Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation xxx (2009) xxx–xxx
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Multi-scale simulations of plasma with iPIC3D

Stefano Markidis [a,b], Giovanni Lapenta [c,*], Rizwan-uddin [a,d]

[a] *Nuclear, Plasma, and Radiological Engineering, University of Illinois at Urbana-Champaign, 104 S. Wright St. Urbana, IL 61801, USA*
[b] *Lawrence Berkeley National Laboratory, 1 Cyclotron Road Berkeley, CA 94720, USA*
[c] *Centrum voor Plasma-Astrofysica and Leuven Mathematical Modeling & Computational Science Centre, Departement Wiskunde, Katholieke Universiteit Leuven, Celestijnenlaan 200B, 3001 Leuven, Belgium*
[d] *National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, 1205 W. Clark St. Urbana, IL 61801, USA*

Received 10 February 2009; received in revised form 22 August 2009; accepted 23 August 2009

Abstract

The implicit Particle-in-Cell method for the computer simulation of plasma, and its implementation in a three-dimensional parallel code, called iPIC3D, are presented. The implicit integration in time of the Vlasov–Maxwell system, removes the numerical stability constraints and it enables kinetic plasma simulations at magnetohydrodynamics time scales. Simulations of magnetic reconnection in plasma are presented to show the effectiveness of the algorithm.

Keywords: Particle-in-Cell; Computational plasma physics; Implicit PIC; 3D Magnetic reconnection

1. Introduction

Computer simulations of a plasma are inherently multi-scale because they require to study phenomena, spanning over a very large time-scale and length-scale range. The presence of electrons, protons and heavier plasma species with different masses and the multitude of different collective modes and instabilities, cause the presence of multiple scales. For instance, the time scales of phenomena occurring in a magnetic confinement fusion reactor range from the picosecond period of electron oscillations (plasma and gyro periods), to the several seconds of the plasma confinement required for self-sustained nuclear fusion [10,37,36]. The large time gap between picosecond and second is filled with characteristic period times of protons and α particles oscillations, with the transit times of waves, and with growth times of plasma instabilities.

Two main approaches are used for plasma computer simulation: the fluid and the kinetic approaches. Fluid methods, such as magnetohydrodynamics (MHD) method, capture the macroscopic evolution of the system, but they become inaccurate for problem in which the detailed kinetic processes (wave–particle interactions, wave trapping phenomena) affect the macroscopic behavior of the plasma. The second approach is kinetic, and it aims at knowing the behavior of plasma using the kinetic equation of plasmas. This approach accurately models the plasma from first principles but it is computationally expensive. Often in plasmas, the macroscopic evolution that develops relatively slowly, is strongly coupled with smaller and faster phenomena where kinetic effects are predominant. For instance, the topological changes

* Corresponding author.
E-mail addresses: smarkidis@lbl.gov (S. Markidis), giovanni.lapenta@wis.kuleuven.be (G. Lapenta), rizwan@illinois.edu (Rizwan-uddin).

0378-4754/$36.00 © 2009 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.matcom.2009.08.038

Please cite this article in press as: S. Markidis, et al., Multi-scale simulations of plasma with iPIC3D, Math. Comput. Simul. (2009), doi:10.1016/j.matcom.2009.08.038

of magnetic field configuration on large scales, such as the magnetic reconnection, are not possible without kinetic mechanisms at small scales [4,25]. For these reasons, to retain the kinetic effects is very important in order to describe correctly the overall evolution of the system. The implicit Particle-in-Cell (PIC) is based on a fully kinetic approach but it removes the need to resolve small time scales (plasma frequencies) without eliminating them. The unresolved scales are kept in an approximate way allowing the coupling with slower scales that are fully resolved by the large time step.

In the present paper, the implementation of the implicit moment PIC method [23,6,20] in the iPIC3D code is presented. After describing the Vlasov–Maxwell system and the PIC method in Sections 2 and 3, the algorithm of the implicit PIC algorithm is discussed in Section 4. Section 5 summarizes the numerical methods used in iPIC3D and it describes the implementation details. Section 6 shows one of the application of the implicit PIC to the study of plasmas in magnetic confinement fusion machines.

2. Governing equations

Plasmas of interest, such as the plasma in use in the magnetic confinement fusion machines, have relatively low density and low collisionality rates [10]. Their collision frequency ($\approx 10^4$ Hz) is orders lower than characteristic plasma frequencies ($\approx 10^{12}$ Hz). Such plasmas can be considered collisionless and the Vlasov equation (transport equation without the collisional term) for the distribution function $f_s(\mathbf{x}, \mathbf{v}, t)$ for each species s applies to them:

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \frac{\partial f_s}{\partial \mathbf{x}} + \frac{q_s}{m_s}\left(\mathbf{E} + \frac{\mathbf{v} \times \mathbf{B}}{c}\right) \cdot \frac{\partial f_s}{\partial \mathbf{v}} = 0 \tag{1}$$

where q_s, m_s are the charge the mass of the species; $\mathbf{E}$, $\mathbf{B}$ are the electric and magnetic fields. The Vlasov equation describes the collisionless transport in phase-space of charged particles moving in the electromagnetic field and it is coupled to the Maxwell's equations. In iPIC3D, the second order formulation of the Maxwell's equations is used. If the div-curl method is used, the first order Maxwell's system can be decomposed by two decoupled second order systems, one involving $\mathbf{E}$ and one involving $\mathbf{B}$[27]. The Maxwell's equation for the electric field in the second order formulation is written as:

$$\nabla^2 \mathbf{E} - \frac{1}{c^2}\frac{\partial^2 \mathbf{E}}{\partial t^2} = \frac{4\pi}{c^2}\frac{\partial \mathbf{J}}{\partial t} + 4\pi \nabla \rho \tag{2}$$

A similar second order equation can be derived for $\mathbf{B}$, but once the electric field is known, the magnetic field $\mathbf{B}$ can be derived simply from the Faraday's Law of induction:

$$\frac{\partial \mathbf{B}}{\partial t} = -c \nabla \times \mathbf{E} \tag{3}$$

The two second order Maxwell's equations provide a solution that satisfy the two divergence equations of the first order formulation ($\nabla \cdot \mathbf{E} = -4\pi\rho$, $\nabla \cdot \mathbf{B} = 0$) at all times also, if appropriate initial and boundary conditions are used [27]. The boundary conditions for the second order formulation for $\mathbf{E}$ can be derived from the natural boundary conditions expressed in both $\mathbf{E}$ and $\mathbf{B}$ using theorems of classical electrodynamics [27]. The charge density ρ and current density, $\mathbf{J}$ provides the link between the Vlasov equation and Maxwell's equations and they are calculated as sum of the zero and first moments of the species distribution function f_s:

$$\rho = \sum_s q_s \int f_s d\mathbf{v} \qquad \mathbf{J} = \sum_s q_s \int \mathbf{v} f_s d\mathbf{v} \tag{4}$$

3. The Particle-in-Cell method

Various numerical techniques have been developed to solve the Vlasov–Maxwell system and they fall in two broad categories: the direct solution of Vlasov equation approach, and the PIC methods. The direct solution approach includes the Fourier–Fourier and Fourier–Hermite transform methods [1] and the finite difference method [24]. These techniques are limited to one- and two-dimensional problems even when supercomputers are used. A three-dimensional problem would require a six-dimensional grid, and it has consequently very high computational and memory cost. The Particle-in-Cell method instead uses computational particles to sample the initial distribution function and moves the particle by the Newton's equation of motion to follow the evolution of distribution function. However, because in practice

Please cite this article in press as: S. Markidis, et al., Multi-scale simulations of plasma with iPIC3D, Math. Comput. Simul. (2009), doi:10.1016/j.matcom.2009.08.038

the number of computational particles is just a small fraction of the number of the real particles in the system, the reconstruction of distribution function is statistically noisy. A number of numerical techniques have been studied to address this problem. For instance in iPIC3D, the computational particles have different statistical weights to provide a good statistical description of the tails of the distribution function [8]. Other techniques include the δ–f PIC [33] and the semi-lagrangian method [32].

The PIC approach is one of the most used numerical method for the solution of the Vlasov–Maxwell system [2,13] because of its capacity to deal with three-dimensional configuration and its implementation simplicity. In the PIC method, the distribution function of each plasma species s of Eq. (1) is described as collection of N_s computational particles with label p. The computational particles represent small elements of phase space with finite size and localized velocity. Each computational particle is characterized by a fixed shape function S and by two parameters, the superparticle position $\mathbf{x}_p$ and velocity $\mathbf{v}_p$. Thus the distribution function f_s can be written as:

$$f_s(\mathbf{x}, \mathbf{v}, t) = \sum_{p=1}^{Ns} S(x - x_p)S(y - y_p)S(z - z_p)\delta(\mathbf{v} - \mathbf{v}_p) \tag{5}$$

where δ is Dirac's delta. The shape function S is symmetric and has a unitary integral by definition. It is typically chosen as a b-spline function of order ℓ [5] in each direction. If Δx is the grid spacing in the x-direction, the shape function is defined as $S(x - x_p) = b_l((x - x_p)/\Delta x)$ and similarly in the other directions. The choice of the first order b-splines as shape functions, leads to the so-called Cloud-in-Cell (CIC) PIC scheme [13]. Because of the linearity of Eq. (5), the evolution of each superparticle p is described by the Vlasov equation also. Substituting Eq. (5) in the Vlasov Eq. (1), the equations for the evolution of the computational particles parameters $\mathbf{x}_p$ and $\mathbf{v}_p$ are derived:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p \qquad \frac{d\mathbf{v}_p}{dt} = \frac{q_s}{m_s}\left(\mathbf{E}_p + \frac{\mathbf{v}_p \times \mathbf{B}_p}{c}\right) \tag{6}$$

The average electric and magnetic fields acting on a computational particle, $\mathbf{E}_p$ and $\mathbf{B}_p$ are defined as the integral of the shape function and of the electromagnetic field over the computational domain V,

$$\mathbf{E}_p = \int_V \mathbf{E}(\mathbf{x})S(\mathbf{x} - \mathbf{x}_p)d\mathbf{x} \qquad \mathbf{B}_p = \int_V \mathbf{B}(\mathbf{x})S(\mathbf{x} - \mathbf{x}_p)d\mathbf{x} \tag{7}$$

Another important characteristic of the PIC algorithm is the use of a grid to solve the Maxwell's equations. In the PIC method the Maxwell's equations are solved on a grid and interpolation functions $W(x_g - x_p)$ to carry information between particles and grid are introduced:

$$W(x - x_p) = \int_{-\infty}^{\infty} S(x - x_p)b_0\left(\frac{x - x_p}{\Delta x}\right)dx = \frac{b_1(x - x_p)}{\Delta x} \tag{8}$$

where the general property of the b-splines $b_{l+1}(\xi) = \int b_l(\xi')b_0(\xi - \xi')d\xi'$ is used [5]. The electric and magnetic fields acting on the particles of Eq. (7) can be expressed more conveniently with the use of the interpolation functions, as:

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

where the cells are labelled with a single index g and the field values in each cell are $\mathbf{E}_g$ and $\mathbf{B}_g$. In addition, the moments of the distribution function ρ_g^n, $\mathbf{J}_g^n$ and Π_g^n (the pressure tensor), can be obtained easily by iterating over the N_s particles of the n_s species:

$$\{\rho^n, \mathbf{J}^n, \Pi^n\}_g = \sum_s^{n_s}\sum_p^{N_s} q_s\{1, \mathbf{v}_p^n, \mathbf{v}_p^n\mathbf{v}_p^n\}W(\mathbf{x} - \mathbf{x}_p^n) \tag{10}$$

In summary, Fig. 1 represents the four steps required for solving numerically the Vlasov–Maxwell system: the equations of motion (Eq. (6)) are solved to advance the computational particles positions and velocities. The moments are then calculated by interpolation from the new computational particles position and velocity, using Eq. (10). Maxwell's equations are solved next and the electromagnetic fields acting on the particle are obtained from Eq. (9) by interpolation. The PIC code integrates explicitly in time the governing equations, Maxwell's equations need only the sources from

Please cite this article in press as: S. Markidis, et al., Multi-scale simulations of plasma with iPIC3D, Math. Comput. Simul. (2009), doi:10.1016/j.matcom.2009.08.038

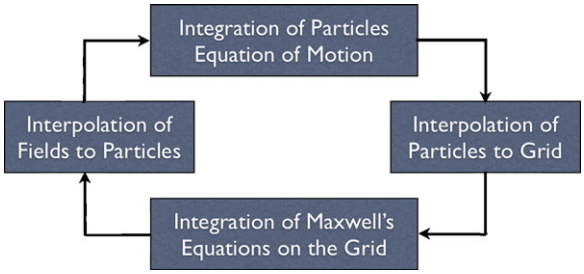

Fig. 1. The PIC computational cycle at each simulation time step.

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 represented in Fig. 1 becomes a direct marching order where each stage is applied after its predecessor and needs only information already available.

3.1. Numerical stability constraints

The numerical stability of the PIC method is studied by comparing the numerical and analytical dispersion relationships that regulate wave propagation in the system [21,16]. The discretization in time and space of the Vlasov equation introduces spurious numerical modes in the simulated model, that if allowed to grow uncontrolled can result in numerical instability. The stability analysis of the PIC method has been carried out extensively in the past using the numerical dispersion relationship [6], showing that the explicit PIC method is subject to three severe numerical stability constraints [2]. First, the explicit differentiation of the Maxwell's equations requires that a Courant condition must be satisfied on the speed of light c: $c\Delta t < \Delta x$. Second, the explicit discretization of the equations of motion introduces a constraint related to the fastest electron response time, the electron plasma frequency ω_{pe}: $\omega_{pe}\Delta t < \Delta x$. Third the interpolation between grid and particles causes a loss of information and an aliasing instability called *finite grid instability* that produces an additional stability constraint: $\Delta x < \varsigma\lambda_D$, where the proportionality constant ς is of order one [2,13]. This requires the grid spacing to be of the order of the Debye length λ_D or smaller. The explicit PIC method achieved remarkable results thanks to the use of modern fast supercomputers, but still there are problems where multiple scales still prevent its use. When slow processes at MHD time scale are the main focus and the kinetic description is required, the implicit PIC is one of the most effective simulation tool.

4. The implicit Particle-in-Cell

The implicit PIC method has been introduced to eliminate the severe numerical stability constraints, described above. In the implicit PIC, the Maxwell's equation source terms ρ and $\mathbf{J}$ are evaluated at the future time level: ρ is evaluated at time level $n+1$, while $\mathbf{J}$ is calculated at time level $n+1/2$ as average of $\mathbf{J}$ at time level $n+1$ and time level n to satisfy consistently the Ampere's law [6]. In the implicit PIC [23,6,20], Eq. (2) is differenced implicitly in time from time level n to time level $n+1$ as follows:

$$\mathbf{E}^{n+1} - (c\Delta t)^2\nabla^2\mathbf{E}^{n+1} = \mathbf{E}^n + c\Delta t\left(\nabla\times\mathbf{B}^n - \frac{4\pi}{c}\mathbf{J}^{n+1/2}\right) - (c\Delta t)^2\nabla 4\pi\rho^{n+1} \tag{11}$$

After the electric field $\mathbf{E}^{n+1}$ has been evaluated, the magnetic field is advanced in time differencing in time the Faraday's law (Eq. (3)):

$$\mathbf{B}^{n+1} = \mathbf{B}^n - c\Delta t\nabla\times\mathbf{E}^{n+1} \tag{12}$$

The difficulty of solving the implicit PIC arises because the new electric field in Eq. (11) depends on the new values of particle position and velocities trough Eq. (9), and vice versa computational particles position and velocity depend on the new electromagnetic field through the equation of motion (Eq. (21)). The exact implicit method iterates on all these equations (particles equations of motions and Maxwell's equation) together. However the computational cost of such exact implicit PIC is prohibitive and there is no clear advantage over simpler explicit PIC methods. The method of evaluating the implicit quantities, ρ^{n+1} and $\mathbf{J}^{n+1/2}$, constitutes the key of the approximated implicit PIC methods.

Please cite this article in press as: S. Markidis, et al., Multi-scale simulations of plasma with iPIC3D, Math. Comput. Simul. (2009), doi:10.1016/j.matcom.2009.08.038

The charge and current density values are extrapolated using a Taylor expansion and are expressed in terms of the present and (unknown) future electromagnetic fields. Once the new values for $\mathbf{E}^{n+1}$ and $\mathbf{B}^{n+1}$ are known by solving the Maxwell's equation, the particles are moved with the predicted fields. The approximated implicit PIC, used in iPIC3D, is based on a Taylor expansion of the interpolation function $W(\mathbf{x}-\mathbf{x}_p^{n+1})$ [34,20]. The Taylor expansion is completed around the particle position at the previous time step $\mathbf{x}_p^n$:

$$W(\mathbf{x}-\mathbf{x}_p^{n+1}) \approx W(\mathbf{x}-\mathbf{x}_p^n) + (\mathbf{x}^n - \mathbf{x}_p^{n+1})\nabla W(\mathbf{x}-\mathbf{x}_p^n) + \cdots \tag{13}$$

$$W(\mathbf{x}-\mathbf{x}_p^{n+1}) \approx W(\mathbf{x}-\mathbf{x}_p^n) - \bar{\mathbf{v}}_p \Delta t \nabla W(\mathbf{x}-\mathbf{x}_p^n) + \cdots \tag{14}$$

where $\bar{\mathbf{v}}_p$ is the particle average velocity, defined as $(\mathbf{v}_p^n + \mathbf{v}_p^{n+1})/2$. If $\bar{\mathbf{v}}_p$ is expressed as function of $\mathbf{E}^{n+1}$ and the extrapolated values for ρ^{n+1} and $\mathbf{J}^{n+1/2}$ values are inserted in Eq. (11), after a series of algebraic manipulations and keeping terms up to the second order in Δt [34], an equation for $\mathbf{E}^{n+1}$ is obtained:

$$(\mathbf{I}+\chi^n)\cdot\mathbf{E}^{n+1} - (c\Delta t)^2(\nabla^2\mathbf{E}^{n+1} + \nabla\nabla\cdot(\chi^n\cdot\mathbf{E}^{n+1})) = \mathbf{E}^n + c\Delta t\left(\nabla\times\mathbf{B}^n - \frac{4\pi}{c}\hat{\mathbf{J}}^n\right) - (c\Delta t)^2\nabla 4\pi\hat{\rho}^n \tag{15}$$

where $\mathbf{I}$ is the identity matrix and χ is called *implicit susceptibility* for similarity of Eq. (15) to the field equation in dielectric media, and it is defined as:

$$\chi\cdot = \sum_{n_s}\chi_s\cdot \quad , \quad \chi_s^n\cdot \equiv \frac{1}{2}(\omega_{ps}\Delta t)^2 R\left(\boldsymbol{\Omega}_s\frac{\Delta t}{2}\right)\cdot \tag{16}$$

for convenience the $\hat{\rho}^n$ and $\hat{\mathbf{J}}^n$ quantities are introduced as modified source terms for the Maxwell's equations:

$$\hat{\rho}^n = \rho^n - \Delta t\nabla\cdot\hat{\mathbf{J}}^n \quad , \quad \hat{\mathbf{J}}^n = \sum_s^{n_s} R\left(\boldsymbol{\Omega}_s\frac{\Delta t}{2}\right)\cdot\left(\mathbf{J}_s^n - \frac{\Delta t}{2}\nabla\Pi_s^n\right) \tag{17}$$

$R(\boldsymbol{\Omega}_s\Delta t/2)\cdot$, is a rotation transformation and it is defined as:

$$\begin{bmatrix} 1+\left(\Omega_{s_x}\frac{\Delta t}{2}\right)^2 & \Omega_{s_z}\frac{\Delta t}{2} + \Omega_{s_x}\Omega_{s_y}\left(\frac{\Delta t}{2}\right)^2 & -\Omega_{s_y}\frac{\Delta t}{2} + \Omega_{s_x}\Omega_{s_z}\left(\frac{\Delta t}{2}\right)^2 \\ -\Omega_{s_z}\frac{\Delta t}{2} + \Omega_{s_x}\Omega_{s_y}\left(\frac{\Delta t}{2}\right)^2 & 1+\left(\Omega_{s_y}\frac{\Delta t}{2}\right)^2 & \Omega_{s_x}\frac{\Delta t}{2} + \Omega_{s_y}\Omega_{s_z}\left(\frac{\Delta t}{2}\right)^2 \\ \Omega_{s_y}\frac{\Delta t}{2} + \Omega_{s_x}\Omega_{s_z}\left(\frac{\Delta t}{2}\right)^2 & -\Omega_{s_x}\frac{\Delta t}{2} + \Omega_{s_y}\Omega_{s_z}\left(\frac{\Delta t}{2}\right)^2 & 1+\left(\Omega_{s_z}\frac{\Delta t}{2}\right)^2 \end{bmatrix}.$$

$\boldsymbol{\Omega}_s \equiv q_s/m_s\mathbf{B}^n/c$, and $\omega_{ps} = \sqrt{(4\pi\rho_s q_s)/m_s}$ are the cyclotron frequency vector and the plasma frequency for species s. The introduction of the implicit susceptibility χ is the main characteristic of implicit Maxwell's solver. Eq. (16) defines the implicit susceptibility as combination of a scaling and a rotation transformation on the future value of the electric field. The effect of scaling by the factor of $1/2(\omega_{ps}\Delta t)^2$ is to reduce the electric field component, due the fast electrons oscillations that cannot be resolved by the large time step. The rotation transformation $R(\boldsymbol{\Omega}_s(\Delta t/2))$ includes the effect of the particle Larmor rotation induced by the magnetic field. After the field equation is solved, the electric field must be corrected to ensure that the charge density continuity equation is satisfied [2]:

$$\tilde{\mathbf{E}}^{n+1} = \mathbf{E}^{n+1} - \nabla\Phi \qquad \nabla^2\Phi = \nabla\cdot\mathbf{E}^{n+1} - 4\pi\rho^n \tag{18}$$

4.1. Spatial differentiation

In iPIC3D, the Maxwell's equations are differenced in space on a uniform Cartesian grid. The electric field $\mathbf{E}^{n,n+1}$, the current densities $\mathbf{J}^n$,$\hat{\mathbf{J}}^n$, and the implicit susceptibility χ^n are evaluated at the vertices of the grid, while the magnetic field $\mathbf{B}^{n,n+1}$ and charge densities ρ^n, $\hat{\rho}^n$ are calculated at the centers of the cells. The simple box scheme is used for

Please cite this article in press as: S. Markidis, et al., Multi-scale simulations of plasma with iPIC3D, Math. Comput. Simul. (2009), doi:10.1016/j.matcom.2009.08.038

the spatial differentiation of spatial operators in the field equations (Eqs. (15) and (18)). If $u_{i,j,k}$ is provided at vertices, the derivative $\partial u/\partial x$ at centers cell with indices $i+1/2$, $j+1/2$, $k+1/2$[[*space*]] is calculated as:

$$\frac{\partial u}{\partial x}|_{i+1/2,j+1/2,k+1/2} \approx \frac{u_{i+1,j+1/2,k+1/2} - u_{i,j+1/2,k+1/2}}{\Delta x} \tag{19}$$

The value u at the centers of the cells are computed by averaging the neighboring vertex values:

$$u_{i,j+1/2,k+1/2} = \frac{1}{4}\left(u_{i,j,k} + u_{i,j+1,k+1} + u_{i,j+1,k} + u_{i,j,k+1}\right) \tag{20}$$

Derivatives calculated at the centers of cells are approximated in a similar way. The Laplacian operator is obtained by combining the divergence and gradient operators. When spatial derivatives are approximated in this way, Eq. (15) results in a non-symmetric matrix with variable coefficients.

4.2. Implicit particle mover

After the electromagnetic fields have been calculated with the Maxwell's solver, the computational particles can be advanced by solving the Newton's equation of motions. Eq. (6) is time-differenced as follows [34,35]:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \mathbf{v}_p^{n+1/2}\Delta t, \qquad \mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \frac{q_s}{m_s}\left(\mathbf{E}_{1/2}^{n+1} + \frac{\mathbf{v}_p^{n+1/2} \times \mathbf{B}_{1/2}^{n+1}}{c}\right)\Delta t \tag{21}$$

$\mathbf{E}_{1/2}^{n+1}$, and $\mathbf{B}_{1/2}^{n+1}$ are the electric and magnetic field, calculated at the midpoint of the orbit $x^{n+1/2} = (x^{n+1} + x^n)/2$. The $\mathbf{v}_p^{n+1/2}$ is expressed as average of $\mathbf{v}_p^n$ and $\mathbf{v}_p^{n+1}$ and intermediate velocity $\tilde{\mathbf{v}}_p$ is introduced for convenience:

$$\tilde{\mathbf{v}}_p = \mathbf{v}_p^n + (q_s\Delta t/m_s)\mathbf{E}_{1/2}^n \tag{22}$$

Taking the dot and cross products of $\mathbf{B}$ respectively with $\mathbf{v}_p^{n+1}$ and with $\tilde{\mathbf{v}}_p$, Eq. (21) for $\mathbf{v}_p^{n+1/2}$ reduces to:

$$\mathbf{v}_p^{n+1/2} = \frac{\tilde{\mathbf{v}}_p + (q_s\Delta t/(2m_s c))\tilde{\mathbf{v}}_p \times \mathbf{B}_{1/2}^{n+1} + (q_s\Delta t/(2m_s c))^2(\tilde{\mathbf{v}}_p \cdot \mathbf{B}_{1/2}^{n+1})\mathbf{B}_{1/2}^{n+1}}{1 + ((q_s\Delta t/(2m_s c))\mathbf{B}_{1/2}^{n+1})^2} \tag{23}$$

Because the electric and magnetic fields depend on the mid-orbit position of the particle, an iterative procedure is needed to solve Eq. (23). iPIC3D solves the equation of motion by an iterative method based on a fixed number of Picard iterations that uses a trapezoidal rule for the quadrature of the equation of motion [29]. A number of 3 Picard iterations are completed in a typical run. Studies of the convergence and energy conservation of the mover are reported in reference [20].

4.3. Accuracy condition

The implicit PIC numerical scheme is linearly unconditionally stable and the stability constraint of explicit PIC do not apply to it [6]. Moreover, it has been shown that the implicit method gives accurate estimate of the evolution of the system when the particle displacement per time step is smaller than the grid spacing $\mathbf{v}_{the}\Delta t/\Delta x = \lambda_D\omega_{pe}/\Delta x < 1$, where the λ_D is the Debye length [7]. This accuracy condition arises from the convergence requirement of the Taylor expansion of Eq. (13). The inequality can be satisfied with large time steps even when the grid spacing is large compared to the Debye length. Furthermore the implicit PIC is more robust against the finite grid instability, allowing a larger grid spacing. In the case of simulations of plasma in fusion devices, the implicit PIC time step and grid spacing are tens to hundreds of times larger than those allowed by an explicit PIC.

5. Implementation of the implicit PIC in iPIC3D

iPIC3D is a three-dimensional parallel PIC code. On multiprocessor architectures, the domain decomposition technique is used to divide the computational workload among processors [9]. For implicit PIC where the cost of particle moving and of field solving are of the same order (unlike explicit PIC where most of the cost resides with

Please cite this article in press as: S. Markidis, et al., Multi-scale simulations of plasma with iPIC3D, Math. Comput. Simul. (2009), doi:10.1016/j.matcom.2009.08.038

the particles), it is crucial that both field solving and particle moving be parallelized efficiently. An important aspect of efficiency is the need to retain the particles and cells belonging to a subdomain on the same processor. Large amounts of information is exchanged between grid and particles residing in the same physical domain and therefore it is important to avoid that this information exchange results in inter processor communication [9]. The simulation box is divided among processors using a generic Cartesian virtual topology [11]. Particles are divided among processors also depending on their location, and communicated to adjacent processors if exiting from the processor domain. The parallelization of the code is based on MPI libraries and blocking parallel communication has been chosen for the communication among processors [11].

If the explicit PIC codes require very large memory to store the information of all the particles and fields, the implicit PIC requires even more memory to store additional intermediate variables, such as variables for $\hat{\rho}^n$, $\hat{\mathbf{J}}^n$, $\hat{\mathbf{x}}_p$, $\hat{\mathbf{v}}_p$. Parallel computer memory is needed in order to run simulations with large number of particles and grid nodes. The choice of shared memory machines is therefore currently not feasible and distributed memory machine (clusters) have to be used to simulate large-scale problems.

iPIC3D is a software package written entirely in C++. An Object-Oriented (OO) design has been followed in writing iPIC3D using the so-called *lite* OO approach presented in a previous work [22]. The variables related to particles, are organized as arrays in particle objects and divided depending on the species (electrons, ions, α particles, ...). The electromagnetic field constitutes a whole object, that comprises the electromagnetic field and field sources variables. OO paradigms, such as class inheritance, and polymorphism are used to make easy for developers to add new code to iPIC3D, without overwriting the existing code.

The discretized equations (Eqs. (15) and (18)) and their boundary conditions form a non-symmetric linear system that is solved using the Generalized Minimal Residual (GMRes) method [30,14]. For the divergence cleaning (Eq. (18)), the Conjugate Gradient (CG) method [30] is used since the discretized equation leads to a symmetric matrix. A discussion about the condition number of the matrix represented by Eq. (15) is reported in references [15,27]. The performance of the GMRes solver without a preconditioner, obtained by scaling the number of grid points and using different time steps, are presented in references [15,20]. In summary, the number of iterations to reduce the norm of initial error by three orders of magnitude is fairly low (from 4 to 10 iterations) and it is insensitive to the number of grid points in the domain. The use of a preconditioner would be surely beneficial, but the GMRes solver performance is still adequate even without it.

6. Magnetic reconnection with iPIC3D

iPIC3D has been tested against the magnetic reconnection problem [4,25] to show the effectiveness of the implicit PIC algorithm to solve multi-scale problems using a kinetic method. Magnetic reconnection is the merging of two distinct regions with different magnetic topology, caused by non-ideal dissipation phenomena. This change of magnetic field configuration is accompanied by an effective conversion of magnetic energy to kinetic particles of plasma. In fact, magnetic reconnection in magnetic confinement fusion devices is a major design concern for two reasons [37]. First, it provides a path for energy exchange between magnetic fields and kinetic energy of the plasma species (as is for example the case in reversed field pinch devices (RFP)). Second the formation of magnetic island during magnetic reconnection increases the outward radial transport towards the machine walls, degrading the plasma confinement (as it is for example the case of neoclassical tearing islands in tokamaks). The reconnection in magnetic confinement machine, results in the saw-thooth oscillations and consequent disruptions of plasmas, compromising the plasma confinement in fusion devices [37].

Even though magnetic reconnection can be a fast explosive phenomenon once it starts, it takes long periods to initiate it [4,25]. The magnetic reconnection builds up relatively slowly. For instance in the case of a fusion device plasma, magnetic reconnection develops on the scale of 10^{-6} s [37], but information from electron oscillations at 10^{-12} s is required to describe correctly the magnetic reconnection in explicit PIC. For problems like magnetic reconnection, explicit PIC codes needs time step of the order of 10^{-13} s (one tenth of the electron plasma frequency) to retain the numerical stability, while an implicit PIC runs typically with a hundreds times larger time step.

In most typical magnetic reconnection problems [3], the simulation box is initialized with two distinct regions with magnetic fields along opposite directions and with an interface between the two regions where the magnetic field is zero (antiparallel magnetic field configuration) as shown in Fig. 2a. A current flows perpendicular to the two regions and a plasma equilibrium distribution function, the so-called *Harris' equilibrium* [12], sustain consistently this magnetic

Please cite this article in press as: S. Markidis, et al., Multi-scale simulations of plasma with iPIC3D, Math. Comput. Simul. (2009), doi:10.1016/j.matcom.2009.08.038

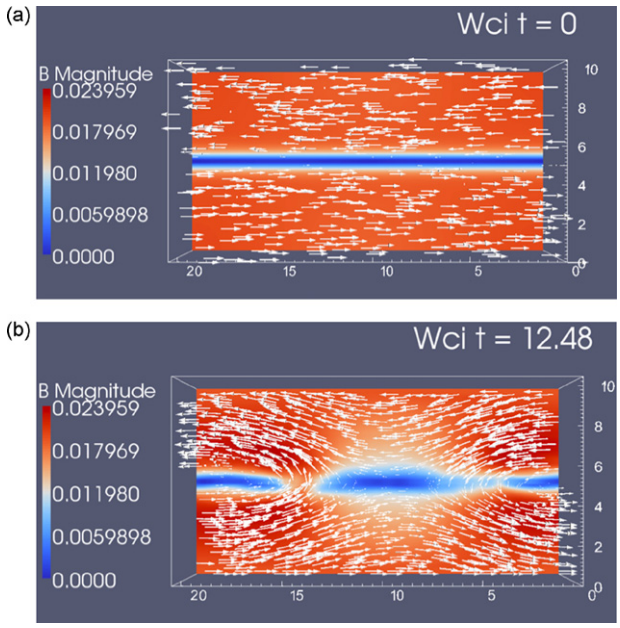

Fig. 2. Top view of the magnetic field quiver plot of the first simulation set. The color represents the magnetic field intensity. (a) Anti-parallel magnetic field configuration as initial simulation set-up. (b) Reconfiguration of the magnetic topology due to magnetic reconnection.

topology [4]. The two regions with the magnetic field in opposite directions merge and reconnect at later stages of the simulation. The magnetic reconnection, simulated with iPIC3D, develops leading to the formation of a line of null magnetic points in the middle of the simulation box and to the merging of the two initially disconnected regions on the sides of the simulation box (Fig. 2b). Two sets of magnetic reconnection simulations with two different system

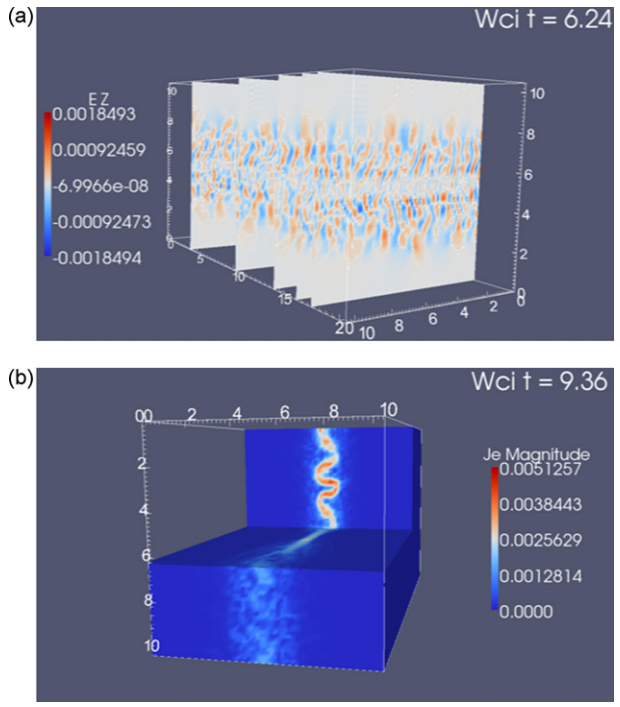

Fig. 3. Current-aligned instabilities in 3D. (a) Electric field contour plot on different slices shows the characteristic LHDI pattern at the edge of the current sheet. (b) Electron current intensity contour plot presents a current kink due to the kink instability.

Please cite this article in press as: S. Markidis, et al., Multi-scale simulations of plasma with iPIC3D, Math. Comput. Simul. (2009), doi:10.1016/j.matcom.2009.08.038

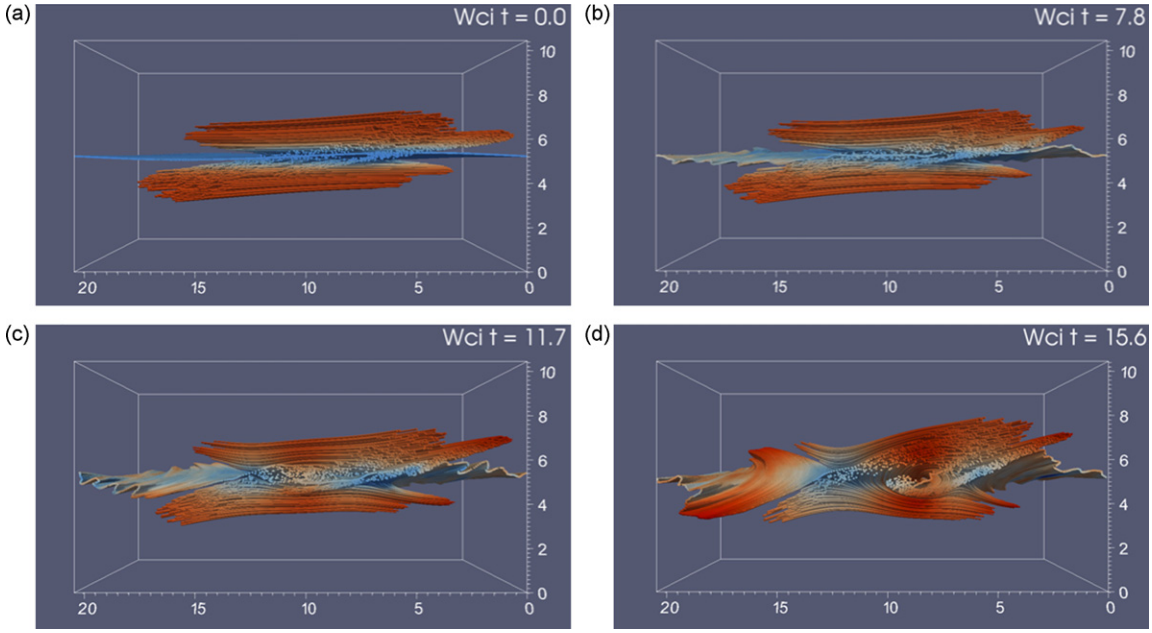

Fig. 4. Different snapshots of the evolution of the magnetic field lines during the magnetic reconnection with the presence of a guide field to mimic the magnetic configuration in a magnetic confinement fusion device.

configurations are presented in Figs. 3 and 4. The first represents the 3D extension of the classic GEM challenge [3], a standardized reconnection problem. The second adds a background uniform magnetic field, the so-called *guide field*, in the direction normal to the magnetic field reversal. Both simulation sets are completed on a $128 \times 64 \times 64$ grid using 100 million particles. The x-direction is the direction of the initial field reversal, the y direction is the direction of the initial density gradient and the z direction is the direction of the initial current (and of the guide field). The simulation box is $20d_i \times 10d_i \times 10d_i$ large, where d_i is the ion skin depth, defined as c/ω_{pi}. The initial current sheet has thickness $0.5d_i$. The parameters of the GEM challenge ($T_i/T_e = 5$, $v_{th,e}/c = 0.1$) are chosen for the other plasma parameters (except the mass ratio that is chosen as $m_i/m_e = 64$ instead of the standard 25) [3]. Additionally, as in the GEM challenge prescriptions, a perturbation is added uniformly in the third dimension. Note that while the implicit PIC method makes physical mass ratios accessible [28], the use of the physical mass ratio in the 3D simulations remains challenging and it is beyond the scope of the present benchmarking runs.

The first run presents the physical processes typical of the onset of reconnection in 3D (see, e.g. [18,31,26,38]). In 3D, reconnection onset is determined by the role of current-aligned instabilities, that develop along the additional dimension direction z [26]. In fact, the development of magnetic reconnection is accompanied by current driven instabilities (Fig. 3) [18]; in particular, the presence of the lower hybrid drift instability (LHDI) at the edge of the current sheet in the electric field plot [18] and of the drift kink instability (DKI) in the electron current plot [17,19] are evident respectively in Fig. 3a and b.

A second set of simulations (Fig. 4) studies the current sheet evolution during the magnetic reconnection in systems with the presence of a background magnetic field, perpendicular to the plane where magnetic reconnection develops [28]. This set-up is representative, for example, of the conditions present in fusion devices where a dominant toroidal magnetic field is present and reconnection is developing in the perpendicular poloidal plane [10,37]. In this case, the presence a magnetic field stabilizes the current sheet and the DKI and other current aligned instabilities are not observed during the magnetic reconnection.

7. Conclusions

The implicit PIC method and its implementation in the iPIC3D code have been presented. iPIC3D is a new parallel code for the kinetic multi-scale simulation of plasmas. It allows large simulation time step (hundreds times the electron plasma frequency) reaching MHD time scales but still retaining plasma kinetic effects. Finally, simulations of three-

Please cite this article in press as: S. Markidis, et al., Multi-scale simulations of plasma with iPIC3D, Math. Comput. Simul. (2009), doi:10.1016/j.matcom.2009.08.038

dimensional magnetic reconnection, an important phenomenon that severely degrades plasma confinement in fusion machines, are presented to show the effectiveness of the implicit PIC method.

Acknowledgments

SM would like to acknowledge support provided by the Department of Nuclear, Plasma and Radiological Engineering at the University of Illinois, and by the DOE through its INIE grant. The authors would also like to acknowledge support provided by the National Center for Supercomputing Applications and the use of ABE and VIC, high performance cluster of the Katholieke Universiteit Leuven. The research leading to these results has received funding from the European Commission's Seventh Framework Programme (FP7/2007-2013) under the grant agreement no. 218816 (SOTERIA project, http://www.soteria-space.eu).

References

[1] T.P. Armstrong, R.C. Harding, Solution of Vlasov's equation by transform methods, Methods Comput. Phys. 9 (1970) 29–86.
[2] C. Birdsall, A. Langdon, Plasma Physics Via Computer Simulation, McGraw-Hill, New York, 2004.
[3] J. Birn, J. Drake, M. Shay, B. Rogers, R. Denton, M. Hesse, M. Kuznetsova, Z. Ma, A. Bhattacharjee, A. Otto, Geospace Environmental Modeling (GEM) magnetic reconnection challenge, J. Geophys. Res. 106 (2001) 3715–3719.
[4] D. Biskamp, Magnetic Reconnection in Plasmas, Cambridge University Press, New York, 2000.
[5] C.D. Boor, A Practical Guide to Splines, Springer, New York, 1978.
[6] J.U. Brackbill, D. Forslund, An implicit method for electromagnetic plasma simulation in two dimensions, J. Comput. Phys. 46 (1982) 271.
[7] J.U. Brackbill, D.W. Forslund, Simulation of low-frequency, electromagnetic phenomena in plasmas, in: J.U. Brackbill, B.I. Cohen (Eds.), Multiple Time Scales, Academic Press, Orlando, 1985, pp. 271–310.
[8] J.M. Dawson, Particle simulation of plasmas, Rev. Modern Phys. 55 (1983) 403–447.
[9] J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, A. White, Sourcebook of Parallel Computing, Morgan Kaufmann, San Francisco, 2003.
[10] J.P. Freidberg, Plasma Physics and Fusion Energy, Cambridge University Press, New York, 2007.
[11] W. Gropp, E. Lusk, A. Skjellum, Using MPI: Portable Parallel Programming with the Message-passing Interface, MIT Press, Cambridge, 1999.
[12] E.G. Harris, On a plasma sheath separating regions of oppositely directed magnetic field, Nuovo Cimento 23 (1962) 115–121.
[13] R. Hockney, J. Eastwood, Computer Simulation using Particles, Taylor & Francis, Bristol, 1988.
[14] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.
[15] D.A. Knoll, G. Lapenta, J.U. Brackbill, A multilevel iterative field solver for implicit kinetic plasma simulation, J. Comput. Phys. 149 (1999) 377–388.
[16] A.B. Langdon, Analysis of the time integration in plasma simulation, J. Comput. Phys. 30 (1979) 202–221.
[17] G. Lapenta, J.U. Brackbill, A kinetic theory for the drift-kink instability, J. Geophys. Res. 102 (1997), 2709927108.
[18] G. Lapenta, J.U. Brackbill, 3D reconnection due to oblique modes: a simulation of Harris current sheet, Nonlinear Process. Geophys. 7 (2000) 151–158.
[19] G. Lapenta, J.U. Brackbill, Nonlinear evolution of the lower hybrid drift instability: current sheet thinning and kinking, Phys. Plasmas 9 (2002) 1544–1554.
[20] G. Lapenta, J.U. Brackbill, P. Ricci, Kinetic approach to microscopic–macroscopic coupling in space and laboratory plasmas, Phys. Plasmas 13 (2006) 055904.
[21] E.L. Lindman, Dispersion relation for computer-simulated plasmas, J. Comput. Phys. 5 (1970) 13–22.
[22] S. Markidis, G. Lapenta, W.B. VanderHeyden, Z. Budimlic, Implementation and performance of a particle-in-cell code written in Java, Concurrency Comput. Pract. Exp. 17 (2005) 821–837.
[23] R. Mason, Implicit moment particle simulation of plasmas, J. Comput. Phys. 41 (1981) 233–244.
[24] G.J. Parker, W.N.G. Hitchon, J.E. Lawler, Numerical solution of the Boltzmann equation in cylindrical geometry, Phys. Rev. E 50 (1994) 3210–3219.
[25] E. Priest, T. Forbes, Magnetic Reconnection: MHD Theory and Applications, Cambridge University Press, New York, 1999.
[26] P. Ricci, J.U. Brackbill, W. Daughton, G. Lapenta, Influence of the lower hybrid drift instability on the onset of magnetic reconnection, Phys. Plasmas 11 (2004) 4489–4500.
[27] P. Ricci, G. Lapenta, J.U. Brackbill, A Simplified Implicit Maxwell Solver, J. Comput. Phys. 183 (2002) 117–141.
[28] P. Ricci, G. Lapenta, J.U. Brackbill, GEM reconnection challenge: implicit kinetic simulations with the physical mass ratio, Geophys. Res. Lett. 29 (2002) 1–3.
[29] K.F. Riley, M.P. Hobson, S.J. Bence, Mathematical Methods for Physics and Engineering, Cambridge University Press, New York, 2006.
[30] Y. Saad, Iteractive Methods for Sparse Linear Systems, SIAM, Philadelphia, 2003.
[31] M. Scholer, I. Sidorenko, C.H. Jaroschek, R.A. Treumann, A. Zeiler, Onset of collisionless magnetic reconnection in thin current sheets: three-dimensional particle simulations, Phys. Plasmas 10 (2003) 3521–3527.
[32] E. Sonnendrucker, J. Roche, B. Pierre, A. Ghizzo, The semi-Lagrangian method for the numerical resolution of the Vlasov equation, J. Comput. Phys. 149 (1999) 201–220.

Please cite this article in press as: S. Markidis, et al., Multi-scale simulations of plasma with iPIC3D, Math. Comput. Simul. (2009), doi:10.1016/j.matcom.2009.08.038

[33] D. Sydora, Low-noise electromagnetic and relativistic particle-in-cell plasma simulation models, J. Comput. Appl. Math. 109 (1999) 243–259.
[34] H.X. Vu, J.U. Brackbill, CELEST1D: an implicit, fully kinetic model for low-frequency electromagnetic plasma simulation, Comput. Phys. Commun. 69 (1992) 253–276.
[35] H.X. Vu, J.U. Brackbill, Accurate numerical solution of charged particle motion in a magnetic field, J. Comput. Phys. 116 (1995) 384–387.
[36] J. Wesley, Operation and control of ITER plasmas, Nucl. Fusion 40 (2000) 485–494.
[37] R.B. White, The Theory of Toroidally Confined Plasmas, World Scientific Publishing Company, New York, 2006.
[38] L. Yin, W. Daughton, H. Karimabadi, B.J. Albright, K.J. Bowers, J. Margulies, Three-dimensional dynamics of collisionless magnetic reconnection in large-scale pair plasmas, Phys. Rev. Lett. 101 (2008) 125001–125005.

Please cite this article in press as: S. Markidis, et al., Multi-scale simulations of plasma with iPIC3D, Math. Comput. Simul. (2009), doi:10.1016/j.matcom.2009.08.038