An Energy- and Charge-conserving, Implicit, Electrostatic Particle-in-Cell Algorithm

L. Chacón, G. Chen
Oak Ridge National Laboratory, Oak Ridge, TN 37831
D. C. Barnes,
Coronado Consulting, Lamy, NM 87540

Classical particle-in-cell (PIC) algorithms [1] employ an explicit approach (leap-frog) to advance Vlasov-Maxwell (or Vlasov-Poisson) system using particles coupled to a grid. A temporal numerical stability constraint of the classical explicit PIC algorithms is that the timestep has to be small enough to resolve the fastest time-scale of the system, e.g., the plasma wave period. A spatial stability constraint is that the grid size has to be smaller than the Debye length to avoid the finite-grid-instability. In many cases, these stringent stability conditions make it impractical to solve realistic multi-scale kinetic problems, even with the aid of modern super-computers.

It is well known that an implicit algorithm, in principle, can employ large timesteps without any temporal instability. Furthermore, it can be more robust against (or even avoid) the finite-grid-instability. Therefore, by taking timesteps and grid sizes that are much larger than the plasma period and the Debye length, respectively, an implicit PIC algorithm can be orders of magnitude more efficient than explicit PIC. This realization motivated the early exploration of various implicit PIC approaches (namely, moment implicit [2,3] and direct implicit [4,5] methods). However, due to the lack of efficient nonlinear solvers for the resulting system of fully implicit, nonlinearly coupled particle-field equations, these early implicit PIC studies employed algorithmic simplifications such as linearization and/or lagging, which led to accumulation of numerical error in long term simulations [6]. As a consequence, significant self-heating/cooling were often observed in implicit simulations, especially when large timesteps were employed.

The goal of this work is to demonstrate a fully implicit, nonlinear PIC algorithm using a Jacobian-Free-Newton-Krylov method on a one-dimensional electrostatic model [7]. The formulation is based on the Vlasov-Ampère formulation (rather than Vlasov-Poisson), because it allows an exact energy-conserving formulation. Exact local charge conservation is ensured by a novel particle mover strategy. While momentum is not exactly conserved, errors are kept small by an adaptive particle sub-stepping orbit integrator, which is instrumental to prevent particle tunneling (a deleterious effect for long-term accuracy). As a result, very large timesteps, constrained only by the dynamical time scale of interest, are possible without accuracy loss.

A main algorithmic development in this study is the nonlinear elimination of the new-time particle variables (positions and velocities). Such nonlinear elimination, which we term particle enslavement, results in a memory-friendly implementation (it only requires a single copy of the particle information, and the nonlinear-solver memory requirements are comparable to those of a fluid residual computation). In addition, particle enslavement affords us substantial freedom in regards to the particle orbit integrator. Standard numerical examples are presented that demonstrate the advertised properties of the scheme. In particular, long-time ion acoustic wave simulations show that numerical accuracy does not degrade even with very large implicit timesteps, and demonstrate significant CPU gains versus explicit simulations.

[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) 233–244.
[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) 151–168.
[7] G. Chen, L. Chacón, D. C. Barnes, *J. Comput. Phys.*, accepted (2011).