High-order implicit particle-in-cell method for plasma simulations at solid densities

D. Wu,[1,*] W. Yu,[2] S. Fritzsche,[3,4] and X. T. He[1,5]

[1]*Institute for Fusion Theory and Simulation, Department of Physics, Zhejiang University, Hangzhou 310058, China*

[2]*State Key Laboratory of High Field Laser Physics, Shanghai Institute of Optics and Fine Mechanics, Shanghai 201800, China*

[3]*Helmholtz Institut Jena, Jena D-07743, Germany*

[4]*Theoretisch-Physikalisches Institut, Friedrich-Schiller-University Jena, Jena D-07743, Germany*

[5]*Key Laboratory of HEDP of the Ministry of Education, CAPT, and State Key Laboratory of Nuclear Physics and Technology, Peking University, Beijing 100871, China*

(Received 30 April 2019; revised manuscript received 21 June 2019; published 26 July 2019)

A high-order implicit multidimensional particle-in-cell (PIC) method is developed for simulating plasmas at solid densities. The space-time arrangement is based on Yee and a leapfrog algorithm for electromagnetic fields and particle advancement. The field solver algorithm completely eliminates numerical instabilities found in explicit PIC methods with relaxed time step and grid resolution. Moreover, this algorithm eliminates the numerical cooling found in the standard implicit PIC methods by using a pseudo-electric-field method. The particle pusher algorithm combines the standard Boris particle pusher with the Newton-Krylov iteration method. This algorithm increases the precision accuracy by several orders of magnitude when compared with the standard Boris particle pusher and also significantly decreases the iteration time when compared with the pure Newton-Krylov method. The code is tested with several benchmarks, including Weibel instability, and relativistic laser plasma interactions at both low and solid densities.

DOI: 10.1103/PhysRevE.100.013207

I. INTRODUCTION

The particle-in-cell (PIC) method [1] has established itself as a state-of-the-art method for solving problems in kinetic plasma physics. The main advantages of the PIC method are that their memory consumption increases linearly with the simulated volume and that the runtime is only of order N (number of used particles). They are also very suitable for the use of large multiprocessor systems. Nowadays, the PIC method is the most-used numerical tool in laser-plasma simulations. The PIC method is employed to provide a great variety of information of complex and dynamic systems, like particle acceleration [2–5], x-ray generation [6], and neutron sources [7].

In recent decades, the rapid development of emerging research fields [8–13] such as high-energy density physics, warm dense matter, and laboratory astrophysics impels the PIC method to be versatile and robust in simulating solid density plasmas. However, the main disadvantages of PIC method are high noise levels and high computational requirements due to the operation on the shortest time- and length scales. The plasma frequency needs to be resolved, and the grid size must be comparable to the Debye length in order to minimize artificial grid heating and suppress numerical instabilities. On the positive side, higher-order interpolation algorithms have been utilized in explicit PIC methods, which is, to some extent, successful in suppressing artificial grid heating and suppressing numerical instabilities. However, for solid density plasmas, usually the electron density can be as high as 10^{24} cm^{-3}. Tremendous challenges still remain for the present high-order explicit PIC methods.

In this paper, in order to completely get ride of numerical instabilities and significantly reduce the simulation burden, a high-order implicit PIC method is presented. Based on this method, a new PIC code named laser plasma interaction for solid (LAPINS), is developed. The space-time arrangement of the code is based on Yee and leapfrog algorithm for electromagnetic fields and particle advancement. The field solver algorithm completely eliminates numerical instabilities found in explicit PIC methods with relaxed time step and grid resolution. Moreover, this algorithm eliminates the numerical cooling found in the standard implicit PIC methods by using a pseudo-electric-field method. The particle pusher algorithm combines the standard Boris particle pusher with the Newton-Krylov iteration method. This algorithm increases the precision accuracy by several orders of magnitude when compared with the standard Boris particle pusher and also significantly decreases the iterations when compared with the pure Newton-Krylov method. The code is tested with several benchmarks, including Weibel instability and relativistic laser plasma interactions at both low and solid densities.

The paper is organized as follows. In Sec. II, the basic algorithms of LAPINS code is introduced. The link between LAPINS and LAPINE is also brief explained. In Sec. III, we present several simulation experiments to benchmark the new code, which include a single-particle benchmark, laser low-density plasma interactions, Weibel instabilities, and laser-solid interactions. Summary and discussion are given in Sec. IV.

*dwu.phys@zju.edu.cn

II. NUMERICAL IMPLEMENTATION

A. Governing equations

The PIC method is an approximate solution to the coupled field-particle system. Before referring to numerical implementation, it is convenient to normalize all quantities with laser wavelength and light velocity. Here quantities with prime are the nondimensional variable. First, length is normalized to the laser wavelength λ_0 with $x =: x'\lambda_0$, and time is normalized to the laser period with $t =: t'\tau_0$, where $\tau_0 = \lambda_0/c$ and c is the light velocity. Then velocity is normalized to laser speed in vacuum with $\mathbf{v} =: \mathbf{v}' \cdot \mathrm{c}$, mass is normalized to static electron mass with $m =: m' \cdot m_0$, and therefore momentum is normalized to $\mathbf{u} =: \mathbf{u}' \cdot m_0 c = \gamma \mathbf{v}' \cdot m_0 c$, where $\gamma = (1 - v'^2)^{-1/2}$. Moreover, charge of particle is normalized to electron static charge with $q =: q'e$, and numerical density is normalized to critical density with $n =: n' n_c$, where $n_c = m_0\omega_0^2/4\pi e^2$. Therefore charge density ρ is normalized to $\rho =: \rho' e n_c$ and current density is normalized to $\mathbf{J} =: \mathbf{J}' \cdot e c n_c$. Finally, electromagnetic field are normalized to $\mathbf{E} =: \mathbf{E}' \cdot m_0\omega_0 c/e$ and $\mathbf{B} =: \mathbf{B}' \cdot m_0\omega_0 c/e$, where ω_0 is the laser circular frequency. The normalized Maxwell equations now read

$$\nabla' \times \mathbf{E}' = -\partial_{t'}\mathbf{B}', \tag{1}$$

$$\nabla' \times \mathbf{B}' = \partial_{t'}\mathbf{E}' + 2\pi\mathbf{J}', \tag{2}$$

$$\nabla' \cdot \mathbf{E}' = 2\pi\rho', \tag{3}$$

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

and the normalized particle equation of motion is

$$m'_\alpha \frac{d\mathbf{u}'_\alpha}{dt'} = 2\pi q'_\alpha \left(\mathbf{E}' + \frac{\mathbf{u}'_\alpha}{\gamma_\alpha} \times \mathbf{B}'\right). \tag{5}$$

The current density and charge density is

$$\mathbf{J}'_\alpha(\mathbf{r}) = \frac{1}{n_c}\sum_\alpha q'_\alpha \mathbf{v}'_\alpha W(\mathbf{r}' - r'_\alpha), \tag{6}$$

$$\rho'_\alpha(\mathbf{r}) = \frac{1}{n_c}\sum_\alpha q'_\alpha W(\mathbf{r}' - r'_\alpha), \tag{7}$$

where shape factor satisfies condition $\int W(\mathbf{r}' - r'_\alpha) = 1$. The Maxwell equations, Eqs. (1)–(4), are coupled to the particle equation of motion, Eq. (5), through the current density and charge density. In the explicit coupling scheme, the current and charge densities carried by computational particle α are computed in Eqs. (6) and (7) and then interpolated to update the Maxwell equations.

In recent decades, Esirkepov [15] proposed a method of local current density assignment. This method ensures exactly charge conservation. Moreover, this method allows one to implement the PIC code without solving the Poisson equation. Following Esirkepov's method [15], the laser plasma interaction (LAPINE) [14] code was developed and in this code, a second-order FDTD field solver was used to advance Maxwell equations. Later, the LAPINE code was extended to a high-order interpolation scheme [16] for both the current density assignment and FDTD field solver. Moreover, many advanced physics modules, like field ionization [17], collision ionization and electro-ion recombination [18], binary collisions including both elastic and inelastic processes [19], and some QED effects [20] were developed and linked to the LAPINE code.

Indeed, the LAPINE code is very successful in a great variety of research branches. However, for solid density plasmas, tremendous challenges still remain for the present high-order explicit PIC methods. In order to study laser plasma interaction at solid densities while keeping unphysical effects to a minimum, recently a new code named LAPINS has been developed. This code completely eliminates numerical instabilities found in explicit PIC methods with relaxed time step and grid resolution. The algorithm of the code is detailed in the following sections.

B. Basic algorithm of LAPINS code

Petrov and Davis used to propose an implicit particle-in-cell algorithm [21]. The most significant advantage of this algorithm is the local solution of all physical quantities. Such an advantage significantly speeds up simulations, as no information from other grids is needed for advancing electric and magnetic fields.

However, some defects remain in their original algorithm, e.g., (a) the lattice arrangement is not inherently divergence-free for the magnetic feld, which could possibly lead to unphysical results; (b) for updating the electric field, relativistic factor $\gamma^{n+1/2}$ at time level $t^{n+1/2}$ is needed; however, it is approximated with γ^n by adding a one-half push using the electric field at time level t^n. This approximation might bring inaccuracy of the electric field, especially when the laser intensity is strong or the laser frequency is high.

Later, a modification of Petrov and Davis's algorithm [21] was proposed by Kempf *et al.* [22]. A reduction of the divergence of the magnetic field by several orders of magnitude was achieved by replacing the original lattice with a staggered lattice. However, other defects in Petrov and Davis's original algorithm still remain.

We here reorganize this algorithm and eliminate both two defects by (1) introducing the staggered Yee mesh and (2) avoiding all inaccuracies of physical quantities by inventing a new time stepping scheme. Moreover, in order to further enhance the numerical instability and precision, we also implement the high-order interpolation scheme (widely used in explicit code) into the implicit PIC code.

The first to construct a PIC code is to choose a spatial and temporal discretization. In order to optimally exploit the resolution, and taking into account the symmetries of the Maxwell equations, we use a Yee lattice (suggested by Kempf *et al.* [22]) to stagger the fields. As shown in Fig. 1, the charge density is located at cell centers. To comply with Gauss's law, Eq. (3), the electric fields and the current density, both entering with the same spatial distribution in Ampere's law, Eq. (2), are staggered upward to cell faces, while magnetic fields, to be consistent with the curl operator in Eq. (2), are placed at cell edges. With this distribution, the derivatives in Eqs. (1)–(4) are automatically calculated at the right spatial positions, and no interpolations are needed. Because of the spatial staggering, the numerical derivatives commute and the evolution of magnetic field conserves divergence to round-off precision. Please note that in the original Petrov and Davis's

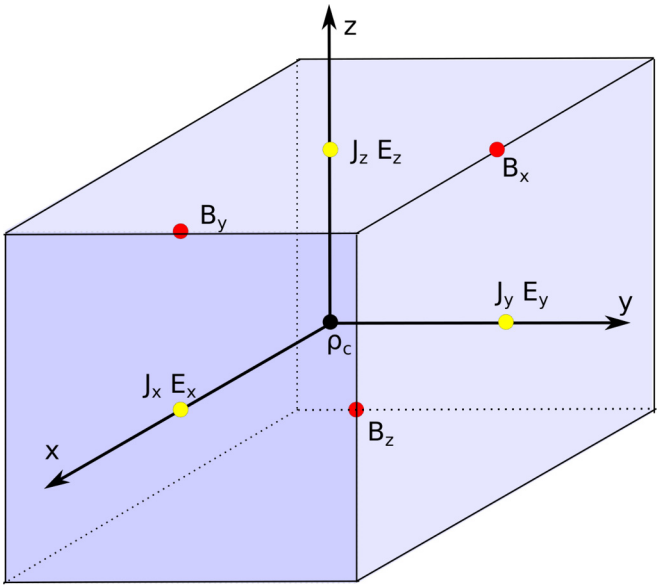

FIG. 1. Spatial staggering in the LAPINS code. The charge density is located at cell centers. The electric fields and the current density are staggered upward to cell faces. The magnetic fields are placed at cell edges. Please note that in the original Petrov and Davis's algorithm, the electric fields, the current densities, and charge densities are located at the cell center, while the magnetic fields are placed at cell edges.

algorithm, the electric fields, the current density and charge densities are located at the cell center, while the magnetic fields are placed at cell edges. This spatial discretization, although simple in algorithm programming, probably leads to a large divergence of the magnetic field.

In our PIC code, the order to do the updates in a time step is (see Fig. 2) as follows

(1) Update the velocity of the macro particles, from $\mathbf{u}_\alpha'^{n-1/2}$ to $\mathbf{u}_\alpha'^{n+1/2}$, using a new algorithm combining Boris's particle pusher and Newton-Krylov iteration method.

(2) Given the velocity of macro particles, $\mathbf{u}_\alpha'^{n+1/2}$, update the position of macro particles from $\mathbf{x}_\alpha'^{n}$ to $\mathbf{x}_\alpha'^{n+1}$ with $\mathbf{x}_\alpha'^{n+1} = \mathbf{x}_\alpha'^{n} + \mathbf{u}_\alpha'^{n+1/2} \cdot \delta t'$.

(3) In the LAPINE code, find the current density $\mathbf{J}'^{n+1/2}$ from the time derivative of the charge density using a charge-conserving method [15].

$$\widehat{T}_\alpha^{n+1/2} = \frac{1}{1+\left|\Delta\boldsymbol{\Omega}_\alpha'^{n+1/2}\right|^2}\begin{bmatrix} 1+\Delta\Omega_{\alpha,x}'^2 & \Delta\Omega_{\alpha,x}'\Delta\Omega_{\alpha,y}' - \Delta\Omega_{\alpha,z}' & \Delta\Omega_{\alpha,x}'\Delta\Omega_{\alpha,z}' + \Delta\Omega_{\alpha,y}' \\ \Delta\Omega_{\alpha,x}'\Delta\Omega_{\alpha,y}' + \Delta\Omega_{\alpha,z}' & 1+\Delta\Omega_{\alpha,y}'^2 & \Delta\Omega_{\alpha,y}'\Delta\Omega_{\alpha,z}' - \Delta\Omega_{\alpha,x}' \\ \Delta\Omega_{\alpha,x}'\Delta\Omega_{\alpha,z}' - \Delta\Omega_{\alpha,y}' & \Delta\Omega_{\alpha,y}'\Delta\Omega_{\alpha,z}' + \Delta\Omega_{\alpha,x}' & 1+\Delta\Omega_{\alpha,z}'^2 \end{bmatrix}, \tag{11}$$

where $\Delta\boldsymbol{\Omega}_\alpha' = q_\alpha'\mathbf{B}_\alpha'^{n+1/2}\delta t'/2m_\alpha'\gamma_\alpha^{n+1/2}$ is one-half the cyclotron frequency times the time step $\delta t'$. Here $\mathbf{u}_\alpha'^{n}$ in Eq. (10) is $\mathbf{u}_\alpha'^{n} = (\mathbf{u}_\alpha'^{n-1/2} + \mathbf{u}_\alpha'^{n+1/2})/2$, and $\mathbf{r}_\alpha'^{n+1/2}$ is $\mathbf{r}_\alpha'^{n+1/2} = (\mathbf{r}_\alpha'^{n} + \mathbf{r}_\alpha'^{n+1})/2$. Due to the unique time staggering and integration order, the relativistic factor $\gamma^{n+1/2}$ is now rigorously known. When compared with the original algorithm by Petrov and Davis [21], this algorithm avoids all inaccuracies of physical quantities.

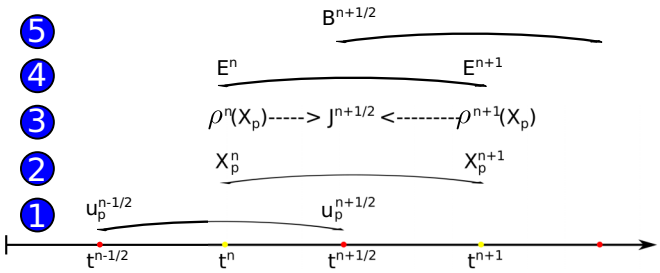

FIG. 2. Time staggering and integration order in the LAPINS code. Note that $t'^{n-1/2} = t'^n - \delta t'/2$, $t'^{n+1/2} = t'^n + \delta t'/2$, and $t'^{n+1} = t'^n + \delta t'$.

(4) In the LAPINE code, given the time-staggered current density $\mathbf{J}'^{n+1/2}$ and magnetic field $\mathbf{B}'^{n+1/2}$, update the electric field using FDTD field solver from $\mathbf{E}'^n$ to $\mathbf{E}'^{n+1}$.

(5) Given the electric field $\mathbf{E}'^{n+1}$, update the magnetic field from $\mathbf{B}'^{n+1/2}$ to $\mathbf{B}'^{n+3/2}$.

(6) Given the electric field $\mathbf{E}'^{n+1}$ and time-centered magnetic field $\mathbf{B}'^{n+1} = (\mathbf{B}'^{n+1/2} + \mathbf{B}'^{n+3/2})/2$, go to step (1).

However, in the newly developed LAPINS code, the electric field update in Eq. (2) of the Maxwell equations is quite different. Instead of the two separate steps, i.e., steps (3) and (4), here the current density and electric field updates are coupled together [21],

$$(\widehat{I} + \widehat{S}^{n+1/2})\mathbf{E}'^{n+1} = (\widehat{I} - \widehat{S}^{n+1/2})\mathbf{E}'^{n} + (\nabla' \times \mathbf{B}'^{n+1/2} - 2\pi\mathbf{J}'^{n+1/2})dt', \tag{8}$$

where

$$\widehat{S}^{n+1/2}(\mathbf{r}') = \sum_\alpha \frac{\pi^2 n_\alpha' q_\alpha'^2 dt'^2}{m_\alpha' \gamma_\alpha^{n+1/2}} \widehat{T}_\alpha^{n+1/2} W\left(\mathbf{r}' - \mathbf{r}_\alpha'^{n+1/2}\right) \tag{9}$$

and

$$\mathbf{J}'^{n+1/2}(\mathbf{r}') = \sum_\alpha \frac{n_\alpha' q_\alpha'}{2m_\alpha' \gamma_\alpha^{n+1/2}} \left[\mathbf{u}_\alpha'^{n} + \widehat{T}_\alpha^{n+1/2}\left(\mathbf{u}_\alpha'^{n} + \mathbf{u}_\alpha'^{n} \times \nabla\boldsymbol{\Omega}_\alpha'^{n+1/2}\right)\right] W\left(\mathbf{r}' - \mathbf{r}_\alpha'^{n+1/2}\right). \tag{10}$$

The tensor $\widehat{T}_\alpha^{n+1/2}$ appears in Eqs. (9) and (10) is defined as

Note that the original algorithm by Petrov and Davis [21] stores electric fields on cell centers and magnetic fields in the cell edges. The vector quantity $\mathbf{J}'$ and the tensor quantity $\widehat{S}$ are deposited on cell centers, as well. This layout was proven to be not inherently divergence-free, possibly leading to unphysical results. In the Yee algorithm, the electric fields are stored on the cell faces. In order to locally solve the electric field, the vector quantity $\mathbf{J}'$ and the tensor quantity $\widehat{S}$ stored on cell

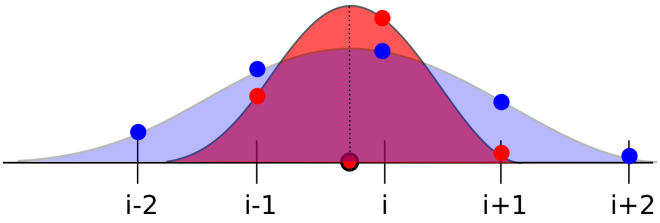

FIG. 3. Red (three point) and blue (five point) shadows define the particles' weight functions. These two functions are all accessible in the code.

centers need to be interpolated linearly for each component of the electric field.

We also implement the high-order interpolation algorithm into LAPINS code. Usually, the deposition of $\mathbf{J}'$ and $\widehat{S}$ on the cells and the interpolation of $\mathbf{E}'$ and $\mathbf{B}'$ to the particle positions are achieved via a weighting function. The standard weighting function is the linear interpolation scheme, while in our LAPINS code, as shown in Fig. 3, both quadratic spline (three-point) and quartic spline (five-point) interpolations are adopted.

Spatial derivatives of the field quantities, in the LAPINS code, are done by finite differencing on a uniform mesh. Staggering of the variables on a Yee lattice leads to highly simplified computations for the difference equations. From Fig. 1, we see that this computation yields the desired value exactly where needed, provided that we compute the central differences at the half-staggered mesh point. A single component of the $\nabla'\times$ operator is illustrated in Fig. 4; this choice of coefficients (based on a Taylor expansion) for the higher-order differential operators has enabled us to significantly reduce the numerical noises appearing in high-density plasmas.

C. Additional refinement of LAPINS code

At present, in the above algorithm, only the Faraday and Ampere equations were considered, while the two divergence equations were not taken in account. It is easy to show that the equation $\nabla'\times\mathbf{B}'=0$ is always satisfied if it is initially. Moreover, Gauss's law $\nabla'\times\mathbf{E}'=2\pi\rho'$ is automatically satisfied if the charge continuity equation $\partial\rho'/\partial t'+\nabla'\cdot\mathbf{J}'=0$ holds true. In fact, the definition of the current density in Eq. (10) does not rigorously satisfy the charge density continuity equation. In this case, the method is said not to conserve

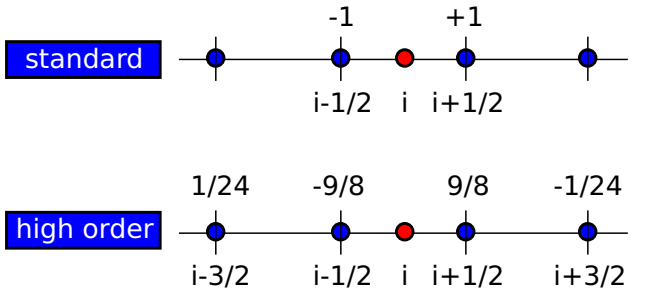

FIG. 4. Example: Second-order and fourth-order difference operation $\nabla'\times$ in 1D geometry. Due to the Yee mesh staggered layout of variables, the central difference is computed exactly where needed.

the charge. Since the implicit algorithms described here are inherently dissipative, energy is not necessarily conserved. Some possible causes or mechanisms of nonconservation of energy is the so-called numerical cooling [23], which is due to the dissipative nature of the implicit time integration scheme. As pointed out in Refs. [24,25], Gauss's law can be regarded as a conservation principle: It is not a strictly necessary equation for describing the evolution of electromagnetic fields, but its violation introduces numerical errors in the simulation and might lead to unphysical behavior of the simulated plasma. All the energy-conserving PIC simulations [26], based on the nonconservative current density definition, have been first initialized solving Gauss's law, ensuring there is no error due to the violation of the charge continuity equation initially. Then the charge conservation has been constantly checked, whether the error grows considerably or leads to unphysical behavior of the simulated plasma.

In the LAPINS code, the numerical error can be significantly reduced by using the pseudoelectric method [24,25]. Here F'^{n+1} is defined as the violation of Gauss's law in the cell at time t^{n+1}, with $F'^{n+1}=\nabla'\cdot\mathbf{E}'^{n+1}-2\pi\rho'^{n+1}$. The additional pseudoelectric field is defined as $\mathbf{E}'^{n+1}_{\rm psd}=2\pi d'\delta t'\nabla' F'^{n+1}$, where d' is a user-defined dimensionless parameter. In each time step, the net electric field imposed on particle is updated by adding the pseudoelectric field $\mathbf{E}'^{n+1}=\mathbf{E}'^{n+1}+\mathbf{E}'^{n+1}_{\rm psd}$. The introduced pseudoelectric field method could significantly eliminate the numerical cooling found in the standard implicit PIC methods and provides enhanced energy conservation for plasmas simulations. The numerical experiments will be detailed in the next section.

To move the macroparticles forward in time, we have to solve Eq. (5). We need a time-centered value for the velocity $\mathbf{v}'^{n}_{\alpha}=\mathbf{u}'^{n}_{\alpha}/\gamma^{n}_{\alpha}$, as this gives an implicit equation for momentum $\mathbf{u}'^{n+1/2}_{\alpha}$. Traditionally, the Boris particle pusher [27] has been used. It is formulated that the time-centered proper velocity is computed as the average, $\mathbf{u}'^{n}_{\alpha}=(\mathbf{u}'^{n-1/2}_{\alpha}+\mathbf{u}'^{n+1/2}_{\alpha})/2$, and from that the three velocity $\mathbf{v}'^{n}_{\alpha}$ is derived. It is still widely used at present in PIC simulation code. However, Boris's particle pushers is of second-order precision, and the accuracy precision depends on squared time step dt'^2. Here in LAPINS code, a new particle pusher algorithm is proposed. It combines the standard Boris algorithm with Newton-Krylov iteration algorithm. It is shown that this new particle pusher can significantly increase the accuracy precision.

For Newton-Krylov algorithm, the aim is to minimize the bellowing residue,

$$\mathscr{R}=\frac{m'_{\alpha}}{2\pi q'_{\alpha}\delta t'}\left(\mathbf{u}'^{n+1/2}_{\alpha}-\mathbf{u}'^{n-1/2}_{\alpha}\right)-\mathbf{E}'^{n}-\frac{\mathbf{u}'^{n+1/2}_{\alpha}+\mathbf{u}'^{n-1/2}_{\alpha}}{\gamma'^{n+1/2}_{\alpha}+\gamma'^{n-1/2}_{\alpha}}\times\mathbf{B}'^{n}. \tag{12}$$

Usually, an initial guess of $\mathbf{u}'^{n+1/2}_{\alpha}$ is put into Eq. (12). Due to the unique time staggering and integration order in the LAPINS code, the initial guess of $\mathbf{u}'_{\alpha}$ is calculated by Boris's particle pusher, and then it is successively minimized by Newton-Krylov iterations. This algorithm increases the precision accuracy by several orders of magnitude when compared with standard Boris's algorithm and also significantly decreases the iteration time when compared with pure

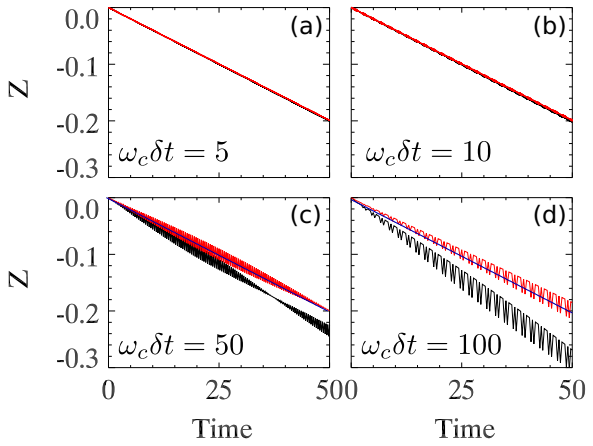

FIG. 5. The $\mathbf{E}\times\mathbf{B}$ drift motion for four cases: (a) $\omega_c'\delta t' = 5$, (b) $\omega_c'\delta t' = 10$, (c) $\omega_c'\delta t' = 50$, and (d) $\omega_c'\delta t' = 100$. The following parameters are the same for the simulation cases: $\mathbf{u}'(t'=0) = (0.1, 0, 0)$, $\mathbf{B}' = (250, 0, 0)$, and $\mathbf{E}' = (0, 1, 0)$. Here the (upper) red lines shows results from Newton-Krylov method, (lower) black lines are results from Boris's method, and the dashed lines are the theoretical predictions.

Newton-Krylov algorithm. In fact, one to two iterations is enough to reach the tolerance specified for the residual reduction. We will detail the numerical experiments in also in the next section.

III. BENCHMARK

Based on the above algorithm, the LAPINS code is developed, which enables all one dimensional in space-three dimensional in velocity (1D-3V), two dimentional in space-three dimensional in velocity (2D-3V) and three dimentional in space-three dimensional in velocity (3D-3V) and simulation capabilities. In this section, the LAPINS code is tested by several benchmarks.

A. Single-particle simulation

In order to benchmark the Newton-Krylov method and compare its performance with the widely used Boris method, we here calculate a single particle's orbits [28] with the Newton-Krylov method and Boris's method. We have chosen to explore the feasibility of running with large $\omega_c'\delta t'$ using the standard $\mathbf{v}\times\mathbf{B}$ rotation scheme of Boris. The reason for doing this, besides simplicity, is that for modeling systems with a varying electromagnetic field, $\mathbf{v}\times\mathbf{B}$ can vary greatly and $\delta t'$ would need to be restricted by the largest value of ω_c'. We begin by showing a single particle's orbits obtained with small $\omega_c'\delta t'$ values. Then we discuss the orbit characteristics when $\omega_c'\delta t'$ becomes large.

Figure 5 shows the $\mathbf{E}\times\mathbf{B}$ drift motion for four cases: $\omega_c'\delta t' = 5$, $\omega_c'\delta t' = 10$, $\omega_c'\delta t' = 50$, and $\omega_c'\delta t' = 100$. The following parameters are the same for the simulation cases: $\mathbf{u}'(t'=0) = (0.1, 0, 0)$, $\mathbf{B}' = (250, 0, 0)$, and $\mathbf{E}' = (0, 1, 0)$. The theoretical $\mathbf{E}'\times\mathbf{B}'$ drift velocity is 0.004 along the z direction. For small $\omega_c'\delta t'$ values, the particle's orbits using Boris's scheme or the Newton-Krylov method matches perfectly with the theoretical predictions. While when $\omega_c'\delta t'$ becomes larger, for example, $\omega_c'\delta t' = 100$, as shown by (lower) black lines, particle orbits using Boris's scheme significantly departs from correct values. However, we note that under such conditions, as shown by (upper) red lines, the Newton-Krylov method is still quite robust.

In Eq. (12), residue $\mathscr{R}$ can be regarded as a quantity to measure the performance of different particle pusher schemes. Table I shows corresponding residues of the Boris method, a typical Newton-Krylov method, and the new method combining Boris and Newton-Krylov. Here one needs to note that the residue is of 1×10^{-4} for the pure Boris method. For a typical Newton-Krylov method, the initial guess of momentum is to set $\mathbf{u}' = 0$. As indicated in Table I, the residue is decreased rapidly after two to three iterations. While for the Newton-Krylov method, the residue can even be reduced to round-off precision (e.g., 1×10^{-19} in our code). The benchmark results in Table I show significant advantages when combining Boris's method with the Newton-Krylov method. Here, for the new method, the initial guess of $\mathbf{u}_\alpha'$ is calculated by Boris's particle pusher, and then it is successively minimized by Newton-Krylov iterations. Compared with the pure Newton-Krylov method, only one to two iterations is enough to reach the tolerance specified for the residual reduction.

Please note that for the Newton-Krylov method, the computational effort when compared with Boris's method is significant. Fortunately, according to our benchmark, only one to two iterations is needed to reach the usual tolerance specified for the residual reduction. This new method, which combines the Boris and Newton-Krylov methods, appears to be an additional refinement for the standard Boris method in the LAPINS code. It is encouraged to be opened only when a particular simulation requires a highlevel of precision.

B. Intense laser plasma interactions

In this benchmark, the interaction of an intense laser with low-density plasmas is tested by using 1D-3V PIC simulations. Following Ref. [29], the simulation box is chosen to be along the z direction with a size of 6 μm, which is divided into 300 uniform grids. Here a helium gas is located at 1 μm $< z <$ 5 μm. The laser is normally focused on the gas target of helium. The normalized laser amplitude is of $E' = 1$, and the polarization of the laser is along the x

TABLE I. The residues, following Eq. (12), of Boris's method, typical Newton-Krylov method, and the new method combining Boris and Newton-Krylov. The parameter is $\mathbf{u}'(t=0) = (0.1, 0.0, 0.0)$, $\mathbf{B}' = (250, 0, 0)$, and $\mathbf{E}' = (0, 1, 0)$. Here we fix $\omega_c'\delta t' = 5$.

Iteration	0	1	2	3	4	5	6
JFNK	1.524	3.34e-4	8.97e-5	4.22e-9	4.12e-09	1.96e-13	1.57e-13
Boris+JFNK	3.45e-4	3.16e-5	1.34e-8	4.25e-10	1.15e-13	1.96e-14	3.46e-19

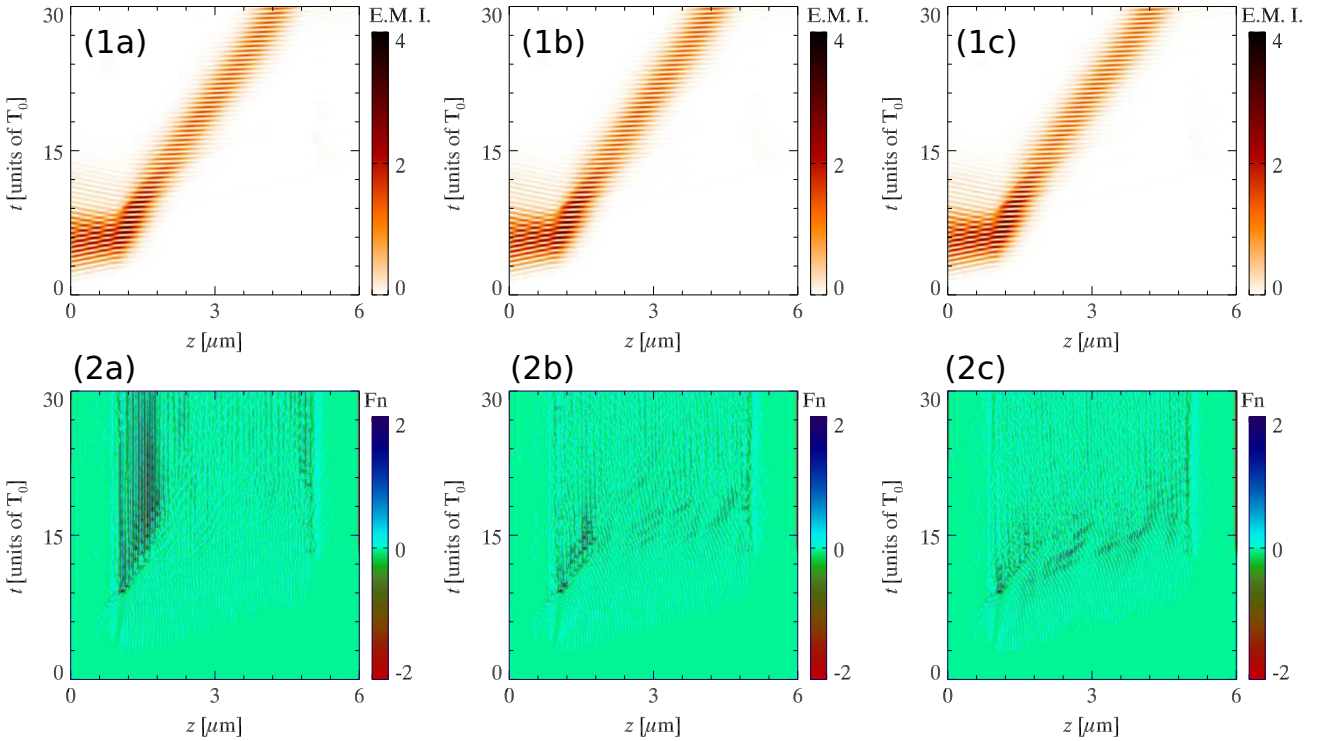

FIG. 6. The first row shows energy distribution of electromagnetic fields as functions of the time and positions. The second row shows the value of F_n' (see text for explanation) as functions of time and positions. The first, second, and third columns correspond to the results carried out by LAPINS code with $d' = 0.0$, $d' = 0.5$, and $d' = 1$, respectively.

direction. This laser consists out of a $\sin^2$ rising in front of $5T_0$ following by a $\cos^2$ declining end also of $5T_0$, where T_0 is the laser-wave period. The central wavelength of this laser is 1 μm, and therefore the laser period T_0 equals 3.33 fs. The simulation time step is 0.04 fs. The helium density is $0.65n_c$, where $n_c = 1.1 \times 10^{21}$ cm^{-3} is normalized by $m_e\omega_0^2/4\pi e^2$. The initial temperature of helium gas is room temperature, and the initial ionization degree of helium is $Z' = 0$. In the simulations, absorbing boundary conditions are applied for both particles and laser fields.

As the implicit numerical scheme used in LAPINS code does not exactly ensure Gauss's law, we therefore apply an additional pseudoelectric field to reduce the numerical errors. This pseudo field is defined as $\mathbf{E}_{\rm psd}^{\prime n+1} = 2\pi d'\delta t'\nabla' F'^{n+1}$, where d' is a user-defined dimensionless parameter. In this benchmark, three simulation cases are carried out, with different values of $d' = 0$, $d' = 0.5$, and $d' = 1.0$, respectively.

In Fig. 6, the first row shows the energy distribution of electromagnetic fields as functions of the time and positions. In the LAPINS code, F_n' is defined as $F_n' = \nabla' \cdot \mathbf{E}' - 2\pi\rho'$. For the second row, the distribution of F_n' as functions of the time and position is presented. When $d' = 0.0$, i.e., no pseudoelectric field is applied, Fig. 7(a) indicates there is a slight violation of Gauss's law, which appears at latter time of simulation. When we set $d' = 0.5$ and $d' = 1.0$, as shown in Figs. 6(2b) and 6(2c), this violation is significantly eliminated.

The correlation between violation of Gauss's law and violation of energy conservation is shown in Fig. 7. In this figure, the residue charge and the energy conservation as a function of time, are simultaneously presented. Thick blue, dashed red, and black lines refer to three different simulations with $d' = 0.0$, $d' = 0.5$, and $d' = 1.0$, respectively. The residue charge at each time step is calculated by summarizing F_n' over all computational grids. The advantage of pseudoelectric field method appears, as shown in Figs. 7(a) and 7(b), when we compare residue charge and the energy conservation as a function of time. By adding a pseudoelectric field with $d' = 0.5$ or $d' = 1.0$, the violation of Gauss's law is significantly eliminated, and in the meantime, the numerical cooling found in the standard implicit PIC methods is also eliminated, which provides enhanced energy conservation for plasma simulations.

C. Weibel instability

The motivation for this benchmark lies in the generic nature of Weibel instabilities. It is of fundamental importance to fusion physics, laser particle acceleration, astrophysical shock generation, γ-ray bursts, and solar flares physics. Following Refs. [30–32], we summarize the initial configuration: Homogeneous, collision-less, and charge- and current-neutral systems. We study the transverse dynamics of two initially uniform currents; the former is a relativistic beam, whereas the second is a return current which guarantees initial charge and current neutrality. The plasma is characterized by density $n_{p0} = 1 \times 10^{21}$ cm^{-3}, while the beam has density $n_{b0}/n_{p0} = 0.1$ and Lorentz factor $\gamma_{b0} = 2.5$. We also assume that electrons have a Maxwellian distribution with thermal velocity 10^{-4} times smaller than the beam velocity, while the ions are initially at rest. We here carry out 2D-3V PIC simulations.

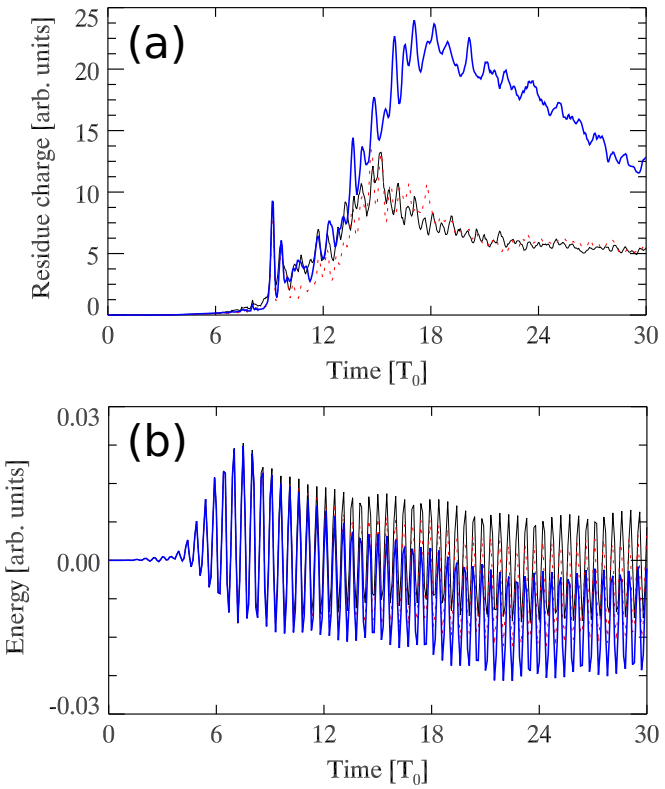

FIG. 7. (a) The residue charge, i.e., integrating F_n' over the whole space, as a function of time. (b) The energy conservation as a function of time. Here (thick) blue line, (dashed) red line, and black line refer to results carried out by LAPINS code with $d' = 0.0$, $d' = 0.5$, and $d' = 1$, respectively.

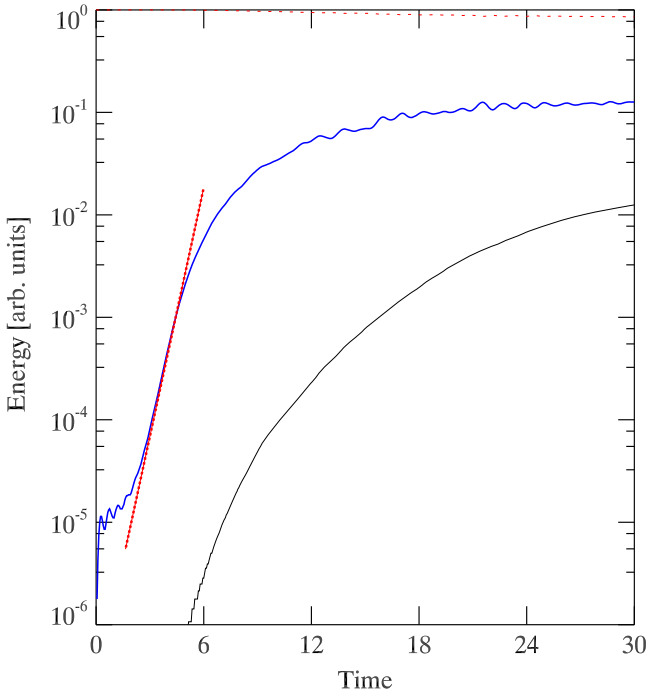

FIG. 8. The energy (transfer) as a function of time. Data are shown for plasma electron kinetic energy (dashed red), ion kinetic energy (black), and electromagnetic energy (thick blue). Here the electromagnetic energy is defined as $(1/2)\sum(\mathbf{E}'^2 + \mathbf{B}'^2)\delta x'\delta y'\delta z'$. The straight line shows the analytical growth rate.

The simulation plane is transverse to the beam propagating direction, where the coupling of the Weibel-type instability with the two stream instability is excluded. In following simulations, the simulation time and length are normalized to 3.33 fs and 1 μm. The size of the simulation box is 4 μm $\times$ 4 μm with 400 $\times$ 400 cells. The simulation time step is 0.08 fs. Periodic boundary conditions are applied in both two directions and 25 particles per cell are used.

In Fig. 8, the electromagnetic field energies, i.e., $(1/2)\sum(\mathbf{E}'^2 + \mathbf{B}'^2)\delta x'\delta y'\delta z'$, as a function of time are presented. It is apparent that the relativistic beam electrons transfer energy and heat the plasma until nonlinear mechanisms dominate the dynamics. The magnetic energy exhibits an exponential behavior as predicted from the linear theory. Under this circumstance, the dispersion relation is $Z'(\omega/kv_{\rm th}) = -(v_{\rm th}/v_b)^2(\omega_p^2 + k^2c^2)/v_{\rm th}^2$ [30]. The linear growth rate of our simulation coincides well with theoretical prediction [30] and other existing simulations [31,32].

In Fig. 9, we show time development of the beam, plasma electron density, and plasma ion density. The purely growing perturbations center on points of enhanced beam density, which magnetically attract nearby beam electrons and repel plasma electrons. Thus, the beam splits into filaments, each of which self-pinches. When n_b reaches $n_{b0} + n_{p0}$, plasma electrons are totally excluded from the filaments.

In order to benchmark the robustness of the new LAPINS code, we here apply second-order LAPINE, fourth-order LAPINE, and recently developed LAPINS codes to the same beam plasma system. For following four groups of simulations, simulation time and length are normalized to 3.33 fs and 1 μm. The size of all simulation box is fixed at 5 μm $\times$ 5 μm with 100 $\times$ 100 cells. The simulation time step is fixed at 0.08 fs. The plasma densities are set up to 1×10^{21} cm^{-3} (for group 1), 1×10^{22} cm^{-3} (for group 2), 1×10^{23} cm^{-3} (for group 3), and 1×10^{24} cm^{-3} (for group 4). The beam has density $n_{b0}/n_{p0} = 0.1$ and Lorentz factor $\gamma_{b0} = 2.5$. In Fig. 10, the energy density distribution of electromagnetic field is shown at $t = 15$. From the first row in Fig. 10, we can see that when the density of background plasma is of $n_p = 1 \times 10^{21}$ cm^{-3}, the calculated instability patterns coincide with each quite well for second-order LAPINE, fourth-order LAPINE, and the newly developed LAPINS codes. From the second row, when we increase the density of background plasma to $n_p = 1 \times 10^{22}$ cm^{-3}, we can see, although the calculated instability patterns show some slight differences, that the filament distribution and size are still of the same for all three codes. While from the third row, when we increase the density of background plasma to $n_p = 1 \times 10^{23}$ cm^{-3}, we can see the calculated instability pattern from the second-order LAPINE code is completely swallowed by numerical noise; the calculated instability pattern from the fourth-order LAPINE code is already disturbed slightly; and, in contrast, the calculated instability pattern from the LAPINES code still shows distinct filament structures. Finally, when we further increase the density of background plasma to $n_p = 1 \times 10^{24}$ cm^{-3}, as shown in the fourth row, only the LAPINS code survives.

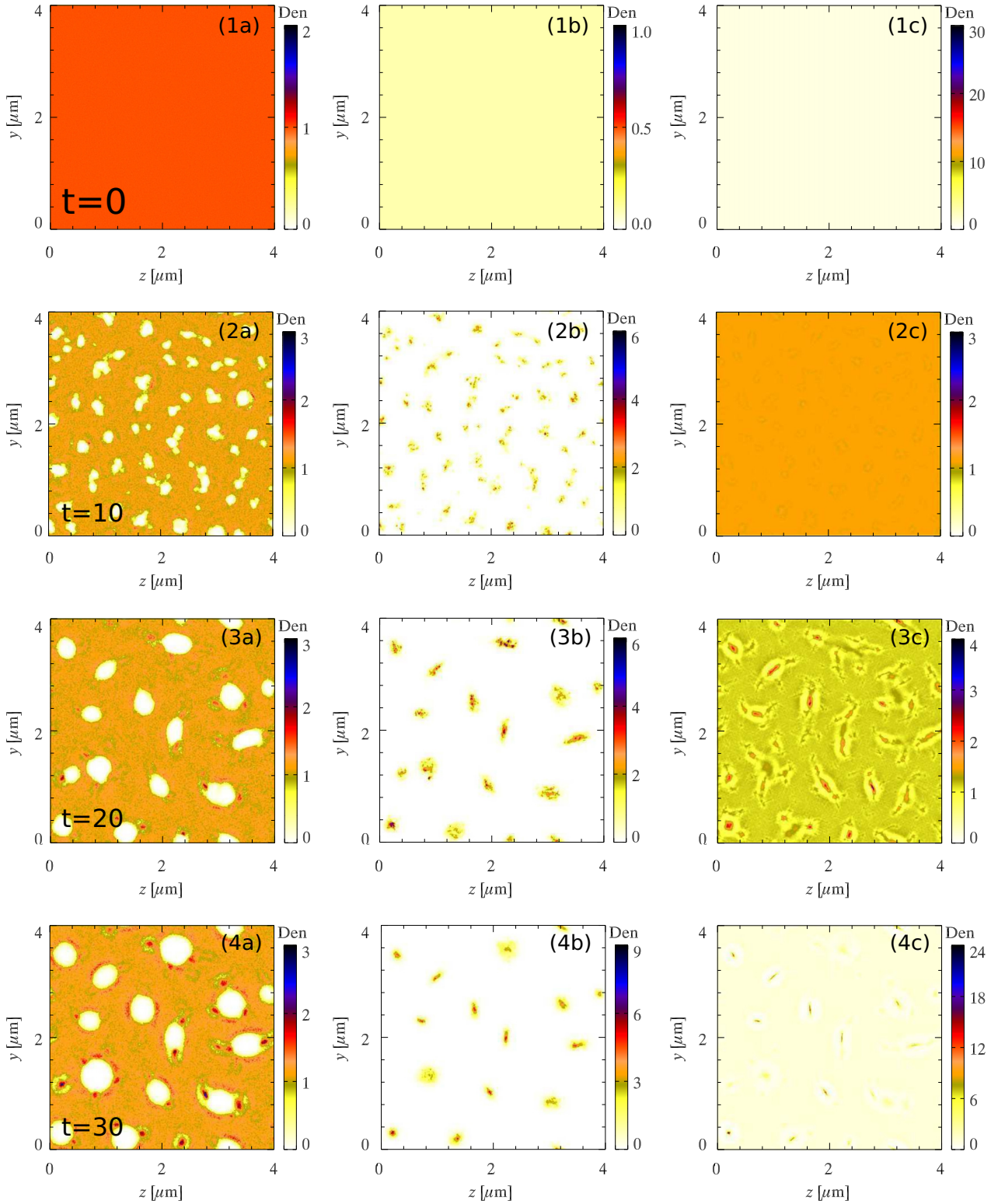

FIG. 9. The evolution of plasma electron density (first column), beam density (second column), and plasma ion density (third column) as a function of time. The densities are in units of 1×10^{21} cm^{-3}, and time is in units of 3.33 fs. For all simulations, the size of the simulation box is fixed as 4 μm $\times$ 4 μm with 400 $\times$ 400 cells and the time step is fixed the same. See text for the detailed explanation.

D. Relativistic intense laser-solid interactions

In this section, we have carried out 2D-3V PIC simulations to benchmark the laser plasma interaction at solid densities. Here, a gold (Au, density of 19.32 g/cm^3) target is selected as an example in this simulation. The target is modeled as a uniform slab of 3 μm thickness with a small preplasma

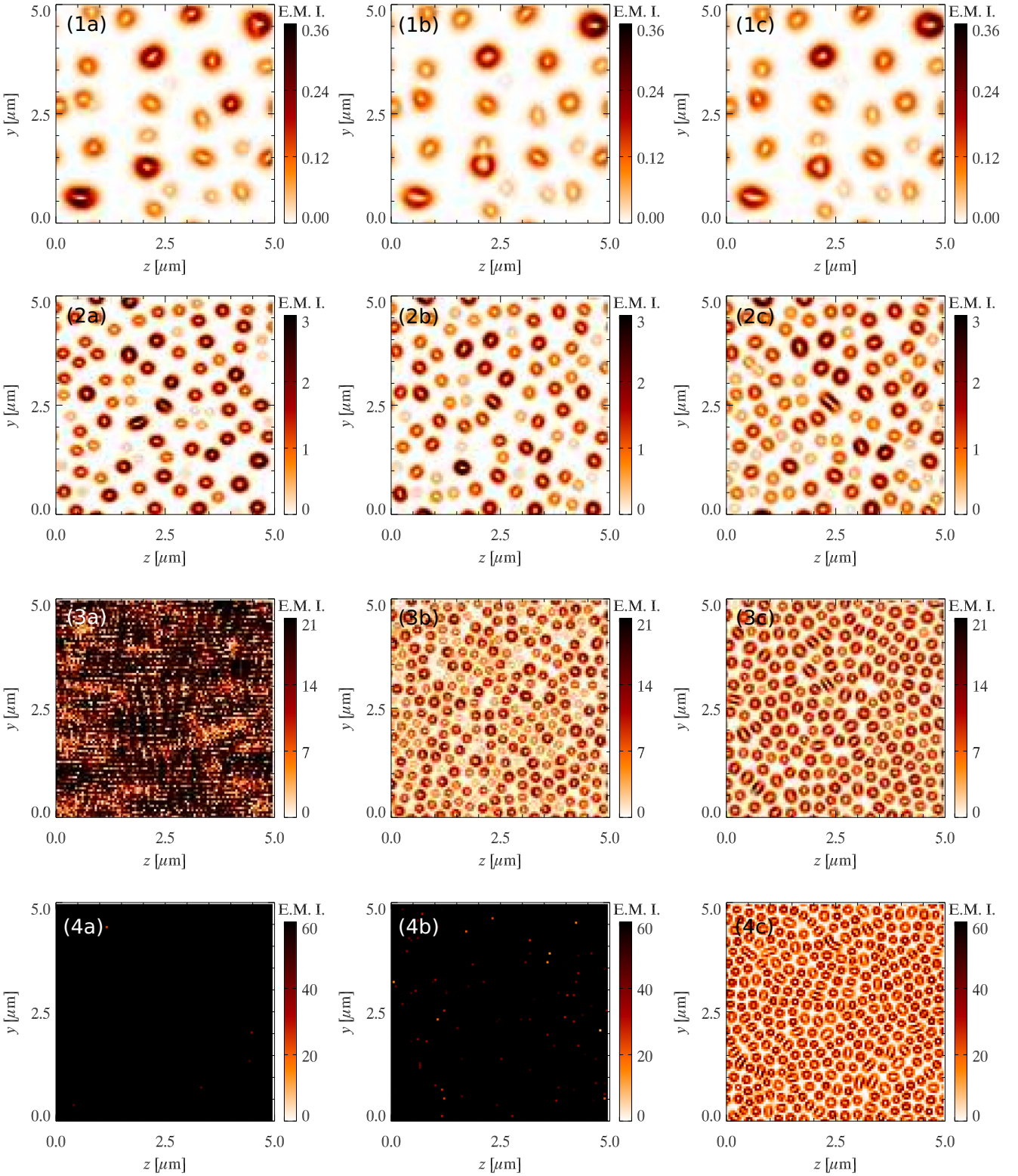

FIG. 10. The energy density distributions of electromagnetic fields at $t = 15$. Here the electromagnetic energy density is defined as $(1/2)(\mathbf{E}'^2 + \mathbf{B}'^2)$. The different rows, (a), (b), (c), and (d), correspond to different simulation cases with background plasma densities $n_p = 1 \times 10^{21}$ cm^{-3}, $n_p = 1 \times 10^{22}$ cm^{-3}, $n_p = 1 \times 10^{23}$ cm^{-3}, and $n_p = 1 \times 10^{24}$ cm^{-3}, respectively. The first column is run by LAPINE code, the second column is run by high-order LAPINE code, and the third column is run by recently developed LAPINS code. For all simulations, the size of the simulation box is fixed as 5 μm $\times$ 5 μm with 100 $\times$ 100 cells and the time step is fixed the same. See text for the detailed explanation.

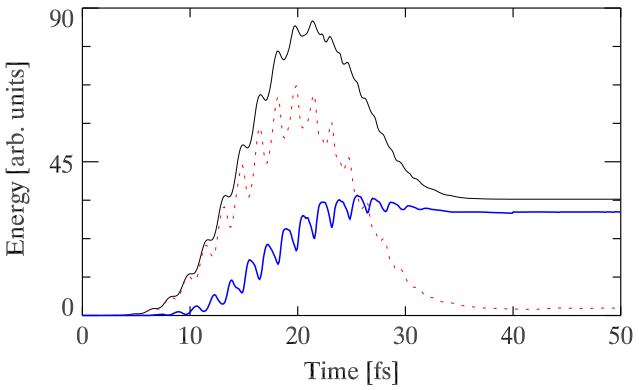

FIG. 11. The energy (transfer) as a function of time. Data are shown for laser energy entering into the simulation box (black), electromagnetic energy (dashed red), and particle kinetic energy (thick blue).

in front of the target. The initial temperature of Au solid is chosen to be 0.1 eV. The initial ionization degrees of Au is $Z' = 5$. The simulations were carried out in the $Z - Y$ Cartesian geometry with laser propagation in the Z direction. The size of the simulation box is chosen to be Z (5 μm) $\times$ Y (10 μm), which is divided into 500 $\times$ 500 uniform grids. The laser pulse has a predefined profile of form $e^{-r^2/r_0^2}\sin^2(\pi t/2\tau_0)$, with intensity 2.5×10^{19} W/cm^2, $r_0 = 2.5\lambda_0$, $r_0 = 2.5\lambda_0$, $\tau_0 = 5T_0$ where T_0 is the laser cycle. The central wavelength of the laser is $\lambda_0 = 1$ μm, and therefore laser period T_0 equals 3.33 fs. The laser pulse is normally incident on the target. In the Z and Y directions, absorbing boundary conditions are applied for both particles and laser field. We here also take advantage of the recently developed ionization [17,18] and collision dynamics [19] models that help calculate the ionization charge state and conductivity of target quite precisely according to the local plasma and electromagnetic fields conditions within the simulations.

Figure 11 shows the temporal evolution of energy (transfer) into the simulation box, including laser energy entering (black), electromagnetic field energy (dashed red), and the particle kinetic energy (thick blue). From this temporal evolution, the energy conservation of the code is well benchmarked for laser-solid interactions. Moreover, based on their time evolution, we obtain further insights into the whole laser-solid interaction processes. Note that the laser energy enters into the simulation box is calculated as $\mathscr{E}'_{\rm pyt} = \sum(\mathbf{E}' \times \mathbf{B}')\delta y'\delta t'$, where $\delta t'$ is the time step of the simulation box. We find that $\mathscr{E}'_{\rm pyt}$ first increases and then arrives at the maximum value at $t = 20$ fs, when the whole laser is entirely injected into the simulation box. Afterward, to $t = 35$ fs, we find that there is a strong energy transfer from the electromagnetic energy to the particle-kinetic energy, which we briefly refer to as the laser plasma interaction stage in front of the target. Afterward, to the end of simulation, the energy transfer from electromagnetic fields to particles is slow, which we refers to as the hot electrons' transport stage within the target.

In Fig. 12, the electron kinetic energy density, free electron density, and static magnetic field (averaged over 3.3 fs) are presented at $t = 35$ fs and $t = 50$ fs, respectively. In

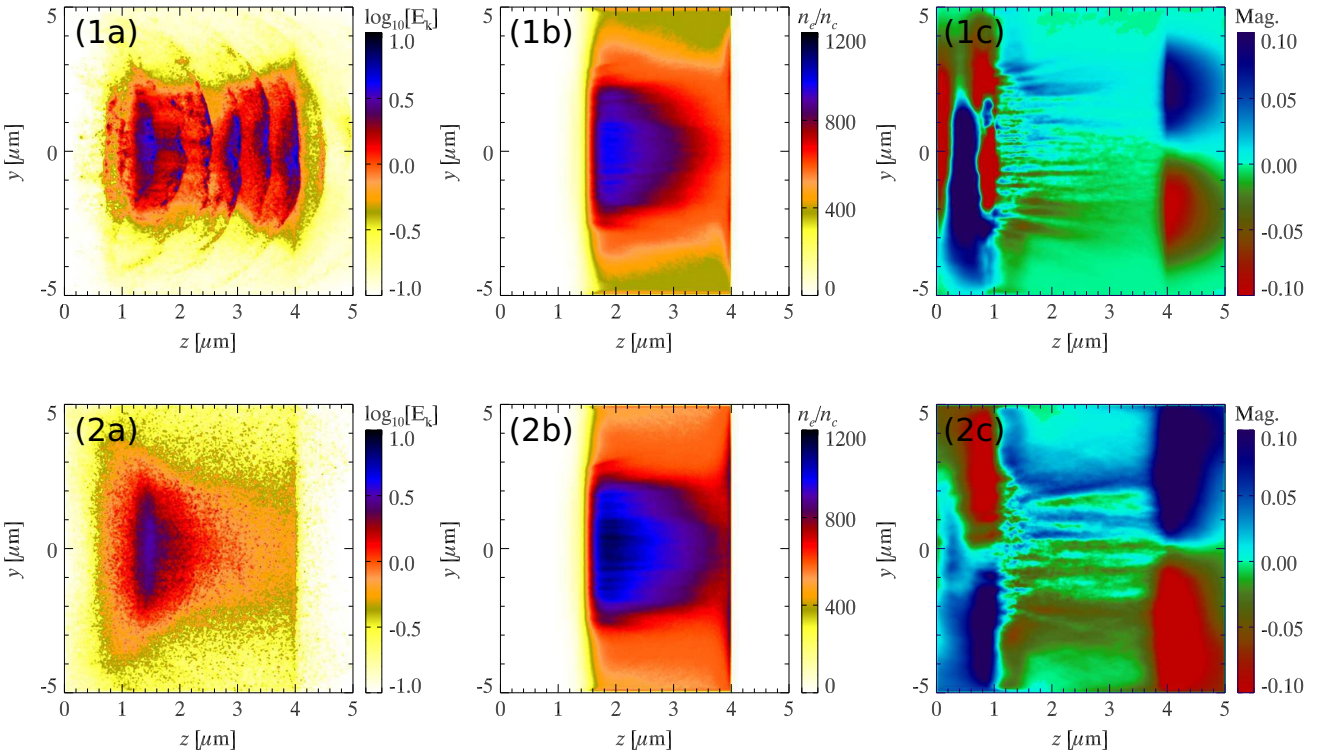

FIG. 12. The distribution of electron kinetic energy, electron density, and static magnetic field. The first row shows values at $t = 35$ fs and the second row shows at $t = 50$ fs. This simulation is produced by LAPINS code.

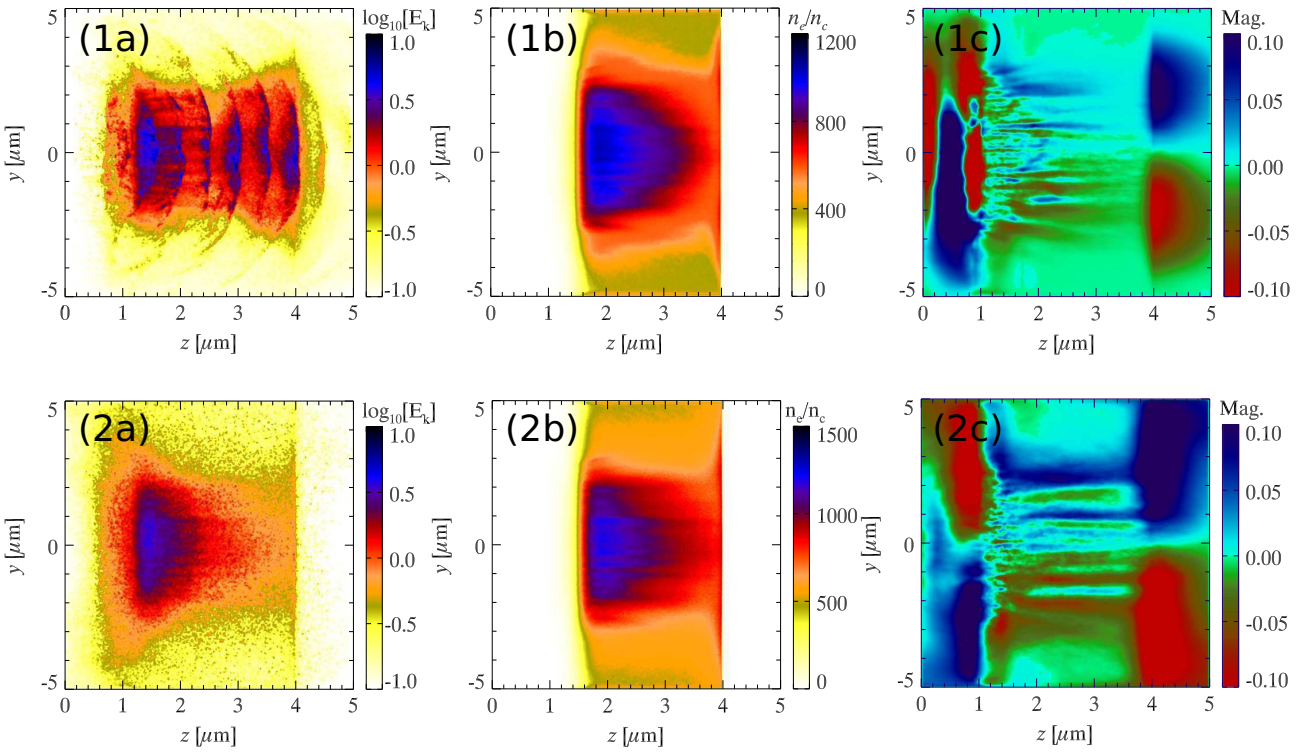

FIG. 13. The distribution of electron kinetic energy, electron density, and static magnetic field. The first row shows values at $t = 35$ fs and the second row shows at $t = 50$ fs. This simulation is produced by high-order LAPINE code approximated with the layered-density method.

Fig. 12(1a), we find several bunches of energetic electrons. Such energetic electrons are produced in front of the target by direct laser acceleration [33–35]. When passing through the target, abundant plasma and atomic interactions take place therein. As shown in Fig. 12(2a), the remaining target is significantly heated up. Moreover, collision ionization would dramatically increase the electron density. Typically, the ionization and plasma energy density decrease rapidly along the electron propagation direction. In Fig. 12(b), we notice some filamentation structures for electron density. In Fig. 12(c), strong static magnetic fields are produced, and some filamentation structures also appear on the magnetic fields. As we have shown in the last benchmark, this filamentation is due to Weibel instabilities, when energetic electron beams pass through high-density plasmas. While Weibel instabilities alone cannot explain why such static magnetic field is so strong, there is another source that might induce strong magnetic fields [36], $-\partial \mathbf{B}'/\partial t' = \nabla' \times \mathbf{E}'$. When such energetic electron beam propagates through the solid, the forward $\mathbf{J}'_e$ is quickly neutralized by the return current. A resistive electric field $\mathbf{E}'_z = \eta' \mathbf{J}'_e$ might be produced, where η' is the target resistivity. Such a resistive electric field then induces a resistive magnetic field.

We here have presented a full physics simulation of intense laser plasma interaction for a Au target. As the new code LAPINS completely eliminates numerical instabilities found in explicit PIC methods, the calculation burden is significantly reduced. For such a simulation, only 1000 CPU h are consumed. For typical PIC code, a full physics simulation of laser-solid Au interactions is a great challenge. Here we need to note that in our previous work, we used to propose a layered-density method, which was especially designed for laser-solid interactions. For a detailed explanation, one can refer to Ref. [16]. In Fig. 13, the electron kinetic energy density, electron density, and static magnetic field are also presented at $t = 35$ fs and $t = 50$ fs, respectively. When we compare Fig. 12 and Fig. 13, two simulation results coincide with each other quite well.

IV. SUMMARY AND DISCUSSION

A high-order implicit multidimensional PIC method is developed for simulating plasmas at solid densities. The field solver algorithm completely eliminates numerical instabilities found in explicit PIC methods. Moreover, this algorithm eliminates the numerical cooling found in the standard implicit PIC methods by using a pseudoelectric field method. The particle pusher algorithm combines the standard Boris's particle pusher with a Newton-Krylov iteration method. This algorithm increases the precision accuracy by several orders of magnitude when compared with a standard Boris particle pusher and also significantly decreases the iteration time when compared with the pure Newton-Krylov method. The code is tested with several benchmarks, including Weibel instability, and relativistic laser plasma interactions at both low and solid densities.

In the future, the LAPINS code will be intensively compared with experiments and applied to uncover the material dependence feature of laser-solid interactions, including laser-driven ion acceleration, fast ignition inertial

confinement fusion, laser-driven x- and γ-ray sources, and the production of warm dense matter. Moreover, it is also planned to predict new physics that are not well revealed but strongly related to applications.

ACKNOWLEDGMENTS

This work was supported by Science Challenge Project (No. TZ2016005) and National Natural Science Foundation of China (No. 11605269).

[1] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (Taylor & Francis, New York, 2005).

[2] A. Macchi, M. Borghesi, and M. Passoni, Rev. Mod. Phys. **85**, 751 (2013).

[3] H. Daido, M. Nishiuchi, and A. S. Pirozhkov, Rep. Prog. Phys. **75**, 056401 (2012).

[4] D. Wu, C. Y. Zheng, C. T. Zhou, X. Q. Yan, M. Y. Yu, and X. T. He, Phys. Plasmas **20**, 023102 (2013).

[5] D. Wu, C. Y. Zheng, B. Qiao, C. T. Zhou, X. Q. Yan, M. Y. Yu, and X. T. He, Phys. Rev. E **90**, 023101 (2014).

[6] D. Wu, W. Yu, Y. T. Zhao, S. Fritzsche, and X. T. He, Matter and Radiation at Extremes **3**, 293 (2018).

[7] G. M. Petrov, D. P. Higginson, J. Davis, B. Petrova, J. M. McNaney, C. McGuffey, B. Qiao, and F. N. Beg, Phys. Plasmas **19**, 093106 (2012).

[8] L. Van Woerkom, K. U. Akli, T. Bartal, F. N. Beg, S. Chawla, C. D. Chen, E. Chowdhury, R. R. Freeman, D. Hey, M. H. Key, J. A. King, A. Link, T. Ma, A. J. MacKinnon, A. G. MacPhee, D. Offermann, V. Ovchinnikov, P. K. Patel, D. W. Schumacher, R. B. Stephens, and Y. Y. Tsui, Phys. Plasmas **15**, 056304 (2008).

[9] A. G. MacPhee, L. Divol, A. J. Kemp, K. U. Akli, F. N. Beg, C. D. Chen, H. Chen, D. S. Hey, R. J. Fedosejevs, R. R. Freeman, M. Henesian, M. H. Key, S. Le Pape, A. Link, T. Ma, A. J. Mackinnon, V. M. Ovchinnikov, P. K. Patel, T. W. Phillips, R. B. Stephens, M. Tabak, R. Town, Y. Y. Tsui, L. D. Van Woerkom, M. S. Wei, and S. C. Wilks, Phys. Rev. Lett. **104**, 055002 (2010).

[10] X. T. He, J. W. Li, Z. F. Fan, L. F. Wang, J. Liu, K. Lan, J. F. Wu, and W. H. Ye, Phys. Plasmas **23**, 082706 (2016).

[11] M. Tabak, J. Hammer, M. E. Glinsky, W. L. Kruer, S. C. Wilks, J. Woodworth, E. M. Campbell, M. D. Perry, and R. J. Mason, Phys. Plasmas **1**, 1626 (1994).

[12] T. Ma, H. Sawada, P. K. Patel, C. D. Chen, L. Divol, D. P. Higginson, A. J. Kemp, M. H. Key, D. J. Larson, S. Le Pape, A. Link, A. G. MacPhee, H. S. McLean, Y. Ping, R. B. Stephens, S. C. Wilks, and F. N. Beg, Phys. Rev. Lett **108**, 115004 (2012).

[13] T. G. White, N. J. Hartley, B. Borm, B. J. B. Crowley, J. W. O. Harris, D. C. Hochhaus, T. Kaempfer, K. Li, P. Neumayer, L. K. Pattison, F. Pfeifer, S. Richardson, A. P. L. Robinson, I. Uschmann, and G. Gregori, Phys. Rev. Lett. **112**, 145005 (2014).

[14] H. Xu, W. W. Chang, H. B. Zhuo, L. H. Cao, and Z. W. Yue, Chin. J. Comput. Phys. **19**, 305 (2002).

[15] T. Esirkepov, Comput. Phys. Comm. **135**, 144 (2001).

[16] D. Wu, X. T. He, W. Yu, and S. Fritzsche, High Power Laser Sci. Eng. **6**, e50 (2018).

[17] D. Wu, B. Qiao, C. McGuffey, X. T. He, and F. N. Beg, Phys. Plasmas **21**, 123118 (2014).

[18] D. Wu, X. T. He, W. Yu, and S. Fritzsche, Phys. Rev. E **95**, 023208 (2017).

[19] D. Wu, X. T. He, W. Yu, and S. Fritzsche, Phys. Rev. E **95**, 023207 (2017).

[20] D. Wu, B. Qiao, and X. T. He, Phys. Plasmas **22**, 093108 (2015).

[21] G. M. Petrov and J. Davis, Phys. Plasmas **18**, 073102 (2011).

[22] A. Kempf, U. Ganse, P. Kilian, and F. Spanier, J. Comput. Phys. **237**, 56 (2013).

[23] B. I. Cohen, A. B. Langdon, and D. W. Hewett, J. Comput. Phys. **81**, 151 (1989).

[24] B. Marder, J. Comput. Phys. **68**, 48 (1987).

[25] A. B. Langdon, Comput. Phys. Commun. **70**, 447 (1992).

[26] S. Markidis and G. Lapenta, J. Comput. Phys. **230**, 7037 (2011).

[27] J. P. Boris, *Proceedings of the Fourth Conference on Numerical Simulation of Plasmas* (Naval Research Laboratory, Washington DC, 1970).

[28] S. E. Parker and C. K. Birdsall, J. Comput. Phys. **97**, 91 (1991).

[29] D. Wu, W. Yu, S. Fritzsche, C. Y. Zheng, and X. T. He, Phys. Plasmas **26**, 063107 (2019).

[30] R. Lee and M. Lampe, Phys. Rev. Lett. **31**, 1390 (1973).

[31] M. Honda, J. Meyer-ter-Vehn, and A. Pukhov, Phys. Plasmas **7**, 1302 (2000).

[32] L. Siddi, G. Lapenta, and P. Gibbon, Phys. Plasmas **24**, 082103 (2017).

[33] D. Wu, S. I. Krasheninnikov, S. X. Luan, and W. Yu, Nucl. Fusion **57**, 016007 (2017).

[34] D. Wu, S. I. Krasheninnikov, S. X. Luan, and W. Yu, Phys. Plasmas **23**, 123116 (2016).

[35] D. Wu, S. X. Luan, J. W. Wang, W. Yu, J. X. Gong, L. H. Cao, C. Y. Zheng, and X. T. He, Plasma Phys. Control. Fus. **59**, 065004 (2017).

[36] P. Leblanc and Y. Sentoku, Phys. Rev. E **89**, 023109 (2014).