AIP Physics of Plasmas

Kinetic approach to microscopic-macroscopic coupling in space and laboratory plasmas

Giovanni Lapenta, J. U. Brackbill, and Paolo Ricci

Citation: Phys. Plasmas **13**, 055904 (2006); doi: 10.1063/1.2173623

View online: http://dx.doi.org/10.1063/1.2173623

View Table of Contents: http://pop.aip.org/resource/1/PHPAEN/v13/i5

Published by the American Institute of Physics.

Kinetic approach to microscopic-macroscopic coupling in space and laboratory plasmas[a)]

Giovanni Lapenta[b)]
Plasma Theory Group, Theoretical Division, Los Alamos National Laboratory, Mail Stop: K717, Los Alamos, New Mexico 87545

J. U. Brackbill
Particle Solutions, Portland, Oregon 97214

Paolo Ricci
Department of Physics and Astronomy, Dartmouth College, Hanover, New Hampshire 03755

(Received 24 October 2005; accepted 19 January 2006; published online 10 May 2006)

Kinetic plasma simulation typically requires to handle a multiplicity of space and time scales. The implicit moment particle in cell (PIC) method provides a possible route to address the presence of multiple scales effectively. Here, a new implementation of the implicit moment method is described. The present paper has two goals. First, the most modern implementation of the implicit moment method is described. While many of the algorithms involved have been developed in the past, the present paper reports for the first time how the implicit moment method is currently implemented and what specific algorithms have been found to work best. Second, we present the CELESTE3D code, a fully electromagnetic and fully kinetic PIC code, based on the implicit moment method. The code has been in use for a number of years but no previous complete description of its implementation has been provided. The present work fills this gap and introduces a number of new methods not previously presented: a new implementation of the Maxwell solver and a new particle mover based on a Newton-Krylov nonlinear solver for the discretized Newton's equations. A number of benchmarks of CELESTE3D are presented to shown the typical application and to investigate the improvements introduced by the new solver and the new mover. © *2006 American Institute of Physics*. [DOI: 10.1063/1.2173623]

I. INTRODUCTION

The presence of multiple scales in space and time is a defining property of plasmas. The macroscopic evolution of the whole system often is relatively slow and develops on system scales, but is tightly coupled with smaller and faster scales. Small scales determine dissipations and enable processes such as topological changes of the magnetic field (e.g., reconnection) that are not otherwise possible. From the perspective of simulation, two approaches can be considered: fluid and kinetic.[1]

Fluid models are most suitable to describe the overall system scales but are not completely consistent and require some information from microscopic kinetic models to deal with the coupling between microscopic and macroscopic scales. Conversely, kinetic models are most suitable for the microscopic scales but are extremely costly to extend to full system scales. However, it must be noted that progress in computer speed and memory, software engineering, and algorithms has extended the applicability of the full kinetic description to significant macroscopic processes. In space plasma physics, global hybrid (fluid electrons and kinetic ions) simulations of large scale problems in the Earth's magnetosphere[1,2] are common and in fusion devices gyrokinetic transport simulations of tokamak plasmas are a crucial tool in modern fusion research.[3,4]

However, hybrid and gyrokinetic models still are based on a some limitations. Gyrokinetic models are based on averaging the kinetic equations over the gyroradius of the plasma species.[3] The length scales resolved by the gyrokinetic approach are limited by the condition that $\rho_s/L_{\mathrm{eq}} \ll 1$, where L_{eq} is the typical equilibrium scale and ρ_s is the gyroradius of species s. The time scales covered by the gyrokinetic model are limited by $\tau\omega_{cs} \gg 1$, where τ is the typical temporal scale of evolution and ω_{cs} is the cyclotron frequency. Furthermore, small electron scales, such as the electron Debye length, are typically eliminated. Hybrid models are based on fluid electrons and cannot properly describe processes that depend on the electron physics, such as electron acceleration, heating, and most importantly resistivity and viscosity. Hybrid models rely on using appropriate models to summarize the electron behavior and can become unreliable when such models fail. As an example, unless proper models of high order moments such as the pressure tensor and heat flux are included, the hybrid model can lead to unphysical results.[5] The predictive capability of hybrid models is limited by their reliance on fluid electrons.

We consider here a fully kinetic approach that removes both the need to gyroaverage and to assume quasineutrality. The assumption of quasineutrality leads to the removal of the fastest (Langmuir waves) and smallest (Debye) scales. In

[a)]Paper GI2b 1, Bull. Am. Phys. Soc. **50**, 137 (2005).
[b)]Invited speaker. Electronic address: lapenta@lanl.gov

most cases, such scales are indeed very small compared with the scales thought to be relevant. However, charge separation can be important in affecting large scales as recent work on reconnection has shown.[6,7]

The gyroaveraging approach is valid in most current experimental fusion devices where the electron and ion gyroradius are smaller than the scales of interest, satisfying the condition of validity of the gyrokinetic approach. In ITER, as in any future thermonuclear burning experiment, the presence of alpha particles (helium nuclei) will change this situation. As an example, in ITER, the gyroradius of deuterium at 10 keV is of the order of a tenth of a centimeter, but the gyroradius of alpha particles generated by fusion at 3.5 MeV is of the order of 10 cm. The latter scale can in some cases be too large to allow a gyroaveraged model. A complete kinetic model will be needed to address the behavior of alpha particles in ITER, while electrons and ions can still be gyroaveraged.

The implicit particle in cell (PIC) method was developed as a general plasma simulation tool based on a fully kinetic approach[8,9] that did not rely on any physical approximation but only on the use of advanced numerical methods to reduce the cost of large scale kinetic simulations. As in any area of scientific simulation, the basic approaches are appealing for their direct natural interpretation but are subject to severe numerical problems. In its simplest form the PIC method uses explicit numerical techniques that require to resolve all scales, including the Debye length and the Langmuir plasma frequency.[10] The need for hybrid and gyrokinetic approach stems precisely from this limitation.[1] However, the implicit PIC method provide an alternative where the smallest scales are not eliminated but are kept in a approximate inexact way sufficient to deal with their coupling with larger scales.[11]

The implicit PIC method has been developed for a number of years since its original introduction. Two main approaches have emerged: the direct implicit method[12] and the implicit moment method.[11] Here we consider the implicit moment method. In the past few years, we have published a number of papers on the application of CELESTE3D (e.g., see Ref. 13 and 14 and references therein) but the code CELESTE3D itself has never been described in the published literature. In the present work, we describe for the first time the combination of algorithms used in CELESTE3D. While some of the algorithms used by CELESTE3D are not new, their combined use and the reasons for selecting the methods we have selected are new.

The present manuscript reports two types of new developments in the implicit moment PIC method. First, some of the methods presented here are new: a new particle mover and a new field solver are presented here and their performance is tested. Moreover, other algorithms used in CELESTE3D are based on previous work but have been modified to obtain better performances. Second, the overall combination of schemes and algorithms that compose CELESTE3D is in itself new. To make CELESTE3D work successfully a number of algorithms and tools needed to be deployed and refined. Here we describe precisely how this is done.

The paper is organized as follows. Section II summarizes the methods used in CELESTE3D, focusing primarily on the specific implementation choices made to obtain the best performances. Section II introduces two new algorithms: the particle mover and the field solver. Section III discusses a number of benchmarks used for the validation and verification of CELESTE3D. Section IV analyzes the performances of the new particle mover and filed solver.

II. MOMENT IMPLICIT PIC

The fundamental plasma description for a large variety of physical systems in laboratory, space and astrophysics is the Boltzmann-Maxwell model.

In the derivations below, we consider the full set of Maxwell's equations in SI units:

$$\begin{aligned} &\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0, \\ &\nabla \times \mathbf{B} - \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} = \mu_0 \mathbf{J}, \\ &\epsilon_0 \nabla \cdot \mathbf{E} = \rho, \\ &\nabla \cdot \mathbf{B} = 0, \end{aligned} \tag{1}$$

and the Boltzmann equation for the distribution function $f_s(\mathbf{x},\mathbf{v},t)$ in phase space for each species s:

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \frac{\partial f_s}{\partial \mathbf{x}} + \frac{q_s}{m_s}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_s}{\partial \mathbf{v}} = \mathrm{St}(\{f_s\}). \tag{2}$$

The last term in the Boltzmann equation is the collision integral.

In the present work we will assume classic and collisionless plasmas. The extension to relativistic (special or general) plasmas is possible[15] but not trivial and beyond the scope of the present paper. The inclusion of a collision term is also possible and has indeed been done within the framework of implicit PIC.[16]

Below we will focus on the implicit PIC solution of the classic Vlasov-Maxwell model.

A. Particle in cell (PIC) approach

The fundamental assumption of the particle in cell method is that the distribution function of each species s can be described by a superposition of N_s finite elements (called superparticles or computational particles) labeled by p and of the form

$$f_s(\mathbf{x},\mathbf{v},t) = \sum_{p=1}^{Ns} S(x-x_p)S(y-y_p)S(z-z_p)\,\delta(\mathbf{v}-\mathbf{v}_p), \tag{3}$$

where S is the so-called shape function and δ is Dirac's delta. The shape function is defined to be symmetric and with a unitary integral and is typically chosen as a b spline of order ℓ[17] in each direction. Introducing a particle and cell size Δx in the x direction, we define $S(x-x_p)=b_l[(x-x_p)/\Delta x]$ and similarly in the other directions. A typical example is the use of b splines of order 1, leading to the so-called cloud in cell (CIC) scheme:[18]

$$b_l(\xi) = \begin{cases} 1 - |\xi| & \text{if } |\xi| < 1 \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

Each superparticle represents a small element of phase space with finite size in each spatial direction but localized in velocity. Each element is characterized by a constant (in time) shape and by two parameters, the superparticle position $\mathbf{x}_p$ and velocity $\mathbf{v}_p$. We note that this has no relationship with Eulerian methods where the distribution function is described in terms of a spline fit.[19]

The linearity of the definition in Eq. (3) ensures that the evolution of each superparticle is still described by the Vlasov equation. Substituting Eq. (3) in the Vlasov equation (2), one derives the equations for the evolution of the parameters $\mathbf{x}_p$ and $\mathbf{v}_p$:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p, \qquad \frac{d\mathbf{v}_p}{dt} = \mathbf{a}_p. \tag{5}$$

The derivation of the PIC method directly from the Vlasov equation provides also a unique definition of the average acceleration acting on the finite-sized superparticle as $\mathbf{a}_p = \mathbf{E}_p + \mathbf{v}_p \times \mathbf{B}_p$, where

$$\mathbf{E}_p = \int_V \mathbf{E}(\mathbf{x}) S(\mathbf{x} - \mathbf{x}_p) d\mathbf{x}, \qquad \mathbf{B}_p = \int_V \mathbf{B}(\mathbf{x}) S(\mathbf{x} - \mathbf{x}_p) d\mathbf{x} \tag{6}$$

over the computational domain V.

The solution of the evolution equations requires the solution of the Maxwell's equation. The second defining property of the PIC method is the use of a grid to do so. The information needs to be carried between particles and grid. To this end, the interpolation function is introduced defined for each spatial direction as

$$W(x_c - x_p) = \int_{-\infty}^{\infty} S(x - x_p) b_0\left(\frac{x - x_c}{\Delta x}\right) dx, \tag{7}$$

where we used the flat-top function (or b spline of order 0) equal to one only within the cell span Δx and zero otherwise. We limit the attention here to uniform grids, but the generalization to adaptive grids is straightforward.[20]

The computation of W is simplified if we recall the following general property of the b splines:

$$b_{l+1}(\xi) = \int_{-\infty}^{+\infty} b_l(\xi') b_0(\xi - \xi') d\xi'. \tag{8}$$

The interpolation function in the x direction simply becomes $W(x - x_p) = b_{l+1}(x - x_p)/\Delta x$, and similarly in the other directions.

Using the interpolation functions, the particle fields can be expressed more conveniently as

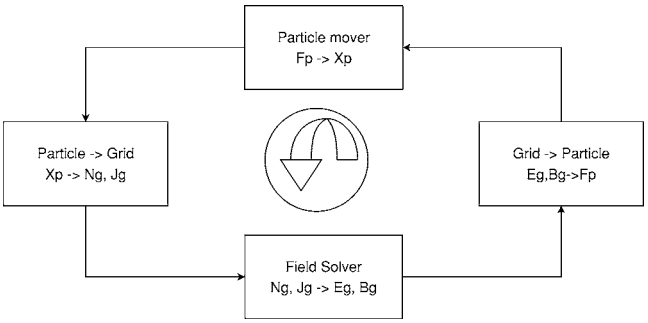

FIG. 1. Blocks needed in a cycle of the PIC scheme.

$$\mathbf{E}_p = \sum_g \mathbf{E}_g W(\mathbf{x} - \mathbf{x}_p), \qquad \mathbf{B}_p = \sum_g \mathbf{B}_g W(\mathbf{x} - \mathbf{x}_p), \tag{9}$$

where the cells are labeled with a single index g and the field values in each cell are $\mathbf{E}_g$ and $\mathbf{B}_g$.

Similarly the moments of the distribution function needed for solving the field equations can be obtained easily integrating the definition of the superparticle:

$$\rho_{s,g} = \sum_{p=1}^{N_s} q_p W(\mathbf{x} - \mathbf{x}_p), \qquad \mathbf{J}_{s,g}(\mathbf{r}) = \sum_{p=1}^{N_s} q_p \mathbf{u}_p W(\mathbf{x} - \mathbf{x}_p), \tag{10}$$

where q_q is the charge of the superparticle p and the sum is extended to particles belonging to species s.

Figure 1 summarizes the steps required for solving the Vlasov-Maxwell model: the equations of motion (5) need to be solved to advance the superparticle positions and velocities, the new superparticle position and velocity are used to compute the moments using Eq. (10). The field equations are solved next and the fields on the particle are obtained from Eq. (9).

The fundamental issue of relevance here is that the motion and field equations are coupled via the fields. The equations of motion need the fields computed over the superparticle ($\mathbf{E}_p$, $\mathbf{B}_p$) and the field equations need the moments $\rho_{s,g}$ and $\mathbf{J}_{s,g}$. The great majority of PIC codes in use today address this problem relying on a explicit method.

In the explicit method, the field equations are discretized with one in a number of different explicit methods available (see Refs. 10 and 18 for a review) and the equations of motion are usually discretized with the leapfrog algorithm.[10,18] The key point is that in the explicit method, the field equations need only the sources from the previous time cycle and the equations of motions need only the fields from the previous time cycle. Even though the equations remain coupled, no iteration is needed and the cycle depicted in Fig. 1 becomes a simple marching order where each block

is applied after its predecessor and needs only information already available. This choice makes the explicit PIC very simple.

As can be imagined this simplicity comes at a price. The explicit PIC approach is subject to three very restrictive stability constraints.

First, the explicit discretization of the field equations requires that a Courant condition must be satisfied on the fastest traveling wave speed, the speed of light:

$$c\Delta t < \Delta x. \tag{11}$$

Second, the explicit discretization of the equations of motion introduces a constraint related to the fastest electron response time, the electron plasma frequency:

$$\omega_{pe}\Delta t < \Delta x. \tag{12}$$

Note that, instead, thanks to the Boris algorithm for the motion in a magnetic field, the gyromotion introduces no stability constraints.[10]

Finally, the interpolation between grid and particles causes a loss of information and a aliasing instability called finite grid instability that results in an additional stability constraint,

$$\Delta x < \varsigma\lambda_{\mathrm{De}}, \tag{13}$$

that requires the grid spacing to be of the order of the Debye length or smaller, the proportionality constant ς being of order one and dependent on the details of the scheme deployed.[10,18]

Thanks to ever faster computers, the explicit PIC method has been able to achieve remarkable results over the 50 years since its invention. But there are problems where multiple scales still prevent its use. When large scales and slow processes are what is of interest other approaches are needed. The implicit PIC approach provides a viable alternative in such problems.

B. Implicit PIC

The implicit plasma simulation method was introduced[8,9,11,12] with the goal of retaining a full kinetic model but eliminating many of the numerical stability constraints on the time and space steps described above. Below we describe the latest implementation of the moment implicit PIC method developed for the CELESTE3D code.

We consider first the time discretization. The solution is advanced in the time domain with discrete time steps, Δt, from the initial time, $t^0=0$, to the final time, $t^N=T$. The generic quantity Ψ at time step n ($t=t^n$) is denoted with Ψ^n.

The equations of motion are discretized as[21,22]

$$\begin{aligned}
&\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \mathbf{v}_p^{n+1/2}\Delta t,\\
&\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \frac{q_s\Delta t}{m_s}[\mathbf{E}_p^{n+\theta}(\mathbf{x}_p^{n+1/2}) + \mathbf{v}_p^{n+1/2}\times\mathbf{B}_p^n(\mathbf{x}_p^{n+1/2})],
\end{aligned} \tag{14}$$

where all quantities evaluated at intermediate values are computed as $\Psi^{n+\theta}=\Psi^n(1-\theta)+\Psi^{n+1}\theta$. Note that the velocity equation is more conveniently rewritten as

$$\mathbf{v}_p^{n+1/2} = \hat{\mathbf{v}}_p + \beta_s\hat{\mathbf{E}}_p^{n+\theta}(\mathbf{x}_p^{n+1/2}), \tag{15}$$

where $\beta_s=q_p\Delta t/m_p$ (independent of the particle weight and unique to a given species). For convenience, we have introduced hatted quantities obtained by explicit transformation of quantities known from the previous computational cycle:

$$\begin{aligned}
&\hat{\mathbf{v}}_p = \boldsymbol{\alpha}_s^n\cdot\mathbf{v}_p^n,\\
&\hat{\mathbf{E}}_s^{n+\theta} = \boldsymbol{\alpha}_s^n\cdot\mathbf{E}_s^{n+\theta}.
\end{aligned} \tag{16}$$

The transformation tensor operators $\boldsymbol{\alpha}_s^n$ are defined as

$$\boldsymbol{\alpha}_s^n = \frac{1}{1+(\beta_s B^n)^2}(\boldsymbol{I}-\beta_s\boldsymbol{I}\times\mathbf{B}^n+\beta_s^2\mathbf{B}^n\mathbf{B}^n), \tag{17}$$

and represent a scaling and rotation of the velocity vector.

Semidiscrete, continuous space, temporal discretization to Maxwell's equations is written as

$$\begin{aligned}
&\nabla\times\mathbf{E}^{n+\theta} + \frac{1}{c}\frac{\mathbf{B}^{n+1}-\mathbf{B}^n}{\Delta t} = 0,\\
&\nabla\times\mathbf{B}^{n+\theta} - \frac{1}{c}\frac{\mathbf{E}^{n+1}-\mathbf{E}^n}{\Delta t} = \frac{4\pi}{c}\mathbf{J}^{n+1/2},\\
&\nabla\cdot\mathbf{E}^{n+\theta} = 4\pi\rho^{n+\theta},\\
&\nabla\cdot\mathbf{B}^n = \nabla\cdot\mathbf{B}^{n+1} = 0.
\end{aligned} \tag{18}$$

The parameter $\theta\in[1/2,1]$ is chosen in order to adjust the numerical dispersion relation for electromagnetic waves (for $\theta<1/2$, the algorithm is not unconditionally stable[11]). We note that for $\theta=1/2$ the scheme is second-order accurate in Δt; for $1/2<\theta\leqslant 1$ the scheme is first-order accurate.

The interpolation formulas derived above provide the expression for the fields in the equations of motion and for the sources in Maxwell's equations. The coupling of the set of equations is evident.

The fundamental problem to address in developing an implicit PIC method is the coupling between the equations of motion and the field equations for the presence of the time advanced electric field (but not magnetic field, that is used from the previous cycle, as no instability is introduced) in the equations of motion and for the appearance [due to the presence of $\mathbf{x}_p$ and $\mathbf{v}_p$ in Eq. (10)] of the particle properties in the sources of the Maxwell equations. In both cases the coupling is implicit, so that the new particle properties need to be known before the fields can be computed and likewise the new fields need to be available before the new particle properties can be computed.

While a direct solution of the coupled system for fields and particle motion is now within the reach of modern computational methods and computer resources,[23] we focus here on the implicit moment method that removes the need for iterative methods and provides a direct method to compute the advanced fields without also having to move the particles.

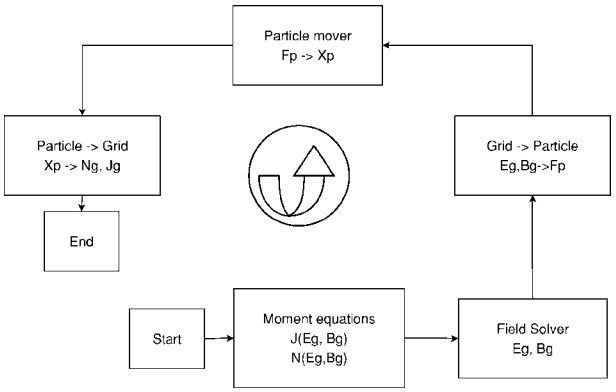

FIG. 2. Blocks needed in a cycle of the implicit moment PIC scheme.

C. Moment implicit method

The implicit moment method reduces the number of equations that must be solved self-consistently to a set of coupled fluid moment and field equations. The solution of these equations implicitly, and the subsequent explicit solution of the particle equations of motion in the resulting fields, is stable and accurate.

The idea is summarized in Fig. 2 where the logic of the sequence of operations involved in the implicit moment method is shown. The coupling due to the implicit discretization of both field and particle equations is broken by approximating the sources of the field equations using the moment equations instead of the particle equations directly. Once the field equations are solved within this approximation, the rest of the steps can be completed directly without iterations: with the new fields, the particle equations of motion can be solved and the new current and density can be computed for the next computational cycle.

The implicit moment method formulation used here is described in Refs. 11 and 21. The key step is to derive a suitable set of moment equations that can approximate the particle motion over a computational cycle. The approach followed in the present implementation of the implicit moment method is based on a series expansion of the interpolation function appearing in the expression for the field sources, eq. (10). The expansion is done with respect to the particle position, choosing the center of the expansion as the particle position at the start of the computational cycle:

$$W(\mathbf{x}-\mathbf{x}_p^{n+1}) = W(\mathbf{x}-\mathbf{x}_p^n) + (\mathbf{x}-\mathbf{x}_p^n)\cdot\nabla W(\mathbf{x}-\mathbf{x}_p^n) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_p^n)(\mathbf{x}-\mathbf{x}_p^n):\nabla\nabla W(\mathbf{x}-\mathbf{x}_p^n) + \cdots, \tag{19}$$

where a tensor notation is used.

The expansion (19) can be used to compute the field sources directly using particle information only from the previous computational cycle and removing the need to iterate the particle and field equations. The details of the simple but tedious algebraic manipulations are provided in Ref. 21, the final answer being

$$\rho_s^{n+1} = \rho_s^n - \Delta t\,\nabla\cdot\mathbf{J}_s^{n+1/2},$$
$$\mathbf{J}_s^{n+1/2} = \hat{\mathbf{J}}_s - \frac{\Delta t}{2}\boldsymbol{\mu}_s\cdot E_\theta - \frac{\Delta t}{2}\nabla\cdot\hat{\boldsymbol{\Pi}}_s, \tag{20}$$

where the following expressions were defined:

$$\hat{\mathbf{J}}_s = \sum_p q_p\hat{\mathbf{v}}_p W(\mathbf{x}-\mathbf{x}_p^n),$$
$$\hat{\boldsymbol{\Pi}}_s = \sum_p q_p\hat{\mathbf{v}}_p\hat{\mathbf{v}}_p W(\mathbf{x}-\mathbf{x}_p^n), \tag{21}$$

with the obvious meaning, respectively, of current and pressure tensor based on the transformed hatted velocities. An effective dielectric tensor is defined to express the feedback of the electric field on the plasma current and density:

$$\boldsymbol{\mu}_s^n = -\frac{q_s\rho_s^n}{m_s}\boldsymbol{\alpha}_s^n. \tag{22}$$

The expression (20) for the sources of the Maxwell's equations provides a direct and explicit closure of Maxwell's equations. When Eq. (20) is inserted in Eq. (18), Maxwell's equations can be solved without further coupling with the particle equations. This is the key property of the moment implicit method and allows the implicit PIC method to retain the once-through approach typical of explicit methods and eliminates the need for expensive iteration procedures that would require to move the particles multiple times per each computational cycle.

D. Field solver

Previous versions of the implicit PIC method have relied on different solution procedures for the Maxwell equations. In the 1D version of CELESTE3D, a potential formulation was applied,[21] while early versions of CELESTE3D were based on a time-staggered solution of the two divergence Maxwell's equations and of a second-order formulation of the two curl Maxwell's equations.[28] More recently a consistent second-order formulation has been presented[24] and is used here.

In the continuum, the two curl Maxwell's equations give a solution with the property of satisfying also the two divergence equations at all times, provided that appropriate initial and boundary conditions are used. Based on this property,[24] we solve the following second-order equation obtained from the two time-discretized (but continuous in space) curl equations in Eq. (18):

$$(c\theta\Delta t)^2\left[-\nabla^2\mathbf{E}^{n+\theta} - \nabla\nabla\cdot(\boldsymbol{\mu}^n\cdot\mathbf{E}^{n+\theta})\right] + \boldsymbol{\epsilon}^n\cdot\mathbf{E}^{n+\theta} = \mathbf{E}^n + (c\theta\Delta t)\left(\nabla\times\mathbf{B}^n - \frac{4\pi}{c}\hat{j}^n\right) - (c\theta\Delta t)^2\nabla 4\pi\hat{\rho}^n, \tag{23}$$

where $\boldsymbol{\mu}^n=\Sigma_s\boldsymbol{\mu}_s^n$ and $\boldsymbol{\epsilon}^n=\mathbf{I}+\boldsymbol{\mu}^n$, where $\mathbf{I}$ is the identity tensor.

As shown in Ref. 24, the second-order formulation for the electric field needs to be coupled with a divergence cleaning step to ensure that

$$\epsilon_0 \nabla \cdot \mathbf{E}^n = \rho^n \tag{24}$$

holds for the initial field of each time step.

The boundary conditions for the second-order formulation for **E** can be derived from the natural boundary conditions expressed in both **E** and **B** using theorems of classical electrodynamics.[24] The magnetic field is computed directly once the new electric field is computed as

$$\frac{\mathbf{B}^{n+1} = \mathbf{B}^n}{\Delta t} + \nabla \times \mathbf{E}^{n+\theta}. \tag{25}$$

Using the temporal second-order formulation above, the field equations are further discretized in space using a finite volume approach described in Ref. 25. The discretized equations and their boundary conditions form a nonsymmetric linear system that is solved using generalized minimal residual algorithm (GMRES).[26] For the divergence cleaning equation, conjugate gradient (CG) [or fast Fourier transform (FFT) on a uniform grid][26] can be used since the discretized equation leads to a symmetric matrix.

The field solver is implemented defining a Krylov vector of unknowns constituted by all three components of the electric field on all nodes, including the boundary nodes. The corresponding residuals are provided by the second-order formulation of Maxwell's equation (23) in the interior and by the boundary conditions indicated in Ref. 24 on the boundary nodes. GMRES is used as solver to minimize the residual within a prescribed relative tolerance.

E. Particle mover

Once the fields are computed from the implicit moment method, the particles can be advanced. The set of equations of motions (14) have an inner coupling among themselves due to the appearance of the new velocity in the equation for the position and of the new position in the equation for the velocity. This coupling is purely local at the level of each particle and does not involve coupling with other particles.

In previous implementations of the implicit PIC method, a predictor-corrector (PC) approach was used to solve the nonlinear system of equations of motions.[22] Here, we employ the Newton-Krylov (NK) method[27] to solve the set of equations of motion to convergence.

Instead of simply solving the six coupled equations of motions considering position and velocity as unknowns, we use the formal linearity of the explicit quadrature of the velocity provided in Eq. (15). Substituting the first of the Newton equations (14) in the second, the nonlinear (vectorial) function expressing the discretized Newton equations is

$$\mathbf{f}(\mathbf{x}_p^{n+1}) = \mathbf{x}_p^{n+1} - \mathbf{x}_p^n - \Delta t \hat{\mathbf{v}}_p - \Delta t \beta_s \hat{\mathbf{E}}_p^{n+\theta}(\mathbf{x}_p^{n+1/2}). \tag{26}$$

The three residual equations above have only three unknowns, the components of $\mathbf{x}_p^{n+1}$, all other quantities being known.

The residual equation $\mathbf{f}(\mathbf{x}_p^{n+1})=0$ is solved using the NK method. A crucial feature of the NK method is the fact that the Jacobian matrix never needs to be computed or formed, and the method uses a series of residual function evaluations.[27] The NK method is based on solving iteratively

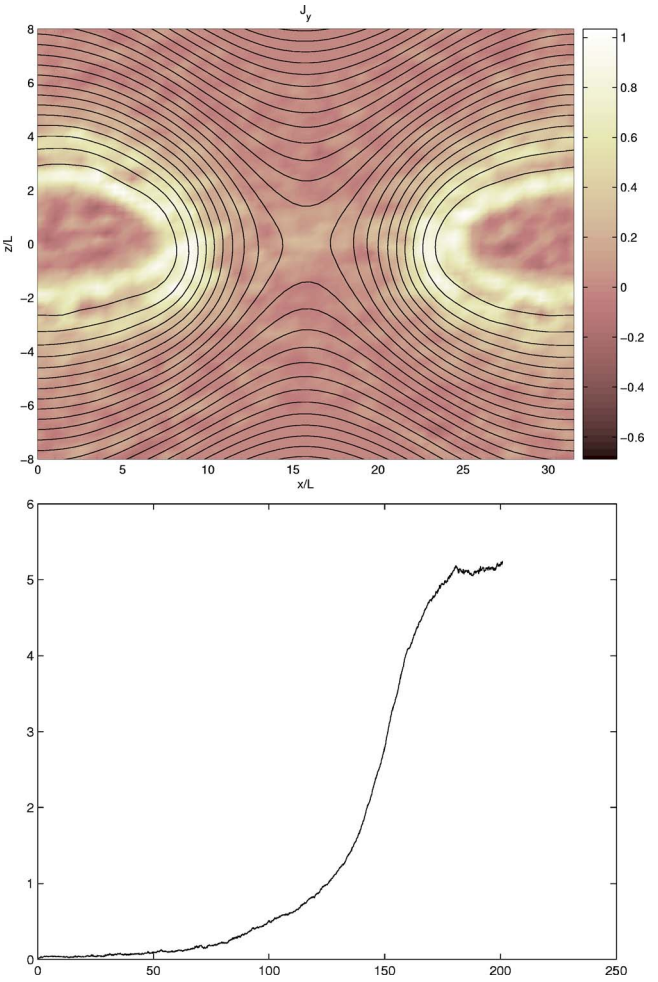

FIG. 3. Newton challenge: (a) Final state at $\omega_{ci}t=200$. The flux surfaces are superimposed over the current density. (b) Time evolution of the reconnected flux.

the residual equation. The new iteration $\mathbf{x}_p^{n+1,(k+1)} = \mathbf{x}_p^{n+1,(k)} + \delta\mathbf{x}_p$ is obtained by computing the displacement $\delta\mathbf{x}_p$ using the Newton method,[27] i.e., using a Taylor series expansion truncated at first order (to ensure linearity of the resulting equation for $\delta\mathbf{x}_p$). To avoid forming and storing the Jacobian matrix needed for the Taylor series expansion, the equation for $\delta\mathbf{x}_p$ is approximated numerically by replacing the derivatives with differences:

$$\mathbf{f}(\mathbf{x}_p^{n+1,(k)}) + \frac{\mathbf{f}(\mathbf{x}_p^{n+1,(k)} + \epsilon\delta\mathbf{x}_p) - \mathbf{f}(\mathbf{x}_p^{n+1,(k)})}{\epsilon} = 0. \tag{27}$$

While formally nonlinear, Eq. (27) is linearized by choosing ϵ small enough to ensure the accurate replacement of the first-order derivative with the difference in the Taylor expansion. In practice, ϵ is chosen according to the machine precision.[27] The Jacobian equation (27) is expressed only in terms of residual function evaluations and there is no need to form or store the matrix. To solve Eq. (27), we use the GMRES algorithm since the Jacobian matrix can be nonsymmetric.[26]

The use of the new mover outlined above ensure solution of the equations of motion to convergence of the nonlinear iteration, the old PC mover instead used a fixed num-

TABLE I. Total energy error, defined as $[\mathcal{E}(T)-\mathcal{E}(0)]/\mathcal{E}(0)$ for the PC and NK scheme for the particle solver.

Δt	$\omega_{ci}/\omega_{pi}=0$		$\omega_{ci}/\omega_{pi}=1$		$\omega_{ci}/\omega_{pi}=10$	
	PC	NK	PC	NK	PC	NK
0.001	2.1×10^{-3}	2.1×10^{-3}	2.6×10^{-11}	2.6×10^{-11}	1.3×10^{-13}	1.3×10^{-13}
0.01	5.5×10^{-3}	5.6×10^{-3}	1.9×10^{-11}	1.9×10^{-11}	1.5×10^{-14}	1.5×10^{-14}
0.1	5×10^{4}	2.7×10^{-3}	1.1×10^{-10}	1.0×10^{-10}	1.1×10^{-13}	1.1×10^{-13}
0.5	1×10^{11}	1.6×10^{-3}	9.7×10^{-4}	7.1×10^{-4}	9.8×10^{-6}	9.8×10^{-6}
1	7×10^{14}	4.9×10^{-3}	2.3×10^{-3}	1.5×10^{-4}	2.1×10^{-5}	2.1×10^{-5}

ber of iterations allowing for an uncontrolled error. This feature leads to an improved energy conservation in the overall scheme. We have recently extended the mover to treat relativistic systems.[15]

F. Stability

The stability properties of the method described above have been studied extensively in the past.[11] All the stability constraints discussed above for the explicit method are removed. The implicit particle mover removes the need to resolve the electron plasma frequency, and the implicit formulation of the field equations removes the need to resolve the speed of light.

The time step constraints are replaced by an accuracy limit arising from the derivation of the fluid moment equations using the series expansion in Eq. (19). This limit restricts the mean particle motion to one grid cell per time step,[11] i.e.,

$$v_{\mathrm{th},e}\Delta t/\Delta x < 1. \tag{28}$$

The finite grid instability limit for the explicit method, $\Delta x < \varsigma\lambda_{\mathrm{De}}$, is replaced by[11]

$$\Delta x/\Delta t < \varsigma v_{th,e}, \tag{29}$$

that allows large grid spacings to be used when large time steps are taken. The gain afforded by the relaxation of the stability limits is twofold.

First, the time step can far exceed the explicit limit. In a typical plasma the electron plasma frequency is far smaller than the time scales of interest and its accurate resolution is not needed. Within the current approach, the processes developing at the sub-Δt scale are averaged and their energy is damped by a numerically enhanced Landau damping. In other approaches, such as the gyrokinetic or hybrid approach,[1] such processes are completely removed and the energy channel towards them is interrupted, removing for example the possibility to exchange energy between sub-Δt fluctuations and particles. In the implicit approach, instead, the sub-Δt scales remain active and the energy channel remains open. This is a crucial feature to retain in full kinetic approach. Furthermore, when additional resolution of the smallest scales is needed, the implicit method can access the same accuracy of the explicit method simply using a smaller time step and grid spacing. This feature is not accessible to reduced model, e.g., gyroaveraged methods, that remove the small scales entirely.

Second, the grid spacing can far exceed the Debye length. Often the scales of interest are much larger than the Debye length. The ability to retain a full kinetic treatment without the need to resolve the Debye length results in a much reduced cost for the implicit PIC method.

III. VERIFICATION AND VALIDATION

The implicit moment PIC method described above is implemented in the CELESTE3D code. The CELESTE3D code was originally conceived for the numerical tokamak project[28,29] but has found its main application in space physics.

Below we summarize first the benchmarks we have conducted to validate and verify CELESTE3D when applied to multiple scale problems. Next we discuss the performance of the code, focusing particularly on the gains afforded by the two innovations introduced in the present paper: the new mover and the new Maxwell solver.

Five types of tests have been conducted to verify and validate CELESTED in full 3D Cartesian geometry and in reduced geometries in 2D and 1D.

First, unit testing of all methods (or subroutines in Fortran terminology) have been conducted. PIC codes are extremely modular and naturally prone to be coded in a object-oriented, component-based framework. CELESTE3D is currently written in Fortran 90 but a new implementation in C++ has been recently developed for the new code project PARSEK.[30]

Second, classic tests have been conducted of well known benchmarks including shocks, the Weibel instability, Landau damping, and ion acoustic waves.[21]

TABLE II. Number of GMRES iterations required to reduce the residual of the discretized Maxwell equation (23) by 3 orders of magnitude. Different time steps, dimensionality, and grids are used. In all cases the standard configuration of the GEM challenge is assumed initially. The number of iterations is averaged over the first 200 iterations.

N_x	N_y	N_z	$\omega_{pi}\Delta t=0.01$	$\omega_{pi}\Delta t=0.1$
25	25	25	4	6
50	50	50	5	9
25	1	25	4	12
50	1	50	4	10
1	25	25	4	14
1	50	50	4	10

Third, we have participated in two community-based benchmarking campaigns: the GEM challenge[24,31,32] and the Newton challenge.[33]

The GEM challenge is a standardized undriven reconnection problem where reconnection in a Harris sheet is initialized by adding a initial perturbation.[31] We have shown that CELESTE3D captures the large scale dynamics correctly, computing the correct reconnection rate and onset time.[24] Furthermore and more importantly, CELESTE3D includes correctly the effect of the electron kinetic response even though the Debye length and the electron skin depth were not resolved. This is a crucial finding supported by two observations. First, we showed the ability to compute correctly the electron acceleration and the divergence of the electron pressure tensor in the outflow region of a reconnection site, even when only the ion skin depth was resolved.[24] Second, we confirmed by simulation the electron scaling laws in the reconnection region, conducting simulations up to a mass ratio of 1836 (the physical value for hydrogen).[32]

The Newton challenge is an extension of the GEM challenge where an initially thicker layer is compressed by a gentle boundary perturbation localized in space and time.[33] The typical onset time and reconnection rate of the Newton challenge are observed in CELESTE3D in agreement with the other kinetic codes used in the campaign.[33] Figure 3 shows the typical reconnected flux evolution and the final reconnected state.

Fourth, we have considered a study of reconnection in systems with low betas. The systems are representative of the conditions present in fusion devices or in some astrophysical systems where a dominant toroidal field is present and reconnection is developing in the poloidal plane. In this case we have investigated the reconnection process both at the macroscopic and microscopic level obtaining agreement with the explicit PIC code NPIC.[14]

Fifth, we validated the results of a 3D stability study of a current sheet equilibrium with satellite observations obtained from the CLUSTER and GEOTAIL mission.[34]

IV. PERFORMANCE

In the present paper we have proposed a new scheme for the particle mover and we analyze the improvements it can provide. Table I shows the error on total energy conservation, for a uniform system composed by electrons and ions, with a mass ratio of $m_i/m_e=1836$ and temperature ration $T_e/T_i=10$ and with $v_{\mathrm{th},e}/c=0.1$. The problem is simulated in a 1D 64 cells grid for a total number of 100 cycles using different time steps. We report the average increase in total energy per time step. Clearly the new scheme based on the Newton-Krylov (NK) method provides an extended range of stability and better energy conservation when compared with the old predictor-corrector (PC) scheme.

The performance of the field solver is illustrated in Table II. Even with no preconditioning the number of iterations remains very low and largely insensitive to the number of cells in any direction. A general trend appearing in Table II and confirmed by other tests not shown can be observed. The field solver tends to require fewer iterations on full 3D grids than in reduced 2D or 1D grids where one or two directions are reduced to just one cell. In the reduced case, the system is still completely determined and the problem well posed, but the coupling is tighter and the number of iterations required is larger. While a preconditioner might be beneficial, the performance without it are adequate and the field solver contributes only a small fraction of the computational time. The large number of particles typically used in a PIC code makes particle operations far dominant over field operations.

ACKNOWLEDGMENTS

Work supported by the United States Department of Energy, under contract No. W-7405-ENG-36 and by the NASA Sun Earth Connection Theory Program.

[1] A. S. S. Lipatov, *The Hybrid Multiscale Simulation Technology* (Springer, Berlin, 2002).
[2] D. Krauss-Varban and N. Omidi, Geophys. Res. Lett. **22**, 3271 (1995).
[3] W. Lee, J. Comput. Phys. **72**, 243 (1987).
[4] Z. Lin, T. S. Hahm, W. W. Lee, W. M. Tang, and R. B. White, Science **281**, 1835 (1998).
[5] G. Lapenta and J. Brackbill, Geophys. Res. Lett. **23**, 1713 (1996).
[6] W. Daughton, G. Lapenta, and P. Ricci, Phys. Rev. Lett. **93**, 105004 (2004).
[7] J. F. Drake, M. Swisdak, C. Cattell, M. A. Shay, B. N. Rogers, and A. Zeiler, Science **299**, 873 (2003).
[8] R. Mason, J. Comput. Phys. **41**, 233 (1981).
[9] J. Denavit, J. Comput. Phys. **42**, 337 (1981).
[10] C. Birdsall and A. Langdon, *Plasma Physics Via Computer Simulation* (Taylor & Francis, London, 2004).
[11] J. Brackbill and D. Forslund, J. Comput. Phys. **46**, 271 (1982).
[12] A. Langdon, B. Cohen, and A. Friedman, J. Comput. Phys. **51**, 107 (1983).
[13] G. Lapenta and J. Brackbill, Nonlinear Processes Gyophys. **7**, 151 (2000).
[14] P. Ricci, J. Brackbill, W. Daughton, and G. Lapenta, Phys. Plasmas **11**, 4102 (2004).
[15] G. Lapenta, G. Zuccaro, and C. Tronci, J. Comput. Phys. (to be published).
[16] C. W. Cranfill, J. U. Brackbill, and S. R. Goldman, J. Comput. Phys. **66**, 239 (1986).
[17] C. De Boor, *A Practical Guide to Splines* (Springer, Berlin, 1978).
[18] R. Hockney and J. Eastwood, *Computer Simulation Using Particles* (Taylor & Francis, London, 1988).
[19] M. Feix, P. Bertrand, and A. Ghizzo, Series Advanc. Math. Appl. Sci. **22**, 45 (1994).
[20] J. Brackbill, J. Comput. Phys. **108**, 38 (1993).
[21] H. X. Vu and J. U. Brackbill, Comput. Phys. Commun. **69**, 253 (1992).
[22] H. Vu and J. Brackbill, J. Comput. Phys. **116**, 384 (1995).
[23] H. Kim, L. Chacòn, and G. Lapenta, Bull. Am. Phys. Soc. **50**, 2913 (2005).
[24] P. Ricci, G. Lapenta, and J. Brackbill, J. Comput. Phys. **183**, 117 (2002).
[25] D. Sulsky and J. U. Brackbill, J. Comput. Phys. **96**, 339 (1991).
[26] Y. Saad, *Iteractive Methods for Sparse Linear Systems* (SIAM, Philadelphia, 2003).
[27] C. T. Kelley, *Iterative Methods For Linear And Nonlinear Equations* (SIAM, Philadelphia, 1995).
[28] J. Brackbill, in *Proceedings of the First International Conference on Supercomputing in Nuclear Applications, Mito, Japan, March 12-16* (JAERI, Kashiwa, 1990), pp. 192–197.
[29] J. Brackbill and G. Lapenta, Bull. Am. Phys. Soc. **39**, 1665 (1994).
[30] S. Markidis and G. Lapenta, Bull. Am. Phys. Soc. **50**, 1995 (2004).

[31] J. Birn, J. Drake, M. Shay, B. Rogers, R. Denton, M. Hesse, M. Kuznetsova, Z. Ma, A. Bhattacharjee, A. Otto, *et al.*, J. Geophys. Res. **106**, 3715 (2001).
[32] P. Ricci, G. Lapenta, and J. Brackbill, Geophys. Res. Lett. **29**, 10.1029/2002GL015314 (2002).
[33] J. Birn, K. Galsgaard, M. Hesse, M. Hoshino, J. Huba, G. Lapenta, P. L. Pritchett, K. Schindler, L. Yin, J. Buchner, *et al.*, Geophys. Res. Lett. **32**, L06105 (2005).
[34] P. Ricci, G. Lapenta, and J. Brackbill, Geophys. Res. Lett. **31**, L06801, doi:10.1029/2003GL019207 (2004).