
Guangye Chen
**An energy- and charge-conserving, fully implicit
particle-in-cell algorithm**

Fusion Energy Division
Oak Ridge National Laboratory
P O Box 2008
Oak Ridge
TN 37831-6169
`gcc@ornl.gov`
Guangye Chen
Luis Chacon
Daniel Barnes

Classical particle-in-cell (PIC) algorithms [1] employ an explicit time integration approach (leap-frog) to advance the Vlasov-Maxwell (or Vlasov-Poisson) system using particles coupled to a grid. As in the fluid context, explicit PIC features a numerical stability constraint which enforces a minimum timestep resolution (i.e, the timestep has to be small enough to resolve the fastest time-scale of the system, e.g., the plasma wave period). Unlike fluid approaches, however, explicit PIC also features a spatial numerical stability constraint that enforces a minimum spatial resolution (i.e., the grid size must be smaller than the Debye length) to avoid the so-called finite-grid instability. These stringent stability constraints make explicit PIC approaches impractical for many realistic multi-scale kinetic problems.

Fully implicit temporal schemes can in principle employ arbitrarily large timesteps without numerical instability. Furthermore, in the context of PIC methods, they can potentially avoid the finite-grid-instability, thus allowing spatial resolution to be dictated by accuracy, not stability. It follows that, by taking time steps and grid sizes much larger than the plasma period and the Debye length, respectively, implicit PIC algorithms can be orders of magnitude more efficient than explicit approaches. This realization motivated the early exploration of various implicit PIC approaches (namely, moment implicit [2,3] and direct implicit [4,5] methods). However, accurately solving the resulting nonlinearly coupled system of particle-field equations proved a daunting task, motivating algorithmic simplifications such as lagging and linearization which failed to converge the nonlinear residual. As a result, large numerical errors developed in long-time simulations [6], with significant self-heating/cooling often observed, especially when large time steps were employed.

The goal of this work is to demonstrate the algorithmic feasibility and accuracy benefits of a fully implicit PIC algorithm. Our solver is based on Jacobian-

Free Newton-Krylov (JFNK) technology. As a proof-of-principle, we focus on a one-dimensional electrostatic model [7] without preconditioning. One potential show-stopper is the very large nonlinear residuals that would result if one considered particle quantities (positions and velocities) as dependent variables. In this case, the residual size, and hence all vectors in the nonlinear solver, would scale with the number of particles in the simulation, resulting in dramatic memory requirements. Our implementation avoids this problem by eliminating nonlinearly particle quantities in favor of macroscopic fields and/or moments. Such nonlinear elimination (which we term particle enslavement) is possible because particle orbits only require macroscopic fields to evolve, and results in a nonlinear field solver with memory requirements comparable to those of a fluid computation.

Our implicit implementation is based on the Vlasov-Ampere system. Our discrete implementation is exactly energy- and charge-conserving, thus avoiding particle self-heating and self-cooling. While momentum is not exactly conserved, errors are kept small by an adaptive particle sub-stepping orbit integrator. Particle moments are orbit averaged to ensure energy conservation to numerical round-off in the presence of multiple substeps. As a result, very large time steps, constrained only by the dynamical time scale of interest, are possible without appreciable loss of accuracy.

As a result of particle enslavement, the particle orbit integration step required in the evaluation of the nonlinear residual becomes segregated. This renders the implicit algorithm ideally suited for emerging heterogeneous architectures which combine CPUs and GPUs [8]. Thus, the field JFNK solver can remain on the CPU, while the particle mover (which is naturally data-parallel and compute-bounded) stays on the GPU. Although the adaptivity of the particle mover creates load imbalances and dynamic control flows (which pose a challenge for the GPU), we demonstrate that a highly efficient GPU implementation of the implicit particle mover (using CUDA) is possible [8]. We obtain 300 to 400 GOp/s (counting floating-point, integer and special function operations) using single precision, depending on the electrostatic PIC model of choice (Ampere vs. Poisson). This is about 20 the GPU (Nvidia GeForce GTX580), and about 200 to 300 times faster than a single-CPU (Intel Xeon@3.16GHz) serial implementation. Furthermore, a mixed-precision implementation is possible that combines the JFNK field solver implemented in double precision on a CPU and the particle mover implemented in single precision on a GPU to produce a double-precision converged result.

Standard numerical examples are presented that demonstrate the properties of the scheme. In particular, long-time ion acoustic wave and ion acoustic shock wave simulations show that numerical accuracy does not degrade with very large implicit time steps, and that a significant increase in computational performance is possible versus explicit simulations, even without a preconditioner. Recent developments to extend the energy and charge-conserving implicit PIC

algorithm to curvilinear geometry will also be discussed.

- [1] C. K. Birdsall, A. B. Langdon, Plasma Physics via Computer Simulation, McGraw-Hill, New York, 2004.
- [2] R. J. Mason, J. Comput. Phys., 41 (2) (1981) 233244.
- [3] J. U. Brackbill, D. W. Forslund, Multiple time scales, Academic Press, 1985.
- [4] A. Friedman, A. B. Langdon, B. I. Cohen, Plasma Phys. Controlled Fusion, 6 (6) (1981) 225–36.
- [5] A. B. Langdon, D. C. Barnes, Multiple time scales, Academic Press, 1985.
- [6] B. I. Cohen, A. B. Langdon, D. W. Hewett, R. J. Procassini, J. Comput. Phys., 81 (1) (1989) 151168.
- [7] G. Chen, L. Chacn, D. C. Barnes, J. Comput. Phys. 230, (2011).
- [8] G. Chen, L. Chacn, D. C. Barnes, J. Comput. Phys. submitted (2011).