

I. INTRODUCTION

Particle-in-cell (PIC) methods first gained popularity for plasma physics simulations in the 1950s and 1960s. For the Vlasov-type systems often considered, we are required to solve in phase-space which makes many common grid methods infeasible due to the large amount of memory that these methods use. Generally, PIC methods can be understood as having four distinct steps [1, 3]:

1. **Deposition:** Deposit the particles charge on the real-space grid.
2. **Grid Solve:** This step requires either solving Poisson's Equation for the field(s).
3. **Interpolate:** Interpolate the field(s) back onto the particles.
4. **Integration:** Integrate the equations of motion.

Recent studies have shown the importance of a fifth step,

5. **Remap:** Deposit the charge on a uniform phase-space grid that spawns new particles, to convergence for long-time simulations [5, 7]. Current state-of-the-art PIC codes often neglect Step 5 and are only second-order accurate. We propose implementing a hybrid MPI-OpenMP implementation and investigating performance of a remapped, high-order PIC method.

II. IMPLEMENTATION

From a parallel implementation standpoint, the most costly steps are generally the particle-mesh interpolation steps (1 & 4 above). Both shared-memory multicore [4] and GPU [2, 6] implementations have been investigated. The difficulty in these steps has to do with the non-uniform memory access pattern of the $O(n)$ implementations. Specifically, race conditions between threads attempting to deposit charge on the same portion of the real-space grid makes parallelization difficult. [4] showed that good performance could be achieved via a coarse-grained locking strategy or through allowing each processor to operate on a local copy of the grid - which is memory intensive but easier from a programming standpoint. We hope to investigate the use of coarse-grained locks if time permits. Specifically, how the order of the method (and thus the size of the stencil of the interpolation kernel) effects the performance.

The focus of this project will be on efficient remapping of the particle distribution to maintain convergence for long-times. [?] has shown the importance of remapping to the convergence and addressed some aspects of implementing a PIC simulation with remapping in MPI. In this paper, the authors found remapping every five time steps was necessary for convergence. We plan to evolve the deformation gradient of the distribution and remap when the deformation is greater than C/h_x .

We plan to use the same physical space domain decomposition strategy used in this paper. More complicated phase-space domain decompositions have the potential for better performance but for linear Landau damping, a physical space decomposition is found to be satisfactory.

The authors found that the most challenging stage to implement was the deposition step. The authors used a 4-point gyro-averaging method. One consequence of this method is that a particle may deposit charge onto as few as 8 grid points and as many as 32 grid points. This can cause data hazards for both shared memory machines and SIMD architectures and bandwidth-intensive MPI communication for distributed memory machines. The requirement to synchronize can be alleviated by constructing thread private copies of the grid at the cost of memory challenges.

To address these issues, the authors binned the particles on the real-space grid at a user defined frequency. Furthermore, for CPUs the grid replication strategy with a merge at the end was used to avoid data hazards. For GPUs the binned particles in each subregion are iteratively processed by the same CUDA thread and adjacent points are solved in consecutive CUDA threads.

-
- [1] Charles K Birdsall and A Bruce Langdon. *Plasma physics via computer simulation*. CRC press, 2004.
 - [2] Ferit Büyükkeçeci, Omar Awile, and Ivo F Sbalzarini. A portable opencl implementation of generic particle-mesh and mesh-particle interpolation in 2d and 3d. *Parallel Computing*, 39(2):94–111, 2013.
 - [3] Roger W Hockney and James W Eastwood. *Computer simulation using particles*. crc Press, 1988.
 - [4] Kamesh Madduri, Jimmy Su, Samuel Williams, Leonid Oliker, Stéphane Ethier, and Katherine Yelick. Optimization of parallel particle-to-grid interpolation on leading multicore platforms. *IEEE Transactions on Parallel and Distributed Systems*, 23(10):1915–1922, 2012.
 - [5] Andrew Myers, Phillip Colella, and Brian Van Straalen. The convergence of particle-in-cell schemes for cosmological dark matter simulations. *The Astrophysical Journal*, 816(2):56, 2016.
 - [6] George Stantchev, William Dorland, and Nail Gumerov. Fast parallel particle-to-grid interpolation for plasma pic simulations on the gpu. *Journal of Parallel and Distributed Computing*, 68(10):1339–1349, 2008.
 - [7] Bei Wang, Gregory H Miller, and Phillip Colella. A particle-in-cell method with adaptive phase-space remapping for kinetic plasmas. *SIAM Journal on Scientific Computing*, 33(6):3509–3537, 2011.

III. PROBLEM

We will solve the Vlasov-Poisson system using a Particle-in-Cell (PIC) method. The system can be written as

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} - \mathbf{E} \cdot \frac{\partial f}{\partial \mathbf{v}} = 0$$

and

$$\nabla^2 \phi = -\rho$$

with $\mathbf{f}(\mathbf{x}, \mathbf{v}, t)$ is the phase-space distribution, $\mathbf{E}(\mathbf{x}, t)$ is the electric field satisfying $\mathbf{E} = -\nabla \phi(\mathbf{x}, t)$, $\phi(\mathbf{x}, t)$ is the potential, and $\rho(\mathbf{x}, t)$ is the charge density. Under the assumptions that our distribution contains negatively charged particles and the ions form a fixed, neutralizing background field, the charge density and phase-space distribution can be related by

$$\rho(\mathbf{x}, t) = 1 - \int f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}.$$

By representing the distribution as a set of Lagrangian particles with charge

$$q_p = f(\mathbf{x}_p^i, \mathbf{v}_p^i, t = 0) h_x^D h_v^D$$

we can express the PDE above as a set of ODE. Advancing these particles requires four steps: deposition of charge, a poisson solve, interpolation of the fields, and integration. We will use a finite difference method for the poisson solve (or could use some multigrid method like hpgmg), an explicit RK4 method for the integration, and do interpolation and deposition as described in [5]. We will implement this method with high-order kernels and in a way that meets the requirements of existing convergence theory. If time permits, we will also do phase-space remapping - which was shown in [7] to be important for long-time convergence.

IV. IMPLEMENTATION

We propose implementing a hybrid MPI-OpenMP implementation and investigating performance of a remapped PIC method. To prescribe work to different nodes, we will decompose the physical domain. The OpenMP threads will then divide the work up in the phase space region associated with physical space region associated with the node. Previous work has implemented the proposed physical space domain decomposition and found this to have good weak scaling results [7]. Others have focused solely on the particle-mesh steps (the deposition and interpolation) for shared-memory architectures [4] and GPUs [2, 6] since these steps often are the most difficult to implement since they involve nonuniform memory

access. We plan to take one of the two

- [1] Charles K Birdsall and A Bruce Langdon. *Plasma physics via computer simulation*. CRC press, 2004.
- [2] Ferit Büyükkeçeci, Omar Awile, and Ivo F Sbalzarini. A portable opencl implementation of generic particle–mesh and mesh–particle interpolation in 2d and 3d. *Parallel Computing*, 39(2):94–111, 2013.
- [3] Roger W Hockney and James W Eastwood. *Computer simulation using particles*. crc Press, 1988.
- [4] Kamesh Madduri, Jimmy Su, Samuel Williams, Leonid Oliker, Stéphane Ethier, and Katherine Yelick. Optimization of parallel particle-to-grid interpolation on leading multicore platforms. *IEEE Transactions on Parallel and Distributed Systems*, 23(10):1915–1922, 2012.
- [5] Andrew Myers, Phillip Colella, and Brian Van Straalen. The convergence of particle-in-cell schemes for cosmological dark matter simulations. *The Astrophysical Journal*, 816(2):56, 2016.
- [6] George Stantchev, William Dorland, and Nail Gumerov. Fast parallel particle-to-grid interpolation for plasma pic simulations on the gpu. *Journal of Parallel and Distributed Computing*, 68(10):1339–1349, 2008.
- [7] Bei Wang, Gregory H Miller, and Phillip Colella. A particle-in-cell method with adaptive phase-space remapping for kinetic plasmas. *SIAM Journal on Scientific Computing*, 33(6):3509–3537, 2011.