NUMERICAL MODELING OF SPACE PLASMA FLOWS: ASTRONUM-2008
ASP Conference Series, Vol. 406, © 2009
Nikolai V. Pogorelov, Edouard Audit, Phillip Colella, and Gary P. Zank, eds.

Parsek2D: An Implicit Parallel Particle-in-Cell Code

S. Markidis,[1] E. Camporeale,[2] D. Burgess,[2] Rizwan-uddin,[1] G. Lapenta[3]

[1]*Nuclear, Plasma, and Radiological Engineering, University of Illinois at Urbana-Champaign, Illinois 61801, USA*

[2]*Astronomy Unit, Queen Mary University of London, UK*

[3]*Centrum voor Plasma-Astrofysica, Katholieke Universiteit Leuven, Celestijnenlaan 200B, 3001 Leuven, Belgium*

Abstract. A Particle-in-Cell (PIC) code, called Parsek2D, for the simulation of astrophysical and space plasmas is presented. Parsek2D enables simulations with large grid spacing and long time step, by means of implicit time differentiation and parallel computing. An implicit formulation of the PIC algorithm removes the severe numerical stability constraints of explicitly time-differenced PIC. Parallel supercomputers are needed to meet the computational and memory storage requirements of large scale problems The implicit PIC algorithm, and its implementation on parallel computers are described. Simulations of magnetic reconnection using physical mass ratio are shown.

1. Introduction

In space and astrophysical plasma physics, there is often interest in studying the long term evolution of large systems. In the magnetospheric plasmas, there are typically over 10^3 Debye lengths(λ_D) in an ion gyroradius, and 10^4 plasma periods(ω_{pe}^{-1}) in an ion cyclotron period(Ω_{ci}^{-1}). For instance, the magnetic reconnection develops over a period of $10\Omega_{ci}^{-1}$, in large (compared to ion gyroradius) systems. To resolve these large span of time and space scales with an explicitly time-differenced Particle-in-Cell(PIC) is not feasible, even on the modern massively parallel computers. The reason is that the explicit PIC method needs to satisfy numerical stability conditions: $c\Delta t/\Delta x < 1$, where c is the speed of light, $\omega_{pe}\Delta t < 2$, and $\Delta x < \lambda_D$. These constraints force the simulation to follow the fastest phenomena present in the system and to resolve the Debye length. The implicit PIC removes the stability condition on the choice of the time step, and allows larger grid spacing (ten to hundred times the grid spacing of explicit PIC).

The stability of the numerical scheme is achieved by means of a more sophisticated algorithm, that makes the parallelization more complex than in a explicit code. A consequence is that more care needs to be put into designing the parallelization of an implicit PIC code. In this paper, the implicit PIC algorithm is presented first, and it is followed by a description of the implementation for parallel computers. Finally results of simulations to study magnetic reconnection in the geomagnetic tail using the physical mass ratio for plasma species are presented.

2. The Implicit Particle-in-Cell

The PIC method mimics electrons and ions with a reduced number of representative computational particles. A grid is introduced in the simulation box, and charge and current densities are interpolated from the particles into the nodes of this grid. The Maxwell's equations are then solved on the grid to calculate the electromagnetic field. Finally, the fields are interpolated from the grid to the particles to provide the force acting on each particle, and the particles are advanced solving the Newton's equation of motion. At the beginning of the simulation, particle positions and velocities and field quantities are initialized sampling the initial configuration of the system. Charge(ρ), current($\mathbf{J}$) and pressure(Π) densities are calculated at the time level n by the interpolation from position $\mathbf{x}_p^n$ and velocity $\mathbf{v}_p^n$ of particle p of the species s using the weight functions $W(\mathbf{x} - \mathbf{x}_p^n)$ (typically b-splines functions):

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

Parsek2D solves the second-order Maxwell's equation for the electric field $\mathbf{E}$ (here in CGS units):

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

the magnetic flux $\mathbf{B}$ is then calculated by integrating the Faraday's Law of induction:

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

The field acting on particles is then calculated by interpolation from the grid to the particles. The particles (with mass m_s and charge q_s where s labels the species of the particles) are advanced using the nonrelativistic equation of motion:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p \quad , \quad \frac{d\mathbf{v}_p}{dt} = \frac{q_s}{m_s}(\mathbf{E} + \frac{\mathbf{v}_p}{c} \times \mathbf{B}) \tag{4}$$

2.1. Implicit Maxwell's Solver

In the implicit PIC (Brackbill & Forslund 1982; Lapenta et al. 2006), Eq. (2) is differentiated implicitly in time from time level n to time level $n+1$ as follows:

$$\mathbf{E}^{n+1} - (c\Delta t)^2 \nabla^2 \mathbf{E}^{n+1} = \mathbf{E}^n + c\Delta t(\nabla \times \mathbf{B}^n - \frac{4\pi}{c}\mathbf{J}^{n+1/2}) - (c\Delta t)^2 \nabla 4\pi \rho^{n+1} \tag{5}$$

After the electric field $\mathbf{E}^{n+1}$ has been evaluated, the magnetic field is advanced in time:

$$\mathbf{B}^{n+1} = \mathbf{B}^n - c\Delta t \nabla \times \mathbf{E}^{n+1} \tag{6}$$

The difficulty of solving the implicit PIC arises because the new electric field in Eq. (5) depends on the new values of particle position and velocities through Eqs. (1), and vice-versa particles position and velocity depend on the new electric

field. The exact implicit method iterates on all these equations together, but the computational cost of such exact implicit PIC would be prohibitive. The approximated implicit PIC used by Parsek2D is called *implicit moment method* and is based on a second order in time Taylor expansion of the interpolation function $W(\mathbf{x}-\mathbf{x}_p^{n+1})$ (Vu & Brackbill 1992). The expansion is made around the particle position at the previous time step:

$$W(\mathbf{x}-\mathbf{x}_p^{n+1}) = W(\mathbf{x}-\mathbf{x}_p^{n}) + (\mathbf{x}^n-\mathbf{x}_p^{n+1})\nabla W(\mathbf{x}-\mathbf{x}_p^{n}) + ... \tag{7}$$

The new values for ρ^{n+1} and $\mathbf{J}^{n+1/2}$ are calculated by extrapolation of interpolation functions, and they are inserted in Eq. (5). After a series of algebraic manipulations (Vu & Brackbill 1992) an equation for $\mathbf{E}^{n+1}$ is obtained:

$$\begin{aligned}(\mathbf{I}+\chi^n)\cdot\mathbf{E}^{n+1} - (c\Delta t)^2(\nabla^2\mathbf{E}^{n+1} + \nabla\nabla\cdot(\chi^n\cdot\mathbf{E}^{n+1})) = \\ \mathbf{E}^n + c\Delta t(\nabla\times\mathbf{B}^n - \frac{4\pi}{c}\hat{\mathbf{J}}^n) - (c\Delta t)^2\nabla 4\pi\hat{\rho}^n\end{aligned} \tag{8}$$

where $\mathbf{I}$ is the identity matrix and χ is called *implicit susceptibility* for similarity of Eq. (8) to the field equation in dielectric media, and it is defined as:

$$\chi\cdot = \sum_{n_s}\chi_s\cdot \quad , \quad \chi_s^n\cdot \equiv \frac{1}{2}(\omega_{ps}\Delta t)^2 R(\mathbf{\Omega}_s\frac{\Delta t}{2})\cdot \tag{9}$$

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

$$\hat{\rho}^n = \rho^n - \Delta t\nabla\cdot\hat{\mathbf{J}} \quad , \quad \hat{\mathbf{J}} = \sum_{s}^{n_s} R(\mathbf{\Omega}_s\frac{\Delta t}{2})\cdot(\mathbf{J}_s - \frac{\Delta t}{2}\nabla\Pi_s) \tag{10}$$

$R(\mathbf{\Omega}_s\frac{\Delta t}{2})\cdot$, is a rotation transformation:

$$\begin{bmatrix} 1+(\Omega_{s_x}\frac{\Delta t}{2})^2 & \Omega_{s_z}\frac{\Delta t}{2}+\Omega_{s_x}\Omega_{s_y}(\frac{\Delta t}{2})^2 & -\Omega_{s_y}\frac{\Delta t}{2}+\Omega_{s_x}\Omega_{s_z}(\frac{\Delta t}{2})^2 \\ -\Omega_{s_z}\frac{\Delta t}{2}+\Omega_{s_x}\Omega_{s_y}(\frac{\Delta t}{2})^2 & 1+(\Omega_{s_y}\frac{\Delta t}{2})^2 & \Omega_{s_x}\frac{\Delta t}{2}+\Omega_{s_y}\Omega_{s_z}(\frac{\Delta t}{2})^2 \\ \Omega_{s_y}\frac{\Delta t}{3}+\Omega_{s_x}\Omega_{s_z}(\frac{\Delta t}{2})^2 & -\Omega_{s_x}\frac{\Delta t}{2}+\Omega_{s_y}\Omega_{s_z}(\frac{\Delta t}{2})^2 & 1+(\Omega_{s_z}\frac{\Delta t}{2})^2 \end{bmatrix}$$

where $\mathbf{\Omega}_s \equiv \frac{q_s}{m_s}\frac{\mathbf{B}^n}{c}$, and $\omega_{ps} = \sqrt{(4\pi\rho_s q_s)/m_s}$ are the cyclotron frequency vector and plasma frequency for species s. To ensure that the charge density continuity equation is satisfied, the electric field must be corrected with:

$$\tilde{\mathbf{E}}^{\mathbf{n+1}} = \mathbf{E}^{\mathbf{n+1}} - \nabla\Phi \quad , \quad \nabla^2\Phi = \nabla\cdot\mathbf{E}^{\mathbf{n+1}} - 4\pi\rho^n \tag{11}$$

The box scheme is used for the spatial differentiation of spatial operators in the field Eqs. (8) and (11). Boundary conditions for the electromagnetic field implemented in Parsek2D include the *perfect conductor*, *open*, and *periodic* boundary conditions. Equations (8) and (11) are solved by matrix-free Generalized Minimal Residual (GMRes) and Conjugate Gradient(CG) iterative linear solvers. It has been shown that the implicit PIC is linearly unconditionally stable, but subject to an accuracy condition $\mathbf{v}_{the}\Delta t/\Delta x < 1$(Brackbill & Forslund 1982). Furthermore the method is more robust against the finite grid instability, allowing a larger grid spacing. In the case of simulations of the geomagnetic tail, the implicit PIC time step and grid spacing are tens to hundreds of times larger (depending on the ion to electron mass ratio used) than those allowed by an explicit PIC.

2.2. Implicit Particle Mover

After the electromagnetic fields have been calculated with the Maxwell's solver, the computational particles are advanced by solving the Newton's equation of motions. The Eqs. (4) are implicitly time differentiated as follows:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \mathbf{v}_p^{n+1/2}\Delta t \quad , \quad \mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \frac{q_s}{m_s}(\mathbf{E}_{1/2}^{n+1} + \frac{\mathbf{v}_p^{n+1/2} \times \mathbf{B}_{1/2}^{n+1}}{c})\Delta t \quad (12)$$

$\mathbf{E}_{1/2}^{n+1}$, and $\mathbf{B}_{1/2}^{n+1}$ are the electric and magnetic field, calculated at the midpoint of the orbit $x^{n+1/2} = (x^{n+1} + x^n)/2$. If $\mathbf{v}_p^{n+1/2}$ is expressed as average of $\mathbf{v}_p^n$ and $\mathbf{v}_p^{n+1}$, then Eq. (12) can be solved for $\mathbf{v}_p^{n+1}$. Taking the dot product and cross product of $\mathbf{B}$ with $\mathbf{v}_p^{n+1}$ and with $\tilde{\mathbf{v}}_p = \mathbf{v}_p^n + (q_s\Delta t/m_s)\mathbf{E}_{1/2}^n$, Eq. (12) for $\mathbf{v}_p^{n+1}$becomes

$$\mathbf{v}_p^{n+1} = (\tilde{\mathbf{v}}_p + \frac{q_s\Delta t}{m_s c}\tilde{\mathbf{v}}_p \times \mathbf{B}_{1/2}^{n+1} + (\frac{q_s\Delta t}{m_s c})^2(\tilde{\mathbf{v}}_p \cdot \mathbf{B}_{1/2}^{n+1})\mathbf{B}_{1/2}^{n+1})/(1 + (\mathbf{\Omega}_s\Delta t)^2) \quad (13)$$

Because the electric and magnetic field depend on the mid-orbit position of the particle, an iterative procedure is needed to solve Eq. (13).

2.3. Implementation of the Implicit Particle-in-Cell

Parsek2D is a 2D$\frac{1}{2}$ PIC: it uses a two dimensional cartesian space, and three components vector quantities. The domain decomposition technique is used on parallel computers. 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 the particles), it is crucial that both field solving and particle moving be parallelized efficiently. A key aspect of efficiency is the need to retain the particles and cells belonging to a subdomain on the same processor. A large amount of information is exchanged between grid and particles residing in the same physical domain and therefore it is crucial to avoid that this information exchange results in communication. The simulation box is divided among processors using a generic cartesian virtual topology. Particles are divided among processors also depending on their location, and communicated to adjacent processors if exiting from the processor domain.

Parsek2D has been written entirely in C++, and compiles on the freely available GNU gcc compiler. An Object-Oriented design has been followed in writing Parsek2D using the so-called lite object orientation approach presented in a previous work (Markidis et al. 2005). The variables related to particles, are organized as arrays in Particle objects and divided depending on the species (electrons, ions,...). The Electromagnetic field constitutes a whole object, that comprises the electromagnetic field and field sources variables. Class inheritance, and polymorphism make easy for developers to add new code to Parsek2D, without overwriting the existing code.

Explicit PIC codes already require very large memory to store the information of all the particles. The implicit PIC requires more computer memory to store additional intermediate variables, such as $\hat{\rho}, \hat{\mathbf{J}}$. 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. The parallelization of the code is based on MPI libraries and blocking communication has been chosen for the communication among processors. Input files are in the *Key Value Format* (KVF) and the output and restart files are written in the HDF5 format.

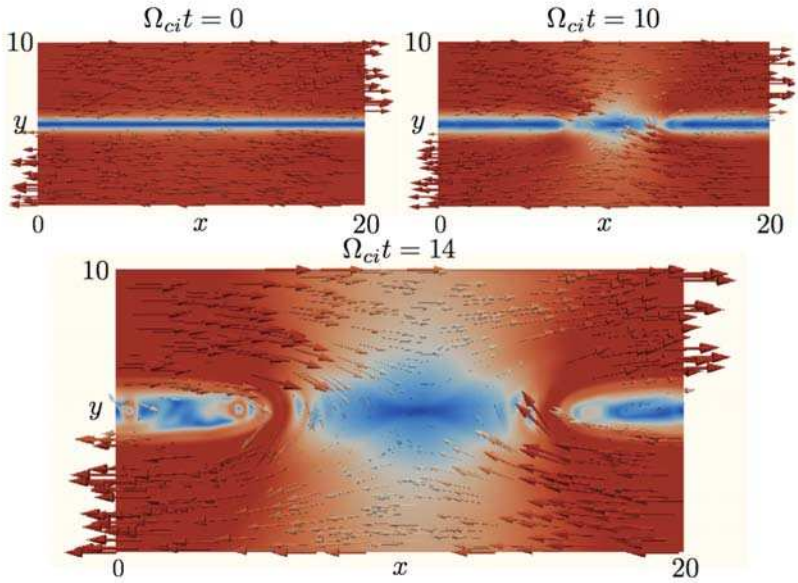

Figure 1. The magnetic field at three subsequent times ($m_i/m_e = 1836$).

3. Results

The implicit moment method has found numerous applications over the years, see Ricci et al. (2002) and Brackbill & Cohen (1985) for a review. Parsek2D has been tested on several problems involving space and astrophysical plasma. As an example, we show a selection of results obtained for a magnetic reconnection simulation in support of the NASA MMS mission. The simulation box has size $20d_i$x$10d_i$, where d_i is the ion skin depth, defined as c/ω_{pi}. The initial current sheet has thickness 0.5 d_i. The prescriptions 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 $m_i/m_e = 1836$ instead of the standard 25).

Figure 1 shows the initial magnetic field configuration and its evolution in response to the standard GEM perturbation (Birn et al. 1987): two regions are initialized with opposite magnetic fields and break up reconnecting and forming two magnetic islands. In presence of a uniform out of plane magnetic field (referred to in the literature as *guide field*), the reconnection process produces very high electron accelerations along the separatrices. In this region, the elec-

trons travel much faster than the ions leading to the onset of a Buneman-like instability, displayed in the contour plot of the parallel electric field as areas of repeating dipolar fields (in small red circles in Figure 2 in a simulation with $m_i/m_e = 64$) (Goldman et al. 2008).

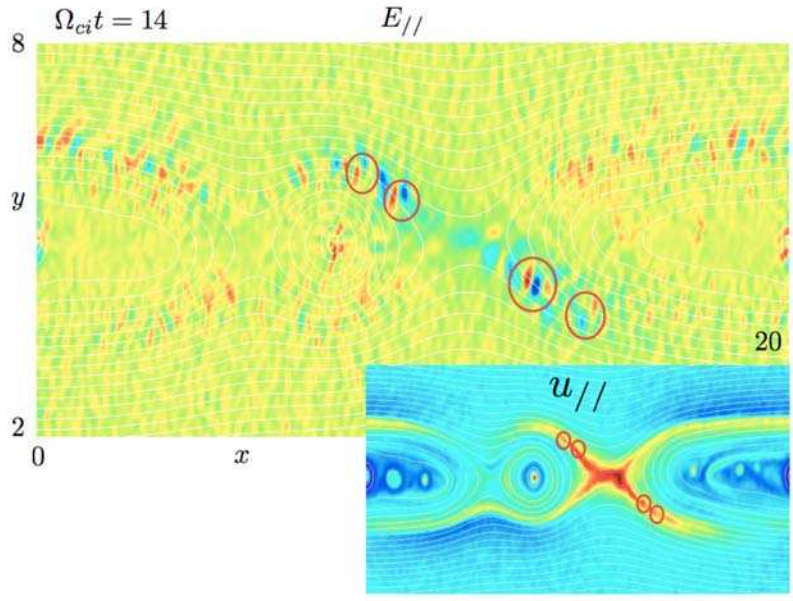

Figure 2. The parallel electric field and electron velocity ($m_i/m_e = 64$).

References

Birn J. et al. 1987, J. Geophys. Res., 92, 11
Brackbill, J.U., & Forslund, D. 1982, J. Comput. Phys., 46, 271
Brackbill, J.U., & Cohen, B.I. 1985, Multiple time scales (Orlando: Academic Press)
Goldman, M. V., Newman, D. L., & Pritchett, P. 2008, Geophys. Res. Lett., 35, in press
Lapenta, G., Brackbill, J.U. & Ricci, P. 2006, Phys. Plasmas, 13, 055904
Markidis, S. et al. 2005, Concurrency Comput. Practice & Experience, 17, 821
Ricci, P., Lapenta, G., & Brackbill, J.U. 2002, J. Comput. Phys., 183, 117
Vu, H. X., & Brackbill, J.U. 1992, Comp. Phys. Comm., 69, 253