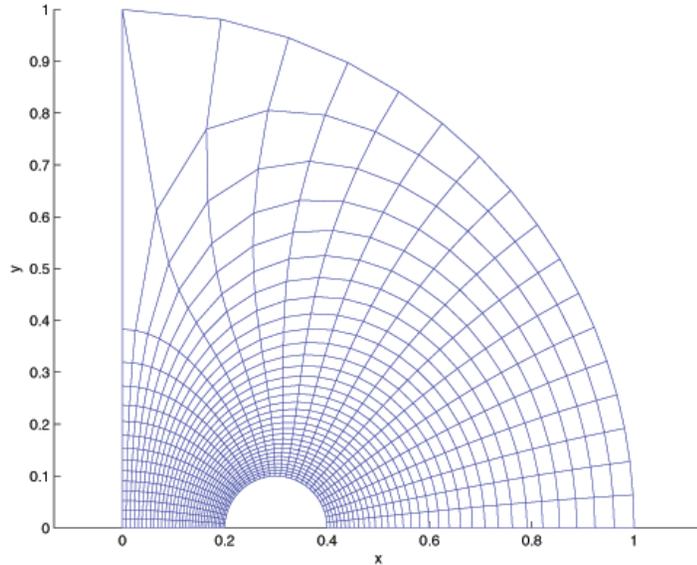


A Particle-in-Cell Code with Arbitrary Curvilinear Mesh

Chris A. Fichtl, University of New Mexico; John M. Finn, T-5

Fig. 1. Physical space grid for studying the interaction of two dust grains in a collisionless plasma.



Particle-in-cell (PIC) codes solve the Vlasov equation, the basic equation of collisionless plasma kinetic theory. The Vlasov equation is a mean field equation, appropriate to hot, collisionless plasmas. That is, it is assumed that the particles interact only through the mean field, which is found by a global field solve from the particle positions and, for the case of an electromagnetic code, the particle velocities. The basic steps in an electrostatic PIC code are: 1) weighting particle positions to the grid, obtaining densities for the electron and ion species, 2) solving the Poisson equation for the electrostatic potential Φ on the grid and obtaining the electric field from Φ , and 3) integrating the particle orbits using this electric field, interpolated to the particle positions.

We are in the process of developing an electrostatic PIC code in 2D (cylindrical or axial symmetry) for an arbitrary structured mesh. The mesh is obtained by mapping from a uniform mesh on the *logical* (computational) domain Ξ , which we take to be the unit square, to an arbitrary non-uniform mesh on the physical space X . At present we are using the Winslow method [1] of mesh generation, a special case of the Laplace-Beltrami approach [2]. This method

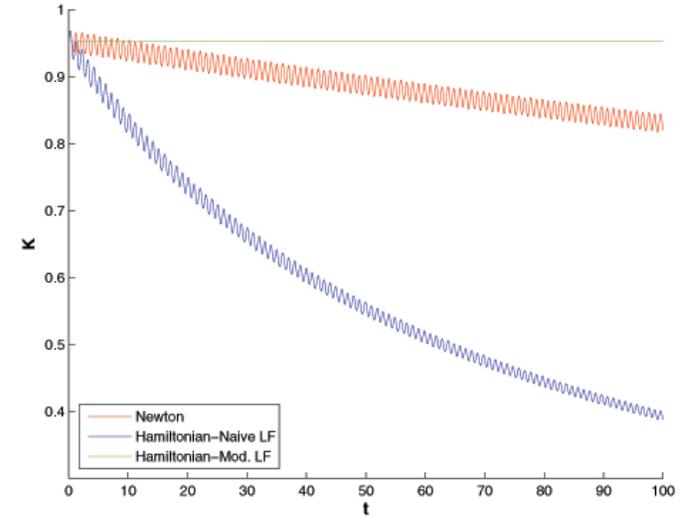


Fig. 2. The energy K as a function of time for three particle integrators.

produces boundary-conforming grids, allowing modeling of curved surfaces without the serious problems of "stair-stepping" in codes that use non-boundary-conforming grids. The example in Fig. 1 shows the mesh appropriate to dusty plasmas, in which the inner sphere corresponds to a dust grain, and the mirror image under a reflection through the vertical axis gives a second grain (the outer boundary represents matching to the rest of a Maxwellian plasma, and needs to be at a sufficiently large radius). Note that this method allows concentration of cells near a dust grain, to help in resolving the dynamics within a few Debye radii of the grain.

In our code all computations are performed on the logical grid on Ξ . When mapped to the logical grid the Poisson equation takes the form

$$\frac{\partial}{\partial \xi^i} \left(J g^{ij} \frac{\partial \Phi}{\partial \xi^j} \right) = -\frac{J \rho}{\epsilon_0}$$

(summation assumed) where $\rho \equiv \rho_{ion} - \rho_{electron}$ is the charge density, J is the determinant of the Jacobi matrix $J^i_j = \partial x^i / \partial \xi^j$, and $g^{ij} = \nabla \xi^i \cdot \nabla \xi^j$ is the contravariant metric tensor on the logical grid. The quantity $J\rho$ is the charge density on the logical grid. The final hurdle is integrating the particle orbits (advancing or

pushing particles). The standard leapfrog integrator for a uniform grid is a symplectic integrator, i.e., conserves the Hamiltonian phase space structure. However, on a nonuniform grid standard (naive), leapfrog is not a symplectic integrator. We introduce a contact transformation $(\mathbf{x}, \mathbf{p}) \rightarrow (\xi, \mathbf{P})$ by a generating function $S(\mathbf{x}, \mathbf{P}) = \xi_j(\mathbf{x})P_j$, leading to a new Hamiltonian (for a time-independent grid)

$$K = \frac{1}{2m} \left(\frac{\partial \xi_j}{\partial x_i} P_j \frac{\partial \xi_k}{\partial x_i} P_k \right) + q\Phi(\mathbf{x}(\xi)) = \frac{1}{2m} (g^{jk}(\mathbf{x}(\xi))P_j P_k) + q\Phi(\xi)$$

where q and m are the particle charge and mass, respectively. The fact that the Hamiltonian is not separable (it is of the form $K = T(P, \xi) + V(\xi)$) prevents naive leapfrog from being a symplectic integrator. To integrate the Hamiltonian equations of motion, we use a generalization (from 2D phase space to 4D) of the modified leapfrog integrator introduced in [3]. This time-centered, second-order accurate integrator involves a combination of implicit and explicit split-steps and preserves the Hamiltonian nature of the phase space exactly.

Figure 2 shows the orbits and a test of the energy conservation for this integrator on a nonuniform grid, compared with naive leapfrog integrators. The latter were performed with the straight Newtonian equations of motion and with the Hamiltonian equations using K , respectively. It is clear that in the nonsymplectic integrators, the orbits spiral in, losing energy, so that PIC runs must be severely limited in time step. The orbits from the symplectic integrator spiral without dissipation or growth for all time. Further, the energy is conserved to $O(\Delta t)^2$, the expected order in Δt and does not decay or grow systematically.

For integration into a full PIC code, an important factor is that, for uniform grids, the particle shapes required for density weighting and for electric field interpolation must be identical [4]. For nonuniform grids this condition must be modified: the two particle shapes must be *adjoints* of each other. If this condition is satisfied, the total momentum is conserved, and the so-called self-force is zero. Since our discretization is on the uniform logical mesh, the particle shapes are symmetric and this condition is automatically

satisfied, eliminating the need for computationally expensive particle search algorithms, which would be necessary if the computations were done on the physical grid.

We have performed the usual battery of tests to benchmark this code, including cold plasma oscillations, warm plasma Langmuir waves, Landau damping, and the linear and nonlinear two-stream instability, for a variety of nonuniform grids. For all, we have found good agreement.

**For more information contact
John M. Finn at finn@lanl.gov.**

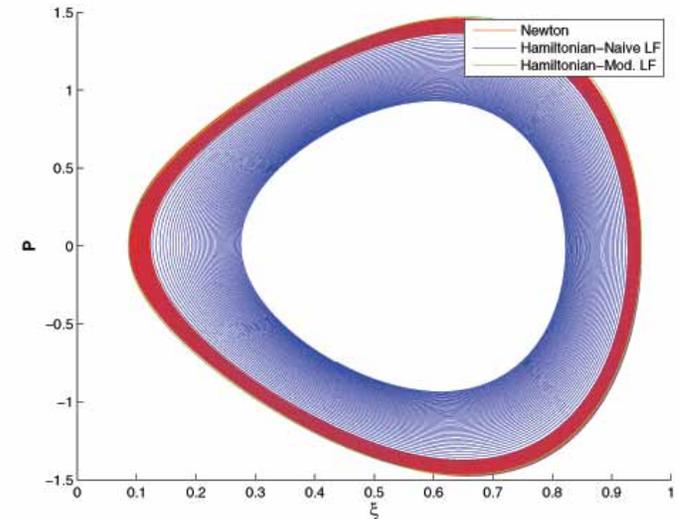


Fig. 3. Phase space for the three particle integrators, showing spiraling in for the two naive leapfrog integrators and good phase space conservation for the modified leapfrog integrator.

- [1] A.M. Winslow, "Adaptive mesh zoning by the equipotential method," *Tech. Rep.* UCID-19062, LLNL (1981).
- [2] V.D. Liseikin, *A Computational Differential Geometry Approach to Grid Generation*, Springer-Verlag (2007).
- [3] J.M. Finn, L. Chacón, *Phys. Plasma* **12**, 054503 (2005).
- [4] C.K. Birdsall, A.B. Langdon, *Plasma Physics via Computer Simulation*, Taylor and Francis Group (1995).

Funding Acknowledgments

DOE Office of Science