
INESTABILIDAD *TWO-STREAM*

PRÁCTICA LIBRE MÉTODOS NUMÉRICOS II

Sara Sales Merino

Febrero, 2024

1 Introducción

Cuando dos haces de electrones que se mueven en direcciones opuestas se interceptan ocurre el fenómeno conocido como inestabilidad *two-stream*. Los haces se consideran fríos, no-colisionales y libres de campo magnético en presencia de un fondo neutralizador de iones. La interacción entre los dos haces genera inestabilidades electroestáticas y inhomogeneidades espaciales que impiden que ambos haces se atraviesen. Parte de la energía cinética de los electrones pasa a generar un aumento del campo eléctrico, dando así lugar a la inestabilidad.

El estudio de esta inestabilidad requiere hacer un modelo del plasma que describa la dinámica a nivel macroscópico de una manera continua y, a nivel microscópico, mediante partículas discretas. El método *particle-in-cell* (PIC) combina una discretización del espacio para los campos con el uso de superpartículas que se mueven continuamente, donde se entiende superpartícula como una nube de partículas reales[4]. De esta manera, este método nos permite modelizar el plasma calculando la fuerza entre las partículas mediante el campo generado por las otras partículas.

2 Modelo matemático

Al tratarse de un caso electroestático, la ecuación de movimiento es:

$$\frac{d^2\mathbf{x}}{dt^2} = -\frac{e}{m}\mathbf{E}, \quad (1)$$

donde e y m son la carga y la masa del electrón, respectivamente. Esta ecuación se resuelve para obtener las posiciones y velocidades de las partículas en cada tiempo. Por otro lado, para obtener el campo electroestático, se usa:

$$\mathbf{E} = -\frac{d\phi}{dx}$$
$$\nabla^2\phi = -\frac{\rho}{\epsilon_0}$$

donde ϕ es el potencial eléctrico, ρ es la densidad de carga y ϵ_0 es la permitividad eléctrica del vacío.

3 Parámetros del plasma y condiciones iniciales

Un gas en equilibrio térmico tiene partículas con todas las velocidades, y la distribución de estas velocidades más probable es la conocida como distribución Maxwelliana[6]. El ancho de la distribución viene dado por la velocidad térmica y está caracterizado por la temperatura, T :

$$v_{th} = \sqrt{\frac{2kT}{m}},$$

donde k es la constante de Boltzmann y m la masa de la partícula.

Cuando un plasma sale de su condición de equilibrio se da lugar a movimientos colectivos de partículas caracterizados por una frecuencia que tienden a restaurar la neutralidad. Dicha frecuencia es la denominada frecuencia del plasma y

viene dada por

$$\omega_r^2 = \frac{n_0 e^2}{\varepsilon_0 m_e}.$$

Otro parámetro fundamental del plasma es la longitud de Debye. Esta indica la distancia a la cual la influencia del campo eléctrico de una sola partícula cargada puede ser sentido por otras partículas cargadas dentro del plasma. Es decir, a una distancia menor a la longitud de Debye los efectos partícula-partícula son mas fuertes y a una distancia menor predominan los efectos colectivos. La longitud de Debye se relaciona con la velocidad térmica y la frecuencia del plasma como:

$$\lambda_D = \frac{v_{th}}{\omega_r}$$

Para el estudio de la inestabilidad *two-stream* se considera un plasma frío y uniforme de electrones móviles y un fondo de iones inmóviles, para neutralizar el sistema. Se inician N partículas distribuidas aleatoriamente por todo el espacio, de longitud L_x en el caso de la simulación 1D y $L_x \times L_y$ en el caso de la 2D. El número de celdas a la simulación 1D es N_x y en la 2D $N_x \times N_y$. Las velocidades iniciales siguen una distribución Maxwelliana con velocidad $\pm v_b$ y velocidad térmica v_{th} , con un haz con velocidad positiva y el otro negativa. En la simulación 2D la velocidad inicial tendrá únicamente componente x . Finalmente, para que el número de partículas se mantenga constante, se usan condiciones de contorno periódicas en ambas direcciones.

En la tabla 1 se encuentran los valores usados en ambas simulaciones.

Table 1: Parámetros de la simulación

	1D	2D
v_b		3
v_{th}		1
N		40000
N_x		100
N_y	-	10, 30, 50
L_x	800	100
L_y	-	10, 30, 50
Δt		1
<i>steps</i>		100

4 Adimensionalización

Usando las constantes de normalización de la tabla 2, se obtienen las ecuaciones adimensionalizadas a resolver:

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}, \quad (2)$$

$$\frac{d\mathbf{v}}{dt} = -\mathbf{E}, \quad (3)$$

$$\mathbf{E} = -\nabla\phi, \quad (4)$$

$$\nabla^2\phi = \frac{n}{n_0} - 1. \quad (5)$$

donde n es la densidad de electrones en los haces y n_0 la densidad de los iones del fondo neutralizador.

De ahora en adelante se asumen todas las magnitudes adimensionales.

5 Implementación. Método *Particle-In-Cell*

El método PIC usa una porción de partículas y luego extrapola su comportamiento a todo el plasma. Esta aproximación se aplica a la descripción cinética en el método PIC haciendo uso de las superpartículas. Se puede resumir el método en 4 operaciones que se repiten a cada tiempo:

- **Densidad de carga en los nodos:** Las densidades de carga y de corriente (en el caso de una simulación electromagnética) de las superpartículas se acumula en el mallado espacial usando un método de interpolación.

Table 2: Constantes de normalización

velocidad	v_{th}
carga	e
masa	m_e
tiempo	ω_r^{-1}
distancia	λ_D
campo eléctrico	$m_e v_{th} \omega_r / e$
campo magnético	$m_e \omega_r / e$

- **Obtención de los campos:** Los campos se resuelven en el mallado.
- **Interpolación de los campos:** Se interpolan los campos a la posición de las superpartículas.
- **Fuerza sobre las partículas:** A partir de los campos en la posición de las partículas se calcula la fuerza que se ejerce sobre estas.

En esta práctica se usa una simulación electroestática, puesto que el fenómeno que se desea estudiar ocurre sin la necesidad de la presencia de campo magnético. De todos modos, la implementación de una simulación PIC electromagnética es similar a esta, aunque evidentemente su complejidad aumenta. Para calcular la fuerza sobre las partículas (fuerza de Lorentz) se usa el método de Boris[1], que se trata de una adaptación del método *leap-frog*. Además, en cada paso del tiempo hay que calcular la corriente en las celdas. Para esto, un método que garantiza la conservación de la carga es el algoritmo de Esirkepov[7]. Finalmente, para la obtención de los campos, se resuelven la ley de Ampère y la ley de Faraday usando *Finite-Differences-Time-Domain* (FDTD). Se pueden encontrar los detalles de esta implementación en el anexo A.

5.1 Densidad de carga en los nodos

Para obtener la densidad de carga en los nodos se usa un algoritmo de interpolación que distribuye la carga de la partícula a sus nodos vecinos. Hay varios algoritmos para esto, siendo el *Nearest Grid Point* el primero en implementarse y el más sencillo. Aún así, este método causa un error excesivo [4]. De este modo, en esta práctica se ha usado un algoritmo de interpolación de primer orden, conocido como *Cloud-In-Cell*. Este método interpreta cada carga como si fuera una "nube" uniformemente cargada del tamaño de una celda, de manera que cada partícula aporta carga a sus nodos vecinos: dos en el caso de una dimensión y cuatro en el caso de dos dimensiones.

Se considera un mallado equiespaciado donde los nodos están localizados en:

$$\mathbf{r}_{i,j} = (i\Delta x, j\Delta y). \quad (6)$$

En una dimensión, una partícula con posición x_p que está entre el nodo i y el nodo $i+1$, contribuye a las densidades n_i y n_{i+1} como:

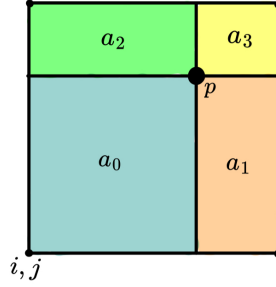
$$n_i = n_i + \frac{x_{i+1} - x_p}{\Delta x} \quad ; \quad n_{i+1} = n_{i+1} + \frac{x_p - x_i}{\Delta x} \quad (7)$$

Para dos dimensiones, el procedimiento se ilustra en la figura 1. La densidad de los nodos vecinos aportada por la partícula es:

$$\begin{aligned} n_{i,j} &= a_3 = \frac{(x_p - x_i)(y_p - y_i)}{\Delta x \Delta y} \\ n_{i+1,j} &= a_2 = \frac{(x_{i+1} - x_p)(y_p - y_i)}{\Delta x \Delta y} \\ n_{i,j+1} &= a_1 = \frac{(x_p - x_i)(y_{j+1} - y_p)}{\Delta x \Delta y} \\ n_{i+1,j+1} &= a_0 = \frac{(x_{i+1} - x_p)(y_{j+1} - y_p)}{\Delta x \Delta y} \end{aligned}$$

La densidad total de carga en los nodos es la suma de las aportaciones de cada una de las partículas. Además, hay que tener en cuenta que se ha fijado un fondo de partículas neutralizador, de modo que para obtener ρ se debe restar a cada uno de los nodos la densidad correspondiente a este fondo. Así:

$$\rho_{i,j} = \frac{n_{i,j}}{n_0} - 1, \quad \text{donde } n_0 = N_{part}/N_{nodos}.$$


 Figura 1: *Cloud-In-Cell* en dos dimensiones

5.2 Resolución de la ecuación de Poisson

Aunque existen métodos directos para resolver la ecuación de Poisson, como el *Fast Fourier Transform*, que es ampliamente utilizado cuando las condiciones de contorno son periódicas, se ha optado por el método iterativo de Gauss-Seidel. La elección de este método se justifica por la simplicidad de implementación de este en comparación a otras alternativas. Para garantizar la convergencia se ha usado una precisión de 10^{-6} en caso 1D y 10^5 iteraciones en caso 2D.

Se discretiza la ecuación (5) utilizando diferencias finitas (central). En una dimensión se tiene:

$$\frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{(\Delta x)^2} = \rho_i, \quad (8)$$

Mientras que en dos dimensiones:

$$\frac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{(\Delta x)^2} + \frac{\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}}{(\Delta y)^2} = \rho_{i,j}.$$

Haciendo uso del método de Gauss-Seidel se obtiene la fórmula iterativa que hay que implementar. En una dimensión:

$$\phi_i^{(k+1)} = -\frac{1}{2} \left(\rho_i (\Delta x)^2 - \phi_{i-1}^{(k+1)} - \phi_{i+1}^{(k)} \right).$$

Y en dos dimensiones:

$$\phi_i^{(k+1)} = -\frac{(\Delta x)^2 (\Delta y)^2}{2((\Delta x)^2 + (\Delta y)^2)} \left(\rho_{i,j} - \frac{1}{(\Delta x)^2} (\phi_{i-1,j}^{(k+1)} + \phi_{i+1,j}^{(k)}) - \frac{1}{(\Delta y)^2} (\phi_{i,j-1}^{(k+1)} + \phi_{i,j+1}^{(k)}) \right).$$

Teniendo en cuenta que en el caso que se trata $\Delta x = \Delta y$, la ecuación anterior se reduce a:

$$\phi_i^{(k+1)} = -\frac{1}{4} \left((\Delta x)^2 \rho_{i,j} - \phi_{i-1,j}^{(k+1)} - \phi_{i+1,j}^{(k)} - \phi_{i,j-1}^{(k+1)} - \phi_{i,j+1}^{(k)} \right).$$

5.3 Campo eléctrico en los nodos

Se obtiene el campo eléctrico a partir del potencial discretizando la ecuación (4) mediante diferencias finitas. En una dimensión:

$$E_i = \frac{\phi_{i-1} - \phi_{i+1}}{2\Delta x}.$$

En dos dimensiones:

$$E_{i,j}^x = \frac{\phi_{i-1,j} - \phi_{i+1,j}}{2\Delta x}, \quad E_{i,j}^y = \frac{\phi_{i,j-1} - \phi_{i,j+1}}{2\Delta y}.$$

5.4 Campo eléctrico en la posición de las partículas

Una vez calculado el campo en el mallado se ha de interpolar en la posición de las partículas. Usando de nuevo el método usado en el apartado 5.1, el campo eléctrico en la posición de la partícula p en una dimensión viene dado por:

$$E_p = \frac{x_{i+1} - x_p}{\Delta x} E_i + \frac{x_p - x_i}{\Delta x} E_{i+1}.$$

Y en dos dimensiones:

$$E_p = a_3 E_{i,j} + a_2 E_{i+1,j} + a_1 E_{i,j+1} + a_0 E_{i+1,j+1}.$$

donde las áreas a_0, a_1, a_2 y a_3 son las dadas en la figura 1 y calculadas en el apartado 5.1.

5.5 Fuerza electrostática sobre las partículas

Existen muchos métodos para resolver ecuaciones diferenciales ordinarias, siendo el método *leap-frog* uno de los más usados en este caso. Aunque existen métodos más precisos como Runge-Kutta de cuarto orden, estos son más difíciles de implementar y requieren de más tiempo de ejecución. Así, se ha elegido el método *leap-frog* ya que es fácil de implementar y suficientemente preciso [4].

Se discretizan las ecuaciones (2) y (3) y se aplica el método *leap-frog* para encontrar las posiciones y velocidades en el siguiente tiempo:

$$\begin{aligned} \mathbf{v}_p^n &= \mathbf{v}_p^{n-1/2} - \mathbf{E}_p^n \frac{\Delta t}{2} \\ \mathbf{x}_p^{n+1} &= \mathbf{x}_p^n + \mathbf{v}_p^n \Delta t \end{aligned} \quad (9)$$

$$\mathbf{v}_p^{n+1/2} = \mathbf{v}_p^n - \mathbf{E}_p^{n+1} \frac{\Delta t}{2} \quad (10)$$

En este método, las velocidades se calculan a medios tiempos. De este modo, antes de iniciar el loop principal, se debe tirar las velocidades medio tiempo atrás y, a la hora de guardar los datos, se deben tirar medio tiempo hacia adelante. Cabe destacar también que entre la ejecución de las ecuaciones (9) y (10) hay que calcular el campo eléctrico (la aceleración) que generan las partículas en tiempo $n + 1$ para poder computar $v^{n+1/2}$.

6 Resultados

En la figura 2 se muestran capturas de la evolución temporal del espacio de fase. Inicialmente los haces tienen velocidades que siguen una distribución Maxwelliana, que podemos ver en la figura 4a como dos bandas uniformes, cada una correspondiente a un haz. Al avanzar el tiempo, se observa la formación espontánea de agujeros de electrones, que son las regiones del espacio de fase donde hay una baja densidad de electrones y una intensificación del campo eléctrico. Vemos así cómo se genera un potencial casi periódico generado por las inhomogeneidades de la densidad de electrones.

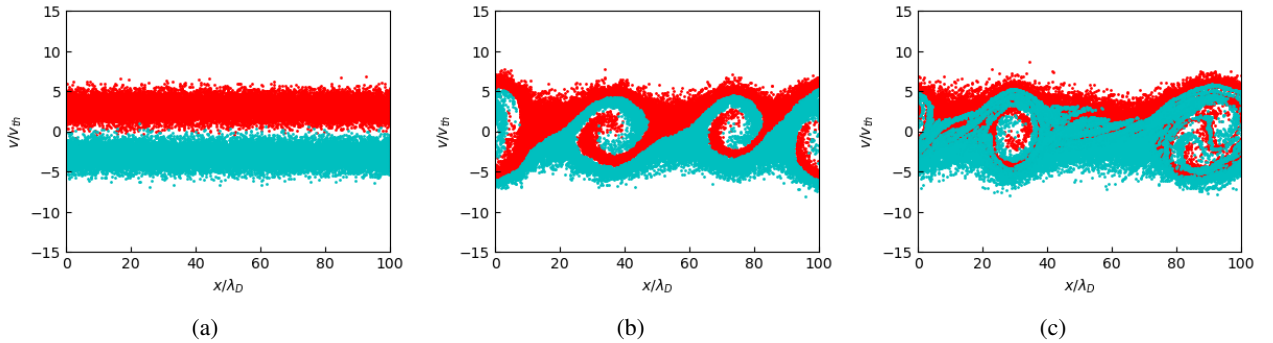


Figura 2: Evolución del espacio de fase en función del tiempo. El color rojo representa el haz viajando hacia la derecha y el color azul el haz viajando hacia la izquierda. (a) $\omega_r t = 0$, (b) $\omega_r t = 20$ y (c) $\omega_r t = 50$.

La figura 3a es una representación gráfica del ratio entre la energía cinética total y la energía cinética total inicial. La figura 3b, en cambio, representa el ratio entre la energía potencial eléctrica y la energía cinética inicial. Estas dos gráficas nos permiten ver cómo inicialmente la energía electrostática es cero. Esto es debido a que los electrones están inicialmente distribuidos uniformemente por el espacio de modo que, dado el fondo de iones inmóviles, la densidad de carga en los nodos es aproximadamente cero en todo el espacio. Rápidamente parte de la energía inicial de los electrones pasa a aumentar el campo eléctrico, que se refleja como una disminución de energía cinética de las partículas y un aumento de la energía potencial electrostática. Vemos después una oscilación entre la energía de las partículas y la energía eléctrica, que va disminuyendo a medida que avanza el tiempo y el sistema va alcanzando el equilibrio.

Los resultados obtenidos en la simulación 2D encajan con los presentados para la simulación 1D. Aún así, para poder obtener resultados equivalentes, la longitud L_y del sistema debe ser mucho más pequeña que la L_y . En la figura 4 se muestra la intensidad del campo eléctrico en el tiempo $t\omega_r = 30$ para diferentes valores de L_y . Podemos ver que a medida que crece la longitud vertical de la simulación, tal y como era de esperar, el campo eléctrico va perdiendo

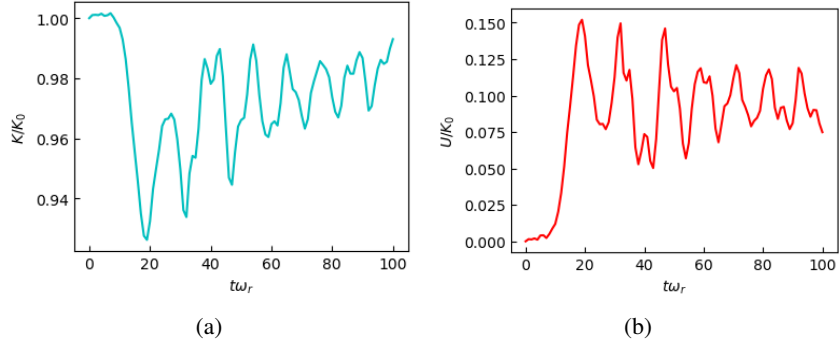


Figura 3: (a) Evolución temporal del ratio de la energía cinética respecto a la energía cinética inicial. (b) Evolución temporal del ratio de la energía electrostática respecto a la energía cinética inicial.

la uniformidad al largo del eje y y, en los puntos de mayor campo, este resulta menos intenso. De este modo, la inestabilidad decrece en intensidad a medida que aumentamos L_y , debido a que las partículas se encuentran menos confinadas.

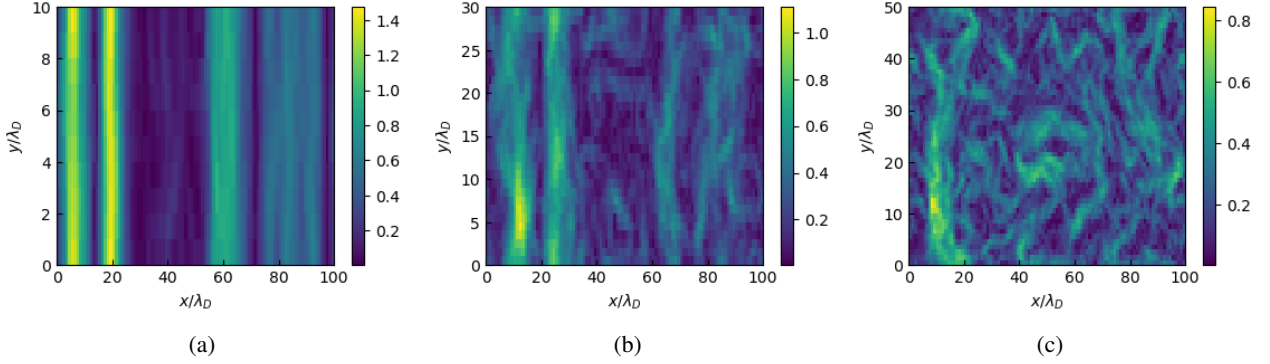


Figura 4: Módulo del campo eléctrico en todo el dominio de la simulación en el tiempo $t\omega_r = 30$ (a) para $L_y = 10$, (b) para $L_y = 30$ y (c) para $L_y = 50$.

7 Conclusiones

El método PIC nos ha permitido simular la inestabilidad *two-stream* de manera satisfactoria y se han obtenido los resultados esperados para la simulación unidimensional tomando como referencias otras hechas en [9, 10, 8]. Se ha observado así cómo la energía oscila entre las partículas y el campo dando lugar a la evolución característica del espacio de fases de esta inestabilidad, marcada por la formación de agujeros de electrones, y que termina por destruir los dos haces.

Por otro lado, en la simulación 2D se ha comprobado que la intensidad de la inestabilidad disminuye cuando el dominio aumenta verticalmente. Esto es debido a que el movimiento de las partículas tiene una menor restricción y, por lo tanto, las interacciones varían respecto a la simulación unidimensional, donde el movimiento y los campos están restringidos a la componente horizontal.

Finalmente, podemos concluir que la elección de los métodos ha sido la apropiada para implementar el modelo del plasma utilizado para simular la inestabilidad. Aún así, existen métodos más precisos que permiten obtener mejores resultados aunque requieren un mayor poder computacional.

A Simulación PIC electromagnética

Para realizar una simulación PIC electromagnética, en primer lugar, hay una fase inicial donde se inician las partículas y se calcula el campo electrostático. En esta primer fase previa al *loop* temporal, se pueden usar los mismos métodos explicados en la simulación electrostática para calcular la densidad de carga en los nodos, resolver la ecuación de Poisson y calcular el campo eléctrico a partir del potencial.

Una vez se ha inicializado las partículas y obtenido el campo eléctrico en los nodos, se inicia un *loop* que se repetirá para cada tiempo. Igual que en el caso electroestático, consiste en cuatro pasos:

- Fuerza sobre las partículas.
- Cálculo de la corriente.
- Resolución de los campos.
- Campos en la posición de las partículas.

A.1 Fuerza sobre las partículas

Mediante el método de Boris[11], se calcula la fuerza de Lorentz y se mueven las partículas un paso temporal. La ecuación a resolver es:

$$\frac{d\mathbf{v}}{dt} = \mathbf{E} + \mathbf{v} \times \mathbf{B}$$

Y discretizada:

$$\frac{\mathbf{r}^{n+1} - \mathbf{r}^n}{\Delta t} = \mathbf{v}^{n+\frac{1}{2}}, \quad (11)$$

$$\frac{\mathbf{v}^{n+\frac{1}{2}} - \mathbf{v}^{n-\frac{1}{2}}}{\Delta t} = \mathbf{E}^n + \frac{\mathbf{v}^{n+\frac{1}{2}} + \mathbf{v}^{n-\frac{1}{2}}}{2} \times \mathbf{B}^n. \quad (12)$$

Tal y como se ve en la discretización realizada, en el método de Boris las posiciones se calculan en tiempos enteros y las velocidades en semienteros. Para resolver la ecuación (12) se separan las fuerzas eléctricas y magnéticas como:

$$\begin{aligned} \mathbf{v}^- &= \mathbf{v}^{n-\frac{1}{2}} + \mathbf{E}^n \frac{\Delta t}{2} \\ \frac{\mathbf{v}^+ - \mathbf{b}^-}{\Delta t} &= \frac{\mathbf{v}^+ + \mathbf{b}^-}{2} \times \mathbf{B}^n \\ \mathbf{v}^{n+\frac{1}{2}} &= \mathbf{v}^+ + \mathbf{E}^n \frac{\Delta t}{2}. \end{aligned}$$

En la primera ecuación se obtiene \mathbf{v}^- añadiendo medio impulso eléctrico a $\mathbf{v}^{n-\frac{1}{2}}$. De la segunda ecuación se obtiene \mathbf{v}^+ como[1]:

$$\mathbf{v}^+ = \mathbf{v}^- + \mathbf{v}' \times \mathbf{s} \quad ; \quad \mathbf{v}' = \mathbf{v}^- + \mathbf{v}^- \times \mathbf{t} \quad ; \quad \mathbf{t} = \frac{\mathbf{B}\Delta t}{2} \quad ; \quad \mathbf{s} = \frac{2\mathbf{t}}{1 + t^2}$$

Finalmente, en la última ecuación se añade la otra mitad del impulso eléctrico para obtener $\mathbf{v}^{n+\frac{1}{2}}$.

Para la nueva posición, directamente de la ecuación (11):

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}^{n+\frac{1}{2}}.$$

A.2 Cálculo de la corriente

Para el cálculo de la corriente se usa el algoritmo de Esirkepov[7]. Las corrientes se calculan, al igual que el campo eléctrico, entre orígenes de celdas para las componentes x e y y en el origen de la celda para la componente z :

$$\begin{aligned} J_{i+\frac{1}{2},j}^x &= -\frac{\Delta x}{\Delta t} W_{i,j}^x \\ J_{i,j+\frac{1}{2}}^y &= -\frac{\Delta y}{\Delta t} W_{i,j}^y \\ J_{i,j}^z &= v_z W_{i,j}^z \end{aligned}$$

Donde W son densidades asociadas a la deposición a lo largo de cada dirección (pesos), y vienen dadas por:

$$\begin{aligned} W_{i,j}^x &= \frac{1}{2}(a_3(x', y') - a_3(x, y') + a_3(x', y) - a_3(x, y)) \\ W_{i+1,j}^x &= \frac{1}{2}(a_2(x', y') - a_2(x, y') + a_2(x', y) - a_2(x, y)) \\ W_{i,j+1}^x &= \frac{1}{2}(a_1(x', y') - a_1(x, y') + a_1(x', y) - a_1(x, y)) \\ W_{i+1,j+1}^x &= \frac{1}{2}(a_0(x', y') - a_0(x, y') + a_0(x', y) - a_0(x, y)) \end{aligned}$$

$$\begin{aligned} W_{i,j}^y &= \frac{1}{2}(a_3(x', y') - a_3(x', y) + a_3(x, y') - a_3(x, y)) \\ W_{i+1,j}^y &= \frac{1}{2}(a_2(x', y') - a_2(x', y) + a_2(x, y') - a_2(x, y)) \\ W_{i,j+1}^y &= \frac{1}{2}(a_1(x', y') - a_1(x', y) + a_1(x, y') - a_1(x, y)) \\ W_{i+1,j+1}^y &= \frac{1}{2}(a_0(x', y') - a_0(x', y) + a_0(x, y') - a_0(x, y)) \end{aligned}$$

$$\begin{aligned} W_{i,j}^z &= \frac{1}{3}a_3(x', y') + \frac{1}{6}a_3(x', y) + \frac{1}{6}a_3(x, y') + \frac{1}{3}a_3(x, y) \\ W_{i+1,j}^z &= \frac{1}{3}a_2(x', y') + \frac{1}{6}a_2(x', y) + \frac{1}{6}a_2(x, y') + \frac{1}{3}a_2(x, y) \\ W_{i,j+1}^z &= \frac{1}{3}a_1(x', y') + \frac{1}{6}a_1(x', y) + \frac{1}{6}a_1(x, y') + \frac{1}{3}a_1(x, y) \\ W_{i+1,j+1}^z &= \frac{1}{3}a_0(x', y') + \frac{1}{6}a_0(x', y) + \frac{1}{6}a_0(x, y') + \frac{1}{3}a_0(x, y) \end{aligned}$$

Las primas indican la coordenada después de Δt y las areas en función de la posición son las dadas por la figura 1.

Vemos que para que tenga sentido (x, y) y (x', y') deben estar en la misma celda. Para conseguir esto, cuando una partícula cambia de celda se divide su trayectoria en dos, de modo que hace su deposición de corriente en diferentes celdas por separado.

A.3 Resolución de los campos

Tratando el caso de un espacio bidimensional, se pueden dar dos tipos de polarización, el modo E y el modo H. Si inicialmente solo existe modo H y corrientes en las direcciones x e y , entonces los campos transversales magnéticos no se excitan[4]. De este modo, se utiliza únicamente el método para los campos transversales eléctricos. En la figura 5 se muestra la celda de Yee[2] con las posiciones de E_x , E_y y B_z . Las componentes x e y del campo eléctrico se encuentran en las posiciones $(i, j + 1/2)$ y $(i + 1/2, j)$ respectivamente, mientras que la componente z del campo magnético se encuentra en (i, j) .

Para obtener los campos se solucionan la ecuación de Faraday y ley de Ampère . Aplicando el método FDTD (*Finite Differences Time Domain*) se tiene:

$$\begin{aligned} B_{i,j}^{zn+1/2} &= B_{i,j}^{zn-1/2} + \frac{\Delta t}{\Delta x}(E_{i,j+1/2}^{xn} - E_{i,j-1/2}^{xn} - E_{i+1/2,j}^{yn} + E_{i-1/2,j}^{yn}) \\ E_{i,j+1/2}^{xn+1} &= E_{i,j+1/2}^{xn} + \frac{\Delta t}{\Delta x}(B_{i,j+1}^{zn+1/2} - B_{i,j}^{zn+1/2}) - \Delta t J_{i,j+1/2}^{xn+1/2} \\ E_{i+1/2,j}^{yn+1} &= E_{i+1/2,j}^{yn} + \frac{\Delta t}{\Delta x}(B_{i+1,j}^{zn+1/2} - B_{i,j}^{zn+1/2}) - \Delta t J_{i+1/2,j}^{yn+1/2} \end{aligned}$$

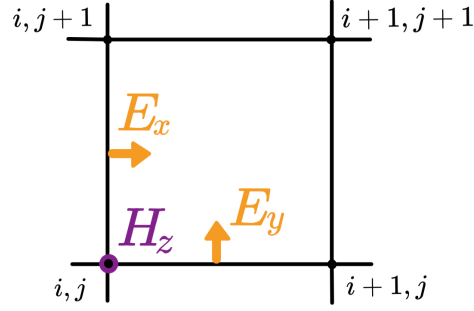


Figura 5: Celda de Yee en 2D para el modo H

A.4 Campos en la posición de las partículas

Se interpolan los campos a la posición de las partículas. Volviendo a usar el método CIC (figura 1), para el campo magnético se tiene:

$$B_p^z = a_3 B_{i,j}^z + a_2 B_{i+1,j}^z + a_1 B_{i,j+1}^z + a_0 B_{i+1,j+1}^z.$$

Para las componentes x e y del campo eléctrico se debe tener en cuenta que, tal y como indica la figura 5, se encuentran en posiciones intermedias. Así:

$$E_p^x = \frac{a_3}{2}(E_{i,j}^x + E_{i,j-1}^x) + \frac{a_2}{2}(E_{i+1,j}^x + E_{i+1,j-1}^x) + \frac{a_1}{2}(E_{i,j+1}^x + E_{i,j}^x) + \frac{a_0}{2}(E_{i+1,j+1}^x + E_{i+1,j}^x)$$

$$E_p^y = \frac{a_3}{2}(E_{i-1,j}^y + E_{i,j}^y) + \frac{a_2}{2}(E_{i,j}^y + E_{i+1,j}^y) + \frac{a_1}{2}(E_{i-1,j+1}^y + E_{i,j+1}^y) + \frac{a_0}{2}(E_{i,j+1}^y + E_{i+1,j+1}^y).$$

References

- [1] Boris method. <https://www.particleincell.com/2011/vxb-rotation/>.
- [2] Yee lattice. http://jdj.mit.edu/wiki/index.php?title=Yee_lattice&printable=yes.
- [3] Carlo Benedetti, Andrea Sgattoni, Giorgio Turchetti, and Pasquale Londrillo. Aladyn: A high-accuracy pic code for the maxwell-vlasov equations. *Plasma Science, IEEE Transactions on*, 36:1790 – 1798, 09 2008.
- [4] C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation*. Adam Hilger, 1991.
- [5] J S Blandón, J P Grisales, and H Riascos. Electrostatic plasma simulation by particle-in-cell method using anaconda package. *Journal of Physics: Conference Series*, 850(1):012007, jun 2017.
- [6] Francis F. Chen. *Plasma Physics and controlled fusion*. Plenum Press, 1974.
- [7] T.Zh. Esirkepov. Exact charge conservation scheme for particle-in-cell simulation with an arbitrary form-factor. *Computer Physics Communications*, 135(2):144–153, 2001.
- [8] Richard Fitzpatrick. Particle-in-cell codes. <https://farside.ph.utexas.edu/teaching/329/lectures/node96.html>.
- [9] F Gbaorun, E John, T Aper, Tomescu Daniel, and Favour Eriba Idoko. Simulation of electron-electron two stream instability (etsi). *NIGERIAN ANNALS OF PURE AND APPLIED SCIENCES*, 1:265–273, 12 2019.
- [10] Philip Mocz. Create your own plasma pic simulation. <https://medium.com/swlh/create-your-own-plasma-pic-simulation-with-python-39145c66578b>.
- [11] Hong Qin, Shuangxi Zhang, Jianyuan Xiao, Jian Liu, Yajuan Sun, and William M. Tang. Why is Boris algorithm so good? *Physics of Plasmas*, 20(8):084503, 08 2013.
- [12] Xavier Sáez Pous. *Particle-in-Cell Algorithms for Plasma Simulations on Heterogeneous Architectures*. PhD thesis, Universitat Politècnica de Catalunya, 2015.