

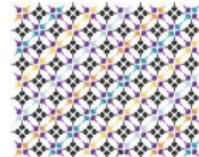
Particle in Cell Simulation: A static approach to electrodynamics

Sebastian Kaloš and Marko Brnović

New York University

PHYS-UA 210 Computational Physics

December 1st



Introduction

oooo

Methodology

○
○
○
○○
○
○

Results

○○
○○
○○○

Discussion

○

References

○

Outline

1 Introduction

2 Methodology

- Object-oriented approach
- Discretizing charge density
- Solving the Poisson Equation
- Leapfrog integration
- Code

3 Results

- Closed box of protons
- Plasma interacting with charged boundary
- Two-stream instability

4 Discussion

5 References

Particle in Cell

- ▶ A go-to approach for modeling plasma, fluids, gravity, accelerator physics, etc.
- ▶ For an N-body problem, there are $\frac{N(N-1)}{2}$ interactions (each particle influences each other particle) \Rightarrow infeasible [Rodríguez-Patiño(2020)]
- ▶ Based on discretizing the domain into a "mesh", and then solving the Poisson equation on each node locally:

$$\nabla^2 \phi = f \tag{1}$$

where $f = -\frac{\rho}{\epsilon_0}$ for electromagnetism, $f = 4\pi G\rho$ for gravity, etc [LLC(2010)].

Goals of the project

General direction

- ▶ Write a scalable electrostatic 2D PIC code from ground up
- ▶ Demonstrate the applicability of the method to plasma simulations

Interesting Simulations

- ▶ Plasma interacting with charged boundary
- ▶ Two stream instability

Algorithm

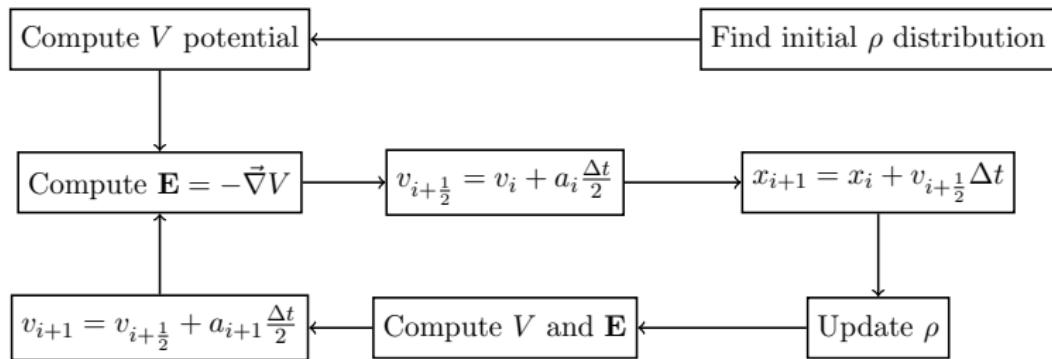


Figure 1: Flowchart for the algorithm

Limitations of our code

- ▶ **Non-relativistic.** Potential changes propagate at finite speed (ignored)
- ▶ **Electrostatic.** Moving particles themselves generate magnetic fields!
- ▶ **2D**
- ▶ **Small N.** Realistic simulation would require $\sim 10^{23}$ particles, world's best simulations achieve $\sim 10^9$. We have a few thousands.

First step: from charge density to potential

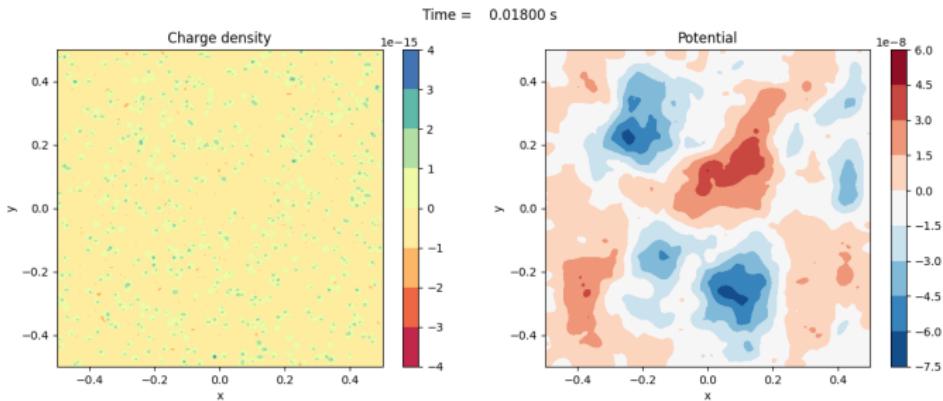


Figure 2: Example of a charge density distribution, and the corresponding potential

How to get there..?

Object-oriented approach

Locating particles:

- ▶ Each particle, node and a cell in the domain is an instance of a class
- ▶ at a step Δt , each particle is located in a cell and associated with a quartet of nodes

Discretizing charge density

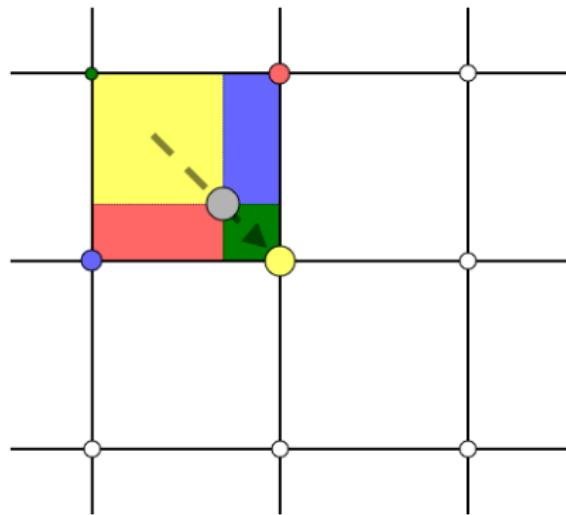


Figure 3: Particle's charge is split among 4 nodes with weights corresponding to the colored areas (normalized)
[Birdsall and Langdon(2018)]

Finite difference method: solving a linear system

Stencil iteration:

$$\frac{\phi_{i-1,j} + \phi_{i+1,1} - 4\phi_{i,j} + \phi_{i,j-1} + \phi_{i,j+1}}{h^2} = -\frac{\rho}{\epsilon_0}$$

$$\begin{bmatrix} 4 & -1 & \dots & -1 & & \dots & 0 \\ -1 & 4 & -1 & & -1 & & \vdots \\ & -1 & 4 & -1 & & -1 & \\ \vdots & & -1 & 4 & -1 & & \ddots \\ -1 & & & -1 & & & \\ & & & & -1 & & \\ & & & & & \ddots & \\ & & & & & & -1 & 4 & -1 \\ \vdots & & & & & & & -1 & 4 & -1 \\ 0 & \dots & & & & & & & -1 & 4 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_n \\ \vdots \\ \phi_m \\ bound. \\ cond. \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_n \\ \vdots \\ \rho_m \\ \epsilon_0^{-1} \end{bmatrix}$$

FFT and scipy-based dstn

- ▶ Spectral method
- ▶ Discrete sine transform of the charge density, and then another one to get the potential
- ▶ Ignores boundary conditions (makes the periodic)
- ▶ Extremely fast (up to 5x the computational efficiency)

Leapfrog integration

- ▶ Calculating force at each particle at $t = idt$ and integrating velocity over dt leads to numerical instabilities
- ▶ Instead, an average velocity between $t = idt$ and $t = (i + 1)dt$ needs to be used to ensure time reversibility

Kick-drift-kick form of the algorithm:

$$v_{i+\frac{1}{2}} = v_i + a_i \frac{\Delta t}{2} \quad (2)$$

$$x_{i+1} = x_i + v_{i+\frac{1}{2}} \Delta t \quad (3)$$

$$v_{i+1} = v_{i+\frac{1}{2}} + a_{i+\frac{1}{2}} \frac{\Delta t}{2} \quad (4)$$

Main loop

```
1  step_counter = 0 #counter used for displaying
2  current time in an animation
3
4  for i in range(n_of_timesteps):
5      update_particle_motion(Update_Position = True) # A
6          function which updates both the position and the
7          velocity
8
9      ---Animation of the particles motion---
10
11     #Compute charge density, potential and E-field at
12     time i:
13
14     rho = update_charge_density()
15     V = solve_potentialifft()
16     update_E_field2()
17
18     ---Animation of the potential and charge density
19     ---
20
21     #Updating just the velocity
22
23     update_particle_motion(Update_Position = False)
24     step_counter += 1
25     print("Solved step",step_counter,"/",
26 n_of_timesteps)
```

Listing 1: Main loop

Introduction

oooo

Methodology

○
○
○
○
○○
○
○

Results

●○
○○
○○○

Discussion

○

References

○

Closed box of protons: animation

Energy considerations

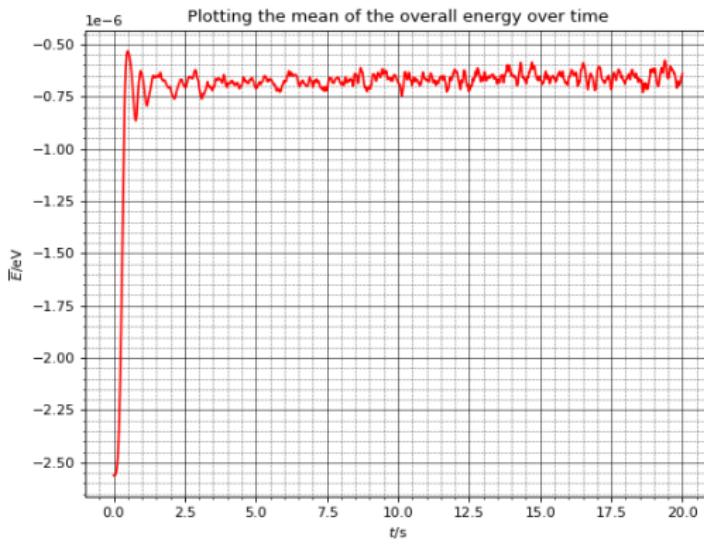


Figure 4: Mean energy propagation of protons in a box

Introduction

oooo

Methodology

○
○
○
○
○○
○
○

Results

○○
●○
○○○

Discussion

○

References

○

Plasma interacting with charged boundary: motion animation

Introduction

oooo

Methodology

○
○
○
○
○○
○
○

Results

○○
○●
○○○

Discussion

○

References

○

Plasma interacting with charged boundary: potential animation

Introduction

oooo

Methodology

○
○
○
○
○○
○
○

Results

○○
○○
●○○

Discussion

○

References

○

Two-stream instability: Particle animation

Introduction

oooo

Methodology

○
○
○
○
○○
○
○

Results

○○
○○
○○○

Discussion

○

References

○

Two-stream instability: Phase space

Introduction

oooo

Methodology

○
○
○
○
○○
○
○

Results

○○
○○
○○●

Discussion

○

References

○

Two-stream instability: Potential animation

Discussion and Conclusions

Further improvements required for real plasma simulations

- ▶ Our "toy" case study does not reflect a realistic scenario
- ▶ $\mathbf{h} \gg \lambda_D$ The cell size needs to be smaller than the Debye length to reach sensible results ("grid instability").
- ▶ Each computational particle should represent a "superparticle", rather than a single electron/proton.
- ▶ Conservation laws!

Conclusions

- ▶ The algorithm works generally very well, and can be used as a starting point towards more realistic simulations.

References



Rodríguez-Patiño.

Implementation of the two-dimensional electrostatic particle-in-cell method.

American Journal of Physics, 88(2):159–167, 2020.



Particle In Cell Consulting LLC.

The electrostatic particle in cell (es-pic) method.

PICC Blog RSS, 2010.

URL http://w3techs.com/technologies/overview/content_language/all.



Charles K Birdsall and A Bruce Langdon.

Plasma physics via computer simulation.

CRC press, 2018.