

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS

Abstract

Bachelor of Engineering (Hons.)Mechanical Engineering

Development of a 2D3V particle-in-cell code

by Tejas MANE

The particle in cell(PIC) method used to solve the Vlasov-Maxwell system of partial differential equations is implemented. Equations for motion for macro particles, charge and current deposition are derived. The violation of the continuity equation is illustrated and a charge conserving current deposition scheme preserving the continuity equation is implemented. A 2D2V/2D3V PIC code implementing periodic boundary conditions is developed. The evolution equations for linear theory of plasmas are derived and the observed Landau damping through analytical calculations is compared with the damping observed via the particle in cell code. Particle noise leading to a high margin of error at higher times is observed. The need for higher order shape factors and alternative methods is established. The 2D3V PIC code¹ is written using high performance scientific computing libraries capable of running on CPU's, OPENCL and Nvidia CUDA devices.

¹Code hosted on <https://github.com/TejasMane/2D2V-2D3V>

Contents

Declaration of Authorship	i
Certificate	ii
Abstract	iii
Acknowledgements	iv
Contents	v
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 The Vlasov-Maxwell system of equations	1
2 Particle in cell simulations	3
2.1 The discrete particle based distribution function	3
2.2 Shape factors and macro particles	4
2.2.1 Why use Macro particles	6
3 Normalization	8
4 Particle pusher	10
4.1 Equations of motion for computational particles	10
4.1.1 Zeroth moment of the distribution function	10
4.1.2 First moment of position of the distribution function	11
4.1.3 Deriving the first moment of velocity of the distribution function	12
4.2 Boris pusher	14
5 Charge deposition	16
5.1 The direct charge deposition scheme	16
6 Current deposition	19
6.1 The direct current deposition scheme	19
6.2 Charge conserving current deposition scheme	22

7 The field solver	25
7.1 The finite difference time domain algorithm.In collaboration with Shyam Sundar Sankaran	25
7.2 Initialization of fields	29
7.2.1 FFT based Poisson solver	30
8 Summary	32
9 Linear theory of Plasmas	35
9.0.1 Linear Theory Current approach	38
9.0.2 Comparison of Linear theory calculations with PIC	39
10 Results	41
A Testing individual components of the code	44
A.1 Testing the particle pusher	45
A.2 Testing the fdtd algorithm	46
A.3 Combined unit test for fdtd and the Boris algorithm	47
A.4 Testing the FFT based Poisson solver	51
A.5 Testing Umeda's charge conserving current deposition	52
B Implementing periodic boundary conditions	53
Bibliography	55

B.1	1D periodic boundary conditions for charge deposition on a central grid	53
B.2	1D periodic boundary conditions for current deposition on a staggered grid	54

B.1	1D periodic boundary conditions for charge deposition on a central grid	53
B.2	1D periodic boundary conditions for current deposition on a staggered grid	54

Chapter 1

Introduction

1.1 The Vlasov-Maxwell system of equations

The Vlasov-Maxwell system of equations describes a weakly interacting collisionless plasma using a distribution function $f(x, v, t)$ where $f(x, v, t)$ is the probability distribution density of finding particles at position x , velocity v at a given time t .

The Vlasov equation, also known as the collisionless Boltzmann equation is a partial differential equation used to describe collisionless plasmas with long range Coulomb interactions:

$$\frac{\partial f_s}{\partial t} + v \cdot \nabla f_s + \frac{F}{m} \cdot \frac{\partial f_s}{\partial v} = 0 \quad (1.1)$$

Where f_s is the distribution function for the particle species.

The long-range interactions are solved for self-consistently using the Maxwell's equations:

$$\nabla \cdot B = 0 \quad (1.2)$$

$$\nabla \cdot E = \rho \quad (1.3)$$

$$\nabla \times E = -\frac{\partial B}{\partial t} \quad (1.4)$$

$$\nabla \times B = +\frac{\partial E}{\partial t} + J \quad (1.5)$$

where ρ and J are as follows:

$$\rho(x, v, t) = \sum_s q_s \int f(x, v, t) dv \quad (1.6)$$

$$J(x, v, t) = \sum_s q_s \int v f(x, v, t) dv \quad (1.7)$$

In this thesis we solve the equations [1.1,1.2,1.3,1.4,1.5,1.6 and 1.7] using the particle-in-cell method and implement it in an efficient solver designed to run on CPU, OPENCL and CUDA devices [1]. Plasmas with long range interactions are often studied using particle in cell simulations. The whole layout of the thesis document is as follows :

- **Chapter 2:** The differential equations solved via PIC simulations, macro particles and the various shape factors associated with these particles is discussed.
- **Chapter 3:** The normalization scheme used in the PIC code is discussed.
- **Chapter 4:** Equations of motion for the macro particles are derived and the Boris algorithm used for the same is discussed.
- **Chapter 5:** The equations for charge deposition used in the PIC method are derived.
- **Chapter 6:** Equations for current deposition via the direct current deposition are derived. The continuity equation and charge conserving current deposition schemes are discussed. The algorithm used to implement Umeda's current deposition scheme[2] is presented.
- **Chapter 7:** The finite difference time domain algorithm used to evolve Maxwell's equations is presented.
- **Chapter 8:** The simulation cycle for the entire particle in cell code is briefly summarized.
- **Chapter 9:** The theoretical calculations behind linear theory of plasmas to obtain linear Landau damping is discussed
- **Chapter 10:** The Landau damping observed via the linear theory calculations, Cheng Knorr and the PIC code are compared.
- **Appendix** Unit tests for the various components of the PIC code and the periodic boundary conditions used for charge and current deposition are presented.

Chapter 2

Particle in cell simulations

2.1 The discrete particle based distribution function

PIC is one of the methods to solve the Vlasov-Maxwell system of equations [1.1, 1.2, 1.3, 1.4, 1.5, 1.6 and 1.7], using a discretized set of equations that govern the motion of a finite set of charged particles:

$$\frac{x^{n+1} - x^n}{\Delta t} = v^{n+\frac{1}{2}} \quad (2.1)$$

$$\frac{v^{n+\frac{3}{2}} - v^{n+\frac{1}{2}}}{\Delta t} = \frac{q}{m} (E(x^{n+1}, t) + v^{n+1} \times B(x^{n+1}, t)) \quad (2.2)$$

$$\nabla \cdot \vec{B} = 0 \quad (2.3)$$

$$\nabla \cdot \vec{E} = \rho \quad (2.4)$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (2.5)$$

$$\nabla \times \vec{B} = +\frac{\partial \vec{E}}{\partial t} + J \quad (2.6)$$

\vec{E} , \vec{B} , ρ and \vec{J} are the electric fields, magnetic fields, charge density and current density represented on a Yee lattice respectively. $E_{i,j,k}$ and $B_{i,j,k}$ are evolved using the Yee algorithm. The distribution function for a discrete computational particle system is represented by[3] :

$$f_{pic}(x, v, t) = A \sum_{p=1}^{N_m} w_p S(x, x_p) S(v_x, v_{x,p}) \quad (2.7)$$

Similarly, the distribution function in 3 dimensions is as follows :

$$f_{pic}(x, v, t) = A \sum_{p=1}^{N_m} w_p S(x, x_p) S(y, y_p) S(z, z_p) S(v_x, v_{x,p}) S(v_y, v_{y,p}) S(v_z, v_{z,p}) \quad (2.8)$$

where N_m , $S(x, x_p)$ and $S(v_x, v_{x,p})$ are the number of macro particles, particle shape factors in position(x direction) and velocity space(x direction) respectively. x_p and $v_{x,p}$ are the positions and velocities of the macro particles. A , w_p and N_m are the normalization factor(if any), weight factor (equal to the number of particles composing the macro particle) and the number of macro particles in the domain respectively.

2.2 Shape factors and macro particles

The shape factor $S(x, x_p)$ is usually a b spline whereas $S(v, v_{x,p})$ is a dirac delta function.

$$S(x, x_p) = \frac{1}{\Delta_p} b_l \left(\frac{x - x_p}{\Delta_p} \right) \quad (2.9)$$

$$S(v, v_{x,p}) = \delta(v - v_{x,p}) \quad (2.10)$$

where Δ_p is the spread of the shape function.

Assumptions for the Shape function [4]:

- Finite support(Finite sized particle)
- Symmetry $\implies S(x, x_p) = S(x_p, x)$
- Normalization

$$\int_{-\infty}^{+\infty} S(x, x_p) dx = 1 \quad (2.11)$$

These finite sized computational particles have a shape factor associated with their velocity and spatial components. PIC methods used Dirac delta shape function for both the spatial and velocity components. Now, B-splines are usually used as shape factors for the spatial components of the particles. B-Splines: The first B-spline is the flat-top function defined as:

$$b_0(\xi) = \begin{cases} 1, & \text{if } |\xi| \leq 1/2 \\ 0, & \text{otherwise} \end{cases} \quad (2.12)$$

Subsequent higher order B-splines, b_l can be defined in the following manner[5]:

$$b_l(\xi) = \int_{-\infty}^{\infty} b_0(\xi - \xi') b_{l-1}(\xi') d\xi' \quad (2.13)$$

These B-splines can be thought of as membership functions where the charge deposition on the nodes occurs at a node proportional to the node's membership value(y – axis value) determined by the B-splines. These are further explained in the upcoming sections.

Few PIC codes use splines of order 1 but most use order 0. Higher order particle shape factors are known to reduce numerical particle noise in PIC codes. $\delta(x)$, b_0 , b_1 and b_2 spline functions as particle shape factors results in the nearest grid point(NGP), cloud in cell(CIC), triangular shaped cloud(TSC) and piecewise cubic polynomial deposition(PCS). The weight functions in a one dimensional case for the same are as follows:

- Nearest grid point(NGP):

$$S(x, x_p) = W^0(x) = \begin{cases} 1, & \text{for } |\delta| < \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \quad (2.14)$$

- Cloud in cell(CIC):

$$S(x, x_p) = W^1(x) = \begin{cases} 1 - \delta, & \text{for } |\delta| < 1 \\ 0, & \text{otherwise} \end{cases} \quad (2.15)$$

- Triangular shaped functions(TSC):

$$S(x, x_p) = W^2(x) = \begin{cases} \frac{3}{4} - \delta^2, & \text{for } |\delta| < \frac{1}{2} \\ \frac{1}{2} \left(\frac{3}{2} - |\delta| \right)^2 & \text{for } \frac{1}{2} \leq |\delta| < \frac{3}{2} \\ 0, & \text{otherwise} \end{cases} \quad (2.16)$$

- Piecewise cubic polynomial deposition(PCS)

$$S(x, x_p) = W^3(x) = \begin{cases} \frac{1}{6} (4 - 6\delta^2 + 3|\delta|^3), & \text{for } 0 \leq |\delta| < 1 \\ \frac{1}{6} (2 - |\delta|)^3 & \text{for } 1 \leq |\delta| < 2 \\ 0, & \text{otherwise} \end{cases} \quad (2.17)$$

Where δ is computed as

$$\delta(x_p, X_i) = \frac{(x_p - X_i)}{dx} \quad (2.18)$$

The following figure illustrates these shape functions:

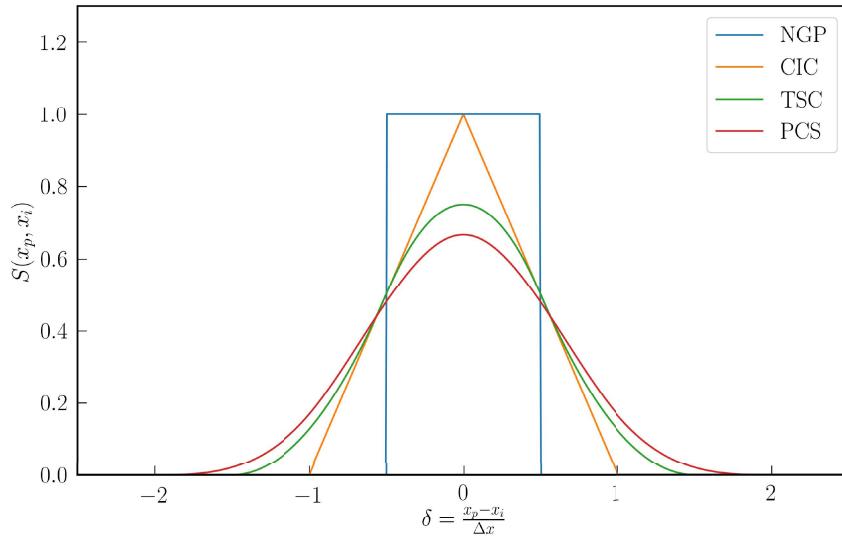


Figure 2.1: The first four b_l splines as shape factors

2.2.1 Why use Macro particles

Modern computers can simulate $\sim 10^{10}$ particles whereas real plasmas can contain number of particles of the order $\sim 10^{20}$ particles. This problem is theoretically solved by coarse graining the system where one macro particle in a PIC code represents several real particles. In this manner, the number density in plasma codes can reflect the real densities found in plasmas.

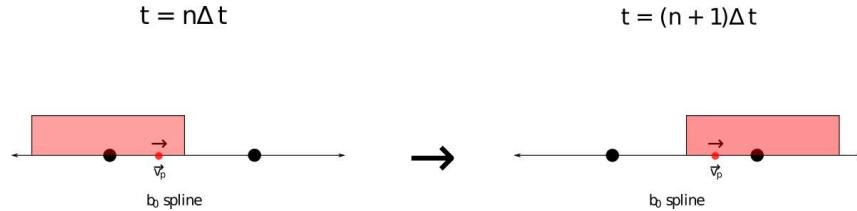


Figure 2.2: b_0 weighted particle viewed by the grid

From the figure shown above[2.2], one can infer that the computational grid views each particle as a finite sized particle. Thus it is viewed as a super particle comprising of several real particles. Coarse graining the system also allows us to handle the extremely low values like charge and mass of the electron well above machine precision depending on the normalization scheme used.

The main advantage of using computational/finite sized particles is the nature of interactions between particles at close ranges. The interpolated electric field acting on the macro particles in the PIC code is Coulombic in nature when the particles are away from each other.

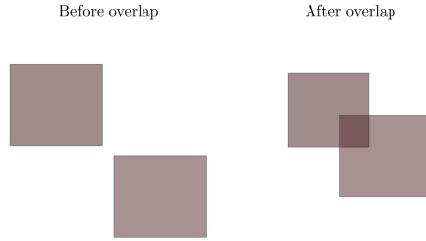


Figure 2.3: Overlap of b_0 weighted particles viewed by the grid

When the two macro particles start approaching each other on the computational grid, the interpolated electric field acting on the particles slowly approaches zero as the macro particles completely overlap each other. This can be verified by taking two particles in the domain with the same position and computing the interpolated electric field acting on the particles in a PIC code. This allows macro particles to approach each other with weak interactions between them at close ranges in PIC simulations. This nature of interaction arises as a numerical effect due to non physical grid forces arising due to use of finite support shape factors on a finite grid. Further details on the same can be found in section 4.8(Page 89) in reference [6]. This makes PIC a viable option to study collisionless plasmas as particles can approach each other with relatively weak interactions between them at close ranges.

The degree of coarse graining or the number of particles represented by the macro particle affects numerical calculations for plasmas with collisions or correlations. Study of collisionless plasmas/Vlasov Maxwell system is not affected by the degree of coarse graining[7].

Chapter 3

Normalization

Normalization used in the PIC code is dependent on user's choice. A suggested normalization scheme is discussed in this section([8] and [9]).

Quantities of interest in PIC code are namely charge (q), mass (m), Temperature (T), distance (x), time (t), velocity (v), Electric field (E) and Magnetic field (B), number density (n_e) and current density (J).

Let the normalized quantity of interested quantity Q be denoted by \bar{Q} . The table below summarizes the normalization used for the fore-mentioned quantities: where e , m_e , ω_{pe}^{-1} , T_{eV} , v_{th} ,

Normalized quantities	Base unit	Evaluation
Charge(\bar{q})	e	$\frac{q}{e}$
Mass(\bar{m})	m_e	$\frac{m}{m_e}$
distance(\bar{x})	λ_d	$\frac{x}{\lambda_d}$
time(\bar{t})	ω_{pe}^{-1}	$\frac{t}{\omega_{pe}^{-1}}$
Temperature(\bar{T})	$m_e c^2$	$\frac{T}{m_e c^2}$
Velocity(\bar{v})	c	$\frac{v}{c}$
Electric field(\bar{E})	$m_e c \omega_{pe} / e$	$\frac{E}{m_e c \omega_{pe} / e}$
Magnetic field(\bar{B})	$m_e \omega_{pe} / e$	$\frac{B}{m_e \omega_{pe} / e}$
Number density(\bar{n}_e)	$\epsilon_0 m_e \omega_{pe}^2 / e^2$	$\frac{n_e}{\epsilon_0 m_e \omega_{pe}^2 / e^2}$
Current density(\bar{J})	$e n_e c$	$\frac{J}{e n_e c}$

Table 3.1: Normalization scheme

n_e and c are the charge of an electron, mass of an electron, plasma period, temperature expressed in energy($K_B T$), thermal velocity, average electron density and speed of light respectively.

The normalization scheme presented in the current section is dependent plasma period (chosen according to the real plasma of interest) is given by the following formula

$$\omega_{pe}^{-1} = \sqrt{\frac{m_e \varepsilon_0}{n_e q_e^2}} \quad (3.1)$$

The temperature is supplied in electron volts which is commonly the norm in plasma physics [10]. This nominal temperature(\bar{T}) corresponds to $k_B T$ of the system of interest. The normalization scheme suggested in this section used the base unit as $m_e c^2$ instead of electron volts(eV).

Note: The normalization scheme presented in this section is used to simulate real plasmas and all the parameters given must correspond to a real plasma of interest.

All the characteristic quantities of the real plasma are evaluated in S.I or c.g.s units and normalized by dividing the computed values by the respective base unit. These normalized quantities are provided as input to the PIC code. The results generated by the PIC code are brought back to S.I units by multiplying the computed results by their respective base units.

Chapter 4

Particle pusher

4.1 Equations of motion for computational particles

The zeroth, x and v moments of the Vlasov equation for the macro particles can be used to derive the equations of motion for the macro particles.

The derivation of equations of motion in a one dimensional scenario is presented in this section. The following notation has been used in the derivation.

$$\langle f \rangle = \int \int f \, dv \, dx \quad (4.1)$$

The Vlasov equation for the distribution function is as follows :

$$\frac{d f_{pic}}{d t} + v \cdot \nabla f_{pic} + \frac{q_m}{m_m} (E + v \times B) \cdot \nabla_{v_m} f_{pic} = 0 \quad (4.2)$$

where $f_{pic}(x, v, t)$ is the distribution function for the macro particles as formerly defined by equation 2.7. The Vlasov equation in x direction is as follows:

$$\frac{d f_{pic}}{d t} + v_x \frac{\partial f_{pic}}{\partial x} + \frac{q_m}{m_m} (E_x + v_{y,z} \times B_{y,z}) \frac{\partial f_{pic}}{\partial v_x} = 0 \quad (4.3)$$

Note: There are other velocities and magnetic fields applicable in y and z dimensions denoted by $v_{y,z}$ and $B_{y,z}$ which are independent of x and v_x in the following derivations.

4.1.1 Zeroth moment of the distribution function

Integrating equation 4.3 with respect to x and v_x results in the following equation.

$$\frac{d \langle f_{pic} \rangle}{d t} + \langle v_x \frac{\partial f_{pic}}{\partial x} \rangle + \langle \frac{q_m}{m_m} (E_x + v_{y,z} \times B_{y,z}) \frac{\partial f_{pic}}{\partial v_x} \rangle = 0 \quad (4.4)$$

Using f_{pic} = zero at $x = \pm\infty$ and integration by parts it can be shown that

$$\left\langle \frac{q_m}{m_m} (E_x + v_{y,z} \times B_{y,z}) \frac{\partial f_{pic}}{\partial v_x} \right\rangle, \left\langle v_x \frac{\partial f_{pic}}{\partial x} \right\rangle = 0 \quad (4.5)$$

$$\implies \frac{d \langle f_{pic} \rangle}{dt} = 0 \quad (4.6)$$

$$\implies \frac{d(A w_p N_m)}{dt} = 0 \quad (4.7)$$

$$\implies w_p N_m = \text{constant} \quad (4.8)$$

In other words, the number of macro particles taken in the domain remains constant.

4.1.2 First moment of position of the distribution function

Multiplying equation 4.3 by x and integrating with respect to x and v results in the following formulation:

$$\frac{d \langle x f_{pic} \rangle}{dt} + \left\langle x v_x \frac{\partial f_{pic}}{\partial x} \right\rangle + \left\langle \frac{q_m x}{m_m} (E_x + v_{y,z} \times B_{y,z}) \cdot \frac{\partial f_{pic}}{\partial v_x} \right\rangle = 0 \quad (4.9)$$

Using integration by parts and $f_{pic}(x, v, t) \rightarrow 0|_{x=\pm\infty}$, it can be shown that $\left\langle \frac{q_m x}{m_m} (E_x + v_{y,z} \times B_{y,z}) \cdot \frac{\partial f_{pic}}{\partial v_x} \right\rangle = 0$.

$$\implies \frac{d \langle x f_{pic} \rangle}{dt} + \left\langle x v_x \frac{\partial f_{pic}}{\partial v_x} \right\rangle = 0 \quad (4.10)$$

The first term on the left hand side of the equation is simplified as follows:

$$\langle x f_{pic} \rangle = A w_p \int S(v_x, v_{x,p}) dv \int \sum_{p=1}^{N_m} x S(x, x_p) dx \quad (4.11)$$

$$\implies \langle x f_{pic} \rangle = A w_p \int \sum_{p=1}^{N_m} x S(x - x_p) dx \quad (4.12)$$

Taking $x - x_p$ equal to x'

$$\implies \langle x f_{pic} \rangle = A w_p \int \sum_{p=1}^{N_m} (x' + x_p) S(x') dx \quad (4.13)$$

$x' S(x')$ being a odd function (Remember $S(x, x_p)$ is symmetric) results in

$$Aw_p \int_{-\infty}^{+\infty} \sum_{p=1}^{N_m} x' S(x') dx = 0$$

$$\implies \langle x f_{pic} \rangle = Aw_p \sum_{p=1}^{N_m} x_p \quad (4.14)$$

The second term of equation 4.10 is simplified in a similar manner as follows:

$$\langle x v_x \frac{\partial f_{pic}}{\partial x} \rangle = \int v_x dv_x \int \sum_{p=1}^{N_m} x \frac{\partial(f_{pic})}{\partial x} dx \quad (4.15)$$

This is simplified in the following manner

$$\int x \frac{\partial f_{pic}}{\partial x} = x f_{pic}|_{-\infty}^{+\infty} - \int f_{pic} dx \quad (4.16)$$

$$\implies \int x \frac{\partial f_{pic}}{\partial x} = - \int f_{pic} dx = -Aw_p S(v_x, v_{x,p}) \quad (4.17)$$

Substituting back in equation 4.15 results in the following expression:

$$\langle x v_x \frac{\partial f_{pic}}{\partial x} \rangle = - \int v_x dv_x \sum_{p=1}^{N_m} S(v_x, v_{x,p}) \quad (4.18)$$

$$\implies \langle x v_x \frac{\partial f_{pic}}{\partial x} \rangle = -Aw_p \sum_{p=1}^{N_m} v_{x,p} \quad (4.19)$$

Equations 4.14 and 4.19 simplifies equation 4.10 as follows arriving at the first equation of motion for macro particles:

$$Aw_p \sum_{p=1}^{N_m} \frac{\partial x_p}{\partial t} = Aw_p \sum_{p=1}^{N_m} v_{x,p} \quad (4.20)$$

$$\implies Aw_p \frac{\partial x_p}{\partial t} = Aw_p v_{x,p} \quad (4.21)$$

$$\frac{\partial x_p}{\partial t} = v_{x,p} \quad (4.22)$$

4.1.3 Deriving the first moment of velocity of the distribution function

Multiplying equation 4.3 by v_x and integrating with respect to x and v_x results in the following equation :

$$\frac{d \langle v f_{pic} \rangle}{d t} + \langle v_x^2 \frac{\partial f_{pic}}{\partial x} \rangle + \langle \frac{q_m v}{m_m} (E_x + v_{y,z} \times B_{y,z}) \frac{\partial f_{pic}}{\partial v_x} \rangle = 0 \quad (4.23)$$

The second term $\langle v_x^2 \frac{\partial f_{pic}}{\partial x} \rangle$ can be evaluated in the following manner:

$$\langle v_x^2 \frac{\partial f_{pic}}{\partial x} \rangle = \int v_x^2 f_{pic} |_{-\infty}^{+\infty} - \int f_{pic} 2v_x \frac{\partial v_x}{\partial x} \quad (4.24)$$

$$\implies \langle v_x^2 \frac{\partial f_{pic}}{\partial x} \rangle = 0 \quad (4.25)$$

This simplifies equation 4.23 as follows:

$$\implies \frac{\partial \langle v f_{pic} \rangle}{\partial t} + \langle \frac{q_m v}{m_m} (E_x + v_{y,z} \times B_{y,z}) \frac{\partial f_{pic}}{\partial v_x} \rangle = 0 \quad (4.26)$$

The second term of equation 4.26 is simplified in the following manner:

$$\langle \frac{q_m v}{m_m} (E_x + v_{y,z} \times B_{y,z}) \frac{\partial f_{pic}}{\partial v_x} \rangle = \frac{q_m}{m_m} \int (E_x + v_{y,z} \times B_{y,z}) dx \int v_x \frac{\partial f_{pic}}{\partial v} dv_x \quad (4.27)$$

Note: $v_{y,z}$ and $B_{y,z}$ in $(E_x + v_{y,z} \times B_{y,z})$ in the above expression is not in the same plane as $v_{y,z}$ and hence can be treated as independent of v_x while evaluating the integral.

$$\begin{aligned} & \langle \frac{q_m v}{m_m} (E_x + v_{y,z} \times B_{y,z}) \frac{\partial f_{pic}}{\partial v_x} \rangle = \\ & \quad - Aw_p \frac{q_m}{m_m} \int dx \int (E_x + v_{y,z} \times B_{y,z}) S(x - x_p) \delta(v_x - v_{x,p}) dv_x \\ & \langle \frac{q_m v}{m_m} (E_x + v_{y,z} \times B_{y,z}) \cdot \frac{\partial f_{pic}}{\partial v} \rangle = -Aw_p \frac{q_m}{m_m} \int dx (E_x + v_{y,z} \times B_{y,z}) S(x - xp) \end{aligned} \quad (4.28)$$

The interpolated Lorentz force is defined by

$$Aw_p \frac{q_m}{m_m} \int (E_x + v_{y,z} \times B_{y,z}) S(x - x_p) dx \quad (4.29)$$

Let the interpolated Lorentz force be

$$Aw_p \frac{q}{m} (E_{i,x} + v_{y,z} \times B_{i,y,z}) \quad (4.30)$$

where

$$E_{i,x} = \int E_x S(x - x_p) dx \quad (4.31)$$

$$B_{i,y,z} = \int B_{y,z} S(x - x_p) dx \quad (4.32)$$

Thus, equation 4.23 is simplified as follows arriving at the second equation of motion for the macro particles:

$$Aw_p \frac{\partial v_{x,p}}{\partial t} = Aw_p \frac{q_m}{m_m} (E_{i,x} + v_{y,z} \times B_{i,y,z}) \quad (4.33)$$

$$\implies \frac{\partial v_{x,p}}{\partial t} = \frac{q_m}{m_m} (E_{i,x} + v_{y,z} \times B_{i,y,z}) \quad (4.34)$$

To summarize, the equations of motion for the macro particles are:

$$\frac{d x_p}{dt} = v_p \quad (4.35)$$

$$\frac{d v_{x,p}}{dt} = \frac{q_m}{m_m} (E_{i,x} + v_{y,z} \times B_{i,y,z}) \quad (4.36)$$

where x_p , $v_{x,p}$, q_m and m_m denote the position, velocity, charge and the mass of the macro particle respectively. One can infer from the above equations that the macro particles move exactly like individual particles as long as the charge to mass ratio($\frac{q}{m}$) remains the same. Use of macro particle/coarse graining the system is discussed in detail in further sections. The particle shape factor for field interpolation, charge and current deposition must be the same for the particle to not experience any self force[6]. For example, one must not implement nearest grid point charge/current deposition and linear field interpolation simultaneously. In other words, the particle shape factor remains consistent through out the code.

4.2 Boris pusher

The Boris algorithm [11] is the standard explicit algorithm used to update velocities in particle in cell simulations. The implementation of the algorithm is as follows: Let $\vec{v} = [v_x, v_y, v_z]$. The velocities are updated by solving the following equation:

$$\frac{\vec{v}^{n+1/2} - \vec{v}^{n-1/2}}{\Delta t} = \frac{q}{m} (\vec{E}^n + \vec{v} \times \vec{B}^n) \quad (4.37)$$

Let \vec{v}^- be defined as:

$$\vec{v}^- = \vec{v}^{n-1/2} + \frac{q \vec{E}}{m} \frac{\Delta t}{2} \quad (4.38)$$

Let \vec{v}' be defined as:

$$\vec{v}' = \vec{v}^- \times \vec{t} \quad (4.39)$$

where $t = \frac{q\vec{B}}{m} \frac{\Delta t}{2}$.

$$\vec{v}^+ = \vec{v}^- + \vec{v}' \times \vec{s} \quad (4.40)$$

where $\vec{s} = \frac{2\vec{t}}{1+|t|^2}$. Now $v^{n+1/2}$ is computed as:

$$\vec{v}^{n+1} = \vec{v}^+ + \frac{q\vec{E}}{m} \frac{\Delta t}{2} \quad (4.41)$$

The positions \vec{x}^n are simply updated in the following manner as the standard Verlet algorithm:

$$\vec{x}^{n+1} = \vec{x}^n + \vec{v}^{n+1/2} \Delta t \quad (4.42)$$

Chapter 5

Charge deposition

5.1 The direct charge deposition scheme

$\rho_{i,j,k}$ is computed using:

$$\rho(x, v, t) = \sum_s q \int f(x, v, t) dv \quad (5.1)$$

In PIC, this is

$$\rho(x, t) = q \int f_{pic} dv + \rho_{ions} \quad (5.2)$$

Using equation 1.6 simplifies it as follows:

$$\rho(x, t) = Aq \sum_{p=1}^{N_m} w_p S(x, x_p) S(v, v_{x,p}) dv_x + \rho_{ions} \quad (5.3)$$

$$\rho(x, t) = Aq \sum_{p=1}^{N_m} w_p S(x, x_p) \delta(v_{x,p}, v_{x,p}) + \rho_{ions} \quad (5.4)$$

$$\rho(x, t) = Aq \sum_{p=1}^{N_m} w_p S(x, x_p) + \rho_{ions} \quad (5.5)$$

On a finite discretized grid(\mathbf{x}_i), the charge deposition on a grid node \mathbf{x}_i in a one dimensional case is computed in the following manner(Assuming ρ_{ion} to be constant through out the domain) :

$$\rho(\mathbf{x}_i, t) = \frac{1}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \rho(x) dx \quad (5.6)$$

$$\rho(\mathbf{x}_i, t) = \frac{Aq \sum_{p=1}^N w_p \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} S(x, x_p) dx}{dx} + \rho_{ions} \quad (5.7)$$

This can be simplified in the following manner using the following properties of b splines and Dirac delta functions:

$$b_{l+1}(\varepsilon) = \int_{-\infty}^{\infty} b_0(\varepsilon - \varepsilon_d) b_{l-1}(\varepsilon - \varepsilon_d) d\varepsilon_d \quad (5.8)$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} S(x, x_p) dx = \int_{-\infty}^{+\infty} b_0\left(\frac{x - \mathbf{x}_i}{dx}\right) S(x, x_p) dx \quad (5.9)$$

If the particle shape factor($S(x, x_p)$) is $\delta(x, x_p)$, equation 5.7 simplifies to:

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} S(\mathbf{x}_i, x_p) dx = \int_{-\infty}^{+\infty} b_0\left(\frac{x - \mathbf{x}_i}{dx}\right) \delta(x, x_p) dx \quad (5.10)$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} S(x_i, x_p) dx = b_0\left(\frac{x_p - \mathbf{x}_i}{dx}\right) \quad (5.11)$$

$$\Rightarrow \rho(\mathbf{x}_i, t) = \frac{1}{\Delta x} A w_p q \sum_{p=1}^N b_0\left(\frac{x_p - \mathbf{x}_i}{dx}\right) + \rho_{ions} \quad (5.12)$$

This is also known as nearest grid point charge deposition.

If the particle shape factor is b_0 spline function, equation 5.9 simplifies to:

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} S(x, x_p) dx = \int_{-\infty}^{+\infty} b_0\left(\frac{x - \mathbf{x}_i}{\Delta x}\right) S(x, x_p) dx \quad (5.13)$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} S(x, x_p) dx = \int_{-\infty}^{+\infty} b_0\left(\frac{x - \mathbf{x}_i}{dx}\right) \frac{1}{\Delta x} b_0\left(\frac{x - x_p}{\Delta x}\right) dx \quad (5.14)$$

Taking $\frac{x - x_p}{\Delta x}$ as ϵ , it can be shown that equations 2.11 and 5.8 simplifies the right hand side of the equation as follows :

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} S(x, x_p) dx = b_{0+1}\left(\frac{\mathbf{x}_i - x_p}{\Delta x}\right) \quad (5.15)$$

$$\Rightarrow \rho(\mathbf{x}_i, t) = \frac{1}{\Delta x} A w_p q \sum_{p=1}^N b_1\left(\frac{x_p - \mathbf{x}_i}{\Delta x}\right) + \rho_{ions} \quad (5.16)$$

This results in linear weighting/cloud in cell charge deposition. This considerably reduces particle noise in the simulation. Subsequently, charge deposition using higher order shape functions can be derived in a similar manner. The charge deposition for arbitrary shape factors can be generalized in the following manner:

$$\rho(\mathbf{x}_i, t) = \frac{1}{\Delta x} Aw_p q \sum_{p=1}^N b_{1+1} \left(\frac{x_p - \mathbf{x}_i}{\Delta_p} \right) \quad (5.17)$$

where $b_{1+1} \left(\frac{x_p - \mathbf{x}_i}{\Delta_p} \right)$ is the shape factor for the particle.

Chapter 6

Current deposition

6.1 The direct current deposition scheme

$$J(x, v, t) = \sum_s q \int v f(x, v, t) dv \quad (6.1)$$

Assuming stationary ions, $J(x, t)$ is computed in a similar manner as charge deposition:

$$J(x, t) = q \int v f_{pic} dv \quad (6.2)$$

$$J(x, t) = q \int \sum_{p=1}^{N_m} v S(x, x_p) S(v, v_{x,p}) dv \quad (6.3)$$

$$J(x, t) = Aw_p q \sum_{p=1}^{N_m} v_{x,p} S(x, x_p) \delta(v_{x,p}, v_{x,p}) \quad (6.4)$$

$$J(x, t) = Aw_p q \sum_{p=1}^{N_m} v_{x,p} S(x, x_p) \quad (6.5)$$

J on a discretized finite one dimensional grid is computed as follows :

$$J(\mathbf{x}_i, t) = \frac{1}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} J(x, t) dx \quad (6.6)$$

$$J(\mathbf{x}_i, t) = \frac{1}{\Delta x} Aw_p q \sum_{p=1}^{N_m} v_{x,p} S(\mathbf{x}_i, x_p) \quad (6.7)$$

where

$$S(\mathbf{x}_i, x_p) = b_0(\mathbf{x}_i, x_p) \implies \text{NGP deposition} \quad (6.8)$$

$$S(\mathbf{x}_i, x_p) = b_1(\mathbf{x}_i, x_p) \implies \text{Linear weighted deposition} \quad (6.9)$$

$$S(\mathbf{x}_i, x_p) = b_{l+1}(\mathbf{x}_i, x_p) \implies b_l \text{ particle shape deposition} \quad (6.10)$$

Figures 6.1, 6.2 and 6.3 illustrate the direct deposition scheme using a b_1 spline shape factor:

$$b_1(\xi) = \begin{cases} 1 - |x - x_i| / \Delta x, & \text{if } |x - x_i| \leq \Delta x \\ 0, & \text{otherwise} \end{cases} \quad (6.11)$$

Charges are deposited at the nodal points using the shape functions:

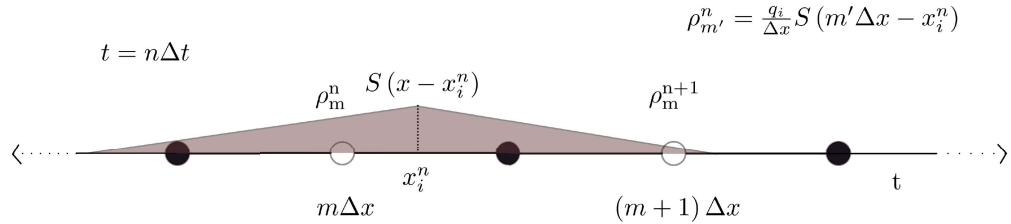


Figure 6.1: Charge and current deposition: Step 1

The currents are then deposited on a staggered temporal and spatial grids using the particle velocities, positions and the same shape function in the following manner:

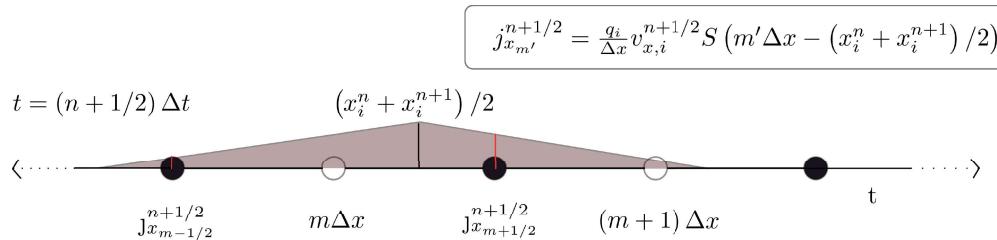


Figure 6.2: Charge and current deposition: Step 2

This cycle is repeated all over again.

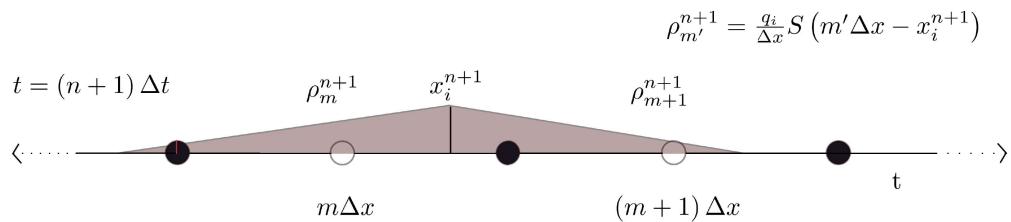


Figure 6.3: Charge and current deposition: Step 3

Shape factors as discussed earlier behave similar to membership functions. Suppose a particle has the membership value of 1 at a nodal point. This implies that the complete charge of the particle is to be deposited on the nodal point. If it has a membership value of 0.5, then half the complete charge is deposited on the grid. Direct current deposition scheme does this in a similar manner as illustrated by the figures shown above [Figures: 6.4, 6.5 and 6.6].

Though Direct current deposition scheme can be easily comprehended, it has a significant drawback because it does not satisfy the continuity equation. The figures shown below clearly show this phenomenon occurring using a b_0 spline shape function. The figures shown below consider a single particle in the domain. **Note:** b_0 spline is given by:

$$b_0(\xi) = \begin{cases} 1, & \text{if } |\xi| \leq 1/2 \\ 0, & \text{otherwise} \end{cases} \quad (6.12)$$

where

$$\xi = (m'\Delta x - x_i) / \Delta x \quad (6.13)$$

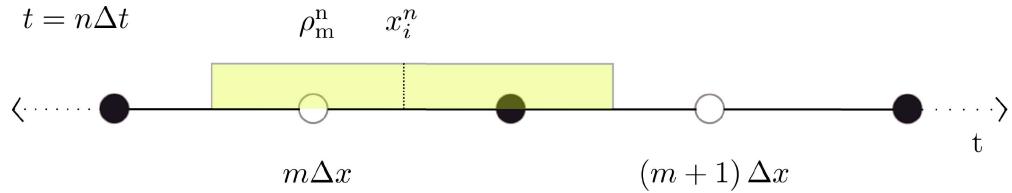


Figure 6.4: Continuity violation: Step 1

Note: Here, there is no spatial variation in the current.

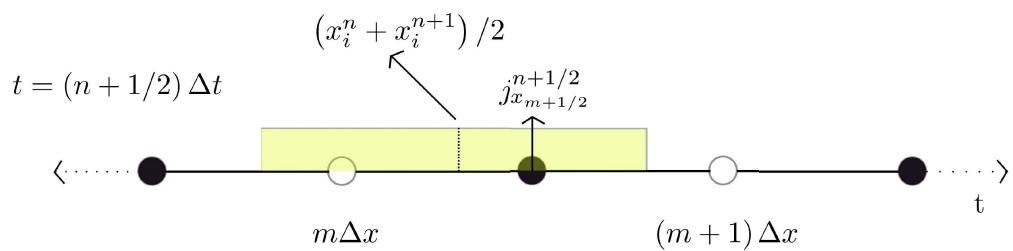


Figure 6.5: Continuity violation: Step 2

Note: Here the ρ deposited on the grid point does not change with time.

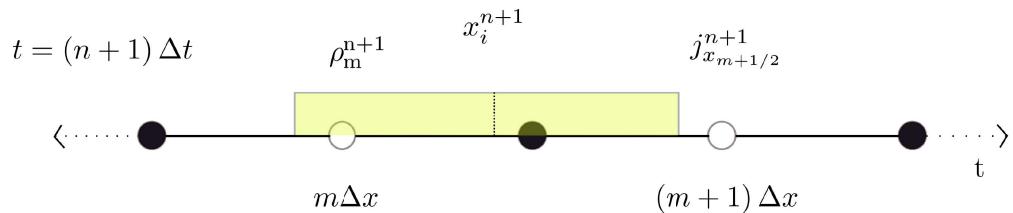


Figure 6.6: Continuity violation: Step 3

Updating the electric fields with the discretized Maxwell-Ampere equation preserves

$$\nabla \cdot E = \rho \quad (6.14)$$

provided that the continuity equation is satisfied at each iteration.

$$\frac{\partial \rho}{\partial t} + j = 0 \quad (6.15)$$

As illustrated by the figures, the PIC loop with direct current deposition does not conserve charge as the continuity equation is not valid at all time steps

There are two alternative solutions adopted to fix this phenomenon namely the Boris correction scheme and to use a charge conserving current deposition scheme instead. Boris correction requires correcting $\nabla \cdot E$ at each iteration. This correction is implemented in the following manner After each iteration(or every few iterations), correct E using:

$$E' = E - \nabla \delta\phi \quad (6.16)$$

where $\delta\phi$ is given by

$$\nabla^2 \delta\phi = \nabla \cdot E - \frac{\rho}{\varepsilon_0} \quad (6.17)$$

The new field E' satisfies the Poisson's equation. To determine $\delta\phi$, There is a need for a Poisson solver to solve the equation shown below

$$\nabla^2 \delta\phi = \nabla \cdot E - \frac{\rho}{\varepsilon_0} \quad (6.18)$$

The Poisson solver can use various techniques such as direct matrix, spectral or relaxation methods to achieve the same.

Another alternative is to instead use a charge conserving current deposition scheme which is discussed in brief in the following subsection.

6.2 Charge conserving current deposition scheme

The simple linear weighting scheme used to deposit currents on the grid does not conserve the continuity equation. Umeda et. al [2] proposed a novel algorithm to do the same. The implementation of the algorithm is discussed in this section.

Umeda et. al's approach assumes that the entire motion of the particle from (x_1, y_1) to (x_2, y_2) is equivalent to one particle moving from (x_1, y_1) to (x_r, y_r) and another particle moving from (x_r, y_r) to (x_2, y_2) . Where x_r and y_r are known as the relay points.

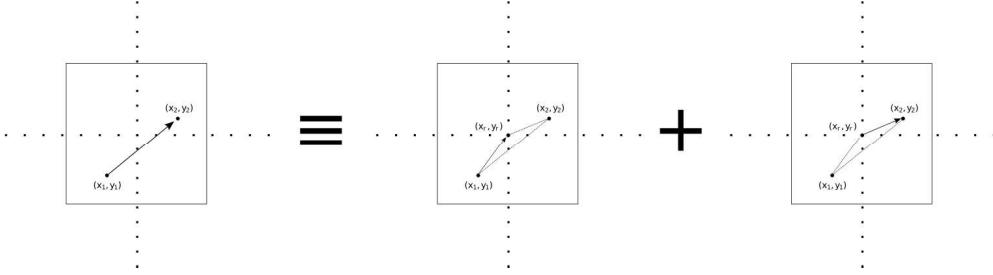


Figure 6.7: Zigzag current deposition scheme

Figure 6.7 illustrates this approach. The notations used in subsequent equations are as described in the original paper[2]).

$$x(t) = x_1, \quad y(t) = y_1 \quad (6.19)$$

$$x(t + \Delta t) = x_2 = x_1 + v_x(t + \Delta t/2)\Delta t, \quad y(t + \Delta t) = y_2 = y_1 + v_y(t + \Delta t/2)\Delta t \quad (6.20)$$

i and j denote the floor indices of the grid cell containing the particle in x and y directions respectively, where i and j are computed as follows:

$$i_1 = [x_1/\Delta x], \quad i_2 = [x_2/\Delta x] \quad (6.21)$$

$$j_1 = [y_1/\Delta y], \quad j_2 = [y_2/\Delta y] \quad (6.22)$$

x_r and y_r are computed in the following manner[2] :

$$x_r = \min \left[\min(i_1\Delta x, i_2\Delta x) + \Delta x, \max \left\{ \max(i_1\Delta x, i_2\Delta x), \frac{x_1 + x_2}{2} \right\} \right] \quad (6.23)$$

$$y_r = \min \left[\min(j_1\Delta y, j_2\Delta y) + \Delta y, \max \left\{ \max(j_1\Delta y, j_2\Delta y), \frac{y_1 + y_2}{2} \right\} \right] \quad (6.24)$$

The currents are computed in terms of fluxes as follows:

$$J_x(i_1 + 1/2, j_1) = \frac{1}{\Delta x \Delta y} F_x(1 - W_y), \quad J_x(i_1 + 1/2, j_1 + 1) = \frac{1}{\Delta x \Delta y} F_x(W_y) \quad (6.25)$$

$$J_y(i_1, j_1 + 1/2) = \frac{1}{\Delta x \Delta y} F_y(1 - W_x), \quad J_y(i_1 + 1, j_1 + 1/2) = \frac{1}{\Delta x \Delta y} F_y(W_x) \quad (6.26)$$

where the fluxes are:

$$F_{x_1} = q \frac{x_r - x_1}{\Delta t}, \quad F_{x_2} = q \frac{x_2 - x_r}{\Delta t} \quad (6.27)$$

$$F_{y_1} = q \frac{y_r - y_1}{\Delta t}, \quad F_{y_2} = q \frac{y_2 - y_r}{\Delta t} \quad (6.28)$$

and the weights are computed as follows:

$$W_{x_1} = \frac{x_1 + x_r}{2\Delta x} - i_1, \quad W_{x_2} = \frac{x_r + x_2}{2\Delta x} - i_2 \quad (6.29)$$

$$W_{y_1} = \frac{y_1 + y_r}{2\Delta y} - j_1, \quad W_{y_2} = \frac{y_r + y_2}{2\Delta y} - j_2 \quad (6.30)$$

Then the currents at the grid nodes are computed as:

$$J_x(i_1 + 1/2, j_1) = \frac{1}{\Delta x \Delta y} F_{x_1} (1 - W_{y_1}), \quad J_x(i_1 + 1/2, j_1 + 1) = \frac{1}{\Delta x \Delta y} F_{x_1} (W_{y_1}) \quad (6.31)$$

$$J_x(i_2 + 1/2, j_2) = \frac{1}{\Delta x \Delta y} F_{x_2} (1 - W_{y_2}), \quad J_x(i_2 + 1/2, j_2 + 1) = \frac{1}{\Delta x \Delta y} F_{x_2} (W_{y_2}) \quad (6.32)$$

$$J_y(i_1, j_1 + 1/2) = \frac{1}{\Delta x \Delta y} F_{y_1} (1 - W_{x_1}), \quad J_y(i_1 + 1, j_1 + 1/2) = \frac{1}{\Delta x \Delta y} F_{y_1} (W_{x_1}) \quad (6.33)$$

$$J_y(i_2, j_2 + 1/2) = \frac{1}{\Delta x \Delta y} F_{y_2} (1 - W_{x_2}), \quad J_y(i_2 + 1, j_2 + 1/2) = \frac{1}{\Delta x \Delta y} F_{y_2} (W_{x_2}) \quad (6.34)$$

Chapter 7

The field solver

7.1 The finite difference time domain algorithm

In collaboration with Shyam Sundar Sankaran

Computational Electrodynamics involves discretization of the fields on a spatial and temporal grid to solve Maxwell's equations. All the equations in the following sub-sections are presented in natural units.

The free space Maxwell's equations are shown below:

$$\nabla \cdot E = \rho/\epsilon_0 \quad (7.1)$$

$$\nabla \cdot B = 0 \quad (7.2)$$

$$\nabla \times E = -\frac{\partial B}{\partial t} \quad (7.3)$$

$$\nabla \times B = \frac{\partial E}{\partial t} + J \quad (7.4)$$

The discretized formulations of Maxwell's equations are used in numerical simulations. The integral forms of Maxwell's equations are:

$$\oint E \cdot dl = -\frac{\partial}{\partial t} \oint B \cdot ds \quad (7.5)$$

$$\oint B \cdot dl = \frac{\partial}{\partial t} \oint E \cdot ds \quad (7.6)$$

Systems of interest are initialized such that the first two Maxwell's equations [7.1 and 7.2] are satisfied. The remaining 2 Maxwell's equations [7.3 and 7.4] evolve the system ensuring that, both the conservation of charge and divergence of B remain conserved throughout the system at all times.

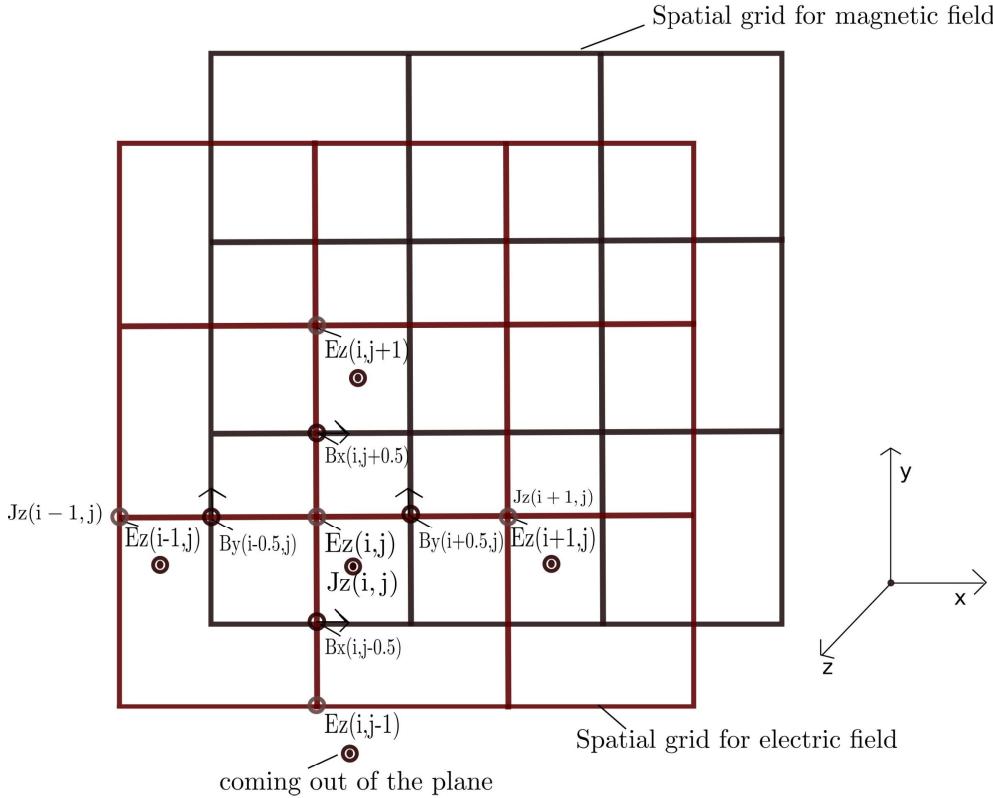


Figure 7.1: Yee Grid for discretizing E and B fields

From Figure 29, We see that the grid for the magnetic field is offset from the grid used for Electric grid by half the grid spacing. Components of electric and magnetic fields get decoupled in a 2 dimensional domain into 2 modes namely TM and TE modes. TM mode corresponds to the set $\{E_z, B_x \text{ and } B_y\}$ and TE mode corresponds to the set $\{E_x, E_y \text{ and } B_z\}$. The mathematical derivation behind this decoupling is shown below: Using the definition of cross product, we can express the 3rd and 4th Maxwell's equations [7.3 and 7.4] in the following manner :

$$\frac{\partial B_x}{\partial t} = - \left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \right) \quad (7.7)$$

$$\frac{\partial B_y}{\partial t} = - \left(- \left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right) \right) \quad (7.8)$$

$$\frac{\partial B_z}{\partial t} = - \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right) \quad (7.9)$$

$$\frac{\partial E_x}{\partial t} + J_x = \left(\frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z} \right) \quad (7.10)$$

$$\frac{\partial E_y}{\partial t} + J_y = - \left(\frac{\partial B_z}{\partial x} - \frac{\partial B_x}{\partial z} \right) \quad (7.11)$$

$$\frac{\partial E_z}{\partial t} + J_z = \left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} \right) \quad (7.12)$$

To solve the Maxwell's equations on a 2D domain, We will now consider variation only along x and y axes thereby eliminating all terms containing $\frac{\partial}{\partial z}$. This simplifies the above set of equations

$$\frac{\partial B_x}{\partial t} = - \left(\frac{\partial E_z}{\partial y} \right) \quad (7.13)$$

$$\frac{\partial B_y}{\partial t} = - \left(- \left(\frac{\partial E_z}{\partial x} \right) \right) \quad (7.14)$$

$$\frac{\partial B_z}{\partial t} = - \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right) \quad (7.15)$$

$$\frac{\partial E_x}{\partial t} + J_x = \left(\frac{\partial B_z}{\partial y} \right) \quad (7.16)$$

$$\frac{\partial E_y}{\partial t} + J_y = - \left(\frac{\partial B_z}{\partial x} \right) \quad (7.17)$$

$$\frac{\partial E_z}{\partial t} + J_z = \left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} \right) \quad (7.18)$$

These equations can now be grouped in the following manner:

- Group 1:

$$\frac{\partial B_z}{\partial t} = - \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right) \quad (7.19)$$

$$\frac{\partial E_x}{\partial t} = \left(\frac{\partial B_z}{\partial y} \right) - J_x \quad (7.20)$$

$$\frac{\partial E_y}{\partial t} = - \left(\frac{\partial B_z}{\partial x} \right) - J_y \quad (7.21)$$

- Group 2:

$$\frac{\partial E_z}{\partial t} = \left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} \right) - J_z \quad (7.22)$$

$$\frac{\partial B_x}{\partial t} = - \left(\frac{\partial E_z}{\partial y} \right) \quad (7.23)$$

$$\frac{\partial B_y}{\partial t} = - \left(- \left(\frac{\partial E_z}{\partial x} \right) \right) \quad (7.24)$$

Thus from the above sets of grouped equations, one can show that E_z , B_x and B_y (Transverse Magnetic(TM)) and E_x , E_y and B_z (Transverse electric (TE)) form two uncoupled sets of coupled fields. The discretized set of equations to be used in numerical simulations for the TM mode are illustrated below. The following set of equations can be derived using figure 7.1 in the following manner:

Integrating the both sides of 4th Maxwell's equation [7.4]:

$$\oint \nabla \times B \cdot ds = \oint -\frac{\partial E}{\partial t} \cdot ds + \oint J \cdot ds \quad (7.25)$$

Using Stoke's theorem to convert the surface integral to a line integral

$$\oint B \cdot dl = \frac{\partial}{\partial t} \oint E \cdot ds + \oint J \cdot ds \quad (7.26)$$

Using a second order quadrature to evaluate both the line and surface integral:

$$By_{i+1/2,j} \Delta y - Bx_{i,j+1/2} \Delta x - By_{i-1/2,j} \Delta y + Bx_{i,j-1/2} \Delta x = \frac{\partial}{\partial t} (E_z \cdot \Delta x \Delta y) + Jz \Delta x \Delta y \quad (7.27)$$

Dividing both sides of the equation by $\Delta x \Delta y$

$$\left(\frac{By_{i+1/2,j} - By_{i-1/2,j}}{\Delta x} \right) - \left(\frac{Bx_{i,j+1/2} - Bx_{i,j-1/2}}{\Delta y} \right) = \frac{\partial}{\partial t} (E_z) + Jz \quad (7.28)$$

The same procedure is extended to arrive at similar expressions for B_x and B_y . The equations for the TM mode are summarized below:

$$\frac{(Ez_{i,j}^{t+1} - Ez_{i,j}^t)}{\Delta t} = \left(\frac{By_{i+1/2,j}^{t+1/2} - By_{i+1/2,j}^{t-1/2}}{\Delta x} \right) - \left(\frac{Bx_{i,j+1/2}^{t+1/2} - Bx_{i,j-1/2}^{t+1/2}}{\Delta y} \right) + Jz_{i,j}^{t+1/2} \quad (7.29)$$

$$\frac{(Bx_{i,j+1/2}^{t+1/2} - Bx_{i,j+1/2}^{t-1/2})}{\Delta t} = - \left(\frac{Ez_{i,j+1}^{t+1} - Ez_{i,j}^{t+1}}{\Delta y} \right) \quad (7.30)$$

$$\frac{(By_{i+1/2,j}^{t+1/2} - By_{i+1/2,j}^{t-1/2})}{\Delta t} = - \left(\frac{Ez_{i,j}^{t+1} - Ez_{i+1,j}^{t+1}}{\Delta x} \right) \quad (7.31)$$

Similarly, the update equations for the TE mode are:

$$\frac{(Ex_{i+1/2,j}^{t+1} - Ex_{i+1/2,j}^{t-1})}{\Delta t} = - \left(\frac{Bz_{i+1/2,j+1/2}^{t+1/2} - Bz_{i+1/2,j-1/2}^{t+1/2}}{\Delta x} \right) + Jx_{i+1/2,j}^{t+1/2} \quad (7.32)$$

$$\frac{(Ey_{i,j+1/2}^{t+1} - Ey_{i,j+1/2}^{t-1})}{\Delta t} = - \left(\frac{Bz_{i+1/2,j+1/2}^{t+1/2} - Bz_{i-1/2,j+1/2}^{t+1/2}}{\Delta x} \right) + Jy_{i,j+1/2}^{t+1/2} \quad (7.33)$$

$$\frac{(Bz_{i+1/2,j+1/2}^{t+1/2} - Bz_{i+1/2,j+1/2}^{t-1/2})}{\Delta t} = \frac{Ex_{i+1/2,j+1} - Ex_{i+1/2,j-1}}{\Delta y} - \frac{Ey_{i+1,j+1/2} - Ex_{i-1,j+1/2}}{\Delta x} \quad (7.34)$$

The grids chosen for all the fields, charge density and current density are listed in the table shown below:

Field	Alignment in space and time				Notation
	x	y	z	t	
E_x	$((i + 1/2) \Delta x)$	$((j) \Delta x)$	$((k) \Delta x)$	$((n) \Delta t)$	$E_{x_{i+1/2,j..k}}^n$
E_y	$((i) \Delta x)$	$((j + 1/2) \Delta x)$	$((k) \Delta x)$	$((n) \Delta t)$	$E_{y_{i..j+1/2,k}}^n$
E_z	$((i) \Delta x)$	$((j) \Delta x)$	$((k + 1/2) \Delta x)$	$((n) \Delta t)$	$E_{z_{i..j..k+1/2}}^n$
B_x	$((i) \Delta x)$	$((j + 1/2) \Delta x)$	$((k + 1/2) \Delta x)$	$((n + 1/2) \Delta t)$	$B_{x_{i..j+1/2..k+1/2}}^{n+1/2}$
B_y	$((i + 1/2) \Delta x)$	$((j) \Delta x)$	$((k + 1/2) \Delta x)$	$((n + 1/2) \Delta t)$	$B_{y_{i+1/2..j..k+1/2}}^{n+1/2}$
B_z	$((i + 1/2) \Delta x)$	$((j + 1/2) \Delta x)$	$((k) \Delta x)$	$((n + 1/2) \Delta t)$	$B_{z_{i+1/2..j..k+1/2}}^{n+1/2}$
ρ	$((i) \Delta x)$	$((j) \Delta x)$	$((k) \Delta x)$	$((n) \Delta t)$	$\rho_{i..j..k}^n$
j_x	$((i + 1/2) \Delta x)$	$((j) \Delta x)$	$((k) \Delta x)$	$((n + 1/2) \Delta t)$	$j_{x_{i+1/2..j..k}}^{n+1/2}$
j_y	$((i) \Delta x)$	$((j + 1/2) \Delta x)$	$((k) \Delta x)$	$((n + 1/2) \Delta t)$	$j_{x_{i..j+1/2..k}}^{n+1/2}$
j_z	$((i) \Delta x)$	$((j) \Delta x)$	$((k + 1/2) \Delta x)$	$((n + 1/2) \Delta t)$	$j_{x_{i..j..k+1/2}}^{n+1/2}$

Table 7.1: Representation on the Yee grid

7.2 Initialization of fields

The fdtd algorithm evolves the fields given the initial conditions. The electric fields in the domain due arbitrary charge and density configurations are determined by solving the Poisson equation.

The Direct current deposition scheme does not conserve Gauss's Law($\nabla \cdot E = \rho$) and requires an implementation of the Boris correction scheme in order to do so. A Poisson solver is essential to implement both the Boris correction scheme and to determine the electric fields for an arbitrary charge configuration. As the name suggests, The Poisson solver solves the Poisson's equation represented by following equation 7.35

$$\nabla^2 V = \rho \quad (7.35)$$

Once V is determined, one can compute E as $-\nabla V$. How do we solve the Poisson's equation for an arbitrary charge configuration on the spatial grid?

A few of the popular techniques[12] used to implement the Poisson solver use Dense LU decomposition, CG (Conjugate gradient),FFT (Fast fourier transform), Multigrid etc.

A FFT based Poisson solver is both computationally efficient and an accurate solver that is suitable for domains with periodic boundary conditions and is chosen as a method of choice to initialize the fields

7.2.1 FFT based Poisson solver

Consider the one dimensional Poisson equation as follows:

$$\frac{\partial^2 V}{\partial x^2} = -\frac{\rho}{\varepsilon} \quad (7.36)$$

It is to be noted that we are dealing with a finite periodic domain of length L . Fourier transform in k space of the function $f(x)$ is:

$$\hat{f}(k) = \frac{1}{\sqrt{2\pi}} \int_0^L f(x) e^{-ikx} dx \quad (7.37)$$

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_0^L \hat{f}(k) e^{ikx} dk \quad (7.38)$$

$$\implies V(x) = \frac{1}{\sqrt{2\pi}} \int_0^L \hat{V}(k) e^{ikx} dk \quad (7.39)$$

$$\implies \rho(x) = \frac{1}{\sqrt{2\pi}} \int_0^L \hat{\rho}(k) e^{ikx} dk \quad (7.40)$$

Using this representation, equation 7.36 is:

$$\frac{\partial^2}{\partial x^2} \frac{1}{\sqrt{2\pi}} \int_0^L \hat{V}(k) e^{ikx} dk = -\frac{1}{\varepsilon} \frac{1}{\sqrt{2\pi}} \int_0^L \hat{\rho}(k) e^{ikx} dk \quad (7.41)$$

Therefore, we have:

$$-k^2 \hat{V}(k) = \hat{\rho}(k) \quad (7.42)$$

$$\hat{V}(k) = \frac{\hat{\rho}(k)}{k^2} \quad (7.43)$$

$$\implies V(x) = -\frac{1}{\sqrt{2\pi}} \int \frac{\hat{\rho}(k)}{k^2} e^{ikx} dk \quad (7.44)$$

The same procedure can be reproduced numerically for the discretized $V(x)$ using a discrete fourier transform. Consider a one dimensional domain with $0 < x < L$. Let the discretization of x be:

$$x(n) = \frac{nL}{N}, \quad n = 0, 1, 2, \dots, N-1 \quad (7.45)$$

$$\implies \hat{V}(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{(\frac{i 2\pi n k}{L})} V(x(n)) \quad (7.46)$$

$$\implies V(x) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{(-\frac{i 2\pi n k}{L})} \hat{V}(k) \quad (7.47)$$

This is the complex fourier transform of $V(x)$. We notice that $V(x)$ is periodic in nature with period L .

Chapter 8

Summary

The following schematics shows the entire process of the Particle in cell method. The schematics found in the current chapter are inspired by reference [13]

The fields are first interpolated at the point on the spatial and temporal grid since the electric and magnetic are computed on staggered grids.

Field gathering

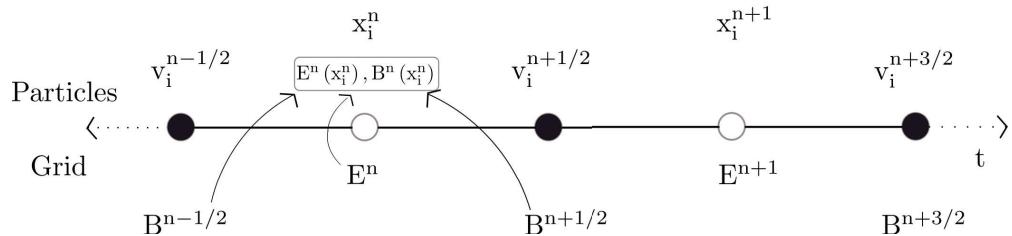


Figure 8.1: PIC cycle: Step 1

Then the interpolated fields are used to push the particle via the Boris algorithm.

Particle pusher

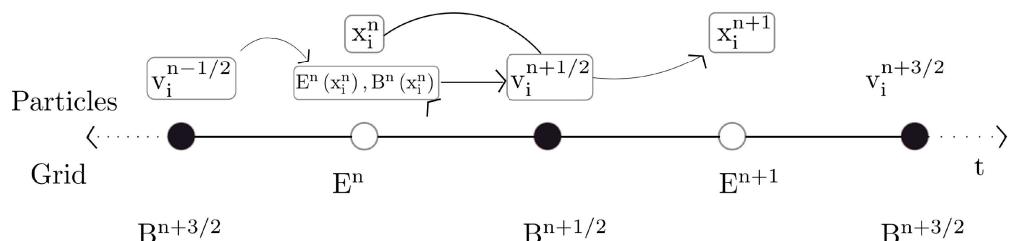


Figure 8.2: PIC cycle: Step 2

The particle velocities and positions are then used to deposit currents on the grids

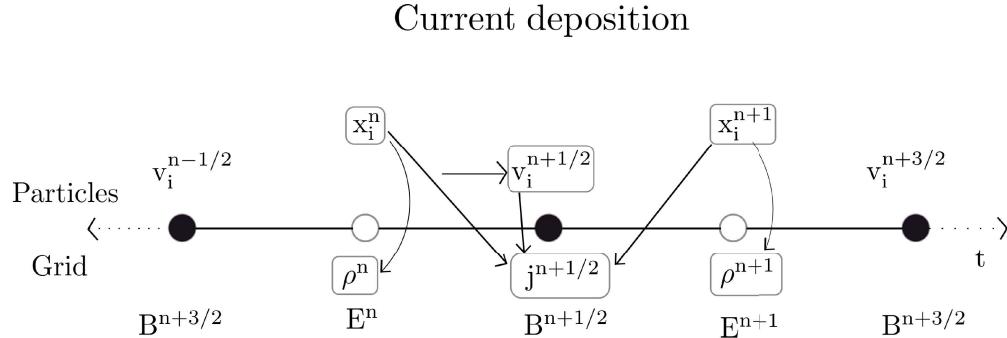


Figure 8.3: PIC cycle: Step 3

The fields are solved using the fdtd scheme using the currents generated.

Field solver

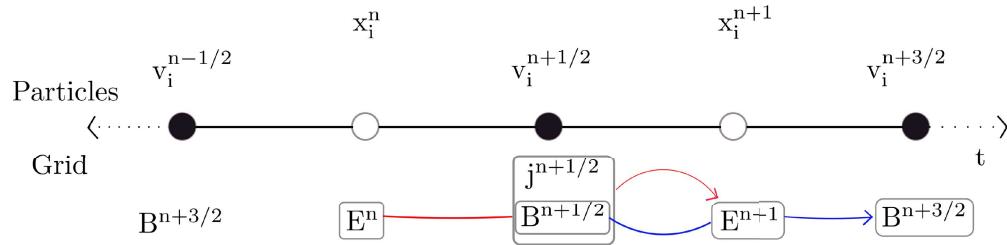


Figure 8.4: PIC cycle: Step 4

The cycle is repeated all over again.

The entire PIC algorithm can be summarized in the following manner depicted by the schematic shown below [13]:

Summary

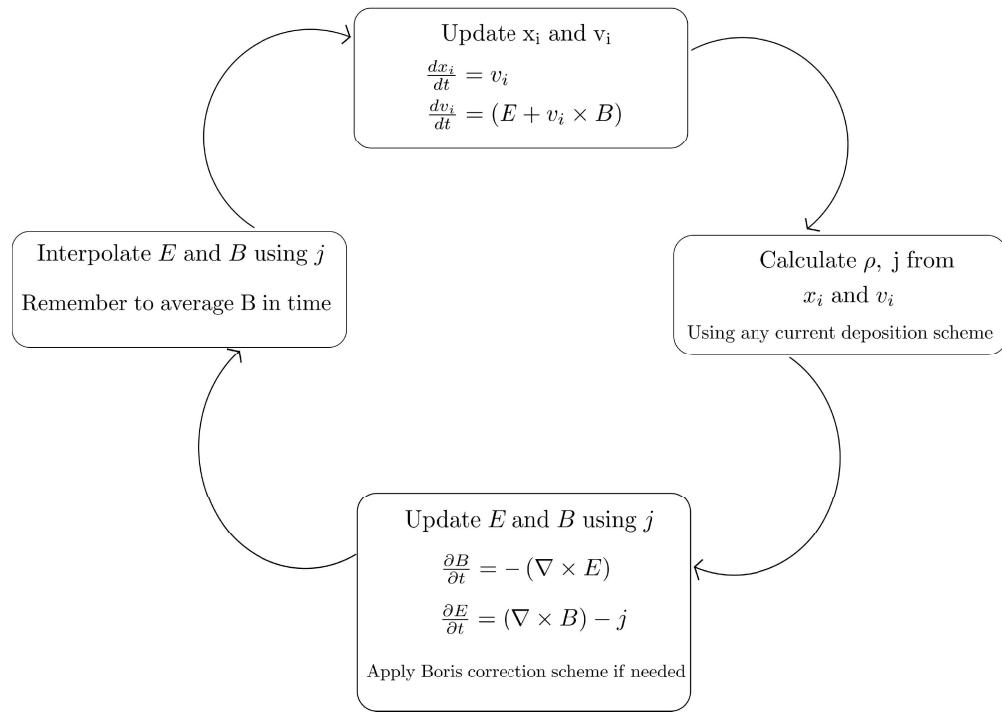


Figure 8.5: The PIC cycle

Chapter 9

Linear theory of Plasmas

In this section, we linearize the Vlasov-Maxwell system of equations [1.1, 1.2, 1.3, 1.4, 1.5, 1.6 and 1.7]. Consider a one dimensional simplified plasma system with variation only along x direction. The Maxwell's equations in such a scenario are:

$$\frac{\partial B_x}{\partial t} = 0 \quad (9.1)$$

$$\frac{\partial B_y}{\partial t} = \left(\frac{\partial E_z}{\partial x} \right) \quad (9.2)$$

$$\frac{\partial B_z}{\partial t} = - \left(\frac{\partial E_y}{\partial x} \right) \quad (9.3)$$

$$\frac{\partial E_x}{\partial t} = -J_x \quad (9.4)$$

$$\frac{\partial E_y}{\partial t} = - \left(\frac{\partial B_z}{\partial x} \right) - J_y \quad (9.5)$$

$$\frac{\partial E_z}{\partial t} = \left(\frac{\partial B_y}{\partial x} \right) - J_z \quad (9.6)$$

Let

$$f_0 \text{ (total)}$$

be the equilibrium Maxwell Boltzmann distribution function describing the charged species at equilibrium. The masses of ions are of several orders higher than that of electrons. Therefore, we assume that they are stationary and that their distribution is uniform and time independent. This fact allows us to safely assume that ions are stationary and their distribution is time independent. Let $\rho_0(t = 0) \text{ (total)} = 0, \forall x$. f_0 being the equilibrium distribution function results

in the following set of equations.

$$\frac{\partial f_0(x, v)}{\partial t} = 0 \quad (9.7)$$

$$\frac{\partial f_0(x, v)}{\partial x} = 0 \quad (9.8)$$

$$F_x = \frac{\partial v_{x,0}}{\partial t} = 0 \quad (9.9)$$

The initial conditions are:

$$E_x = 0, E_y = 0, E_z = 0 \quad (9.10)$$

$$B_y = 0, B_z = 0 \quad (9.11)$$

$$J_y = 0, J_z = 0 \quad (9.12)$$

The initial conditions chosen satisfy the constraint equations:

$$\nabla \cdot E = \rho = 0 \quad (9.13)$$

$$\nabla \cdot B = 0 \quad (9.14)$$

One can evolve the fields and currents for the chosen system using the Vlasov-Maxwell system of equations. Let us introduce a perturbation in the distribution function along x such that:

$$f(x, v, t = 0) = f_0(x, v) + \delta f(x, v, t = 0) \quad (9.15)$$

where $\delta f(x, v, t)$ is expressed in its Fourier series expansion as $\hat{\delta f}(v, t) e^{ikx}$. This implies:

$$f(x, v, t) = f_0(x, v) + \hat{\delta f}(v, t) e^{ikx} \quad (9.16)$$

$E_x(x, t)$ can be expressed in the similar manner:

$$E_x(x, t) = E_{x0} + \hat{\delta E_x}(t) \cdot e^{ikx} \quad (9.17)$$

From the initial conditions chosen, $E_{x0} = 0$ simplifies the right hand side of the equation 9.17 as follows:

$$E_x(x, t) = \hat{\delta E_x}(t) \cdot e^{ikx} \quad (9.18)$$

Now $F_x = qE_x$ can be expressed as

$$F_x = q\hat{\delta E_x}(t) e^{ikx} \quad (9.19)$$

We can now use this formulation and equation 1.1 to arrive at the following equation.

$$\frac{\partial f(x, v, t)}{\partial t} + \frac{\partial r}{\partial t} \cdot \frac{\partial f(x, v, t)}{\partial r} + q\delta E_x(t) e^{ikx} \frac{\partial f(x, v, t)}{\partial v} = 0 \quad (9.20)$$

The Vlasov equation for the plasma system with the density perturbation is:

$$\frac{\partial (f_0 + \delta \hat{f}(v, t) e^{ikx})}{\partial t} + v_x \frac{\partial (f_0 + \delta \hat{f}(v, t) e^{ikx})}{\partial x} + q\delta E_x(t) e^{ikx} \left(\frac{\partial (f_0 + \delta \hat{f}(v, t) e^{ikx})}{\partial v} \right) = 0 \quad (9.21)$$

f_0 being the equilibrium Maxwell-Boltzmann distribution function, simplifies the equation 9.21 as follows

$$\frac{\partial (\delta \hat{f}(v, t) e^{ikx})}{\partial t} + v_x \frac{\partial (\delta \hat{f}(v, t) e^{ikx})}{\partial x} + q\delta \hat{E}_x(t) e^{ikx} \left(\frac{\partial (\delta \hat{f}(v, t) e^{ikx})}{\partial v} \right) = 0 \quad (9.22)$$

Let the perturbations be of order ε where $\varepsilon \ll 1$.

$$\Rightarrow \delta \hat{f}(v, t) \sim O(\varepsilon) \quad (9.23)$$

$$\Rightarrow \delta E_x \sim O(\varepsilon) \quad (9.24)$$

From the above set of equations, one can make the following set of observations:

$$\frac{\partial (\delta \hat{f}(v, t) e^{ikx})}{\partial t} \sim O(\varepsilon) \quad (9.25)$$

$$v_x \frac{\partial (\delta \hat{f}(v, t) e^{ikx})}{\partial x} \sim O(\varepsilon) \quad (9.26)$$

$$q\delta \hat{E}_x(t) e^{ikx} \left(\frac{\partial (f_0)}{\partial v} \right) \sim O(\varepsilon) \quad (9.27)$$

$$q\delta \hat{E}_x(t) e^{ikx} \left(\frac{\partial (\delta \hat{f}(v, t) e^{ikx})}{\partial v} \right) \sim O(\varepsilon^2) \quad (9.28)$$

We neglect the 2nd order terms for obtaining the 1st order solution. Collecting all the first order terms results in the following equation:

$$\frac{\partial (\delta \hat{f}(v, t) e^{ikx})}{\partial t} + v_x \frac{\partial (\delta \hat{f}(v, t) e^{ikx})}{\partial x} + q\delta \hat{E}_x(t) e^{ikx} \left(\frac{\partial (f_0)}{\partial v_x} \right) = 0 \quad (9.29)$$

$$\Rightarrow \frac{e^{ikx} \partial (\delta \hat{f}(v, t))}{\partial t} + v_x ik e^{ikx} \delta \hat{f}(v, t) + q\delta \hat{E}_x(t) e^{ikx} \left(\frac{\partial (f_0)}{\partial v_x} \right) = 0 \quad (9.30)$$

Dividing the whole equation by e^{ikx} , we arrive at the following equation:

$$\frac{\partial(\delta\hat{f}(v,t))}{\partial t} + ik v_x \cdot \delta\hat{f}(v,t) + q\delta\hat{E}_x(t) \left(\frac{\partial(f_0)}{\partial v} \right) = 0 \quad (9.31)$$

Equation 9.31 can be split into a coupled set of ordinary differential equations in the manner shown below: The expression for $\delta\hat{E}_x(t)$ used in the above equation can be derived in the following manner. The Gauss's Law is as follows

$$\nabla \cdot E = \rho \quad (9.32)$$

where ρ is the charge density. This equation can be further simplified to arrive at the following equations:

$$\frac{\partial E_x}{\partial x} = \int q f(x, v, t) dv \quad (9.33)$$

$$\implies \delta E_x(t) = \frac{q}{ik} \int \delta\hat{f}(v, t) dv \quad (9.34)$$

Let the real and imaginary parts of $\delta\hat{f}(v, t)$ and $\delta\hat{E}_x(t)$ be denoted by \hat{f}^r , \hat{f}^i , \hat{E}_x^r and \hat{E}_x^i respectively. Equations 9.31 and 9.34 results in the following set of equations:

$$\implies \frac{\partial \hat{f}^r}{\partial t} = kv_x \hat{f}^i - q\hat{E}_x^r \frac{\partial f_0}{\partial v_x} \quad (9.35)$$

$$\hat{E}_x^r = \frac{q}{k} \int \hat{f}^i dv_x \quad (9.36)$$

$$\frac{\partial \hat{f}^i}{\partial t} = -kv_x \hat{f}^r - q\hat{E}_x^i \frac{\partial f_0}{\partial v_x} \quad (9.37)$$

$$\hat{E}_x^i = -\frac{q}{k} \int \hat{f}^r dv_x \quad (9.38)$$

9.0.1 Linear Theory Current approach

An alternative procedure to solve the set of equations discussed in the previous section is discussed in this section. Proceeding from equation 9.31

$$\implies \frac{\partial(\delta\hat{f}(v,t))}{\partial t} + ikv_x \delta\hat{f}(v,t) + q\delta\hat{E}_x(t) \left(\frac{\partial(f_0)}{\partial v_x} \right) = 0 \quad (9.39)$$

where $\delta\hat{E}_x(t)$ is governed by the following differential equations:

$$\frac{\partial E_x}{\partial t} = -J_x \quad (9.40)$$

$$\Rightarrow e^{ikx} \frac{\partial(\delta\hat{E}_x(t))}{\partial t} = - \int q v_x \delta\hat{f}(v, t) e^{(ikx)} dv_x \quad (9.41)$$

$$\Rightarrow \frac{\partial(\delta\hat{E}_x(t))}{\partial t} = - \int q v_x \delta f(v, t) dv_x \quad (9.42)$$

Let the real and imaginary of $\delta\hat{f}(v, t)$ and $\delta\hat{E}_x(t)$ be \hat{f}^r , \hat{f}^i , \hat{E}^r and \hat{E}^i respectively. Equation 9.34 can be used to initialize the electric field and equations 9.39 and 9.42 allows us to split the above equation into a set of coupled ordinary differential equations.

$$\frac{\partial \hat{f}^r}{\partial t} = kv_x \hat{f}^i - q\hat{E}_r \frac{\partial f_0}{\partial x} \quad (9.43)$$

$$\frac{\partial \hat{E}^r}{\partial t} = - \int q v_x \hat{f}^r dv_x \quad (9.44)$$

$$\frac{\partial \hat{f}^i}{\partial t} = -kv_x \hat{f}^r - q\hat{E}^i \frac{\partial f_0}{\partial x} \quad (9.45)$$

$$\frac{\partial \hat{E}^i}{\partial t} = - \int q v_x \hat{f}^i dv_x \quad (9.46)$$

We now solve the linearized equations [9.35, 9.36, 9.37 and 9.38 or 9.43, 9.44, 9.45 and 9.46] in order to obtain a reference solution to compare against the solution obtained from the PIC scheme, which solves the full nonlinear Vlasov-Maxwell system. How do we interpret the results generated by the PIC simulation?

9.0.2 Comparison of Linear theory calculations with PIC

The distribution function $f(x, v, t)$ used in both PIC and linear theory calculations must be statistically equivalent to compare the results between analytical calculations and particle in cell calculations. Let the distribution function in analytical calculations be $f_a(x, v, t)$ and $f_{pic}(x, v, t)$ respectively, where $f_{pic}(x, v, t)$ is given as follows:

$$f_{pic}(x, v, t) = A \sum_{p=1}^{N_m} w_p S(x, x_p) S(v, v_{x,p}) \quad (9.47)$$

where A be the normalization factor. The results generated by the PIC code or analytical codes is normalized to draw a comparison between the results. The normalization factor for comparing

the analytical and PIC results is given as follows:

$$f_{pic} = f_a \quad (9.48)$$

$$\Rightarrow A = \frac{\int \int f_a dv dx}{\int \int f_{pic} dv dx} \quad (9.49)$$

$$A = \frac{\int \int f_a dv dx}{\int \int \sum_{p=1}^{N_m} w_p S(x, x_p) S(v, v_{x,p}) dv dx} \quad (9.50)$$

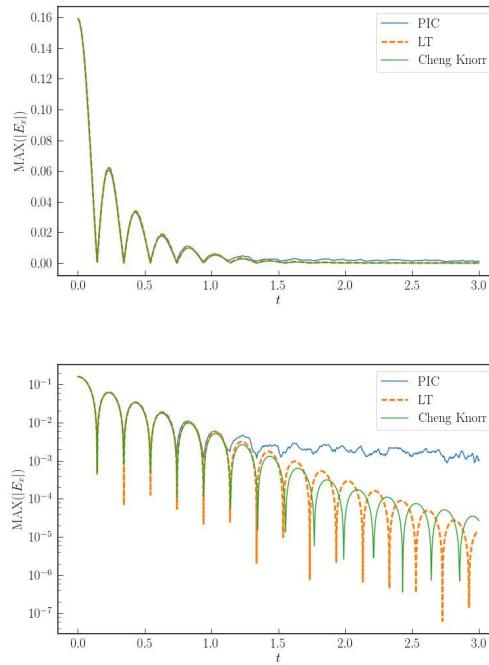
$$A = \frac{\int \int f_a dv dx}{N_m * w_p} \quad (9.51)$$

Chapter 10

Results

Linear Landau is a standard test used to evaluate the collective functioning of all the various components of the particle in a cell code. The Landau damping observed via the PIC code is compared with the Landau damping obtained via the analytical linear theory calculations and the Cheng-Knorr code developed by Shyam Sankaran.

The plots comparing the results generated by the one dimensional particle in cell code with the results generated by the Cheng-Knorr and the linear theory calculations are presented below.



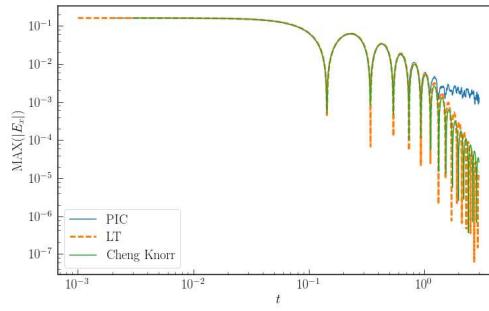


Figure 10.1: 1D PIC vs Cheng-Knorr vs Linear theory for $A = 0.1$

One can infer that the accuracy of the Landau damping observed via the PIC code is limited by the particle noise whereas both the the Cheng Knorr and the linear theory calculations can resolve the damping observed to a much higher accuracy. There is deviation in the landau damping observed between the results obtained via Cheng Knorr code and Linear theory calculations at higher times.

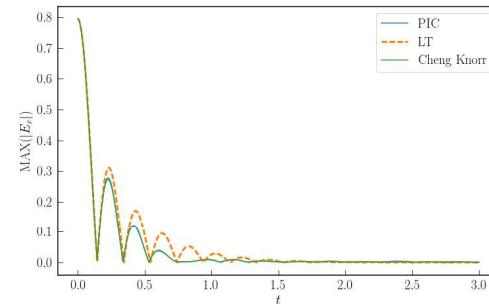
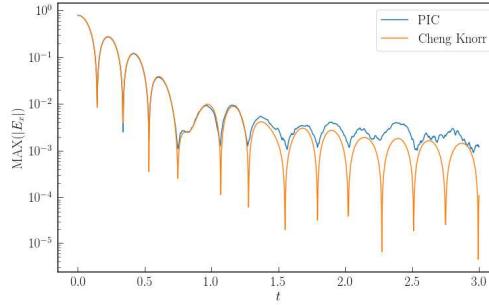


Figure 10.2: 1D PIC vs Cheng Knorr vs Linear theory for $A = 0.5$

One can see the higher damping is observed both via the PIC and the Cheng Knorr code compared to the linear theory calculation possibly due to non-linear effects as the linear theory calculations does not capture the same.

The figures shown below [10.3 and 10.4] compare the one dimensional Landau damping observed via the 2D2V and 2D3V codes against the 1D Cheng Knorr code.

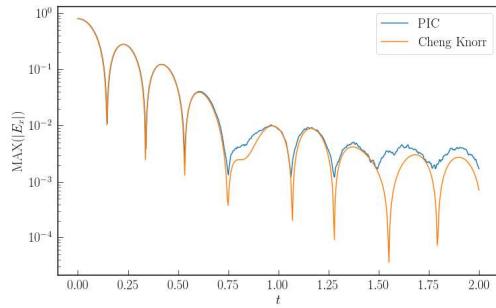


Figure 10.3: 2D2V PIC vs Cheng Knorr for $A = 0.5$

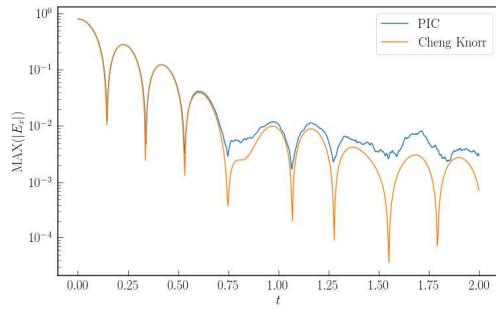


Figure 10.4: 2D3V PIC vs Cheng Knorr for $A = 0.5$

Appendix A

Testing individual components of the code

A.1 Testing the particle pusher

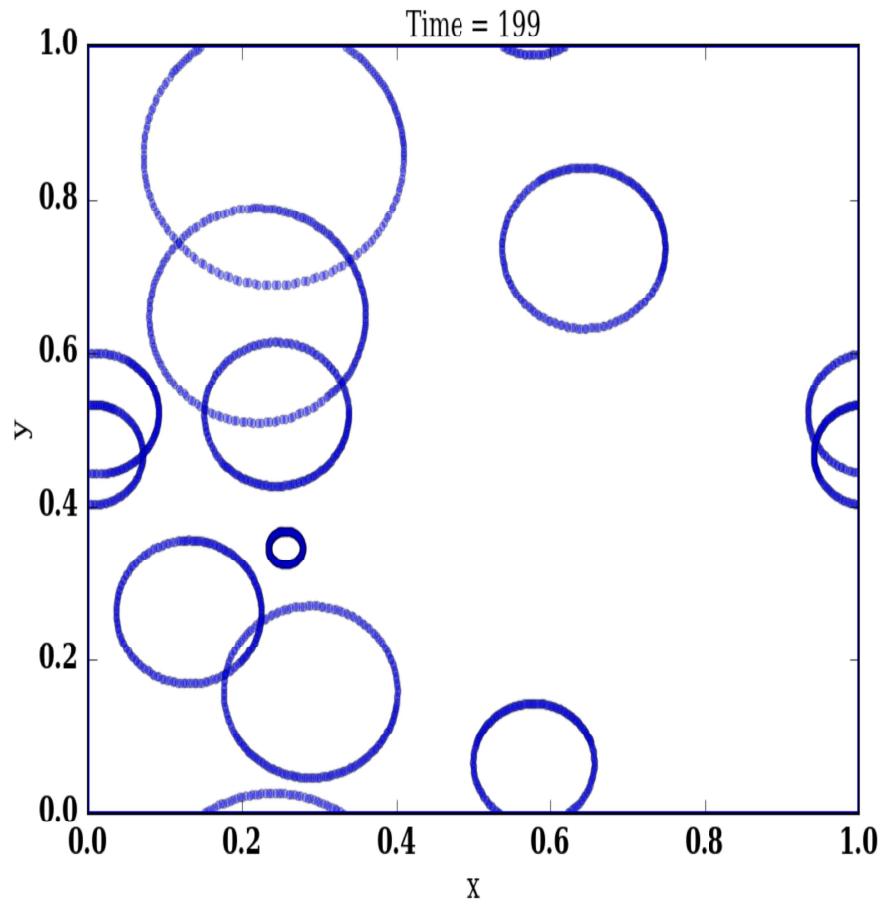


Figure A.1: Particle moving under the influence of a constant perpendicular magnetic field

Particles are found moving in circular paths of constant radius with a constant perpendicular magnetic field acting on them as seen in Figure A.1.

A.2 Testing the fdtd algorithm

Maxwell's equations have wave solutions, which we solve for in a periodic domain. We then perform convergence checks comparing the the solution at the initial time and the solution after a single time period.

Initial conditions of the test are:

$$E_x = 0 \quad (\text{A.1})$$

$$E_y = 1.5e^{-((\frac{y^2}{2\sigma^2}))} \quad (\text{A.2})$$

$$E_z = 2.0e^{-((\frac{y^2}{2\sigma^2}))} \quad (\text{A.3})$$

$$B_x = 0 \quad (\text{A.4})$$

$$B_y = 0.5e^{-((\frac{x^2}{2\sigma^2}))} \quad (\text{A.5})$$

$$B_z = 3e^{-((\frac{y^2}{2\sigma^2}))} \quad (\text{A.6})$$

The above initial conditions lead to a wave solution with propagation speed($c = 1$). Figure A.2 shows the convergence plot obtained:

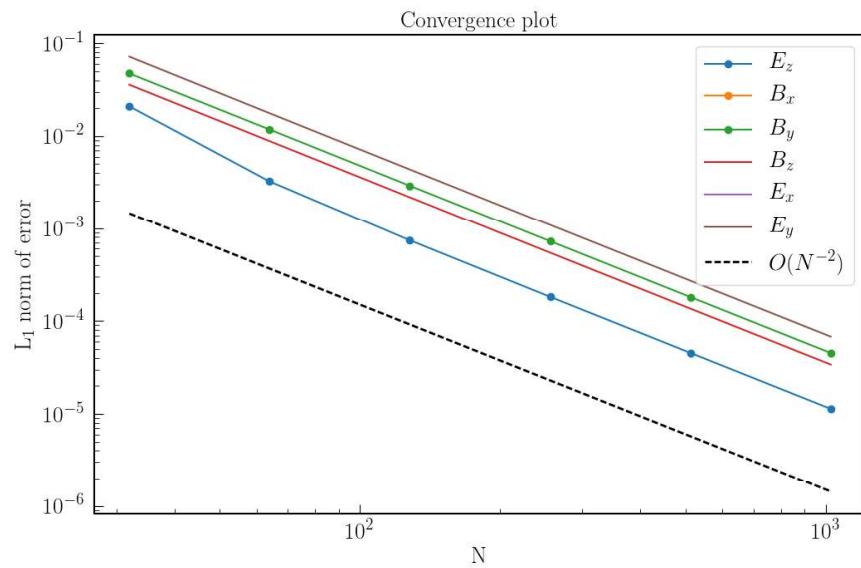


Figure A.2: Numerical vs Analytical(v_y)

A.3 Combined unit test for fdtd and the Boris algorithm

The one dimensional case with variation along z direction has been considered with E_x and B_y acting as the relevant fields. The Maxwell's equations for this simplified case reduce to:

$$\frac{\partial E_y}{\partial t} = - \left(\frac{\partial B_z}{\partial x} \right) \quad (\text{A.7})$$

$$\frac{\partial B_z}{\partial t} = - \frac{\partial E_x}{\partial x} \quad (\text{A.8})$$

$$F = \frac{q}{m} \cdot (E + (\vec{v} \times B)) \quad (\text{A.9})$$

where

$$q = 1$$

$$m = 1$$

$$\vec{x} = [x, y, z]$$

$$\vec{v} = [v_x, v_y, v_z]$$

$$B = [B_x, B_y, B_z]$$

$$E = [E_x, E_y, E_z]$$

Let $E_x = E_z = B_x = B_y = v_z = 0$ and $E_y = B_z = \sin(2 * \beta * (t - x))$.

Substituting for these variable simplifies the Lorentz equation as follows:

$$\frac{\partial^2 \vec{x}}{\partial t^2} = [0, \sin 2\pi(t - x), 0] + [v_y \sin 2\pi(t - x), -v_x \sin 2\pi(t - x), 0] \quad (\text{A.10})$$

$$\frac{\partial^2 \vec{x}}{\partial t^2} = [v_y \sin 2\pi(t - x), \sin 2\pi(t - x)(1 - v_x), 0] \quad (\text{A.11})$$

$$\frac{\partial v_x}{\partial t} = \frac{\partial^2 x}{\partial t^2} = \sin 2\pi(t - x) (v_y) \quad (\text{A.12})$$

$$\frac{\partial v_y}{\partial t} = \frac{\partial^2 y}{\partial t^2} = (1 - v_x) \sin 2\pi(t - x) \quad (\text{A.13})$$

The solutions for the positions and velocities generated by the PIC code and the analytical solutions are compared in the following plots.

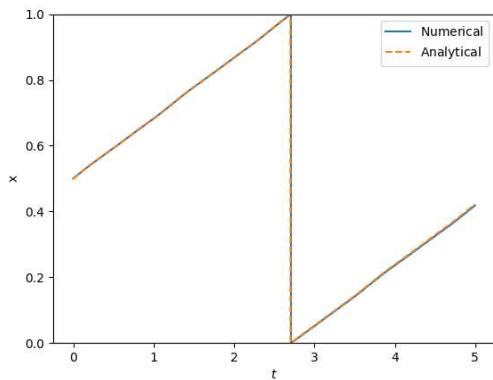


Figure A.3: Numerical vs Analytical(x)

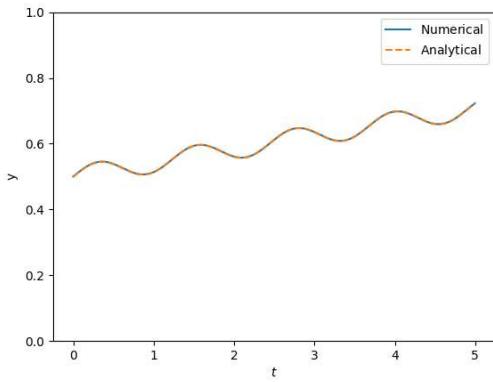


Figure A.4: fdtd and Boris algorithm test:Numerical vs Analytical(y)

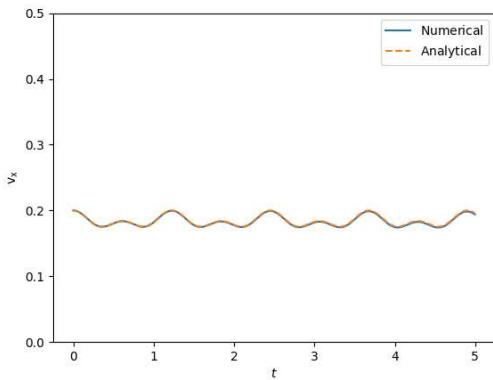


Figure A.5: fdtd and Boris algorithm test:Numerical vs Analytical(v_x)

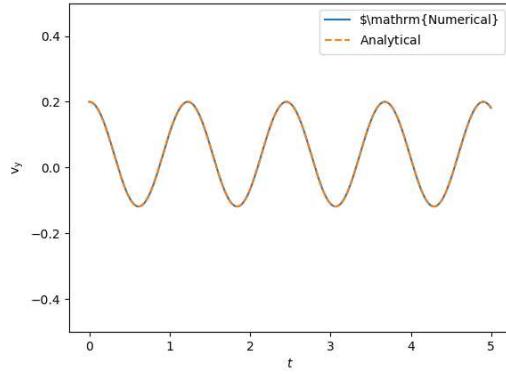


Figure A.6: fdtd and Boris algorithm test:Numerical vs Analytical(v_y)

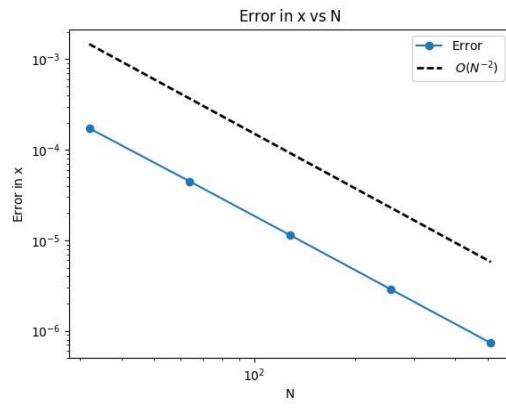


Figure A.7: fdtd and Boris algorithm test:Error in (x)

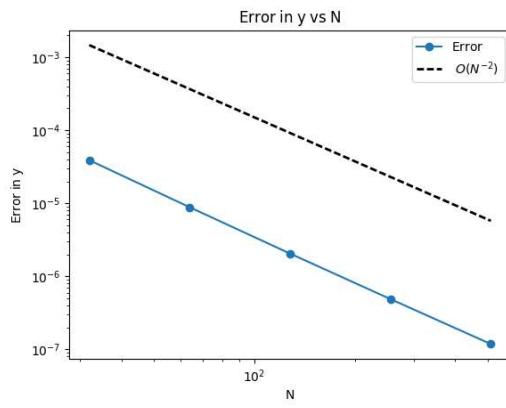


Figure A.8: fdtd and Boris algorithm test:Error in (y)

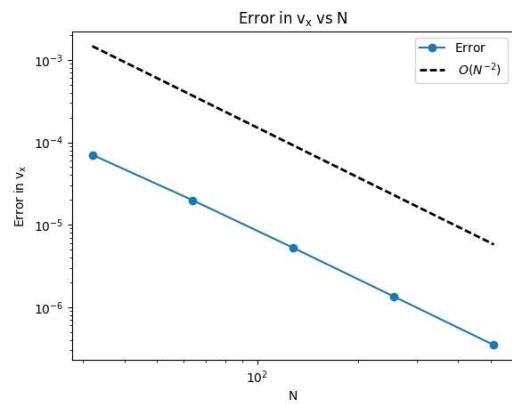


Figure A.9: fdtd and Boris algorithm test:Error in (v_x)

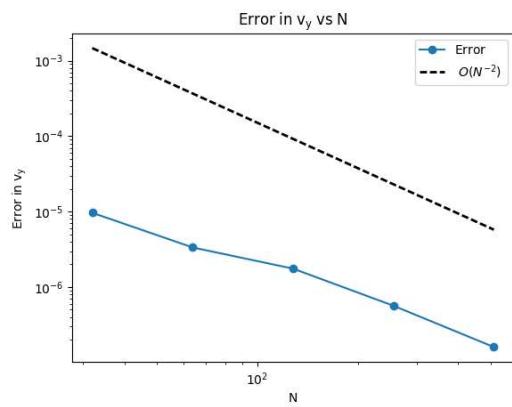


Figure A.10: fdtd and Boris algorithm test:Error in (v_y)

A.4 Testing the FFT based Poisson solver

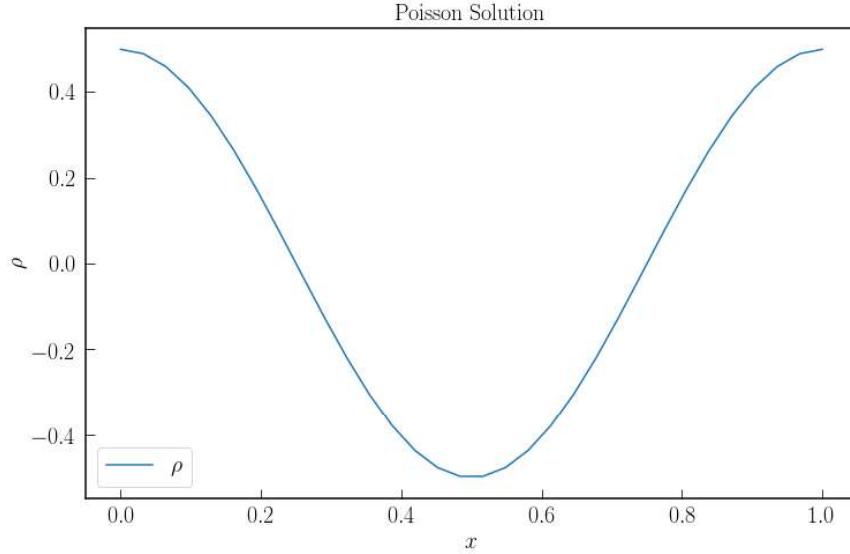


Figure A.11: FFT test: Charge density in the domain

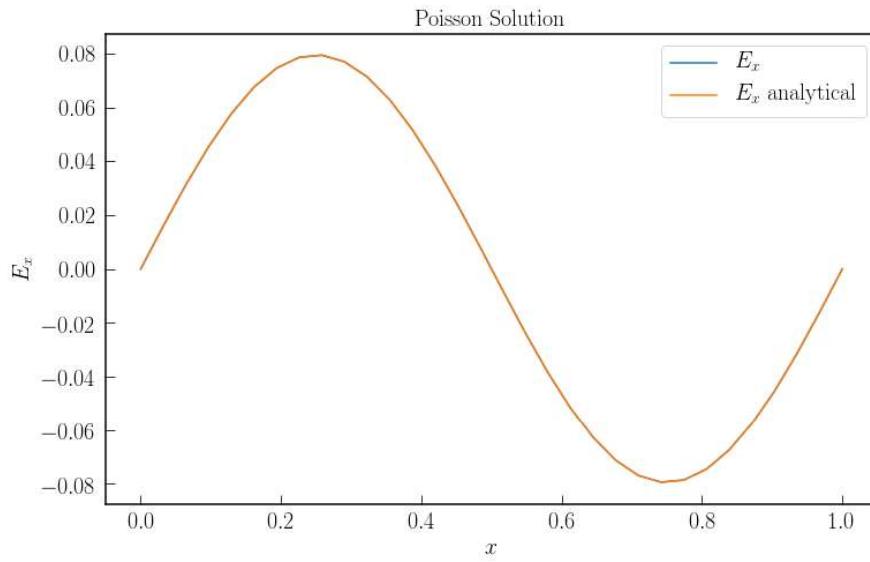


Figure A.12: FFT test: Numerical vs Analytical(E_x)

Figure A.12 shows the comparison between the numerical solution obtained via the FFT solver and the analytical solution for the electric field in the domain with a one dimensional charge

density as follows:

$$\rho(x) = 0.5 \cos\left(\frac{2\pi x}{L}\right) \quad (\text{A.14})$$

A.5 Testing Umeda's charge conserving current deposition

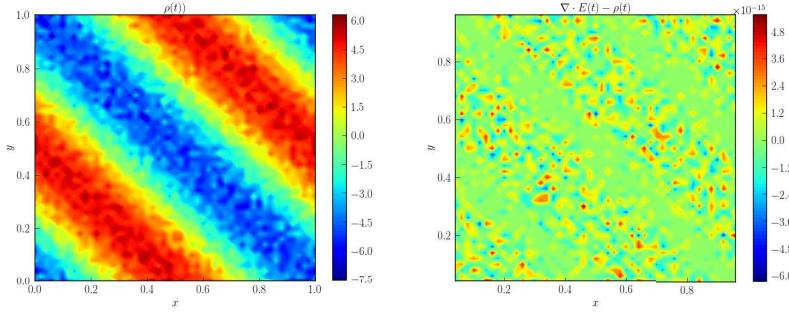


Figure A.13: Initial $(\nabla \cdot E - \rho)(t = 0)$

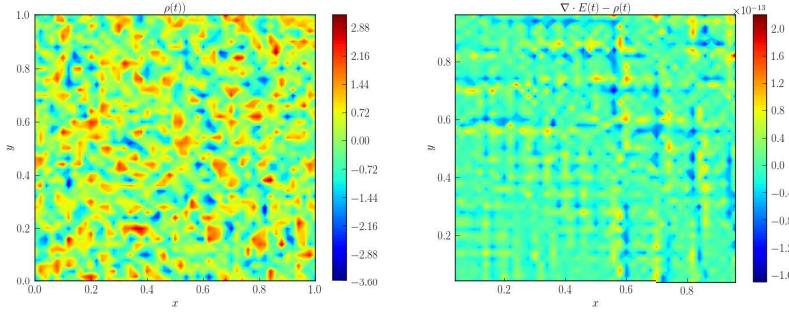


Figure A.14: $(\nabla \cdot E - \rho)(t)$ after 400 time steps

Figures A.13 and A.14 show that the continuity equation is conserved to machine precision via the chosen current deposition[2].

Appendix B

Implementing periodic boundary conditions

Coloured grid nodes of the same colour are the same grid nodes due to periodic boundary conditions

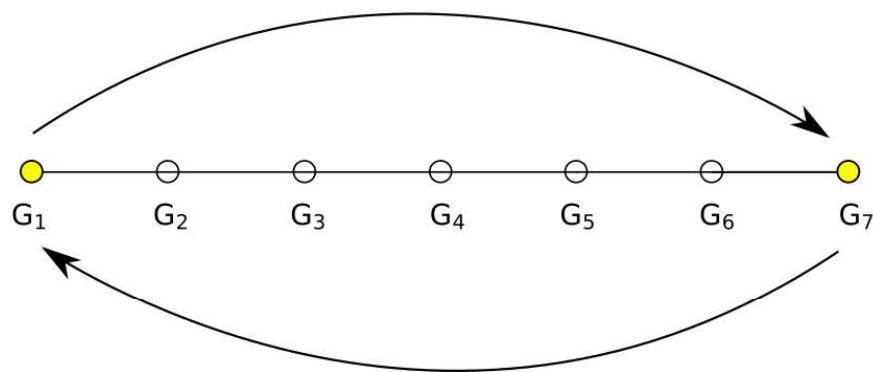


Figure B.1: 1D periodic boundary conditions for charge deposition on a central grid

Periodic boundary conditions for the charge density ρ on a one dimensional grid is illustrated in Figure B.1.

```
rho_electrons[0] = rho_electrons[0] + rho_electrons[-1]
rho_electrons[-1] = rho_electrons[0].copy()
```

Listing B.1: Python example:

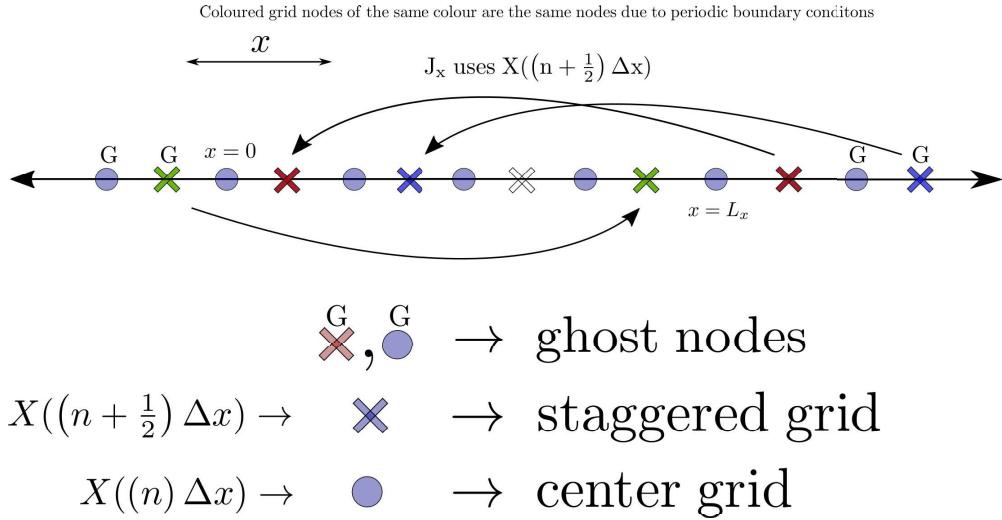


Figure B.2: 1D periodic boundary conditions for current deposition on a staggered grid

Periodic boundary conditions for current deposition(J_x) on a one dimensional staggered grid is illustrated in Figure B.2. The pseudo code for the same is:

```

Jx[ghost_cells]      = Jx[ghost_cells]      + Jx[-1 - ghost_cells]
Jx[-2 - ghost_cells] = Jx[-2 - ghost_cells] + Jx[ghost_cells - 1]
Jx[-1 - ghost_cells] = Jx[ghost_cells].copy()
Jx[ghost_cells + 1] += Jx[-ghost_cells]

```

Listing B.2: Python example: Periodic boundary conditions for J_x

Bibliography

- [1] Yalamanchili, P., Arshad, U., Mohammed, Z., Garigipati, P., Entschev, P., Kloppenborg, B., Malcolm, J. and Melonakos, J. (2015). ArrayFire - A high performance software library for parallel computing with an easy-to-use API. Atlanta: AccelerEyes. Retrieved from <https://github.com/arrayfire/arrayfire>
- [2] Umeda, T., Omura, Y., Tominaga, T., and Matsumoto, H. (2003). A new charge conservation method in electromagnetic particle-in-cell simulations. Computer Physics Communications, 156(1), 73-85.
- [3] http://theory.ipp.ac.cn/~yj/research_notes/particle_simulation.pdf
- [4] <http://www.tp1.ruhr-uni-bochum.de/~grauer/lectures/CompII/PIClecture>
- [5] DOI: 10.13140/RG.2.1.3319.2801. Conference: XII Carolus Magnus Summer School on Plasma and Fusion Energy Physics
- [6] Charles K. Birdsall, A.Bruce Langdon-Plasma physics via computer simulation-McGraw-Hill (1991)
- [7] <https://arxiv.org/abs/1311.4709>
- [8] <http://ssl.mit.edu/files/website/theses/PhD-2001-SzaboJames.pdf>
- [9] <https://arxiv.org/pdf/1702.05128.pdf>
- [10] <https://en.wikipedia.org/wiki/Electronvolt#Temperature>
- [11] J. Boris, in Proceedings of the Fourth Conference on Numerical Simulation of Plasmas (Naval Research Laboratory, Washington DC, 1970)
- [12] <https://people.eecs.berkeley.edu/~demmel/cs267/lecture24/lecture24.html>
- [13] https://people.nscl.msu.edu/~lund/uspas/scs_2016/lec_adv/A1a_EM_PIC.pdf