Advancements in GPU-accelerated particle-in-cell simulations with iPIC3D

Pranab J Deka[1] and Fabio Bacchini[1]

[1] *KU Leuven, Centre for mathematical Plasma Astrophysics, Department of Mathematics, Leuven, Belgium*

Particle-in-cell (PIC) simulations are essential for studying kinetic-scale plasma behaviour in astrophysical and laboratory environments, where processes like magnetic reconnection and collisionless shocks can accelerate particles to extreme energies. We present the recent advancements in the semi-implicit iPIC3D code, which now supports GPU computation using CUDA (NVIDIA GPUs) and HIP (AMD GPUs). Comparisons with the CPU version of the code show a factor of 30 improvement in computational speed. We have integrated the exact energy-conserving semi-implicit method and the relativistic semi-implicit method that ensures energy conservation up to machine precision. This is essential to avoid artificial energy growth in the system over long time scales and obtain physically viable results. Furthermore, the code has undergone several methodological improvements that further boost its speedup by 10x. Owing to the implicit nature of the code and the recent algorithmic advancements, we are on course for exascale simulations of relativistic and nonrelativistic plasma, enabling unprecedented spatial resolution and temporal duration.

Introduction

iPIC3D [1] is a semi-implicit PIC code developed primarily to study collisionless plasma dynamics at kinetic scales. Macroparticles, used to represent an ensemble of plasma particles, are evolved in a Lagrangian framework, whereas the moments (such as plasma density, current, pressure, etc.) and the self-consistent electric and magnetic fields are tracked on an Eulerian grid. The three main kernels of iPIC3D are (a) Particle Mover, (b) Moment Gatherer, and (c) Field Solver. Due to the implicit nature of the underlying algorithms, unlike explicit PIC methods, insufficiently resolved scales do not result in numerical instabilities. This allows us to choose time step sizes and spatial grid sizes 10-100 times greater than those used in traditional explicit PIC codes.

iPIC3D is fundamentally based on two algorithms: the **Implicit Moment Method** (**IMM**) is designed to eliminate the stringent numerical stability constraints inherent to explicit PIC methods. It allows for large time steps and coarse spatial grids by implicitly evolving Maxwell’s equations in time. It decouples particle motion and field updates by approximating charge and current densities using Taylor expansion, resulting in a field equation that includes a dielectric-like response, called “implicit susceptibility". This allows the Maxwell solver to account for unresolved fast timescales while still preserving essential kinetic physics, resulting in unconditional linear stability and accurate multi-scale coupling. This makes it highly effective for large-scale, long-duration plasma simulations. However, it introduces approximations in the particle-field coupling (owing to the usage of Taylor expansion) that compromises the energy conservation of the system. This can lead to nonphysical numerical heating or damping over long time scales, necessitating an algorithm like ECSIM that conserves energy exactly to round-off.

The **Energy Conserving Semi-Implicit Method** (**ECSIM,** [2]), combines the simplicity of explicit schemes with the robustness of implicit methods. Unlike traditional semi-implicit PIC algorithms, ECSIM conserves energy exactly up to machine precision without requiring nonlinear iterations between particles and fields. It does so by employing a new particle mover and computing a mass matrix that linearly relates the electric field to the particle current response, enabling a self-consistent field solution. This method is unconditionally stable and allows arbitrary choices of time step and grid size. Typically, iterative linear solvers (e.g., GMRes) are used to compute the solution to the field equations, where, the accuracy of energy conservation is controlled by the tolerance of the linear solver.

The **Relativistic Semi-Implicit Method** (**RelSIM**, [3]) is an extension of the ECSIM algorithm designed to handle relativistic plasmas. It is unconditionally stable with respect to time step and grid size, allowing it to bypass constraints like the resolution of the Debye length or the relativistic plasma frequency. RelSIM supports arbitrary relativistic velocities, making it particularly suitable for astrophysical and high-energy plasma applications where relativistic effects are significant. This combination of physical fidelity and numerical efficiency makes RelSIM a powerful tool for kinetic plasma simulations in extreme regimes.

Latest Developments

Scalable Parallel Astrophysical Codes for Exascale (SPACE; https://www.space-coe.eu/) is a Centre of Excellence (CoE) focused on redesigning seven astrophysical codes, including iPIC3D, and bringing them to exascale performance, i.e., 10^{18} operations per second. The numerical and algorithmic innovations in iPIC3D within SPACE CoE will open doors to large-scale, long-duration, high-fidelity (relativistic) kinetic plasma simulations on state-of-the-art high performance computers (such as GPUs) for a wide range of physical scenarios.

Module	iPIC3D-GPU	iPIC3D-CPU	**Speedup**
Particle Mover	0.542 s	21.891s	40.4
Moment Gatherer	0.123 s	12.271 s	99.8
Field Solver	0.185 s	0.183 s	0.98
Total	0.870 s	35.007 s	**40**

Table 1: A comparison of the runtimes of iPIC3D-GPU and iPIC3D-CPU.

The scalability of iPIC3D depends on the number of particles, which determines the runtime of the Particle Mover (updating position and velocity of the particles at each cycle). The Moment Gatherer module interpolates particle attributes to the grid (and vice versa) to compute charge, current, and the mass matrix. Both the Particle Mover and Moment Gatherer tend to be compute-bound for most input configurations. As such the Particle Mover and Moment Gatherer modules, for the IMM, were offloaded to GPUs (by our collaborators at KTH Sweden). The Field Solver uses an iterative method (GMRes) to update the electromagnetic fields. GMRes is based on matrix-free Krylov subspace projections which can easily saturate the memory on GPUs, thereby reducing the efficiency of the algorithm, and so the field solver continues to run on CPUs. This introduces an overhead of data communication between the GPU and CPU at every time step. However, this is an acceptable trade-off considering the speedups obtained in the computation of the GPU-offloaded modules (Table 1). We have recently implemented ECSIM and RelSIM in the publicly-available CPU and GPU versions of iPIC3D (https://github.com/Pranab-JD/iPIC3D-CPU-SPACE-CoE and https://github.com/Pranab-JD/iPIC3D-GPU-SPACE-CoE), showcasing excellent scalability (Fig 1).

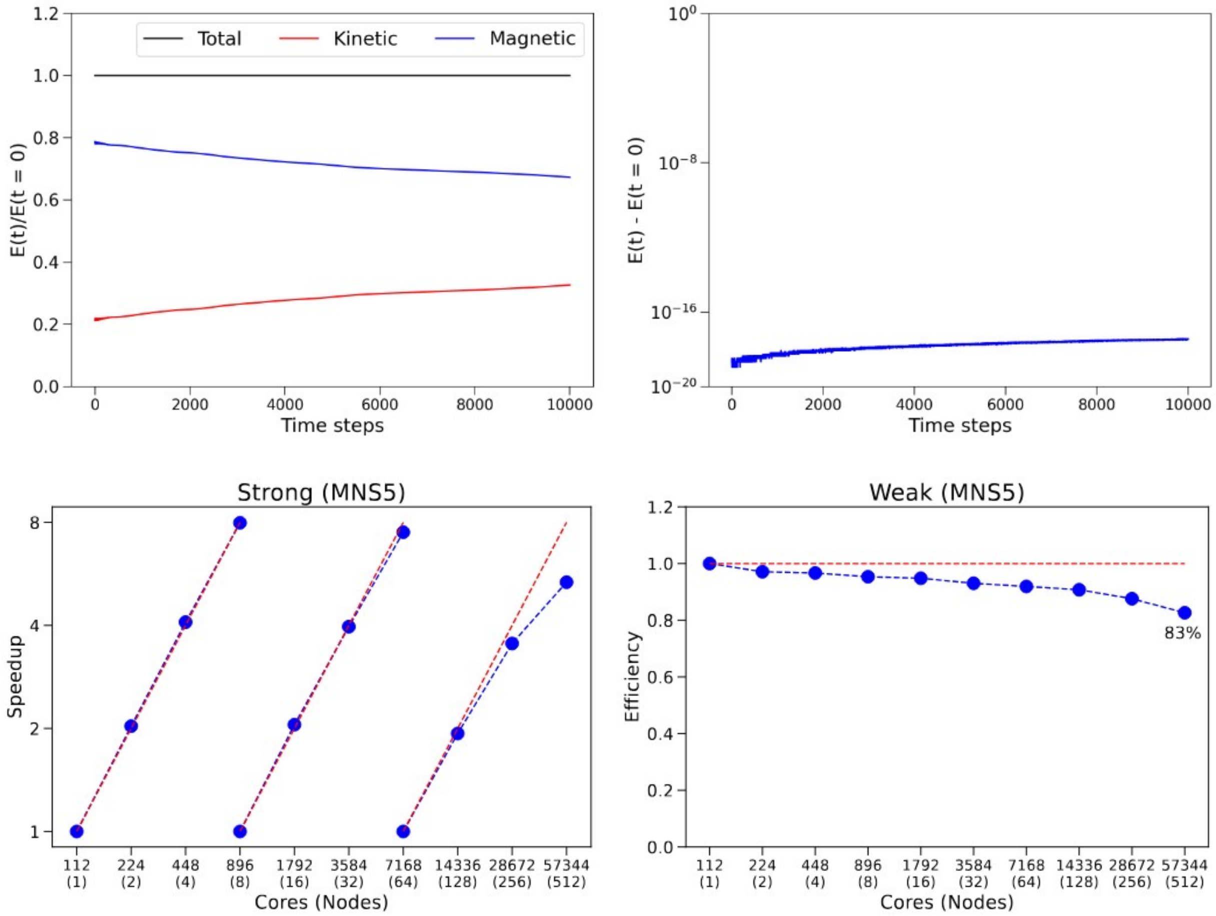

Fig 1: Top: Test case of magnetic reconnection leading to conversion of magnetic energy to kinetic energy of particles whilst conserving energy exactly. Bottom: Strong (left) and weak (right) scaling of iPIC3D on the MareNostrum cluster for a 2D Maxwellian.

Resolution	iPIC3D-CPU (new code)	ECSIM (old code)	Ratio
8192 x 4096 (2D)	48 x 48	30 x 30	2.5
128 x 128 x 128 (3D)	33 x 33 x 33	25 x 25 x 25	2.3

Table 2: Maximum numbers of particles per cell for selected resolution on 64 nodes.

Additionally, we list the following developments in the code.

1. We have developed, for the first time, a **documentation** of the code, consisting of a user guide and a developers' guide.

2. We have added the option of storing output data in **single or double precision**. Our tests show a reduction in the memory requirements by a factor of 1.8 – 1.9, should the data be saved in single over double precision.

3. We have incorporated user-defined restart output cycles, i.e., frequency with which restart data is written to files. In cases of premature abruption of simulations, one can restart the simulations from a given checkpoint, improving the **resilience** of the code.

4. As PIC simulations can consist of millions to billions (or even trillions) of particles, we have implemented **downsampling** the particles by a user-defined downsample factor which allows for consolidating a large number particles to a modest number without loss of physics, thereby significantly reducing the overall memory requirements.

5. Owing to improvements in the data structures and optimisations of arrays, the new code supports over twice as many particles per cell over the old code (Table 2).

The new CPU version of iPIC3D is currently being used for production runs of simulations of the firehose instability in the heliosphere, Kelvin–Helmholtz instability in the Earth's magnetosphere, magnetic reconnection in the Earth's magnetotail, and relativistic magnetic reconnection in highly-magnetised relativistic plasmas around black holes and neutron stars.

References

[1] Markidis, S., Lapenta, G., & Rizwan-uddin. 2010, Math. Comput. Simul., 80, 1509

[2] Lapenta, G., 2017, J. Comput. Phys., 334, 349

[3] Bacchini, F., 2023, 2023, ApJS, 268, 60