–(30), quantities are either known at half or full integer grid values of ξ, i.e., $\xi = (m+\frac{1}{2})\Delta\xi$ or $\xi = m\Delta\xi$, where m is an integer labeling the grid values. We chose that at an integer grid value the particle proper velocities are known. Then it is natural to presume that the particle positions will be known on half integer grid values. To second order in $\Delta\xi$ the particle positions can be computed for the $(m+1/2)$ step by using Eq. (20) for a half time step $\Delta\xi/2$. However, the only fields that can be computed straightforwardly are $\mathbf{A}_\perp$ and hence $\nabla_\perp \cdot \mathbf{A}_\perp$ at index m by using Eqs. (3) and (4), and ψ which can be computed at the $(m+1)$ step by using the gauge condition (Eq. (5)).

Therefore, the need for some type of iteration method is clear. In order to compute the new proper velocity at $(m+1)$ the fields at the $(m+1/2)$ step are needed; but in order to compute the fields at the $(m+1/2)$ step the currents and charge density at $(m+1/2)$ and hence the proper velocity at $(m+1/2)$ are needed.

The iteration starts by predicting $\mathbf{J}_\perp^{m+1/2}$, $\mathbf{J}_{\perp\xi}^{m+1/2}$ and $\rho^{m+1/2}$, it continues by using these predictions to compute all of the fields at $(m+1/2)$, which are then used to advance the particles, and then the proper velocity at index $(m+1)$ is used to correct the prediction for $\mathbf{J}_\perp^{m+1/2}$, $\mathbf{J}_{\perp\xi}^{m+1/2}$ and $\rho^{m+1/2}$. This is summarized in Table 1 and the details are given next.

We assume that the derivative of a quantity in ξ is zero if no information about the derivative is known. The predictions for $\mathbf{J}_\perp^{m+1/2}$, $\mathbf{J}_{\perp\xi}^{m+1/2}$ are therefore,

$$\mathbf{J}_{\perp,l=0}^{m+1/2} = \mathbf{J}_\perp^m + (\Delta\xi/2)\cdot\mathbf{J}_{\perp\xi}^{m-1/2}, \tag{31}$$

$$\mathbf{J}_{\perp\xi,l=0}^{m+1/2} = \mathbf{J}_{\perp\xi}^{m+1/2}, \tag{32}$$

where l is an iteration index.

The prediction of $\rho^{m+1/2}$ is done as follows:

$$\rho^{m+1/2} = \rho_{\mathrm{b}}^{m+1/2} + \rho_{\mathrm{p}}^{m+1/2} \approx \left(\rho_{\mathrm{b}}^{m} + \rho_{\mathrm{b}}^{m+1}\right)/2 + \frac{1}{\text{Volume}}\sum_i \frac{\gamma_{\mathrm{p}i}^{m+1/2}\cdot q_{\mathrm{p}i}}{1 - q_{\mathrm{p}i}\cdot\psi^{m+1/2}/m_{\mathrm{e}}c^2}. \tag{33}$$

Note that for each value of s the beam quantities are known at all integer values of m so that to compute ρ_{b} at a half integer index we take the average. In Eq. (33), only $\gamma^{m+1/2}$ is unknown and needs to be predicted. The prediction is done using Eq. (23) and the previous value of $\mathbf{u}_{\mathrm{p}\perp}$,

$$\gamma_{\mathrm{p}i,l=0}^{m+1/2} = \frac{1 + (\mathbf{u}_{\mathrm{p}\perp}^m/c)^2 + (1 - q_{\mathrm{p}i}\psi^{m+1/2}/m_{\mathrm{e}}c^2)^2}{2(1 - q_{\mathrm{p}i}\psi^{m+1/2}/m_{\mathrm{e}}c^2)}. \tag{34}$$

The other field-related quantities such as ϕ, A_z, $\mathbf{A}_\perp$, $\mathbf{A}_{\perp\xi}$, $\nabla_\perp\cdot\mathbf{A}_\perp$, $\mathbf{B}_\perp$, B_z, $\mathbf{E}_\perp$ can now be solved using Eqs. (3) and (25)–(30) at index $(m+1/2)$ and ψ^{m+1} can also be obtained through the gauge condition, Eq. (5). Using the predicted forces, we advance the particles' proper velocities and positions to their next time index, with $\mathbf{x}_{\mathrm{p}\perp}$ at $(m+3/2)$, and $\mathbf{u}_{\mathrm{p}\perp}$ and $\mathbf{J}_\perp$ at $(m+1)$. The results $\mathbf{u}_{\mathrm{p}\perp}^{m+1}$ and $\mathbf{J}_\perp^{m+1}$ are then used to make a correction to the previous prediction,

$$\mathbf{J}_{\perp\xi,l}^{m+1/2} = (\mathbf{J}_{\perp,l-1}^{m+1} - \mathbf{J}_\perp^m)/\Delta\xi, \tag{35}$$

$$\mathbf{J}_{\perp,l}^{m+1/2} = \left(\mathbf{J}_\perp^m + \mathbf{J}_{\perp,l-1}^{m+1}\right)/2, \tag{36}$$

$$\gamma_{\mathrm{p}i,l}^{m+1/2} = \frac{1 + \left[(\mathbf{u}_{\mathrm{p}i\perp}^m + \mathbf{u}_{\mathrm{p}i\perp,l-1}^{m+1})/2c\right]^2 + (1 - q_{\mathrm{p}i}\psi^{m+1/2}/m_{\mathrm{e}}c^2)^2}{2(1 - q_{\mathrm{p}i}\psi^{m+1/2}/m_{\mathrm{e}}c^2)}. \tag{37}$$

Table 1
Quantities and their roles in the 2D cycle and the corresponding 2D time step at which they are defined

	$m-1/2$	m	$m+1/2$	$m+1$	$m+3/2$
Presumed quantities	ψ, $\mathbf{J}_{\perp\xi}$	$\mathbf{u}_{\mathrm{p}\perp}$, $\mathbf{J}_\perp$	$\mathbf{x}_{\mathrm{p}\perp}$		
Quantities calculated before iteration		$\mathbf{A}_\perp$, $\nabla_\perp\cdot\mathbf{A}_\perp$	ψ		
Predicted quantities			$\mathbf{J}_\perp$, $\mathbf{J}_{\perp\xi}$, γ, ρ		
Quantities known after iteration			φ, $A_\parallel$, $\mathbf{A}_\perp$, $\mathbf{A}_{\perp\xi}$, $\nabla_\perp\cdot\mathbf{A}_\perp$, $\mathbf{B}_\perp$, B_z, $\mathbf{E}_\perp$	$\psi, u_{\mathrm{p}\parallel}$, $\mathbf{u}_{\mathrm{p}\perp}$, $\mathbf{J}_\perp$	$\mathbf{x}_{\mathrm{p}\perp}$

The predictor–corrector loop can be repeated an arbitrary number of times. One way to terminate the loop is to wait until a desired accuracy is reached. However, the algorithm presented above suffers from one kind of numerical instability similar to one that occurs in Darwin codes [18,19]. The low $k_\perp$ modes of $\mathbf{A}_{\perp\xi}$ are the most unstable, they grow rapidly and soon dominate the whole electric and magnetic fields in a few iterations. The problem actually exists in the field solver. If one views the iteration number as a pseudo-time variable, the numerical instability arises from the instantaneous nature of the solution to the elliptical field equation in the pseudo-time. To remedy this, several methods including the moment method [18,20,21] and the canonical momentum method [18,22] have been proposed in the literature. However, incorporating these methods into QuickPIC would involve significant modifications and they are sometimes impractical. Instead, in QuickPIC we modified the Poisson solver in the iteration loop so that they resemble diffusion equations. Eqs. (30) and (3) are modified as follows, respectively:

$$D_J^{-1}\frac{\partial}{\partial T}\mathbf{A}_{\perp\xi} = \nabla_\perp^2\mathbf{A}_{\perp\xi} + 4\pi\mathbf{J}_{\perp\xi}/c, \tag{38}$$

$$D_\rho^{-1}\frac{\partial\phi}{\partial T} = \nabla_\perp^2\phi + 4\pi\rho, \tag{39}$$

where D_J^{-1} and D_ρ^{-1} are reciprocals of the diffusion coefficients. We define the pseudo-time as $T = l\Delta T$, then the above equations can be rewritten in the discrete variable l,

$$(1 - D_J\nabla_\perp^2)\mathbf{A}_{\perp\xi,l} = \mathbf{A}_{\perp\xi,l-1} + D_J\mathbf{J}_{\perp\xi,l}, \tag{40}$$

$$(1 - D_\rho\nabla_\perp^2)\phi_l = \phi_{l-1} + D_\rho\rho_l. \tag{41}$$

In the above equations, ΔT is absorbed into D_J and D_ρ so there is no need to consider it anymore. A diffusion equation is parabolic, thus any local error in the source term $\mathbf{J}_{\perp\xi}$ or ρ cannot propagate across the simulation box instantly in one iteration. Local errors are indeed damped over the pseudo-time and the calculation can converge to the correct solution in a few iterations if D_J and D_ρ are chosen appropriately. For diffusion equations like Eqs. (40) and (41), the characteristic diffusion length is $L_D = \sqrt{4Dl}$ where we have chosen $D_J = D_\rho = D$ for simplicity. Because normally a particle will not move more than one transverse grid size Δx during one real time step $\Delta\xi$, the scale length of the changes in $\mathbf{J}_{\perp\xi}$ and ρ during this time will be on the order of Δx. Thus, we should allow diffusion over this distance to happen in the whole iteration process, i.e., $L_{D,\max} = \sqrt{4Dl_{\max}} = \Delta x$, which gives an estimate for the choice of D, which is $D = (\Delta x)^2/4l_{\max}$. While it is true that a predictor–corrector is essentially second order accurate the coefficient in front of the error can be reduced by using more iterations, i.e., increasing $l_{\max}$. For problems of interest, a full PIC code can also be used to obtain the "correct" answer for the same cell size. Through experimentation, it is then possible to determine how many iterations are needed to obtain the desired accuracy for a fixed longitudinal cell size (effective time step). Alternatively, the accuracy could be improved by using a smaller $\Delta\xi$ (time step). On the other hand, since numerous calculations are involved in an iteration loop, it is desirable to have as few iterations as possible. In practice, we found that $l_{\max} = 2$ and $D = (\Delta x)^2$ gives the most rapid convergence and the best accuracy over a wide range of parameters. These choices are used for the QuickPIC runs that are presented in Section 6 (except that 4 iterations are used in the laser driver benchmark). An area for future work is examining the trade-offs in accuracy, speed, and memory between reducing $\Delta\xi$ and using fewer iterations vs. increasing $\Delta\xi$ and using more iterations.

In the present implementation of QuickPIC, all transverse spatial derivatives are done in Fourier space using fully parallelized FFTs. In principle we could use k dependent diffusion coefficients but we have not investigated this in detail. Eqs. (38) and (39) are integrated implicitly, in the future it might be worth trying a Crank–Nicholson approach.

As ξ is incremented from the front to the end of the moving window, the plasma response and all the fields are solved for and stored at each transverse 2D slice using the above numerical algorithm. Then the drive beam, which exists in 3D space, should be pushed using these fields for a large time step Δs. The beam momenta are known at half integer steps in s and the beam positions are known at full integer steps. When the 2D loop finishes it returns the necessary fields to the 3D loop to update the beam at a full integer value of s. This update uses the standard leap-frog algorithm and is therefore time centered with second order accuracy in Δs. The step Δs only needs to resolve the betatron motion of the beam particles. The equations used are Eqs. (6a)–(7b) with proper normalization. Then the charge density is deposited and the 2D cycle is started again for

the updated beam driver. The 3D beam update and the charge deposition were taken directly from the UPIC framework to be described shortly. The 3D loop does not require any field-solves so it typically uses a small fraction of the total computation time.

If there is a laser driver, the above algorithm only needs to be modified in three places. In the momentum advance, the ponderomotive force needs to be included in Eq. (19)

$$\frac{\mathrm{d}\mathbf{u}_{\mathrm{p}i\perp}}{\mathrm{d}\xi} = \frac{q_{\mathrm{eff}}}{m_{\mathrm{e}}}\left[\gamma_{\mathrm{p}i}\mathbf{E}_{\perp} - \frac{m_{\mathrm{e}}c^2}{q_{\mathrm{p}i}}\nabla_{\perp}\frac{|\hat{\mathbf{a}}|^2}{4} + \left(\frac{\mathbf{u}_{\mathrm{p}i}}{c}\times\mathbf{B}\right)_{\perp}\right] \tag{42}$$

so that the effective electric field in the non-relativistic Boris pusher is now $\gamma_{\mathrm{p}i}\mathbf{E}_{\perp} - \frac{m_{\mathrm{e}}c^2}{q_{\mathrm{p}i}}\nabla_{\perp}\frac{|\hat{\mathbf{a}}|^2}{4}$.

In addition, the expression for $\gamma_{\mathrm{p}i}$ is now

$$\gamma_{\mathrm{p}i} = \frac{1 + \mathbf{u}_{\mathrm{p}i\perp}^2/c^2 + |\hat{\mathbf{a}}|^2/2 + (1 - q_{\mathrm{p}i}\psi/m_{\mathrm{e}}c^2)^2}{2(1 - q_{\mathrm{p}i}\psi/m_{\mathrm{e}}c^2)}. \tag{43}$$

Values of the ponderomotive potential on the half integer grid values of x are calculated according to

$$[|\hat{\mathbf{a}}|^2/2]^{m+1/2} = \frac{1}{2}\{[|\hat{\mathbf{a}}|^2/2]^{m+1} + [|\hat{\mathbf{a}}|^2/2]^{m}\}.$$

Last, we need to deposit the new quantity χ_{p}, the plasma susceptibility,

$$\chi_{\mathrm{p}} = -\frac{4\pi}{m_{\mathrm{e}}\omega_0^2\cdot\mathrm{Volume}}\sum_i \frac{q_{\mathrm{p}i}^2}{(1 - V_{\mathrm{p}zi}/c)\gamma_{\mathrm{p}i}} = -\frac{4\pi}{m_{\mathrm{e}}\omega_0^2\cdot\mathrm{Volume}}\sum_i \frac{q_{\mathrm{p}i}^2}{1 - q_{\mathrm{p}i}\psi/m_{\mathrm{e}}c^2}. \tag{44}$$

4. Implementation of laser pulse propagation equations

The propagation of the laser pulse is described by Eq. (16) for the evolution of the envelope $\hat{\mathbf{a}}$ of the normalized vector potential. This equation applies to the case of pulse propagation in a fully ionized plasma; that is, it is assumed that there is no other matter in the form of neutral or partially ionized gas present that would modify the dispersion relation for the laser light. To make the equations more general we imagine that there is also a tenuous background medium present, characterized by a small correction to the dielectric constant, $\delta\varepsilon(\omega)$, that modifies the group velocity and adds dispersion. In this case Eq. (16) becomes

$$2\frac{\partial}{\partial s}\left(-\mathrm{i}k_0 + \frac{\partial}{\partial\xi}\right)\hat{\mathbf{a}} - \mathrm{i}k_0\beta_1\frac{\partial}{\partial\xi}\hat{\mathbf{a}} + \beta_2\frac{\partial^2}{\partial\xi^2}\hat{\mathbf{a}} - \nabla_{\perp}^2\hat{\mathbf{a}} = k_0^2\chi_{\mathrm{T}}\hat{\mathbf{a}}, \tag{45}$$

where $\beta_1 = 2(v_{\mathrm{f}} - v_{\mathrm{g}})/c$, v_{f} is the frame velocity $\xi = v_{\mathrm{f}}t - z$ (now assumed to be different from the speed of light) and

$$v_{\mathrm{g}} \cong c\left[1 - \frac{1}{2}\frac{\mathrm{d}(\omega\delta\varepsilon)}{\mathrm{d}\omega}\right]_{\omega_0}$$

is the group velocity in the background medium in the limit in which the medium is tenuous. The coefficient of dispersion is given by

$$\beta_2 = \frac{v_{\mathrm{f}}^2 - v_{\mathrm{g}}^2}{c^2} + \omega\frac{\mathrm{d}}{\mathrm{d}\omega}\left(\frac{v_{\mathrm{g}}}{c}\right)\Bigg|_{\omega_0}.$$

The quantity $\chi_{\mathrm{T}} = \chi_{\mathrm{p}} + 2n_2 I$ represents the modified susceptibility including the plasma contribution and the nonlinear susceptibility of the background medium. Here I is the local intensity and n_2 the second order coefficient of nonlinearity in the susceptibility of the background medium. The linear portions of the background susceptibility are described by $\delta\varepsilon$ and contribute to the coefficients β_1 and β_2. The coefficient β_1 can be set to zero by picking the frame velocity to equal the group velocity of the background medium. If no background medium is present, then the frame velocity should equal the speed of light. The plasma reduction of the group velocity is captured in Eq. (45) due to the interplay of the mixed, second order s–ξ derivative and the plasma

contribution to the total susceptibility. Likewise, the reduction of the group velocity for waves propagating obliquely to the z-axis is captured by the interplay of the mixed derivative and the transverse Laplacian. Retention of the mixed derivative term also leads to separate conservation laws for wave energy and wave action [12]. This allows for a correct treatment of pulse depletion due to the excitation of plasma waves.

We wish to find a stable numerical approach to solve Eq. (45) in a fully three-dimensional simulation. In addition to the computational cost associated with solving a three-dimensional non-linear partial differential equation, we have the added complication of performing this calculation on a parallel computer system. To this end, we include the constraint that our computational algorithms have an efficient parallel implementation. We note that the wave operator involves differentiation with respect to all three space-like coordinates. This will make finding an s-centered fully implicit numerical implementation of Eq. (45) difficult. Instead we will use a split step algorithm [23] in which Eq. (45) is separated into two operators each of which can be advanced in an s-centered implicit way. Specifically, we separate Eq. (45) into separate equations involving the two operators L_0 and L_1 so that each equation can be solved with the same tri-diagonal matrix solver.

$$2\frac{\partial}{\partial s}\left(-\mathrm{i}k_0+\frac{\partial}{\partial\xi}\right)\hat{\mathbf{a}}=L_0(\hat{\mathbf{a}})\equiv \mathrm{i}\frac{k_0\beta_1}{2}\frac{\partial}{\partial\xi}\hat{\mathbf{a}}-\frac{\beta_2}{2}\frac{\partial^2}{\partial\xi^2}\hat{\mathbf{a}}+k_0^2\chi_{\mathrm{T}}(\mathbf{x}_\perp,\xi,s)\hat{\mathbf{a}} \tag{46}$$

and

$$2\frac{\partial}{\partial s}\left(-\mathrm{i}k_0+\frac{\partial}{\partial\xi}\right)\hat{\mathbf{a}}=L_1(\hat{\mathbf{a}})\equiv \mathrm{i}\frac{k_0\beta_1}{2}\frac{\partial}{\partial\xi}\hat{\mathbf{a}}-\frac{\beta_2}{2}\frac{\partial^2}{\partial\xi^2}\hat{\mathbf{a}}+\nabla_\perp^2\hat{\mathbf{a}}. \tag{47}$$

Here we have assumed that the coefficients β_1 and β_2 are independent of transverse coordinate and we have included half of their effect in each operator. If these coefficients were spatially varying we would include their full effect in Eq. (46) and eliminate them from Eq. (47). The separation is made so that Eq. (46) can be solved in ξ with $\boldsymbol{x}_\perp$ as a parameter, and Eq. (47) can also be solved in ξ by Fourier transforming in $\boldsymbol{x}_\perp$ and treating $\boldsymbol{k}_\perp$ as a parameter. It is necessary to include the mixed derivative on the left-hand side in both steps so that the interplay between the mixed derivative and the susceptibility in the case of Eq. (46) and the mixed derivative and the transverse Laplacian in the case of Eq. (47) that gives rise to the reduction of the group velocity due to plasma and oblique propagation is recovered.

The sequence in which Eqs. (46) and (47) are solved is illustrated in Fig. 4. The top portion of this figure illustrates the part of the code that solves for the laser field, while the bottom portion illustrates the part that evolves the particles and calculates the plasma wake as described in the previous section. The communication between the two parts of the code is illustrated by the arrows connecting the upper and lower portions of the figure. The laser propagation part of the code must supply the normalized vector potential (actually, its magnitude squared $|\hat{\mathbf{a}}|^2$) evaluated on the s-grid, $s_n=n\Delta s$, to the particle part; while the particle part must supply the susceptibility χ_{T} on the s-grid to the laser propagation part of the code. Given that the susceptibility χ_{T} is known on the s-grid, Eq. (46) is centered about $s_n=n\Delta s$. The finite difference in s version of Eq. (46) is thus written:

$$2\left(-\mathrm{i}k_0+\frac{\partial}{\partial\xi}\right)\frac{\tilde{\mathbf{a}}_1^{n+1/2}-\tilde{\mathbf{a}}^{n-1/2}}{\Delta s}=\frac{1}{2}L_0(\tilde{\mathbf{a}}_1^{n+1/2}+\tilde{\mathbf{a}}^{n-1/2}), \tag{48}$$

where $\tilde{\mathbf{a}}^{n-1/2}$ and $\tilde{\mathbf{a}}_1^{n+1/2}$ are intermediate values of the normalized vector potential that are illustrated in Fig. 4. The operater L_0 contains the time and spatially dependent susceptibility evaluated at s_n. This depends on the vector potential at this same time, and this will have to be determined from $\tilde{\mathbf{a}}^{n-1/2}$. We will discuss this point subsequently. In between applications of operator L_0 we must apply operator L_1, which includes the transverse derivatives. This will connect values of the intermediate quantities $\tilde{\mathbf{a}}^{n+1/2}$ and $\tilde{\mathbf{a}}_1^{n+1/2}$ as indicated in Fig. 4,

$$2\left(-\mathrm{i}k_0+\frac{\partial}{\partial\xi}\right)\frac{\tilde{\mathbf{a}}^{n+1/2}-\tilde{\mathbf{a}}_1^{n+1/2}}{\Delta s}=\frac{1}{2}L_1(\tilde{\mathbf{a}}^{n+1/2}+\tilde{\mathbf{a}}_1^{n+1/2}). \tag{49}$$

In the operator L_1 the coefficients are assumed to be independent of $\boldsymbol{x}_\perp$. Thus Eq. (49) is solved in Fourier space, and periodic boundary conditions in the two transverse directions are applied.

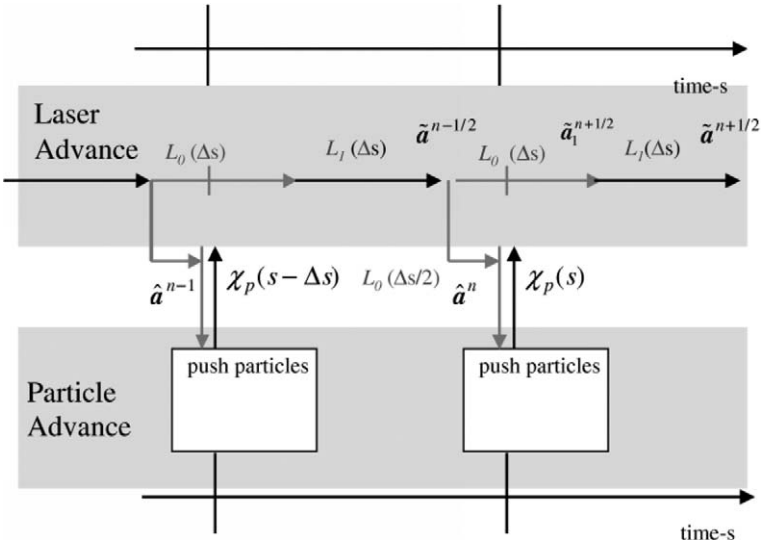

Fig. 4. Schematic representation of second order accurate split step algorithm for advancing the laser field in s. Also shown is the communication between the laser propagation part of the code and the particle and wake part of the code.

Both operators L_0 and L_1 are second order with respect to differentiation by ξ. These operators are then represented as second order finite difference equations. Both equations are thus equivalent to tridiagonal matrices, which are solved by the dual sweep algorithm [24]. The issue of boundary conditions is complicated by the addition of the dispersion coefficient β_2. Dispersion allows that information can propagate both faster and slower than the frame velocity. Thus, information can propagate to both large and small values of ξ. Rather than attempt to impose an absorbing boundary condition at the minimum and maximum values of ξ, we simply require that the vector potential vanish at these points and further require that the simulation domain be large enough such that no reflection of waves from these points can occur. We note that in the case in which dispersion is absent, then information can only propagate in the direction of increasing ξ. In this case the required boundary condition is that all values of $\hat{\mathbf{a}}(\xi)$ for ξ less then the minimum value in the simulation domain must vanish.

The split step algorithm just described gives a second order in Δs accurate solution for the wave equation provided a second order accurate value of the susceptibility is in the operator L_0 in Eq. (48). The required susceptibility is evaluated on the s-grid, while the intermediate quantities determined in Eqs. (48) and (49) are known on the half grid. To evaluate the vector potential at the required time we apply a half time step advancement of the operator L_0 to generate $\hat{\mathbf{a}}^n$ on the grid $n\Delta s$ starting from the intermediate value $\tilde{\mathbf{a}}^{n-1/2}$,

$$2\left(-\mathrm{i}k_0+\frac{\partial}{\partial\xi}\right)\frac{\hat{\mathbf{a}}^n-\tilde{\mathbf{a}}^{n-1/2}}{\Delta s/2}=\frac{1}{2}L_0(\hat{\mathbf{a}}^n+\tilde{\mathbf{a}}^{n-1/2}). \tag{50}$$

Since $\tilde{\mathbf{a}}^{n-1/2}$ is already second order accurate, and we are only advancing s by a fraction of a grid spacing we can use a first order evaluation of the susceptibility in L_0 for this step. Typically, we extrapolate the susceptibility from the two previous s-steps for this estimate. Thus, to advance the laser field, only one call to the part of the code that advances the particles is needed per s-step.

Eqs. (48) and (49) involve only finite differencing in the axial grid, thus each transverse location or wavenumber can be solved independently. This separation motivates the parallel implementation used to solve this system. The normal layout of the field involves each node having all axial data and all the data for one transverse coordinate. The second transverse coordinate is divided among the different processors. This layout allows the evolution of Eqs. (48) and (49) to be local to a processor. Unfortunately this layout requires two large communications to perform the transverse FFT necessary to evolve Eq. (49). We take an FFT in the coordinate that is local to the processors then redistribute the data so the other transverse coordinate is local. We can then perform the second FFT in this direction and then solve Eq. (49). Finally we reverse the process leaving the data in the original layout.

5. Construction of code and parallelization

The QuickPIC code is built from the UCLA parallel particle-in-cell (UPIC) framework [25] and a separate laser solver module described in the previous section. This framework provides trusted components for the rapid construction of new, parallel particle-in-cell (PIC) codes, using object-oriented ideas. It is designed in layers. The lowest layer consists of highly optimized Fortran77 routines from 25 year legacy of PIC codes. The upper layers are written in Fortran95. The middle layer primarily provides a much safer and simpler interface to the complex Fortran77 legacy subroutines by encapsulating many details, such as data layouts on parallel machines. The upper layer consists of powerful high level classes that enable the reuse of large blocks of code. The framework also provides several sample main codes.

The QuickPIC code is a fusion of a 3D and 2D parallel PIC code. The 3D code contains the beam particles and treats the 2D code as a transverse field solver, where it passes the beam density as input, and obtains the electric and magnetic fields as output. The framework uses the GCPIC algorithm for domain decomposition [26], but with different decompositions. The 3D framework distributes the ξ coordinate across different processors, while the 2D framework distributes the y coordinate and treats the ξ coordinate as time. Therefore, the 3D code transposes the beam density which has dimensions (nx, ny, nz) from the form $f(nx, ny, nz/nproc)$ to the form $f(nx, ny/nproc, nz)$, where $nproc$ is the number of processors, before it calls the 2D code, and transposes the potentials back to the original form after the 2D code returns.

The 2D framework distributes the y coordinate across processors, but the number of grids per processor can vary. This allows one to keep the number of particles per processor approximately constant by adjusting the sizes of the domains, and thus permits load balancing of the calculation. The framework is spectral and uses FFTs to solve the fields. However, in addition to periodic boundary conditions, conducting boundary conditions are also supported using various combinations of sine and cosine transforms.

The framework provides many of the functions that QuickPIC uses, such as solvers for the potentials, management of guard cells, FFTs, and a particle manager which ensures particles are in the correct domain. To make such functions easy and safe to use, classes have been defined to hide the implementation details of complicated data structures. To illustrate the importance of this, the conducting boundary potential solver doubles the size of the grid in each dimension and creates image charges before the FFT is called. The data in Fourier space are transposed and have a different domain decomposition than in real space. All these details are hidden from the user who uses this solver.

In addition to the components provided by the UPIC framework, some special functions had to be custom written for QuickPIC, usually using a function provided in the framework as a starting point. For example, both the particle push and charge deposit subroutines were modified. The new functions were placed in their own modules, but also had available functions in the original modules, so that the relationship was similar to that of inheritance in object-oriented languages. This kind of structure allows QuickPIC to make use of upgrades to the UPIC framework with relatively little modification.

6. Comparison with full PIC(OSIRIS) simulations

In this section, we present the results of benchmarks of QuickPIC against a 3D fully electromagnetic code OSIRIS. The greatest challenge is correctly computing the wakes for given driver shapes. Once the wake is accurately computed then the accuracy of the advance of the beam particles or laser is not an issue. Therefore, the benchmarks to be presented are on the calculation of the wake for electron, positron, and laser beams. In each case, we assume the beam is azimuthally symmetric and it does not have any head to tail misalignment, i.e., there is no tilt. We have benchmarked QuickPIC against OSIRIS for beams with asymmetric spot sizes and tilts and the agreement is within the same accuracy as those for the symmetric cases. Also the parameters of the benchmarks are chosen to be similar to those in the current or near future experiments; therefore these benchmarks are of great practical relevance.

The first benchmark is done for an electron beam driver, for which we are mostly interested in the focusing force on the beam and the longitudinal wakefield. The electron beam has a Gaussian profile in both transverse and longitudinal directions, the spotsizes are $\sigma_r = 7\ \mu\text{m}$ and $\sigma_z = 45\ \mu\text{m}$, respectively, the emittances are $\varepsilon_x = \varepsilon_y = 15\ \text{mm mrad}$. The total number of electrons in the beam is $N = 1.8 \times 10^{10}$. The beam is

ultra-relativistic with $\gamma = 55{,}800$. The plasma density is $n_0 = 2.0 \times 10^{16}\ \mathrm{cm}^{-3}$, the corresponding plasma skin depth is $c/\omega_p = 37.5\ \mu\mathrm{m}$. This is much larger than σ_r, so the beam is considered narrow and the benchmark results should depend weakly on σ_r. The peak beam density is much higher than the background plasma density, $n_b/n_0 = 25.9$. For these parameters, some electrons reach maximum velocities near $0.96c$ and maximum axial velocities near $0.83c$ and the maximum blowout radius is $\sim 1.5\ c/\omega_p$. Thus the beam–plasma interaction is in the highly non-linear blow-out regime and the assumptions of Whittum do not work [27]. The 3D OSIRIS and QuickPIC simulations are carried out in moving windows with box size $16c/\omega_p \times 16c/\omega_p \times 13.35c/\omega_p$ and conducting boundaries in the transverse directions. The plasma provides strong screening of the driver's fields. So, although the electric and magnetic fields of the beam are huge in its vicinity, they do not penetrate into the plasma beyond a few c/ω_p. The transverse box size is chosen to be large enough so that at the boundaries all fields go to zero, thus the boundaries play little role here. Both simulations use $256 \times 256 \times 256$ cells; the spatial resolution is $0.0625c/\omega_p$ for the transverse direction and $0.052c/\omega_p$ for the longitudinal direction. For the OSIRIS simulation, the time step is chosen to be $0.025c/\omega_p$ to satisfy the Courant condition. During the simulation the beam has propagated $20c/\omega_p$ into the plasma. The beam is stiff due to its large Lorentz factor; its shape almost does not evolve in this short distance. So the quasi-static approximation should be well satisfied.

Shown in Fig. 5 are the longitudinal wakefields from QuickPIC and OSIRIS simulations. Results from QuickPIC simulation with 2 and 4 iterations are both shown and they differ slightly from each other and from the OSIRIS result only at the back of the wake field structure. The agreement is improved with more iterations. However, the simulation with two iterations already gives very satisfactory results even in this highly non-linear case. The discrepancy at the positive spike is caused by the fact that the charge and current deposition schemes diverge for plasma particles with $V_{pz} \sim c$, therefore a smaller 2D step $\Delta\xi$ is required to make the deposition schemes more accurate.

To show that the additional approximations of Whittum are not valid for the highly non-linear interaction, we carried out the same benchmark using basic QuickPIC in which the same algorithm used by Whittum is implemented. The basic QuickPIC result shown in Fig. 5 is dramatically different in regards to the shape, wavelength and amplitude.

Fig. 6 shows the radial electric fields and the azimuthal magnetic fields along the x direction at the center of box ($\xi = 0$). Excellent agreement is observed for both comparisons. There is a plasma sheath at the blow-out

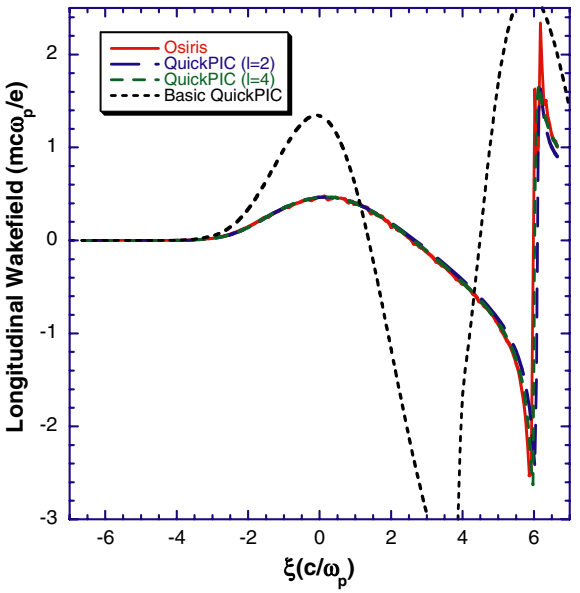

Fig. 5. Longitudinal wakefields in QuickPIC and OSIRIS simulations for an electron drive beam. Both two iterations ($l = 2$) and four iterations ($l = 4$) are used for the QuickPIC simulations. The driver moves from right to left in this plot. Also shown in dashed black curve is the result from basic QuickPIC.

channel boundary which carries high charge density and current density. The fields from the plasma sheath almost cancel the fields from the beam. Therefore, both E_x and B_y are small outside the sheath.

For the second benchmark we use a positron beam driver. A positron beam attracts plasma electrons to the axis and creates a density compression region near the axis. This process is non-linear in that electrons move towards the axis with trajectories that cross. The resulting wake structure is small in scale, which requires fine spatial resolution, thus increasing the problem size. The lack of strong shielding from a sheath layer also makes a wide simulation box necessary. Therefore, the positron benchmark is a more stringent test for the algorithm in QuickPIC.

In the positron benchmark, we once again use a beam containing $N = 1.8 \times 10^{10}$ positrons and set $\gamma = 55{,}800$. The beam spot sizes are $\sigma_r = 25\ \mu\text{m}$ and $\sigma_z = 600\ \mu\text{m}$, respectively, and the emittances are $\varepsilon_x = \varepsilon_y = 15$ mm mrad. The plasma density is $n_0 = 2.0 \times 10^{14}\ \text{cm}^{-3}$, the peak density ratio is $n_b/n_0 = 15.2$. The OSIRIS simulation uses a $4c/\omega_p \times 4c/\omega_p \times 16c/\omega_p$ box with $512 \times 512 \times 256$ cells. The QuickPIC simulation uses a $8c/\omega_p \times 8c/\omega_p \times 16c/\omega_p$ box with $1024 \times 1024 \times 256$ cells. Here a larger box in the QuickPIC simulation is needed due to the fact that a particle reflecting boundary is used in QuickPIC. To show the fine wake

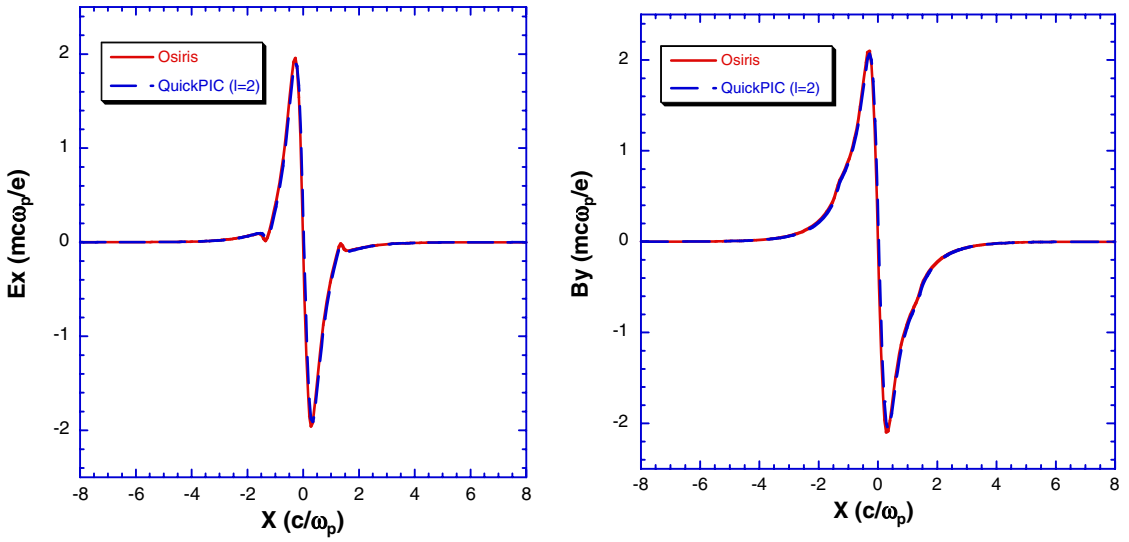

Fig. 6. Radial electric and azimuthal magnetic fields comparisons for electron drive beam.

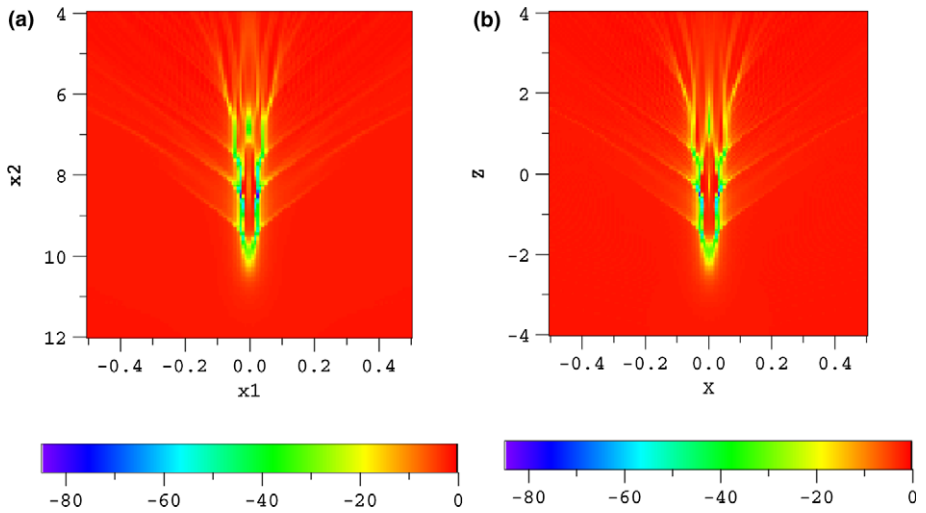

Fig. 7. The plasma electron charge density (ρ_p/ρ_{ion}) in the x–z plane at the center of the beam is shown for (a) an OSIRIS simulation and (b) a QuickPIC simulation. In both cases, the driver moves from top to bottom.

structure, we plot the plasma density in both simulations in Fig. 7. The plots show a small region of the simulation domain to give a better view of the positron wake. The plots show a low density area near the axis and wing-like structures formed around it. The QuickPIC simulation once again reproduces very faithfully most of the detailed features in the OSIRIS results.

Figs. 8 and 9 are comparisons of the wake fields for the positron driver. Both simulations show fine scale fluctuations in the longitudinal wakefield E_z and multiple spikes in E_x (for the center slice with $\xi = 0$). These are the results of the fluctuations of local charge density near the beam core due to the strong phase-mixing as the electrons are drawn to the axis by the positron beam. However, this process does not generate strong plasma currents to cancel the beam current. Therefore, the azimuthal magnetic field is dominated by the beam current. As seen in this benchmark, QuickPIC achieves very good agreement in the non-linear regime. Finally, it should be noted that the positron wake fields are sensitive to the spatial resolution, simulation box size, beam evolution after propagation into the plasma and other boundary effects. A careful simulation setup is

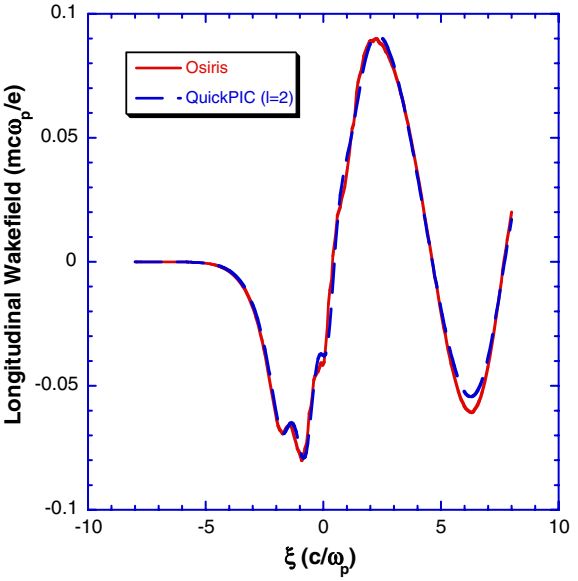

Fig. 8. Comparison of the longitudinal wakefield for a positron driver.

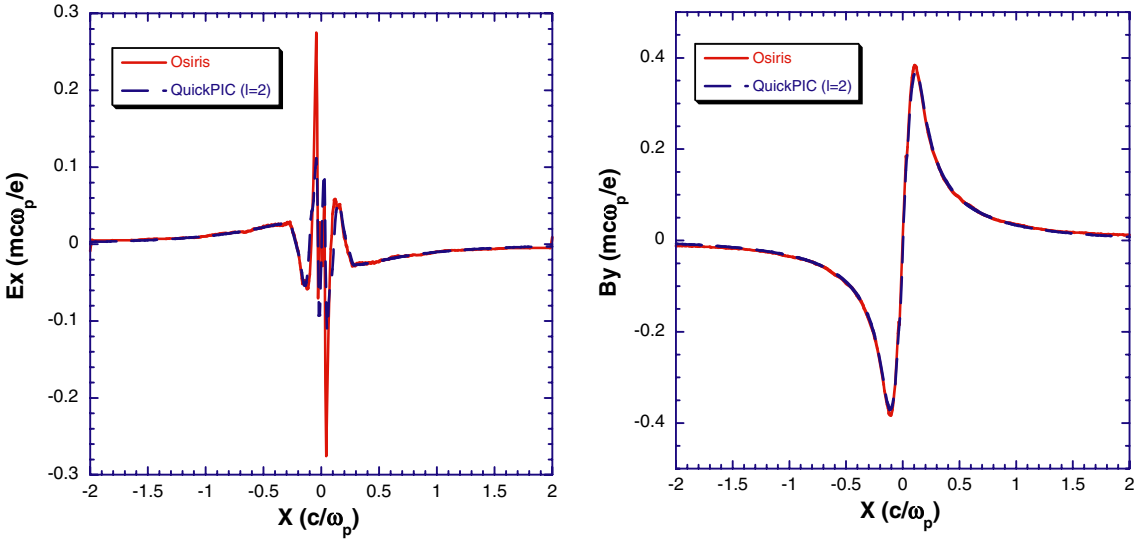

Fig. 9. Radial electric field and azimuthal magnetic field in the positron benchmark simulations.

necessary for the benchmark. The subtle differences between the two simulations observed in Figs. 8 and 9, such as a small plateau region of the first decelerating field, the amplitudes of the second decelerating field and the peak radial electric field, are mostly due to these four factors.

The last benchmark is for a laser driver. The benchmark problem is adopted from the standard problem proposed in Ref. [28] with some modifications to the laser parameters. The laser pulse has a longitudinal profile of the polynomial form $10t'^3 - 15t'^4 + 6t'^5$, where

$$t' = [1 - H(t - t_0)](t_0 - t)/t_{\text{rise}} + H(t - t_0)(t - t_0)/t_{\text{fall}},$$

and $H(t)$ is the Heaviside function. Here the laser has $t_{\text{rise}} = t_{\text{fall}} = 30$ fs and the wavelength is 800 nm. The profile is defined for the electric field, not the intensity. The laser propagates in a uniform plasma with matching density $n_0 = 1.38 \times 10^{19}\ \text{cm}^{-3}$. For this density, $\omega_0/\omega_p = 11.24$ and the non-relativistic plasma skin depth equals the FWHM of the laser pulse longitudinal profile. The transverse profile is Gaussian and the focused spot size is $w_0 = 13.66\ \mu\text{m}$ FWHM at the plasma entrance. The condition for the validity of the ponderomotive guiding center approach is $|\hat{\mathbf{a}}| \ll \sqrt{22.48} = 4.74$. The normalized vector potential of the laser is $|\hat{\mathbf{a}}| = 2$ which satisfies this condition. The laser–plasma interaction is in the non-linear blow-out regime.

The size of the simulation box is $320c/\omega_0 \times 320c/\omega_0 \times 321.8c/\omega_0$ or $28.47c/\omega_p \times 28.47c/\omega_p \times 28.63c/\omega_p$. The number of the grids is $256 \times 256 \times 1024$. For the 3D OSIRIS simulation, $\Delta t = 0.235/\omega_0$, which satisfies the Courant condition. And the laser pulse has traveled $312.55c/\omega_0$ into the plasma so that the laser pulse has no noticeable evolution yet for the comparison of the initial wake. The QuickPIC simulation has a shorter box $320c/\omega_0 \times 320c/\omega_0 \times 157.2c/\omega_0$ to show the first wake bucket as the OSIRIS simulation shows trapped particles in the second bucket. $256 \times 256 \times 256$ grids are used in the QuickPIC simulation. For the relatively small frequency ratio used the two simulations use a similar number of grids. However, if the frequency ratio were increased by a factor r, the OSIRIS simulation would need r times as many cells in z while the QuickPIC simulation would not. Fig. 10 shows the longitudinal wakefield benchmark for the OSIRIS and QuickPIC simulations. The curve from the QuickPIC simulation closely resembles the OSIRIS result, and it is smooth while the OSIRIS curve is modulated by the laser oscillation. We have found that for larger $|\hat{\mathbf{a}}|$ that QuickPIC does not agree with OSIRIS. Understanding the limitations of the ponderomotive guiding center approach is an area for future research.

It is interesting to consider how the required number of iterations depends on the beam parameters and how the QuickPIC algorithm breaks down. Fig. 11 shows two additional longitudinal wakefield benchmarks for identical beam parameters to the case in Fig. 5 except in one case we reduce the beam charge by a factor of 5 and in the other we increase it by a factor of 5. In Fig. 11(a) we show E_z for a weakly nonlinear case where the maximum blow-out radius is $\sim 1c/\omega_p$ and the maximum $V_{pz} \sim 0.7c$. Excellent agreement is obtained between QuickPIC and OSIRIS for 2 (and even 1) iterations. It is seen that the basic QuickPIC result deviates substantially from the full QuickPIC and OSIRIS results even when there are no trapped particles in this case.

In Fig. 11(b) we show E_z for a very extreme example. For these parameters the maximum blow-out radius is $\sim 4c/\omega_p$ and $V_{pz} \sim 0.993c$ for some electrons. Nevertheless QuickPIC gives reasonable results for two iterations and excellent agreement for four iterations except for the spike where particles begin to move forward near the speed of light. Some of these particles can be trapped in the wake under certain circumstances. The basic QuickPIC result for this case is useless. This example shows that the accuracy of the predictor–corrector loop can be improved by using more iterations for a fixed $\Delta\xi$ and the major failure of the code is caused by the particles that can become trapped in the wake. When there are few trapped particles, the code does not become grossly inaccurate no matter much charge is in the beam.

We close this section with an estimate of the CPU savings for the QuickPIC algorithm versus a full PIC algorithm such as OSIRIS. We start with considerations for a beam driver. We assume both types of algorithms use a "moving" 3D window of length L_z composed of $N_x \times N_y \times N_z$ grids and the number of particles per cell, N_p, is the same. It is instructive to calculate the total number of particle pushes. Suppose one wants to calculate the wake from an identical beam, in a full PIC code the beam starts outside the plasma and then must propagate through the box in the z direction. This requires N_z timesteps times a factor of $K_c = \Delta z/c\Delta t$ due to the Courant condition ($K_c = 3^{1/2}$ assuming cells are the same size in each direction). So the number of particle pushes is $N_x \times N_y \times N_z \times N_p \times N_z \times \sqrt{3}$ (it is really a factor of ~ 2 less since not all of the box is filled with particles at the beginning). On the other hand, using QuickPIC we only need to go through the 2D part of

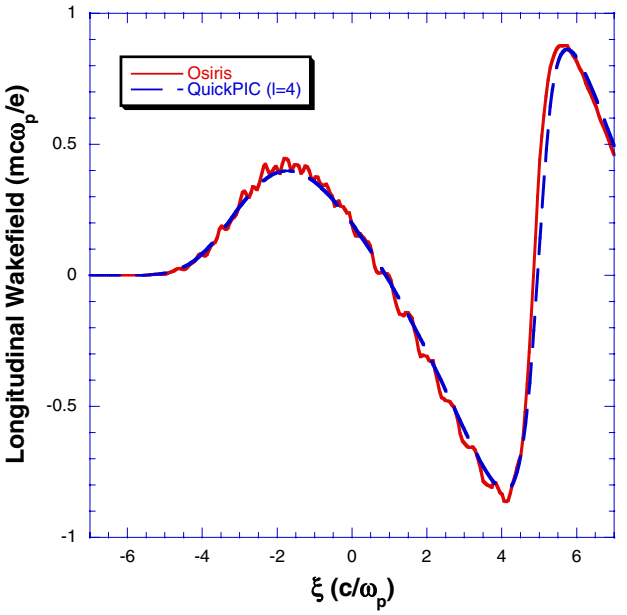

Fig. 10. Longitudinal electric field comparison for a laser driver.

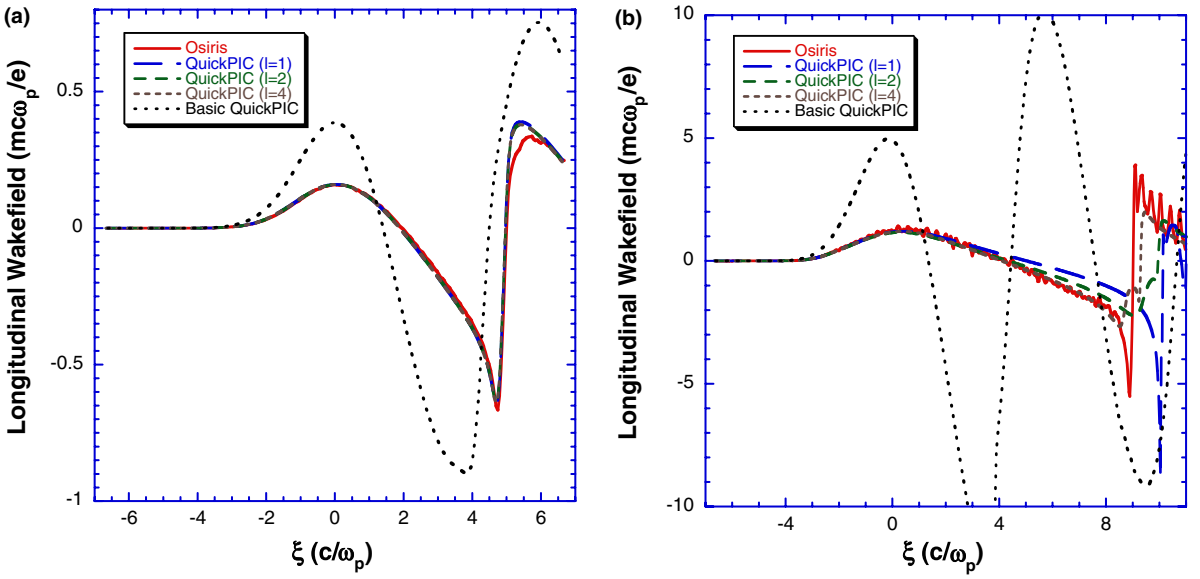

Fig. 11. Longitudinal electric field comparisons for an electron beam driver with (a) $N = 0.36 \times 10^{10}$ and (b) $N = 8 \times 10^{10}$ electrons. Other parameters of the beam and the plasma are the same as in the benchmark in Fig. 5. The accuracy of full QuickPIC is improved with more iterations.

the code once. The number of 2D particle pushes is $N_x \times N_y \times N_\mathrm{p} \times N_z$ where the N_z factor is the number of 2D time steps. Therefore, assuming N_p is the same there is a savings of $N_z \times \sqrt{3}$ in particle pushes. For example, if $N_z = 256$ this is a factor of $\sim$450.

There is additional savings that comes from taking large 3D steps in s after the initial wake is obtained. Suppose one wants to propagate a beam through one betatron oscillation into the plasma, $\lambda_\beta \equiv 2\pi k_\beta^{-1} = 2\pi\sqrt{2\gamma}\frac{c}{\omega_\mathrm{p}}$. Using a full PIC code, a total of $(\lambda_\beta/L_z)N_z \times K_\mathrm{c}$ updates are required to simulate one betatron wavelength. This gives $N_x \times N_y \times N_z \times N_\mathrm{p} \times \lambda_\beta \times (N_z/L_z) \times K_\mathrm{c}$ particle pushes. The factor N_z/L_z is the cell size in z. On the other hand, using QuickPIC requires only $N_x \times N_y \times N_\mathrm{p} \times N_z \times K_\beta$ particle pushes where K_β is the number of steps

needed to resolve one wavelength. We typically use $K_\beta = 20$–30. Therefore, when modeling the propagation of a beam through a fixed distance (here one λ_β), there is a savings of $(\lambda_\beta \times N_z \times K_c)/(L_z \times K_\beta)$. For $N_z \sim 256, K_c \sim \sqrt{3}$, $L_z \sim 2.5\lambda_p$, $K_\beta \sim 20$ and $\gamma_b \sim 10^5$, $\lambda_\beta N_z K_c/(K_\beta L_z) \sim \sqrt{2\gamma_b} N_z K_c/(K_\beta) \sim 1000$. However, the number of particle pushes is not the only issue. The QuickPIC loop is less efficient than a full PIC loop because each 2D update has 2–4 iterations and the field solve involves numerous FFTs. We find from detailed timings that the actual speedup is between 100 and 1000.

For a laser driver, the cell size in z is typically much smaller in a full PIC simulation, so the savings can be considerably higher. When modeling a beam driver the cell size in z is $\sim 0.05c/\omega_p$, while when modeling a laser driver the cell size must be $\lessapprox 0.25c/\omega_0$. Therefore, the savings is typically $[\omega_0/5\omega_p]^2$ times larger when modeling a laser driver. This factor is proportional to the ratio of the cell size squared because there are $\omega_0/5\omega_p$ times more particles and $\omega_0/5\omega_p$ times more time steps.

The above discussion raises the possibility of running a full PIC code to calculate the wake and then using the wake to advance beam particles a large distance forward in z and then repeating this loop. This subcycling would be advantageous over a full PIC code when the distance of the beam advance is larger than the window size. This would still be a factor of $N_z K_c$ less efficient than a quasi-static code but it could be complimentary in some cases [29].

7. Future development

In the immediate future, we will improve QuickPIC by adding ionization and load balancing. Other directions for future code development are the use of k dependent diffusion coefficients in the iteration loop, adaptive time steps in the 2D loop [14] and the use of pipelining. In QuickPIC, the 2D code views ξ as a time variable so the calculation is sequential in ξ, and because the speed of the moving window is c, no information is passed backward in ξ in the 3D code where ξ is treated as a spatial variable. This feature allows QuickPIC to use software pipelining to achieve greater parallelism. Using this technique multiple copies of the code can be started simultaneously, each works on different parts in ξ of the beam and passes the results to the one working on the immediately following part. This is an analogy to the instruction-level pipelining technique used in modern CPU design where speedup is achieved by adding more execution units to the workflow.

If the number of copies used is N, the code speeds up by approximately N times over its current speed. For typical QuickPIC runs, N can be as large as 64. And since the code currently scales to 32 processors, it would be possible with software pipelining to run efficiently on as many as 2048 processors.

8. Summary

In this paper, we have described in detail the algorithm and structure of a new 3D parallelized quasi-static PIC code called QuickPIC for efficiently modeling PWFA and LWFA problems. The main assumption in the QuickPIC model is that there are two distinct time scales for the driver and the plasma evolution. For PWFA, this means that $\sqrt{2\gamma_b} \gg 1$ and for LWFA this means $\omega_0/\omega_p \gg 1$. During the transit time of the driver, plasma particles' longitudinal displacements are also assumed to be small compared to the bunch length. In addition, for the ponderomotive guiding center description of LWFA to be accurate, $|\hat{\mathbf{a}}| \ll \sqrt{2\frac{\omega_0}{\omega_p}}$. Under these approximations, the calculation of the plasma response can be greatly simplified. For the parameters of the current E167 experiment at SLAC and the "afterburner" concept, these approximations are well satisfied and QuickPIC is capable of achieving at least a factor of 100 savings in CPU needs as compared to the fully explicit electromagnetic PIC code OSIRIS. We have presented comparisons of QuickPIC with OSIRIS and the agreement is excellent.

Acknowledgements

This work was supported by the US Department of Energy through Grants DE-FC02-01ER41179, DE-FG02-03ER54721, DE-FG03-92-ER40727, DE-FG02-03NA00065, DE-FG02-97ER41039, DE-FG02-92ER40745 and DE-FC02-01ER41192, by ILSA at LLNL under W-07405-ENG48 and by NSF under PHY0317271.

References

[1] E. Esarey, P. Sprangle, J. Krall, A. Ting, Overview of plasma-based accelerator concepts, IEEE Trans. Plasma Sci. 24 (1996) 252–288, and referenced therein.
[2] T. Tajima, J.M. Dawson, Laser electron accelerator, Phys. Rev. Lett. 43 (4) (1979) 267–270.
[3] C. Joshi et al., Phys. Rev. Lett. 47 (1981) 1285–1288;
T.A. Antonsen Jr., P. Mora, Phys. Rev. Lett. 69 (1992) 2204;
J. Krall et al., Phys. Rev. E 48 (1993) 2157;
N.E. Andreev et al., Pis'ma Zh. Eksp. Teor. Fiz. 55 (1992) 551.
[4] C. Joshi, W.B. Mori, T. Katsouleas, J.M. Dawson, J.M. Kindel, D.W. Forslund, Ultra-high gradient particle acceleration by intense laser-driven plasma density waves, Nature 311 (1984) 525.
[5] P. Chen, J.M. Dawson, R.W. Huff, T. Katsouleas, Acceleration on electrons by the interaction of a bunched electron beam with a plasma, Phys. Rev. Lett. 54 (1985) 693.
[6] C. Joshi, Review of beam driven plasma wakefield accelerators, AIP Conf. Proc. 737 (2004) 3.
[7] S.P.D. Mangles, C.D. Murphy, Z. Najmudin, et al., Nature 431 (2004) 535.
[8] C.G.R. Geddes, Cs. Toth, J. van Tilborg, E. Esarey, C.B. Schroeder, D. Bruhwiler, C. Nieter, J. Cary, W.P. Leemans, High-quality electron beams from a laser wakefield accelerator using plasma-channel guiding, Nature 431 (2004) 538.
[9] J. Faure, Y. Glinec, A. Pukhov, et al., Nature 431 (2004) 538.
[10] C.K. Birdsall, A.B. Langdon, Plasma Physics via Computer Simulation, McGraw-Hill, New York, 1985.
[11] W.B. Mori, Recent advances and some results in plasma-based accelerator modeling, in: C.E. Clayton, P. Muggli (Eds.), Advanced Accelerator Concepts, Tenth Workshop, AIP Conference Proceedings No. 647, 2002, pp. 11–28.
[12] P. Mora, T.M. Antonsen, Kinetic modeling of intense, short laser pulses propagating in tenuous plasmas, Phys. Plasmas 4 (1997) 217.
[13] D.H. Whittum, Transverse two-stream instability of a beam with a Bennett profile, Phys. Plasmas 4 (1997) 1154.
[14] K.V. Lotov, Fine wakefield structure in the blowout regime of plasma wakefield accelerators, Phys. Rev. ST Accel. Beams 6 (2003) 061301.
[15] R. Hemker, Particle-in-cell modeling of plasma-based accelerators in two and three dimensions, Ph.D. thesis, University of California, Los Angeles, 2000;
R.A. Fonseca, L.O. Silva, F.S. Tsung, V.K. Decyk, W. Lu, C. Ren, W.B. Mori, S. Deng, S. Lee, T. Katsouleas, et al., Lect. Notes Comput. Sci. 2331 (2002) 342.
[16] W.B. Mori, The physics of the nonlinear optics of plasmas at relativistic intensities for short-pulse lasers, IEEE J. Quantum Electron. 33 (11) (1997) 1942–1953, and references therein.
[17] J.P. Boris, Relativistic plasma simulation-optimization of a hybrid code, presented at the Proceedings of the 4th Conference on Numerical Simulation Plasmas, 1970.
[18] C.W. Nielson, H.R. Lewis, Particle-Code Methods in the Nonradiative LimitMethods in Computational Physics, vol. 16, Academic Press, New York, 1976.
[19] E. Sonnendrücker, J.J. Ambrosiano, S.T. Brandon, A finite element formulation of the Darwin PIC for use on unstructured grids, J. Comput. Phys. 121 (1995) 281.
[20] E. Sonnendrücker, A. Friedman, D.P. Grote, Progress towards simulating heavy ion beams for inertial fusion energy based on (1) A Darwin model field solver, and (2) A semi-Lagrangian Vlasov solver, in: Proceedings of the 1999 Particle Accelerator Conference, New York, 1999.
[21] J. Busnardo-Neto, P.L. Prichett, A.T. Lin, J.M. Dawson, A self-consistent magnetostatic particle code for numerical simulation of plasmas, J. Comput. Phys. 23 (1977) 300.
[22] W. Wei-li Lee, Edward Startsev, Hong Qin, Ronald C. Davidson, Electromagnetic (Darwin) model for three-dimensional perturbative particle simulation of high intensity beams, in: Proceedings of the 2001 Particle Accelerator Conference, Chicago.
[23] R.W. Boyd, Nonlinear Optics, Academic Press, Boston, 1992.
[24] R.D. Richtmeyer, K.W. Morton, in: L. Bers, R. Courant, J.J. Stoker (Eds.), Interscience Tracts in Pure and Applied Mathematics, Number 4 – Difference Methods for Initial-Value Problems, Wiley, New York, 1967.
[25] V.K. Decyk, C.D. Norton, UCLA parallel PIC framework, Comput. Phys. Commun. 164 (2004) 80–85.
[26] P.C. Liewer, V.K. Decyk, General concurrent algorithm for plasma particle-in-cell codes, J. Comput. Phys. 85 (1989) 302.
[27] J.B. Rosenzweig, B. Breizman, T. Katsouleas, J.J. Su, Acceleration and focusing of electrons in two-dimensional nonlinear plasma wake fields, Phys. Rev. A 44 (1991) R6189.
[28] John R. Cary, Courtlandt L. Bohn, Computational accelerator physics working group summary, in: AIP Conference Proceedings, vol. 737, 2004, p. 231.
[29] S. Deng, X. Wang, T. Katsouleas, Developing a multi-timescale PIC code for plasma accelerators, in: Proceedings of the 2005 Particle Accelerator Conference, Knoxville, Tennessee, USA, 2005.