CATEGORY: COMPUTATIONAL PHYSICS - CP05

POSTER P6183 | CONTACT NAME John Osorio: john@sirius.utp.edu.co

GPU TECHNOLOGY CONFERENCE

Universidad Tecnológica de Pereira

Implementation of the Vlasov-Poisson Equations in 2D Using PIC on GPUs

{ John Osorio[1], Yensy Gomez[2], Julio Vargas[3] and Oscar Henao[4]} Universidad Tecnologica de Pereira - Sirius[1,2], GIGA[3] and Automatica[4]

Abstract

The Particle-in-cell Method (PIC) is a computational method that allows to solve theoretical models such as the kinetic description of plasma [1]. To study plasma it is necessary to understand the behaviour of the particles through differential equations such as the Vlasov-Poisson equations. The simulation of methods such as PIC consumes several computational resources due to the amount of particles that are used, hence, the processing capacity of regular CPUs does not satisfy the execution time requirements.

This work presents the implementation of the PIC code in 2D to simulate two-stream instability in a cyclic and conservative simulation space while introducing 2D PIC CUDA [2] implementation for the GPU usage achieving an improvement (17x) in the execution time for a grid of 64*64 cells for 800,000 particles and up to 13x for a grid of 517*517 for 800,000 particles.

Introduction

Plasma is an ionized hot gas consisting of positive ions, free electrons and neutral atoms that is produced when a laser beam hits a solid. Plasma, due to its collective behaviour expands in a non-reactive space which makes it a hydrodynamic problem that can be described by the Vlasov, Poisson and Maxwell equations [1]. Vlasov-Poisson equations describe the movement of charged particles that interact in electrostatic fields which represents their movement [3].

Since the behaviour of the fields that interact among themselves is non-linear, it becomes necessary the implementation of computational methods based on particles for its simulation [4], since the calculation with a big amount of data is difficult from an analytical point of view.

Equations

Implemented Equations:

$$\frac{\partial f}{\partial t} + v_x \frac{\partial f}{\partial x} + v_y \frac{\partial f}{\partial y} + E_x \frac{\partial f}{\partial v_x} + E_y \frac{\partial f}{\partial v_y} = 0 \quad (1)$$

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = \phi(x, y) \quad (2)$$

$$\frac{\partial v_p x}{dt} = \frac{q_s}{m_s} E_p(\vec{r}) \quad (3)$$

$$f(p, v) = \frac{n_0}{2} \left\{ \frac{1}{\sqrt{2\pi}} e^{-(v-v_b)^2/2} + \frac{1}{\sqrt{2\pi}} e^{-(v+v_b)^2/2} \right\} \quad (4)$$

Algorithm

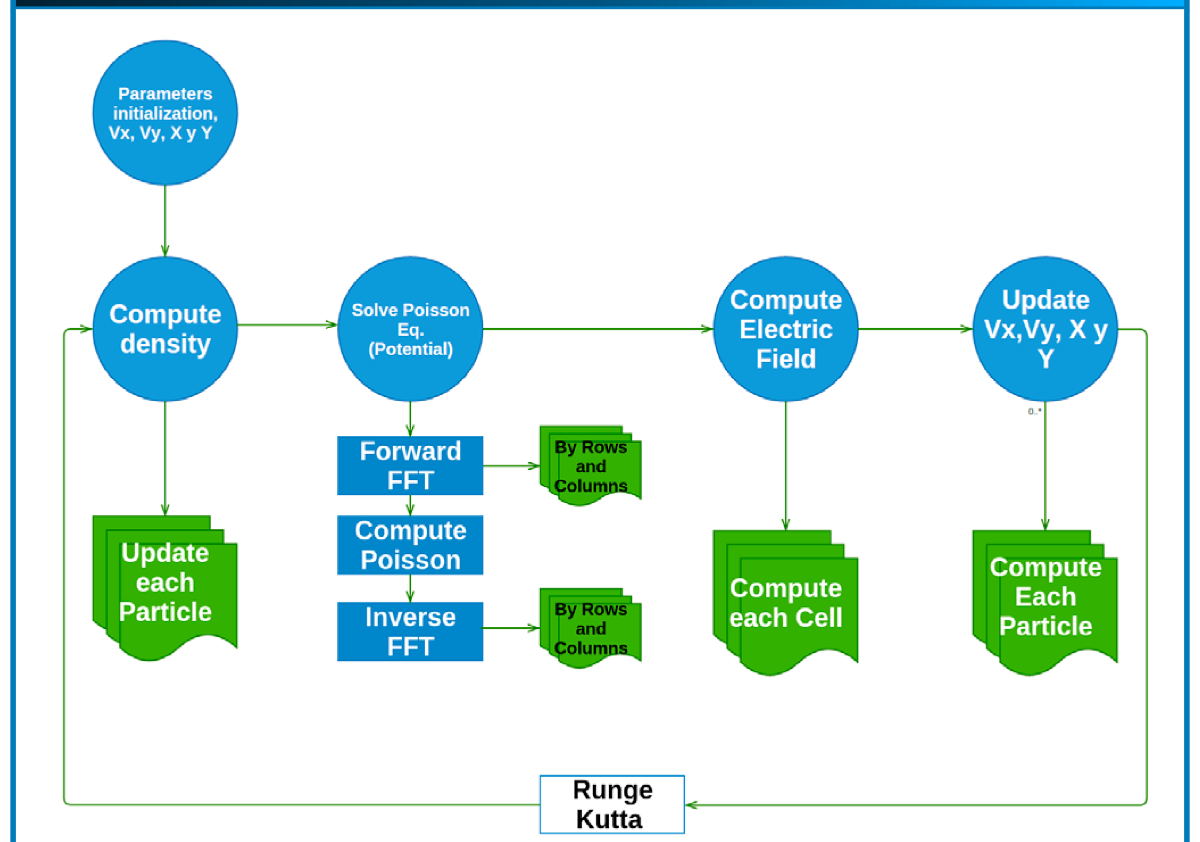

Figure 1: Parallel Algorithm Implemented

Results

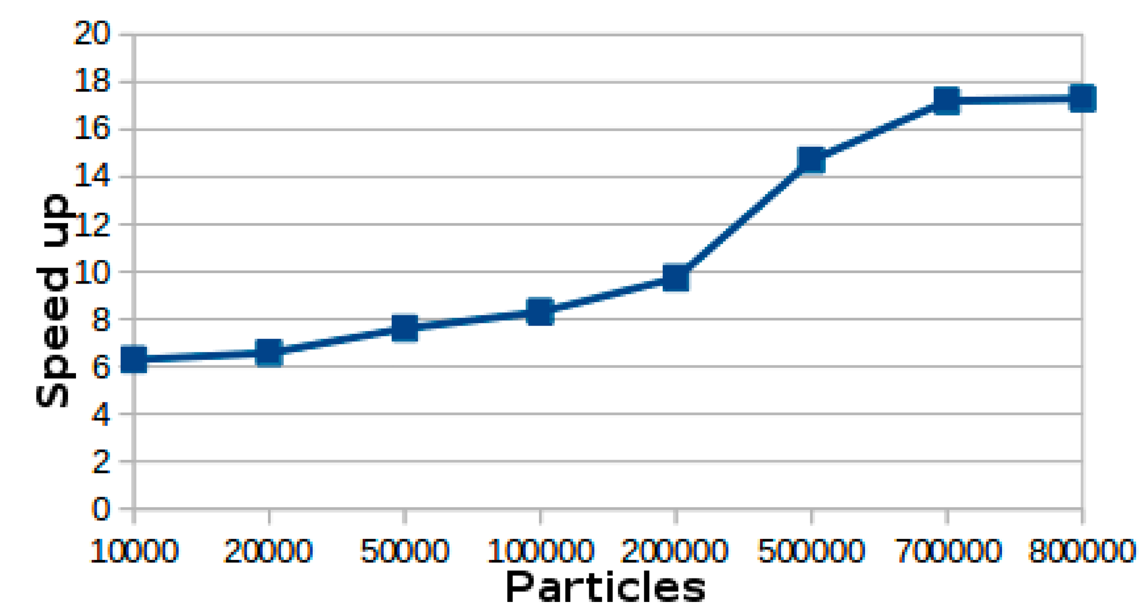

Figure 2: Speed Up

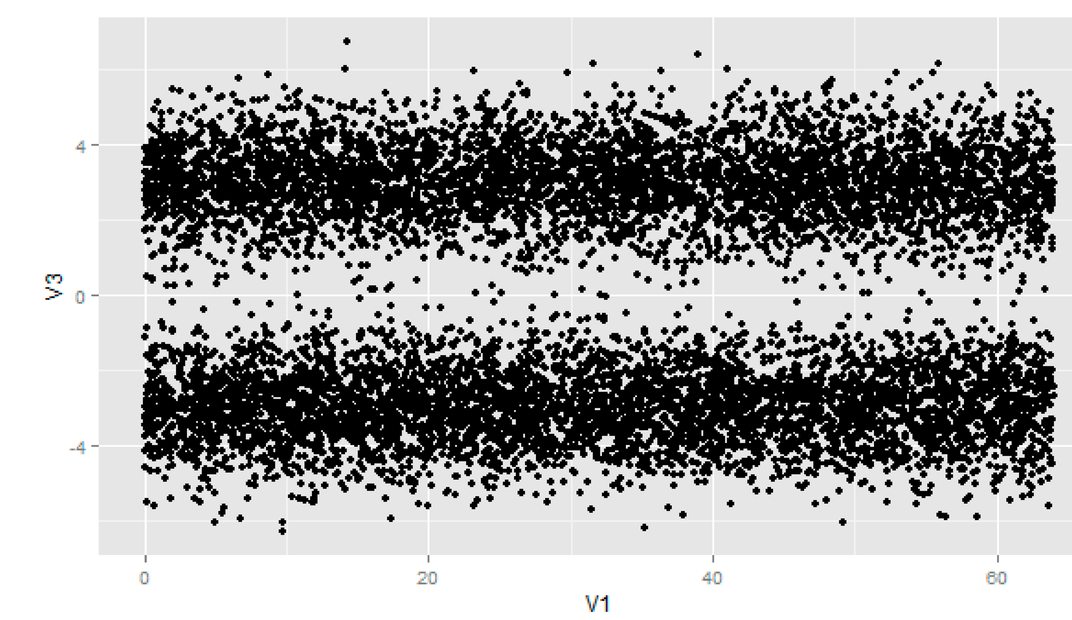

Figure 3: Iteration 0

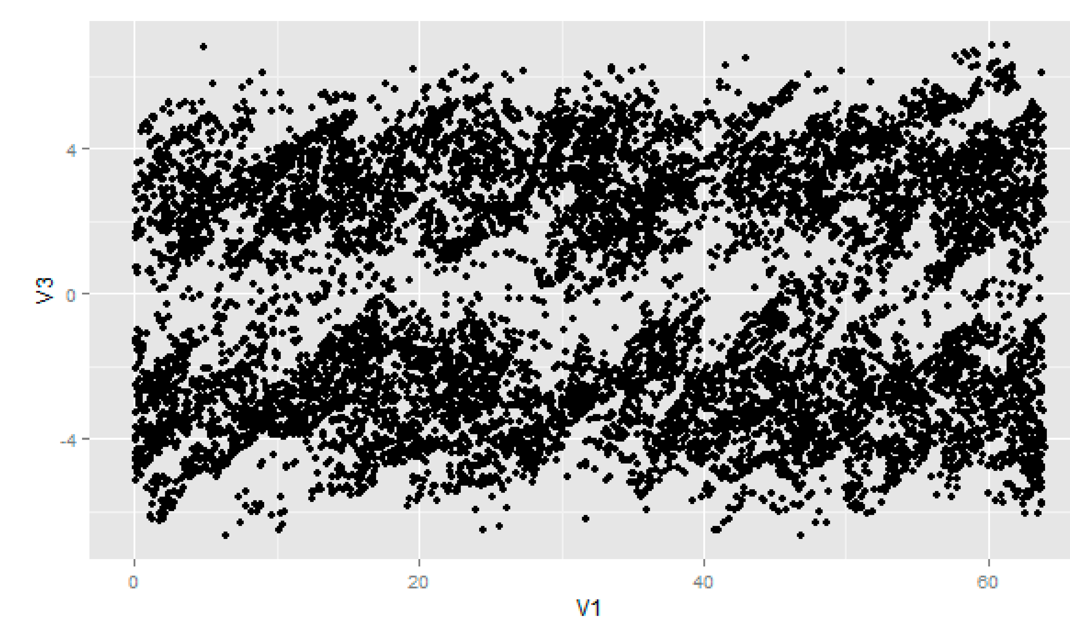

Figure 4: Iteration 200,000

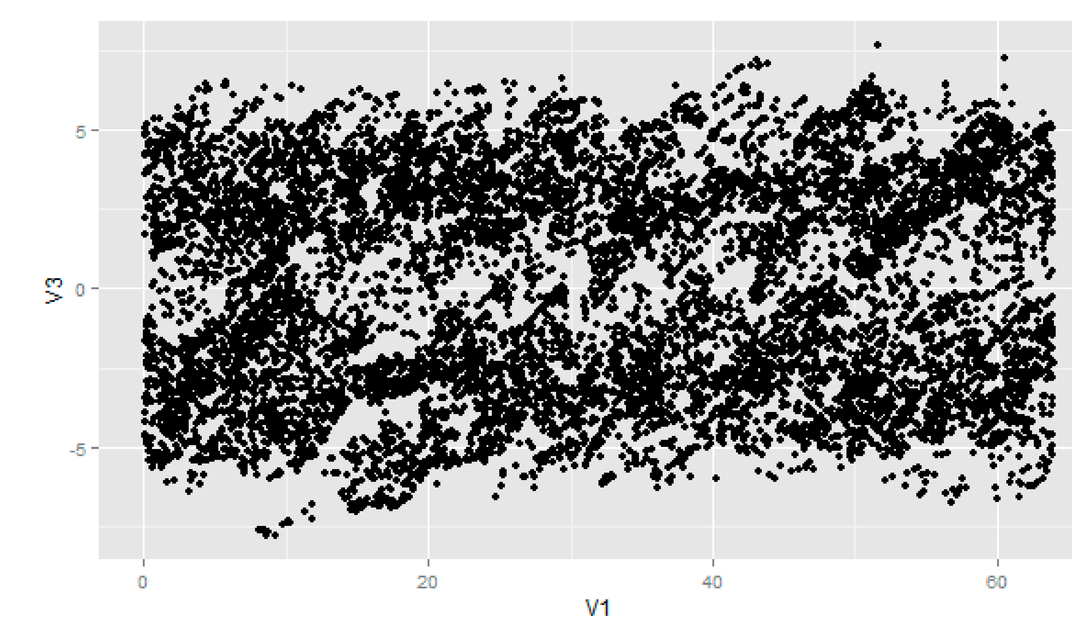

Figure 5: Iteration 400,000

Future Research

- Study a memory-management scheme. Its limit has been reached due to the big amount of data.
- Implementation of the algorithm in a distributed memory environment. The parallel and sequential algorithms consume a lot of resources.
- Implementing the Vlassov-Poisson equations in 3D.

Conclusion

- It was possible to execute the simulation in grid and particle sizes that were not possible in previous simulations.
- The evolution of the simulation model allowed us to determine how the electric fields of some particles affect others.

Contact Information

Web sirius.utp.edu.co

Email john@sirius.utp.edu.co

References

[1] *Plasma Physics via Computer Simulation.*

[2] *Programming Massively Parallel Processors.*

[3] Vlasov on gpu.

[4] The expansion of a plasma into a vacuum.