

**ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE CIENCIAS**

**SIMULACIÓN CINÉTICA DE UN SHOCK RELATIVISTA PARA  
ESTUDIAR LOS MECANISMOS DE ACELERACIÓN EN LA  
EMISIÓN TEMPRANA DE UN GRB**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
FÍSICO**

**PROYECTO DE INVESTIGACIÓN**

**OSCAR FERNANDO BUSTOS GANCHOZO**

**oscarphysics@gmail.com**

**DIRECTOR: NICOLÁS ALEJANDRO VASQUEZ PAZMIÑO, PH.D.**

**nicolas.vasquez@epn.edu.ec**

**Quito, Abril 2018**

## DECLARACIÓN

Yo, OSCAR FERNANDO BUSTOS GANCHOZO, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual, correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.



---

Oscar Fernando Bustos Ganchozo.

## CERTIFICACIÓN

Certifico que el siguiente trabajo fue desarrollado por OSCAR FERNANDO BUSTOS GANCHOZO, bajo mi supervisión



Nicolás Alejandro Vásquez Pazmiño, Ph.D.

Director del Proyecto.

## **AGRADECIMIENTOS**

Agradezco enormemente al Dr, Nicolás Alejandro Vásquez Pazmiño, por la colaboración durante el tiempo que duro este trabajo de investigación.

## **DEDICATORIA**

Este trabajo esta dedicado a la gran paciencia de mis padres, tíos y demas familiares que siempre están pendientes de mi bienestar.

# Índice general

<b>Lista de Figuras</b>	<b>2</b>
<b>Resumen</b>	<b>3</b>
<b>Abstract</b>	<b>4</b>
<b>1. Marco teórico</b>	<b>5</b>
1.1. Introducción . . . . .	5
1.2. Ráfagas de Rayos Gamma (GRBs) . . . . .	6
1.2.1. ¿Que son los GRBs? . . . . .	6
1.2.2. Modelo Estandar del Fireball . . . . .	7
1.2.3. Los shock relativistas como plasmas sin colisiones . . . . .	10
1.3. Descripción cinética del shock . . . . .	12
1.3.1. Las Ecuaciones Vlasov-Maxwell Acopladas . . . . .	12
1.3.2. Ecuaciones adimensionales del plasma . . . . .	14
1.3.3. Ecuaciones de Vlasov-Maxwell para el caso unidimensional . . . . .	15
<b>2. Integración de las ecuaciones Vlasov-Maxwell</b>	<b>18</b>
2.1. Método PIC . . . . .	18
2.1.1. Descripción del método PIC . . . . .	18
2.1.2. Integración de las Ecuaciones de Movimiento . . . . .	21
2.1.3. Interpolación de los campos sobre las partículas . . . . .	22
2.1.4. Integración de las ecuaciones de Maxwell . . . . .	24
2.1.5. La fuentes del campo electromagnético . . . . .	25
2.1.6. Corrección del campo Eléctrico . . . . .	25
2.2. Pseudo algoritmo tipo PIC . . . . .	26

<b>3. Experimentos computacionales y Resultados</b>	<b>29</b>
3.1. Inestabilidad de dos corrientes: caso no relativista . . . . .	29
3.2. Inestabilidad de dos corrientes: caso relativista . . . . .	30
3.3. Shock relativista netamente eléctrico . . . . .	31
3.4. Shock relativista electromagnético . . . . .	36
<b>4. Conclusiones</b>	<b>42</b>
<b>A. Código PIC en C++</b>	<b>44</b>
<b>Bibliografía</b>	<b>68</b>

# Índice de figuras

1.1. Modelo del Fireball . . . . .	9
1.2. Shock Relativista . . . . .	11
1.3. Sistema de Referencia . . . . .	16
1.4. Condiciones de bordes periódica para los shocks. Esta condición de borde genera dos frentes de interacción entre las capas. El frente 1 y 2, donde se realiza el shock, y el frente 3 y 4 que se genera por las condiciones de borde. . . . .	17
2.1. Esquematzación de los campos en una malla unidimensional. . . . .	23
3.1. Inestabilidad de dos corrientes: caso clásico . . . . .	30
3.2. Inestabilidad de dos corrientes: caso relativista . . . . .	32
3.3. Shock netamente eléctrico . . . . .	33
3.4. Evolución de la intensidad del campo eléctrico durante el shock netamente eléctrico . . . . .	34
3.5. Conservación de la energía en el shock eléctrico . . . . .	35
3.6. Evolución de la distribución espectral de partículas en funcion del $\Gamma$ para el shock eléctrico . . . . .	36
3.7. Shock electromagnético . . . . .	38
3.8. Variación del campo electromagnético en el shock electromagnético . . . . .	39
3.9. Conservación de la energía en un shock electromagnético . . . . .	40
3.10. Variación del espectro de las partículas, durante el shock electromagnético. . . . .	41



# Resumen

Los GRBs (ráfagas de rayos gamma) son procesos radiactivos altamente energéticas. El modelo mas aceptado para describirlo es el modelo estándar del fireball. Este modelo describe regiones posibles donde se producen los GRBs por medio de shock relativistas, generados por capas que se mueven con distintos factores de Lorentz  $\Gamma$ . Es bastante aceptado que en estos shocks las partículas cargadas sean aceleradas por fluctuaciones en el campo electromagnético. Al tener un composición de electrones, protones y bajas densidades, las partículas no interactúan binariamente, si no por medio de la interacción campo-partícula. Por lo cual estos shock se los considera como un plasma no colisional. En este trabajo de investigación implementamos un algoritmo PIC para la descripción de estos shock, considerando que inicialmente no están magnetizados.

# Abstract

Gamma Ray Bursts (GRBs) are highly energetic radiation processes. The most accepted model to describe them is the standard Fireball model. This model describes possible regions where GRBs are produced, originated from relativistic shocks, generated by layers that move with different Lorentz factors  $\Gamma$ . It is well accepted that in these shocks the charged particles are accelerated by fluctuations of the electromagnetic field. Having a composition of electrons, protons with low densities, the particles do not interact mutually, but by the interaction field-particle. Therefore, these shock are considered as collisionless plasma. In this research work we implemented a PIC algorithm for the description of these shock, considering that initially they are not magnetized.

# Capítulo 1

## Marco teórico

### 1.1. Introducción

En Agosto del año 2017, se observó simultáneamente las ondas gravitacionales GW170817 y el GRB170817A en la fusión de dos estrellas de neutrones (NS-NS) [1]. Este evento cósmico confirmó: a la fusión binaria de estrellas de neutrones como “en al menos uno” de los progenitores de las ráfagas de rayos gamma (por su siglas en Inglés GRBs) [1, 5, 12]. Así esta detección marcó un hito en la astrofísica y extendió el campo de estudio de los fenómenos transiente en altas energías. La evidencia observacional permitió extender la validez de los modelos clásicos de los GRBs y en este trabajo se revisa los posibles procesos de aceleración que intervienen en la fotoproducción de rayos gamma en los GRBs [19, 28].

El modelo más aceptado es el “fireball”: considera que instantáneamente después del cataclismo cósmico, un material es eyectado a velocidades relativistas [15]. Este material esta compuesto de electrones, positrones, protones, etc. Debido a esta composición es considerado como un plasma [14, 15]. Durante su expansión, este plasma interactúa con su ambiente que posee densidades bajas [22, 23], esto frena al material generando una serie de shocks relativistas entre capas, ya sea con el ambiente o internas al material en expansión. Debido a las densidades bajas presentes entre las distintas capas en estos shocks, las partículas cargadas no colisionan entre ellas [29], si no que interactúan con el efecto colectivo generado por las mismas partículas, es decir interactúan con las variaciones de campo producido por las partículas en el shock. Estas variaciones de campo aceleran y frenan a los electrones, emitiendo radiación no térmica [22, 23]. Es la

razón, por la cual estos shocks sin colisiones juegan un papel importante en las ráfagas de rayos gamma (GRBs) [19]. En estos escenarios está presente radiación no térmica que constituye parte de la población de los rayos cósmicos observados [19].

Puesto que estamos interesados en el incremento y las fluctuaciones del campo electromagnético producido en los shocks. En este trabajo se propone realizar un estudio de estos shocks por medio de simulaciones cinéticas mediante el uso de un algoritmo PIC (particle in cell) [16, 17]. PIC es un método numérico que permite simular plasmas sin colisiones integrando el sistema de ecuaciones acoplado de Vlasov-Maxwell [3, 8, 9]. Por motivo de sencillez, se optó para modelizar estos shock un algoritmo PIC 1D3P (una dimensión espacial y las tres componentes de momento lineal en el espacio de fase) [13], considerando dos capas no magnetizadas que van hacia su encuentro. Cada capa se desplazan con distribuciones de Maxwell de momento relativista de acuerdo a sus factores de Lorentz  $\Gamma$  [29].

En este capítulo se hace una revisión sobre modelo estándar de las ráfagas de rayos gamma. Además, la descripción de los shocks relativistas como plasma sin colisiones por medio de las ecuaciones de Maxwell-Vlasov, y la reducción de este sistema a 1D3P. En el segundo capítulo presentamos el método PIC para la resolución de las ecuaciones de Maxwell-Vlasov 1D3P y los pseudo algoritmos que serán implementados. En el tercer capítulo comprobamos el funcionamiento del algoritmo y los resultados de las simulaciones de los shocks relativistas. Finalmente se concluye sobre los mecanismos que influyen en la aceleración de partículas.

## 1.2. Ráfagas de Rayos Gamma (GRBs)

### 1.2.1. ¿Que son los GRBs?

Los GRBs (por sus siglas en inglés: gamma-ray burst) son destellos repentinos e intensos de fotones gamma  $\gamma$  y son detectados en el rango energético de los keV hasta MeV. Cuando esto sucede, opacan por poco segundos otras fuentes de fotones  $\gamma$  presentes en el cielo. Los GRBs pueden ser divididos en dos grupos: ráfagas largas con  $T_{90} > 2s$  y ráfagas cortas con  $T_{90} < 2s$  [21], donde  $T_{90}$  corresponde al periodo de tiempo donde la fluencia alcanza el 90 % de la energía total liberada [21].

Los GRBs fueron detectados por primera vez por el satélite Vela en el año 1967. No se tiene conocimiento exacto de sus progenitores, a pesar de las observaciones mas

avanzadas del telescopio espacial FERMI y otros telescopios ópticos. Sin embargo las posibles fuentes (o progenitores) de los GRBs pueden ser: el colapso del núcleo de una estrella masiva (Supernova SN) para los GRBs largos y la fusión de dos objeto masivos (la fusión entre dos estrella de neutrones (NS-NS) o la fusión de una estrella de neutrones y un agujero negro (NS-BH)) para los GRBs cortos [12]. Estos escenarios generarían las explosiones mas grandes presente en el Universo después del Big Bang [5].

Después de la explosión, la energía liberada conduce el flujo saliente de un plasma ultra-relativista, con factores de Lorentz  $\Gamma > 10$ . La emisión de rayos  $\gamma$  se asume que se debe a shocks sin colisiones dentro de este flujo relativista. Estos shocks ocurren a distancias muy lejanas del centro de la explosión debido a la variabilidad en el viento emitido desde la fuente y a su interacción con el medio ambiente que le rodea [29]. Comúnmente, se asume que los electrones por ser partículas cargadas livianas son aceleradas a altas energías dentro de estos shocks sin colisiones y emiten radiación  $\gamma$  por medio de la emisión sincrotrón [20, 22]. Posterior a la emisión temprana del GRB, otra emisión ocurre llamada “afterglow”. Esta emisión es retrasada y esta compuesta de rayos X, radiación óptica y de radio, su descubrimiento demostró que proviene de la misma fuente del GRB. Este hallazgo permitió determinar con gran precisión las distancias cosmológicas a las que se encuentran las fuentes de los GRBs [26, 29]. El modelo que sugiere que la radiación es generada por shocks es denominado “fireball”, y lo describimos a continuación [10, 19].

### 1.2.2. Modelo Estandar del Fireball

Poco instantes después de la catástrofe cósmica, gran cantidad de energía gravitacional es liberada y convertida en energía libre en escala de tiempo de milisegundos dentro de un volumen del orden de unos pocos kilómetros cúbicos [22]. El principal resultado de esta liberación es la conversión de una fracción de esta energía hacia neutrinos inicialmente en equilibrio térmico, y ondas gravitacionales (las cuales no están en equilibrio térmico), mientras que una fracción mas pequeña va hacia el fireball caliente con una temperatura  $k_B T \sim \text{MeV}$  compuesto de pares  $e^\pm$ , rayos  $\gamma$ , bariones y campo electromagnético [14].

Inicialmente, a un radio  $R_{in}$  el fireball empieza a expandirse adiabáticamente, y logra ser transparente al escape de ondas gravitacionales y de los neutrinos, liberando

energías del orden de  $\sim 10^{53}$  ergs [14]. Los bariones, los pares  $e^\pm$  y fotones  $\gamma$  están en equilibrio térmico. Los pares  $e^\pm$  dominan la profundidad óptica, esto no permite que la radiación térmica y  $\gamma$  escapen, y sea cedida a los bariones del fireball para transformarla en energía cinética. El factor de Lorentz del fireball  $\Gamma$  incrementa linealmente en función de  $r$ . A medida que el fireball se expande los pares  $e^\pm$  se aniquilan y su densidad decae hasta valores de casi neutralidad entre electrones y protones  $n_e \approx n_p$  [14]. Esta expansión se mantiene hasta el radio de saturación  $R_{sat}$  (radio hasta el cual, el  $\Gamma$  del fireball obtiene su valor máximo). Hasta este radio la profundidad óptica ha decrecido pero sigue siendo mayor 1, y la radiación aún no puede escapar. En este punto, se define el radio fotoesférico  $R_{ph}$  cuando la profundidad óptica alcanza la unidad, y termina la aniquilación de pares  $e^\pm$ , manteniendo el equilibrio entre el número de electrones y protones. La temperatura del fireball desciende hasta  $k_B T \sim \text{keV}$  [28] y el  $\Gamma$  del fireball se mantiene constante. Mas allá de  $R_{ph}$  la radiación es capaz de escapar y puede llegar a un observador distante.

Mientras el fireball se sigue expandiendo a velocidades ultrarelativistas, su temperatura desciende hasta el orden de los keV. En presencia de un medio externo, se forman shocks relativistas sin colisiones, que se propagan con factores de Lorentz  $\Gamma \gg 1$ . Un shock sin colisiones se define como una onda de choque. En el cual, la transición de los estados previo y posterior al shock ocurre en una escala de longitud mas pequeña que el camino libre medio de las partículas. La razón por el cual dicha estructura puede existir es porque las partículas no interactúan entre si a través de las colisiones de Coulomb, si no por los efectos colectivos del campo electromagnético generado por las partículas.

Si estos shocks ocurren después de que el fireball se ha hecho ópticamente delgado (la profundidad óptica es pequeña), entonces los shocks podría reconvertir la energía cinética del flujo en un aumento de la energía cinética de las partículas individuales [23]. Aquí es importante que haya un número suficiente de electrones, los cuales después de ser acelerados por la interacción con el campo electromagnético, radiarán energía no térmica a través de la emisión sincrotrón o por combinación de sincrotrón y emisión de inverso Compton para producir el espectro de rayos  $\gamma$  observado [28]. Se espera que los siguientes shocks tomen lugar en este modelo.

- **Shocks Externos (ES)**: es el shock del fireball que se produce al interactuar con el medio externo que rodea al fireball. Este shock explica bastante bien la

radiación del afterglow de múltiples longitudes de onda. En efecto, el modelo del ES anticipó la presencia del afterglow [22]. El ES genera un shock reverso que atraviesa de regreso el volumen del fireball.

- **Shocks Internos (IS):** ocurren cuando sucesivas capas salen de la fuente con diferentes valores de  $\Gamma$ , e interactúan con las capas más lentas o con el shock reverso. Para explicar la rápida variación que presentan las curvas de luz en la emisión temprana de los GRBs, se presume que ocurren múltiples shocks, en la región ópticamente delgada [23].

En los ES, las partículas del fireball interactúan con el medio externo, mientras que en los IS, las partículas interactúan con otras partículas del mismo fireball. Los rayos  $\gamma$  se formarán en los shocks internos en una región que es ópticamente delgada. Dicha región debe estar por fuera de la fotosfera y anterior a los choques externos, para que mucha de su energía salga como radiación no térmica. La radiación que logre escapar, llegará al observador solo si el vector velocidad de las partículas radiantes hace un ángulo mas pequeño que  $\theta \sim \Gamma^{-1}$ , con la línea de visión [14].

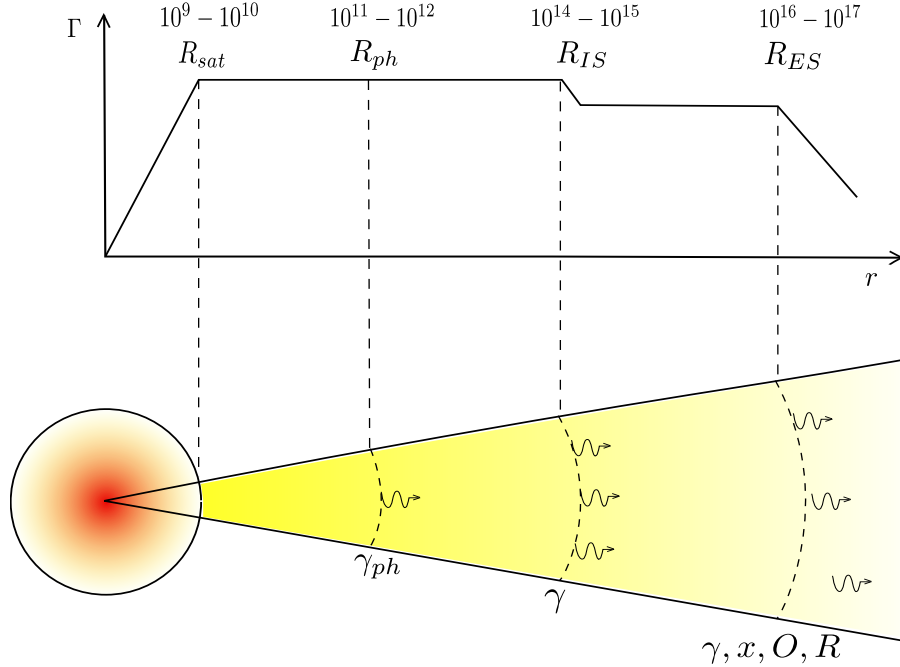


Figura 1.1: **Modelo del fireball.** Esquematización de la variación de  $\Gamma$  en función de los radios,  $R_s$  radio de saturación,  $R_{ph}$  radio de la fotosfera,  $R_{IS}$  radio donde se generan los IS y  $R_{ES}$  los ES. La fotosfera produce rayos  $\gamma$  térmicos, los IS producen rayos  $\gamma$  no térmicos, mientras que los ES producen la radiación del afterglow (adaptada de Mészáros 2006).

En la Figura 1.1, indica la esquematización del comportamiento de  $\Gamma$  versus  $r$ , la distancia radial desde el evento, en el modelo estándar del fireball [14]. Se muestra los radios donde los IS y ES aparecen, ligados con la radiación que pueden emitir y el orden de los valores de estos radios.  $R_{sat}$  es la distancia que define el fin de la aceleración, durante el cual el factor de Lorentz del fireball incrementa con  $r$ .  $R_{ph}$  caracteriza el radio de la fotosfera, donde el fireball se convierte en ópticamente transparente y  $\Gamma$  permanece constante.  $R_{IS}$  define el fin de la fase con  $\Gamma$  constante,  $R_{IS}$  y  $R_{ES}$  son usada para definir las fases entre los IS y ES. Después de  $R_{ES}$ ,  $\Gamma$  decrece, este radio define el inicio de la región donde se genera el afterglow, y la dependencia de  $\Gamma$  en función del radio es decreciente:  $\Gamma \propto r^{-3/2}$  para una expansión adiabática o  $\Gamma \propto r^{-3}$  en el caso radiactivo, considerando que el medio interestelar tiene una densidad uniforme.

En particular, a Figura (1.1) también muestra los radios para la producción de rayos  $\gamma$  [14]. En la fotosfera están indicados como  $\gamma_{ph}$ , y son fotones térmicos absorbidos por algunas partículas del fireball para ganar energía cinética, debido a la opacidad de la región estos fotones no son observados. En los IS en  $R_{IS}$  se producen la mayoría de los rayos  $\gamma$  que llegarán al observador, mientras que en los ES en  $R_{ES}$  se producen rayos  $\gamma$ ,  $x$ , visible  $O$  y de radio  $R$  que pertenecen al afterglow, que también llegaran al observador.

### 1.2.3. Los shock relativistas como plasmas sin colisiones

Los shocks relativistas se generan cuando interactúan dos capas con distintos valores de  $\Gamma$ , comúnmente  $\Gamma_1 > \Gamma_2$ . En la figura (1.2) se esquematiza un shock relativista. Se muestra dos capas interactuando separadas por una línea que representa una discontinuidad en el factor  $\Gamma$  entre las capas. En este trabajo asumimos que el fireball es un plasma caliente, donde se generan ES y IS. Hipotetizamos, que estos shocks son responsables de la generación de radiación no térmica por la aceleración de las partículas cargadas con la presencia de un campo electromagnético, sobre todo electrones puesto que presentan menor inercia que los protones [14, 21].



# Shock

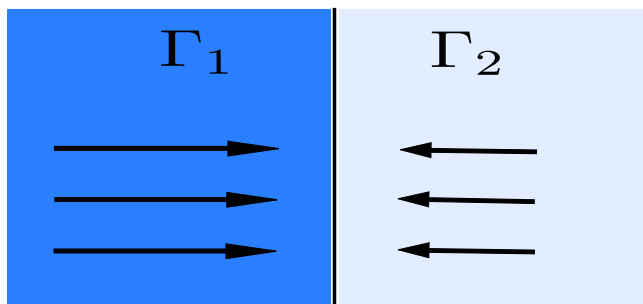


Figura 1.2: Representación de un shock relativista. Dos capas se encuentran con valores de  $\Gamma_1$  vs  $\Gamma_2$ . La línea negra que presentan una discontinuidad en el factor  $\Gamma$

Se conoce que durante el shock las partículas dispersadas son llevadas hacia isotropización del momento, sin embargo esto no puede ser mediado por colisiones binarias entre partículas. Para demostrar esto, se parte desde la colisión binaria de Coulomb, y por el carácter central que posee la fuerza de Coulomb [29]. En el shock las partículas con más probabilidad de chocar son las bariónicas, es decir los protones; por su tamaño y densidad de partículas  $n_p \sim 10^6 \text{cm}^{-3}$  [22]. En el marco de referencia del shock, la densidad de partículas es  $n = \Gamma n_p$  [23], y la sección eficaz para la colisión de Coulomb puede ser estimada por  $\pi d^2$ , con  $d \sim e^2 / (\Gamma m_p c^2)$  [29]. El camino libre medio resultante es  $\lambda \sim 1 / (n \pi d) \sim 10^{31} \Gamma n_p^{-1} \text{cm}$ . Mucho mayor que las dimensiones de  $R_{IS}$ , donde ocurren los shocks que produce la radiación (ver Figura 1.1). Por lo cual es claro que las colisiones de Coulomb no pueden dispersar las partículas en el shock [29].

Con lo deducido anteriormente se puede decir que el shock es un plasma sin colisiones; al igual que todo el fireball en las regiones cercanas a  $R_{IS}$ . La isotropización del momento de las partículas dispersadas se debe a efectos de inestabilidad electromagnética, que generan campos magnéticos y eléctricos en el shock. Estas inestabilidades se desarrollan en una escala de tiempo comparable al inverso de la frecuencia del plasma relativista  $t \sim 1/\omega_p$ . Por lo cual se espera que la distribución sea isotropizada en una escala de  $1/\omega_p$ , mientras que el ancho del shock, el tamaño de la región sobre la cual las partículas están siendo dispersadas, se espera que sea del orden de  $c/\omega_p$ , donde

$$\frac{c}{\omega_p} = c \left( \frac{4\pi\Gamma n_p e^2}{\Gamma m_p} \right)^{1/2} = c \left( \frac{4\pi n_p e^2}{m_p} \right)^{-1/2} \sim 10^7 n_p^{-1/2} \text{cm} [29], \quad (1.1)$$

este cálculo es realizado para protones.

Con estas inestabilidades los shocks relativistas, generan campos, que aceleran a las partículas que posteriormente generaran radiación. Es de nuestro interés estudiar estos mecanismos de aceleración, considerando inicialmente dos capas de electrones y protones, no magnetizados encontrándose frontalmente la una con la otra a distintos valores de  $\Gamma$ .

Como se ha demostrado, la presencia de partículas cargadas en el shock relativista (suponiendo la ausencia completa de partículas neutras), no son desviadas por medio de colisiones entre partículas. Mas bien, las partículas son desviadas por el campo electromagnético autogenerado por mismas partículas. Esto hace que no sea válida una descripción de fluidos (por medio de la magnetohidrodinámica), donde importa la interacción entre partículas. Esto justifica, que en los shocks relativistas se utilice la descripción cinética de los plasmas sin colisiones. Por lo cual, se hace uso de la ecuación de Vlasov relativista acoplada a las ecuaciones electromagnéticas de Maxwell para determinar la evolución de un plasma no colisional.

## 1.3. Descripción cinética del shock

### 1.3.1. Las Ecuaciones Vlasov-Maxwell Acopladas

Dado que consideramos a estos shock como plasma sin colisiones, en la región donde se generan, y con altos factores  $\Gamma$ , podemos modelizar por la teoría cinética de plasma, que acopla la evolución de la función de distribución y las ecuaciones de la electrodinámica. La función de distribución de las partículas en el plasma será denotada como  $f_s(\mathbf{r}, \mathbf{p}, t)$ , que describe las partículas del tipo  $s =$  electrones, protones, en el espacio de fases  $(\mathbf{r}, \mathbf{p})$  en un instante  $t$  [6], y su evolución la describe la ecuación de Vlasov relativista, en el sistema Gaussiano

$$\frac{\partial f_s}{\partial t} + \frac{\mathbf{p}}{m_s \Gamma} \cdot \nabla f_s + q_s \left( \mathbf{E} + \frac{1}{c} \frac{\mathbf{p}}{m_s \Gamma} \times \mathbf{B} \right) \cdot \nabla_{\mathbf{p}} f_s = 0, \quad (1.2)$$

donde  $c$  es la velocidad de la luz,  $m_s$  es la masa de la partícula tipo  $s$ ,  $q_s$  es la carga de la partícula tipo  $s$ ,  $\mathbf{p} = \Gamma m_s \mathbf{v}$  es el momento lineal asociado a la partícula de masa  $m_s$  y  $\Gamma = \sqrt{1 + p^2/(m_s c)^2}$  es el factor de Lorentz. Para completar esta descripción es

necesario acoplarla a las ecuaciones de Maxwell

$$\nabla \cdot \mathbf{E} = 4\pi\rho, \quad (1.3)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (1.4)$$

$$\nabla \times \mathbf{E} = -\frac{1}{c} \frac{\partial \mathbf{B}}{\partial t}, \quad (1.5)$$

$$\nabla \times \mathbf{B} = \frac{1}{c} \frac{\partial \mathbf{E}}{\partial t} + \frac{4\pi}{c} \mathbf{j}, \quad (1.6)$$

donde  $\mathbf{E}$  y  $\mathbf{B}$  son el campo eléctrico y magnético respectivamente. Las fuentes del campo electromagnético densidad de carga  $\rho$  y la densidad de corriente  $\mathbf{j}$  están determinadas por los momentos de la función de distribución,

$$\rho = \sum_s q_s \int f_s d^3p, \quad (1.7)$$

$$\mathbf{j} = \sum_s q_s \int f_s \mathbf{v}(\mathbf{p}) d^3p. \quad (1.8)$$

En este trabajo, las ecuaciones de la divergencia (1.3) y (1.4), son necesarias para formular las condiciones iniciales del sistema. Por ello, si se satisfacen en algún tiempo inicial, entonces la Ley de Faraday (1.5) y la Ley de Ampère (1.6) se cumplirán para cualquier tiempo, siempre que la ecuación de continuidad de la carga se cumpla:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{j} = 0. \quad (1.9)$$

Considerando las escalas descritas en la subsección 1.2.3, un shock presenta inestabilidades a escala de tiempo comparables al inverso de la frecuencia de un plasma relativista  $t \sim 1/\omega_p \sim 10^{-8}$ s, y un espesor de unos cuantos  $c/\omega_p \sim 10^3 - 10^7$ cm. Debido al tiempo y el tamaño que tiene el sistema es más factible presentarlo en un sistema de escala más conveniente que describa el mismo fenómeno físico, por lo cual en la siguiente sección se introduce un sistema adimensional equivalente para describir estos shock relativistas.

### 1.3.2. Ecuaciones adimensionales del plasma

Normalizamos el espacio y el tiempo, usando la relajación de las inestabilidades en el shock como el tiempo característico  $T_o = 1/\omega_p$ , y para la longitud característica usamos skin depth para la absorción de radiación en el plasma  $L_o = c/\omega_p$ . Por lo tanto la velocidad característica del sistema  $V_o = c$ , mientras que la masa característica  $M_o = m_e$ . Adicionalmente el momento lineal característico será  $P_o = m_e c$  [24]. Por lo tanto las variables adimensionales tomarían la siguiente forma:

$$t' = \frac{t}{T_o} = \omega_p t, \quad \mathbf{r}' = \frac{\mathbf{r}}{L_o} = \frac{\omega}{c} \mathbf{r}, \quad \mathbf{v}' = \frac{\mathbf{v}}{V_o} = \frac{\mathbf{v}}{c}, \quad \mathbf{p}' = \frac{\mathbf{p}}{P_o} = \frac{\mathbf{p}}{m_e c}, \quad (1.10)$$

aplicando esto sobre la diferencial espacial y temporal obtenemos

$$\nabla' = L_o \nabla = \frac{c}{\omega_p} \nabla, \quad \frac{1}{\partial t'} = \frac{T_o}{\partial t} = \frac{1}{\omega_p \partial t}. \quad (1.11)$$

Por otra parte, tomamos la carga, densidad de la carga y corriente como:

$$q' = \frac{q}{e}, \quad \rho' = \frac{4\pi}{n_o e} \rho, \quad \mathbf{j}' = \frac{4\pi}{n_o e c} \mathbf{j} \quad (1.12)$$

donde  $n_o = n_e$  es la densidad de los electrones,  $n_o e$  es la densidad característica eligiendo la carga elemental del electrón. Reemplazando las ecuaciones 1.10 y 1.11 en la ecuación 1.3 con el tenemos el campo eléctrico  $\mathbf{E} = \mathbf{E}/(en_o L_o)$ . De la misma forma podemos obtener la reducción para el campo magnético  $\mathbf{B} = \mathbf{B}/(en_o L_o)$  a partir de la ecuación 1.5. Para la reducción de la ecuación de Vlasov debemos considerar:

$$f'_s(\mathbf{r}', \mathbf{p}', t) = \frac{(m_e c)^3}{4\pi n_o} f_s(\mathbf{r}, \mathbf{p}, t) \quad (1.13)$$

la introducción de esta reducción de unidades nos permite obtener el siguiente sistema adimensional de Vlasov-Maxwell:

$$\frac{\partial f'_s}{\partial t'} + \frac{\mathbf{p}'}{\Gamma} \cdot \nabla' f'_s + \frac{4\pi n_o e^2 L_o^2}{m_s c^2} \left( \mathbf{E}' + \frac{\mathbf{p}'}{\Gamma} \times \mathbf{B}' \right) \cdot \nabla'_{\mathbf{p}'} f'_s = 0, \quad (1.14)$$

$$\nabla' \cdot \mathbf{E}' = \rho', \quad (1.15)$$

$$\nabla' \cdot \mathbf{B}' = 0, \quad (1.16)$$

$$\nabla' \times \mathbf{E}' = -\frac{\partial \mathbf{B}'}{\partial t'}, \quad (1.17)$$

$$\nabla' \times \mathbf{B}' = \frac{\partial \mathbf{E}'}{\partial t'} + \mathbf{j}', \quad (1.18)$$

mientras que las fuentes del campo electromagnético adquieren la forma:

$$\rho' = \sum_s q'_s \int f'_s d^3 p', \quad (1.19)$$

$$\mathbf{j}' = \sum_s q'_s \int f'_s \frac{\mathbf{p}'}{\Gamma} d^3 p', \quad (1.20)$$

y la ecuación de continuidad

$$\frac{\partial \rho'}{\partial t'} = \nabla' \cdot \mathbf{j}'. \quad (1.21)$$

En particular la elección de este sistema nos permite reducir la carga  $e'$ ,  $m'_e$  y  $c'$  con valores iguales a la unidad, así el factor de Lorentz en este sistema es  $\Gamma = \sqrt{1 + p'^2}$ . De aquí en adelante, sin perjuicio elegimos cualquier cantidad  $A'$  como  $A$ .

### 1.3.3. Ecuaciones de Vlasov-Maxwell para el caso unidimensional

Este proyecto de investigación desarrolla el estudio de shocks relativistas con las siguientes consideraciones: Las capas o regiones en promedio tiene densidad uniforme y se encuentran ubicadas cerca de  $R_{IS}$ . A estas distancias, estas capas tienen frentes planos, y se desplazan unidireccionalmente con distintos factores  $\Gamma$ . Respecto a un observador externo y estacionario. El shock es un plasma cuasineutro, inicialmente no magnetizado, compuesto de electrones y protones de acuerdo al modelo del fireball  $n_e \approx n_p \approx n_o$ . Debido a que  $m_p \approx 1836m_e$ , consideramos que estos se mantienen estáticos. Se considera que la física del sistema es descrita únicamente por la función de distribución  $f_e(\mathbf{r}, \mathbf{p}, t)$ . Además, consideramos que esta función es igualmente valida en ambas regiones del shock. Por la simplicidad del caso usaremos una descripción 1D3P, que implica el movimiento de las partículas en una dimensión espacial manteniendo las tres componentes del momento. El sistema Vlasov-Maxwell en el sistema de referencia

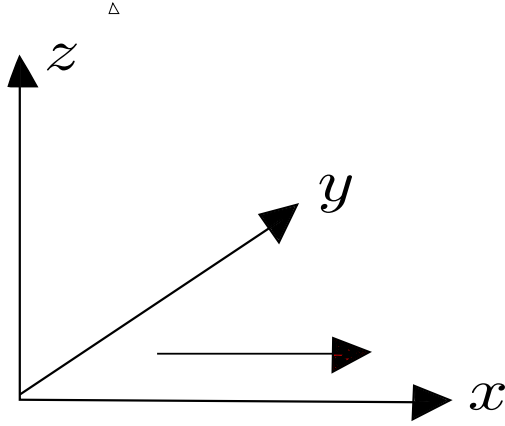


Figura 1.3: En el sistema de referencia, eligiendo el eje  $x$ , reduce el sistema Vlasov-Maxwell a 1D3P.

de la Figura 1.3 se reduce a:

Bajo las consideraciones antes mencionada, el factor en la ecuación de Vlasov para los electrones  $4\pi n_o e^2 L_o^2 / (m_e c^2) = 1$ , Por lo cual el sistema Vlasov-Maxwell adquiere en 1D3P la forma final

$$\frac{\partial f_e}{\partial t} + \frac{\mathbf{p}}{\Gamma} \cdot \nabla f_e + \left( \mathbf{E} + \frac{\mathbf{p}}{\Gamma} \times \mathbf{B} \right) \cdot \nabla_{\mathbf{p}} f_e = 0, \quad (1.22)$$

$$\frac{\partial B_x}{\partial x} = 0, \quad \frac{\partial B_x}{\partial t} = 0, \quad (1.23)$$

$$\frac{\partial E_x}{\partial t} = -j_x, \quad \frac{\partial E_y}{\partial t} = -\frac{\partial B_z}{\partial x} - j_y, \quad \frac{\partial E_z}{\partial t} = \frac{\partial B_y}{\partial x} - j_z. \quad (1.24)$$

$$\frac{\partial B_y}{\partial t} = \frac{\partial B_z}{\partial x}, \quad \frac{\partial B_z}{\partial t} = \frac{\partial E_x}{\partial x}. \quad (1.25)$$

La densidad de carga es  $\rho = \int f_p dp_x - \int f_e dp_x$ . Puesto que tenemos protones cuasies-táticos y cuya densidad es cuasiconstante, su normalización de densidad de carga será cercana a la unidad es decir  $\int f_p dp_x \approx n_p e^+ / n_o |e| \approx 1$ . Mientras que los electrones se mueven libremente entre los protones. Por lo cual la densidad de carga se simplifica a:

$$\rho \approx 1 - \int f_e dp_x, \quad (1.26)$$

y densidad de corriente  $\mathbf{j}$  están determinada únicamente por el movimiento de los

electrones:

$$\mathbf{j} = - \int f_e \frac{\mathbf{p}}{\Gamma} d^3 p. \quad (1.27)$$

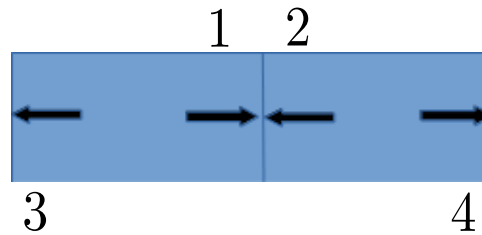


Figura 1.4: Condiciones de bordes periódica para los shocks. Esta condición de borde genera dos frentes de interacción entre las capas. El frente 1 y 2, donde se realiza el shock, y el frente 3 y 4 que se genera por las condiciones de borde.

Con la ayuda de la Figura 1.4 podemos trabajar sobre las condiciones de frontera. Puesto que las capas de un shock están acotadas por otras capas externas con las cuales también interactúan, es necesario incluir esa interacción. La idea más simple es hacer interactuar los frentes opuestos 3 y 4 al shock 1 y 2 de las capas como se muestra en la Figura 1.4. Después de la interacción entre los frentes 1 y 2 las capas retrocederán y por consiguiente se generará un nuevo frente interacción entre las capas pero con los frentes opuestos 3 y 4. Para ello utilizaremos condiciones de bordes periódicas. Sin la necesidad de inyectar más capas de partículas en el dominio computacional y buscar condiciones de bordes más complicadas.

# Capítulo 2

## Integración de las ecuaciones

### Vlasov-Maxwell

#### 2.1. Método PIC

En esta capítulo resumiremos un algoritmo, que permite describir un plasma bajo condiciones de borde periódicas. Empezamos con una descripción general del método Particle in Cell (PIC), para la obtención de las ecuaciones de movimiento para las partículas computacionales a partir de la ecuación de Vlasov y la respectiva discretización de la ecuaciones de Maxwell.

##### 2.1.1. Descripción del método PIC

El método PIC es usualmente utilizado en la simulaciones de plasmas sin colisiones, tiene la ventaja de representar muchas partículas puntuales en una simple cuadrilla computacional o partícula computacional. Las suposiciones consideradas en el modelo físico del algoritmo PIC son mínimas. Las partículas computacionales son rastreadas en el espacio de fase en un marco de referencia Lagrangiano, resolviendo las ecuaciones de sus movimientos. Mientras que las macro cantidades tales como la densidad de carga y densidad de corriente son calculadas simultáneamente, por medio de los momentos de la función de distribución  $f_s$  Valentini et al 2007.

Para resolver el sistema de ecuaciones Vlasov-Maxwell con el algoritmo PIC se parte de aproximar la función de distribución en el espacio de fase como la suma de



las funciones de distribución de las partículas computacionales del tipo  $s$

$$f_s(\mathbf{x}, \mathbf{p}, t) = \sum_i^s f_i^s(\mathbf{x}, \mathbf{p}, t) \quad (2.1)$$

$$= \sum_i^s N_i^s S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_i(t)) S_{\mathbf{p}}(\mathbf{p} - \mathbf{p}_i(t)), \quad (2.2)$$

donde  $N_i^s$  es el número de partículas puntuales en la partícula computacional  $i$  de la especie  $s$ ,  $S_{\mathbf{x}}$  es la llamada función de forma para la posición y  $S_{\mathbf{p}}$  la función de forma para el momento [13].

Para la función de forma que representa el momento se ha elegido la función delta de Dirac,

$$S_{\mathbf{p}}(\mathbf{p} - \mathbf{p}_i(t)) = \delta^3(\mathbf{p} - \mathbf{p}_i(t)). \quad (2.3)$$

Esta elección se la hace considerando que todas las partículas dentro del elemento del espacio de fase descrito por una partícula computacional tienen la misma velocidad, y ellas se mantienen próximas en el espacio de fase durante el subsecuente evolución [13].

La parte espacial de la función de forma es reducida al caso unidimensional

$$S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_i(t)) = S_x(x - x_i(t)), \quad (2.4)$$

y hacemos uso de las funciones llamadas b-splines. Esta particular elección es por la sencillez y por tener valores de cero fuera de la forma espacial de la partícula computacional. Además es conveniente para promediar los campos electromagnéticos en la posición de la partícula computacional, cuando son definidos a trozos contantes en el dominio de una malla computacional. Estas funciones b-splines son una serie de funciones consecutivas de alto orden obteniendo la una de la otra por integración. La primera b-spline es la llamada función flat-top  $b_0(\xi)$  definida como [13]

$$b_0(\xi) = \begin{cases} 1 & \text{si } |\xi| < 1/2 \\ 0 & \text{otros,} \end{cases} \quad (2.5)$$

Los subsecuentes splines b-splines,  $b_l$ , son obtenidos por integración sucesivas por medio de la siguiente formula:

$$b_l(\xi) = \int_{-\infty}^{+\infty} b_0(\xi - \xi_p) b_{l-1}(\xi_p) d\xi_p. \quad (2.6)$$

Basado en los b-splines, la forma espacial de la función para el método PIC es elegida como:

$$S_x(x - x_i(t)) = b_l \left( \frac{x - x_i(t)}{\Delta l} \right) \quad (2.7)$$

donde  $\Delta l$  es la longitud de la partícula computacional. Pocos códigos PIC usan los splines de orden 1, la mayoría usa b-spline de orden 0 [8]. La forma de la partícula en si es una nube uniforme cuadrada en el espacio de fase, con envergadura infinitesimal en la dirección de la velocidad y un tamaño finito en el espacio.

Si reemplazamos la aproximación de la función de distribución (2.2) y tomando los respectivos momentos de la ecuación de Vlasov, podemos obtener las ecuaciones de movimiento para cada partícula computacional [9]

$$\frac{dN_i^s}{dt} = 0, \quad (2.8)$$

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i, \quad (2.9)$$

$$\frac{d\mathbf{p}_i}{dt} = q_s (\mathbf{E}_i + \mathbf{v}_i \times \mathbf{B}_i). \quad (2.10)$$

La primera ecuación representa la conservación del número de partículas físicas por partícula computacional. La segunda y tercera ecuación determinan la posición y el momento de la partícula computacional, respectivamente. El momento de la partícula esta relacionado con la velocidad por  $\mathbf{p}_i = \Gamma m_s \mathbf{v}_i$ . Sobre cada partícula computacional actúa la fuerza promedio generada por los campos eléctrico y magnético que están dados por [13]

$$\mathbf{E}_i = \int_V \mathbf{E} S_x(\mathbf{x} - \mathbf{x}_i(t)) dV \quad \mathbf{B}_i = \int_V \mathbf{B} S_x(\mathbf{x} - \mathbf{x}_i(t)) dV. \quad (2.11)$$

Los campos están promediados sobre la partícula computacional. Estas aproximaciones en el algoritmo PIC dan una gran ventaja desde el punto de vista computacional, puesto que el orden temporal que tiene este el algoritmo es proporcional  $N$ , diferente a la interacción de  $N$  cuerpos cuyo orden es  $N \log(N)$  [13].

### 2.1.2. Integración de las Ecuaciones de Movimiento

En nuestro caso de estudio. Las Ecuaciones de movimiento para las partículas computacionales pueden ser discretizadas por medio del método de leap-frog, el cual determina las coordenadas espaciales en una grilla Euleriana a un tiempo  $t_{n+1} = (n+1)\Delta t$ . Mientras que las velocidades son determinadas en el tiempo  $t_{n+1/2} = (n + 1/2)\Delta t$ . El esquema para cada paso de tiempo con intervalo  $\Delta t$  para la posición es:

$$\frac{x_p^{n+1} - x_p^n}{Dt} = v_{x_p}^{n+1/2}, \quad (2.12)$$

con  $Dt = \Delta t/2$ , y  $\Delta t$  es el paso temporal. considerando, condiciones de bordes periódicas:

$$x = \begin{cases} x = x + L & \text{si } x < 0, \\ x = x - L & \text{si } x > L. \end{cases} \quad (2.13)$$

Donde  $L$  es la longitud del dominio computacional. La ecuación para la posición es integrada dos veces en cada paso de tiempo  $Dt$ , puesto que el error es de segundo orden en la trayectoria de las partículas. La integración del momento lineal sigue el esquema:

$$\frac{\mathbf{p}_p^{n+1/2} - \mathbf{p}_p^{n-1/2}}{\Delta t} = q_s \left( \mathbf{E}_i^n + \frac{\mathbf{p}_p^{n+1/2} + \mathbf{p}_p^{n-1/2}}{2\Gamma_p^n} \times \mathbf{B}_i^n \right). \quad (2.14)$$

donde  $\mathbf{v}_i^{n+1/2} = \mathbf{p}_i^{n+1/2}/(\Gamma_i^{n+1/2})$ . El método de leap-frog es un esquema explícito que depende de las fuerzas calculadas previamente en el paso  $n$ . Sin embargo, tiene una definición implícita para el momento para el nuevo paso de tiempo  $n+1/2$ . Usualmente, el método de Boris [4] se utiliza para resolver la ecuación de diferencias finitas (2.14), introduciendo las variables  $\mathbf{p}^-$  y  $\mathbf{p}^+$  en la siguiente forma

$$\mathbf{p}^- = \mathbf{p}^{n-1/2} + \frac{q_s \mathbf{E}^n \Delta t}{2}, \quad (2.15)$$

$$\mathbf{p}^+ = \mathbf{p}^{n+1/2} - \frac{q_s \mathbf{E}^n \Delta t}{2}, \quad (2.16)$$

reduciendo el campo eléctrico en la ecuación 2.14,

$$\frac{\mathbf{p}^+ - \mathbf{p}^-}{\Delta} = \frac{q_s}{2\Gamma^n} (\mathbf{p}^+ + \mathbf{p}^-) \times \mathbf{B}^n. \quad (2.17)$$

Esto genera un sistema de tres ecuaciones escalares que son resueltas analíticamente, con una interpretación geométrica en la obtención de  $\mathbf{p}^+$ . Se obtiene el vector  $\mathbf{p}^+$  a través de una rotación del vector  $\mathbf{p}^-$ , perpendicular del vector  $\mathbf{B}$ , de manera que las magnitudes de  $\mathbf{p}^+$  y  $\mathbf{p}^-$ , sean iguales. Boris determino en pocos pasos  $\mathbf{p}^+$  [4], que satisface (2.17)

$$\mathbf{t}^n = \frac{q_s \mathbf{B}^n \Delta t}{2\Gamma^n}, \quad (2.18)$$

$$\mathbf{p}' = \mathbf{p}^- + \mathbf{p}^- \times \mathbf{t}^n, \quad (2.19)$$

$$\mathbf{s} = \frac{2\mathbf{t}}{(1 + t^2)}, \quad (2.20)$$

$$\mathbf{p}^+ = \mathbf{p} + \mathbf{p}' \times \mathbf{s}, \quad (2.21)$$

entonces, por medio de (2.16) podemos obtener la actualización del momento,

$$\mathbf{p}^{n+1/2} = \mathbf{p}^+ + \frac{q_s \mathbf{E}^n \Delta t}{2}. \quad (2.22)$$

Finalmente, las partículas que salen de los borde son reiniciadas en la celdas con las misma velocidad con la que salieron del dominio computacional.

### 2.1.3. Interpolación de los campos sobre las partículas

La ubicación de las componentes de los campos en una malla unidimensional se muestran como en la figura (2.1). Las componentes  $E_y$  y  $B_y$  son ubicadas en los nodos de la malla unidimensional, mientras que las componentes  $E_z$ ,  $B_z$  y  $E_x$  son calculadas en el centro de la malla, para los cálculo de las derivadas espaciales en las ecuaciones de Maxwell.

$i$	$i + 1/2$	$i + 1$	$i + 3/2$
$E_y$	$E_z$	$E_y$	$E_z$
$\dot{J}_y$	$\dot{J}_z$	$\dot{J}_y$	$\dot{J}_z$
$B_y$	$B_z$	$B_y$	$B_z$
$\rho$	$E_x$	$\rho$	$E_x$
	$\dot{J}_x$		$\dot{J}_x$

Figura 2.1: Esquematización de los campos en una malla unidimensional.

Pero, antes de calcular el campo que actúa sobre las partículas, las componentes  $E_z$ ,  $B_z$  y  $E_x$  son reubicados en los nodos [18]. Los campos eléctricos y magnéticos que actúan en la posición de la partícula, se los calcula de acuerdo a

$$\mathbf{E}_p = \int_L \mathbf{E}(x)S(x - x_p)dx \quad (2.23)$$

$$\mathbf{B}_p = \int_L \mathbf{B}(x)S(x - x_p)dx, \quad (2.24)$$

donde  $L$  representa el dominio computacional en una dimensión y se lo puede aproximar con límites finitos. Como aproximación se considera que la dimensión de la partícula es de la misma longitud que una celda en la malla  $\Delta x$  y que los campos se mantienen constante sobre una celda de la malla, siguiendo el procedimiento de Lapenta et al. 2006, se obtiene que la interpolación de los campos sobre la partícula tienen la forma

$$E_{x,p} = \sum_i E_{x,i}W(x_i - x_p) \quad (2.25)$$

$$E_{y,p} = \sum_i E_{y,i}W(x_i - x_p) \quad (2.26)$$

$$E_{z,p} = \sum_i E_{z,i}W(x_i - x_p) \quad (2.27)$$

$$B_{y,p} = \sum_i B_{y,i}W(x_i - x_p) \quad (2.28)$$

$$B_{z,p} = \sum_i B_{z,i}W(x_i - x_p) \quad (2.29)$$

donde  $W$  es la función de interpolación de los campos sobre la partícula, y se define

como

$$W(x_i - x_p) = b_1 \left( \frac{x_i - x_p}{\Delta x} \right) = \begin{cases} 1 + \frac{x_i - x_p}{\Delta x} & \text{si } -1 \leq \frac{x_i - x_p}{\Delta x} < 0 \\ 1 + \frac{x_i - x_p}{\Delta x} & \text{si } 0 \leq \frac{x_i - x_p}{\Delta x} < 1 \\ 0 & \text{otros} \end{cases} \quad (2.30)$$

La función de interpolación determina la influencia de los campos entre las celdas  $i$  e  $i + 1$  para una partícula dentro del rango  $x_i \leq x_p < x_{i+1}$  [13].

#### 2.1.4. Integración de las ecuaciones de Maxwell

La discretización de las ecuaciones (1.24) se lo realiza para cada paso de tiempo  $\delta t$  y como se muestra a continuación

$$E_{x,i} = E_{x,i} - \Delta t j_{x,i}, \quad (2.31)$$

$$E_{y,i} = E_{y,i} - \frac{\Delta t}{dx} (B_{z,i} - B_{z,i-1}) - \Delta t j_{y,i}, \quad (2.32)$$

$$E_{z,i} = E_{z,i} + \frac{\Delta t}{dx} (B_{y,i+1} - B_{y,i}) - \Delta t j_{z,i}, \quad (2.33)$$

con las condiciones de borde

$$E_{x,0} = E_{x,L}; \quad (2.34)$$

$$E_{y,0} = E_{y,L}; \quad (2.35)$$

$$E_{z,0} = E_{z,L}. \quad (2.36)$$

Mientras que las ecuaciones del campo magnético 1.25 son discretizadas

$$B_{y,i} = B_{y,i} + \frac{Dt}{dx} (E_{z,i} - E_{z,i-1}), \quad (2.37)$$

$$B_{z,i} = B_{z,i} - \frac{Dt}{dx} (E_{y,i+1} - E_{y,i}), \quad (2.38)$$

con las condiciones de borde

$$B_{y,0} = B_{y,L}; \quad (2.39)$$

$$B_{z,0} = B_{z,L}. \quad (2.40)$$

Las ecuaciones 2.37 y 2.38 al ser integradas con pasos de tiempo  $Dt$ , por lo cual en cada ciclo de iteración, este desarrollo será usado dos veces como se muestra en el algoritmo tomado de [18].

### 2.1.5. La fuentes del campo electromagnético

Una partícula entre os nodos de la malla  $x_i < x_p < x_{i+1}$ , la densidad de carga calculada es

$$\rho_i = \sum_{p=1}^N qW(x_i - x_p); \quad (2.41)$$

y sera calculada al tiempo  $n + 1$ . Una vez realizado este cálculo se debe añadir  $+1$  a cada elemento  $\rho_i$ , para asegurar la presencia de los protones estáticos en el medio. La densidad de corriente  $\mathbf{j}_i^{n+1/2}$ , es calculada a partir del uso de la velocidad  $\mathbf{v}^{n+1/2}$  multiplicado por la función de interpolación  $W$ , y la posición al paso de tiempo  $x^{n+1/2}$  de la partícula p [13], es decir;

$$j_{x,i}^{n+1/2} = \sum_q^N qv_{x,p}^{n+1/2} \frac{W(x_i - x_p^{n+1/2})}{2\Gamma^{n+1/2}} \quad (2.42)$$

$$j_{y,i}^{n+1/2} = \sum_q^N qv_{y,p}^{n+1/2} \frac{W(x_i - x_p^{n+1/2})}{2\Gamma^{n+1/2}} \quad (2.43)$$

$$j_{z,i}^{n+1/2} = \sum_q^N qv_{z,p}^{n+1/2} \frac{W(x_i - x_p^{n+1/2})}{2\Gamma^{n+1/2}} \quad (2.44)$$

Los últimos cálculos son realizados respecto a los nodos de la malla. Las densidades de corriente  $j_x$  y  $j_z$  debe ser reubicadas en el centro de las celdas con  $(j_i + j_{i+1})/2$  [4].

### 2.1.6. Corrección del campo Eléctrico

Debido a los errores de redondeo en el avance temporal, es necesario implementar un método de conservación de carga. Para el caso unidimensional, el mas empleado es obtenido de la solución de la ecuación de Poisson  $\partial E_x / \partial x = \rho$ . En este método se asume que el campo eléctrico  $E_x$  contiene errores, por lo cual se debe calcular un

campo eléctrico de corrección  $E_{x,c}$  el cual satisface la siguiente ecuación

$$\frac{\partial}{\partial x} (E_x + E_{x,c}) = \rho. \quad (2.45)$$

La carga de corrección se la obtiene de  $\rho_c = \rho - \partial E_x / \partial x$  entonces se debe resolver la ecuación  $\partial E_{x,c} / \partial x = \rho_c$ , o la ecuación de Poisson

$$\frac{\partial \phi^2}{\partial x^2} = -\rho_c. \quad (2.46)$$

En nuestro estudio usamos la discretización por diferencia finita como

$$\frac{\phi_{c,i+1} - 2\phi_{c,i} + \phi_{c,i-1}}{\Delta x^2} = -\rho_c \quad (2.47)$$

Para un sistema periódico, se aplica la Fast Fourier Transform (FFT) [7]. Entonces para iniciar el campo eléctrico, corregimos la carga del campo eléctrico y calculamos el campo eléctrico en una simulación puramente eléctrica [18]

## 2.2. Pseudo algoritmo tipo PIC

Este pseudo algoritmo está basado en el código Kyoto University ElectroMagnetic Particle Code One dimension (KEMPO1), adaptado a una versión relativista. El pseudo código se presenta a continuación:

**Pseudo algoritmo 1** PIC electromagnético.

Introducir los parámetros y constantes del sistema: el dominio computacional  $l$ ,  $\Delta x$  discretización de la malla,  $N_p$  número de partículas,  $t_{max}$  el tiempo máximo,  $\Delta t$  el paso del tiempo,  $p_{th}$  el momento térmico, y los factores de Lorentz  $\Gamma_1$  y  $\Gamma_2$ ;

Introducir velocidades  $\mathbf{v}^{-1/2}$  y posiciones  $x^0$  iniciales de las partículas;

Calcular la densidad carga  $\rho^0$ ;

Iniciar el campo eléctrico  $\mathbf{E}^0$  por medio de la corrección del campo eléctrico, realizando los pasos de la subseccion 2.1.6;

Iniciamos  $t=0$ ;

**while**  $t < t_{max}$  **do**

Calcular el campo magnético  $\mathbf{B}^{n+1/2}$ ;



Calcular la velocidad  $\mathbf{v}^{n+1/2}$ ;  
 Calcular la posición  $x^{n+1/2}$ ;  
 Calcular la corriente  $\mathbf{j}^{n+1/2}$ ;  
 Calcular posición  $x^n$ ;  
 Calcular el campo magnético  $\mathbf{B}^n$ ;  
 Calcular el campo eléctrico  $\mathbf{E}^n$ ;  
 Calcular la densidad del carga  $\rho^n$ ;  
 Corregir el campo eléctrico;

**end while**

En el caso netamente eléctrico, el pseudo algoritmo se reduce a

### **Pseudo algoritmo 2** PIC eléctrico

Introducir los parámetros y constantes del sistema: el dominio computacional  $l$ ,  $\Delta x$   
 discretización de la malla,  $N_p$  número de partículas,  $t_{max}$  el tiempo máximo,  $\Delta t$  el  
 paso del tiempo, los factores de Lorentz  $\Gamma_1$  y  $\Gamma_2$ ;

Introducir velocidades  $\mathbf{v}^{-1/2}$  y posiciones  $x^0$  iniciales de las partículas;

Calcular la densidad carga  $\rho^0$ ;

Iniciar el campo eléctrico  $\mathbf{E}^0$ ;

Iniciamos  $t=0$ ;

**while**  $t < t_{max}$  **do**

Calcular la velocidad  $\mathbf{v}^{n+1/2}$ ;

Calcular la posición  $x^{n+1/2}$ ;

Calcular posición  $x^n$ ;

Calcular la densidad del carga  $\rho^n$ ;

Corregir el campo eléctrico (determina  $\mathbf{E}^n$ );

**end while**

Este pseudo algoritmo fue implementado en C++ y se muestra en el Apéndice A. Los siguientes casos de estudio fueron implementados:

1. Caso no relativista, que simula la inestabilidad de dos corrientes en el espacio de fases.
2. Caso relativista, que simula la inestabilidad de dos corrientes en el espacio de fase.

3. Shock netamente eléctrico.

4. Shock con campo electromagnético.

El primer y el segundo caso son fenómenos puramente eléctricos y son utilizados a modo de prueba del algoritmo. Puesto que estos casos son ejemplos clásicos. El tercer y cuarto caso son de interés, ya que son candidatos a mostrar, las fluctuaciones en el campo eléctrico y magnético, cuando dos capas de electrones interactúan a velocidades relativistas. Como las partículas cargadas son aceleradas en las fluctuaciones del campo electromagnético entonces, son posibles candidatas a generar radiación no térmica presente en los GRBs.

# Capítulo 3

## Experimentos computacionales y Resultados

### 3.1. Inestabilidad de dos corrientes: caso no relativista

Como prueba del algoritmo anteriormente expuesto, simulamos la inestabilidad de dos corrientes de electrones que fluyen en dirección contraria y condiciones de borde periódicas. Para ello se emplea el algoritmo PIC en el régimen clásico. Se utilizó un número de partículas  $N = 20000$  repartidas en igual número en cada capa. Se discretizó el espacio en  $N_x = 400$  puntos, cada uno a distancia  $\Delta x = 0,5c/\omega_p$ . Se empleó un tiempo total  $t = 25/\omega_p$ , discretizados a  $\Delta t = 0,01/\omega_p$ . Se tomó inicialmente la distribución de velocidades de Maxwell para las dos corrientes con velocidad térmica 3,0 [7].

En la Figura 3.1, se puede apreciar como dos corrientes uniformes de electrones y dirección contraria, evolucionan desde  $t = 0/\omega_p$  panel *a*) hasta  $t = 25/\omega_p$  panel *f*), con saltos de tiempos  $t = 5/\omega_p$ . A medida que el tiempo progresa las corrientes pierden su uniformidad, panel *b*). Las bandas espontáneamente desarrollan estructuras ondulatorias paneles *c*) y *d*). Finalmente evolucionan a vórtices que se conectan, paneles *e*) y *f*). En este estado final los electrones circundan los vórtices, rebotando hacia atrás y hacia adelante por un potencial cuasi periódico a causa de las densidades no uniformes de los electrones [7]. Las inestabilidades que se forman, efectivamente destruyen la uniformidad de los haces.

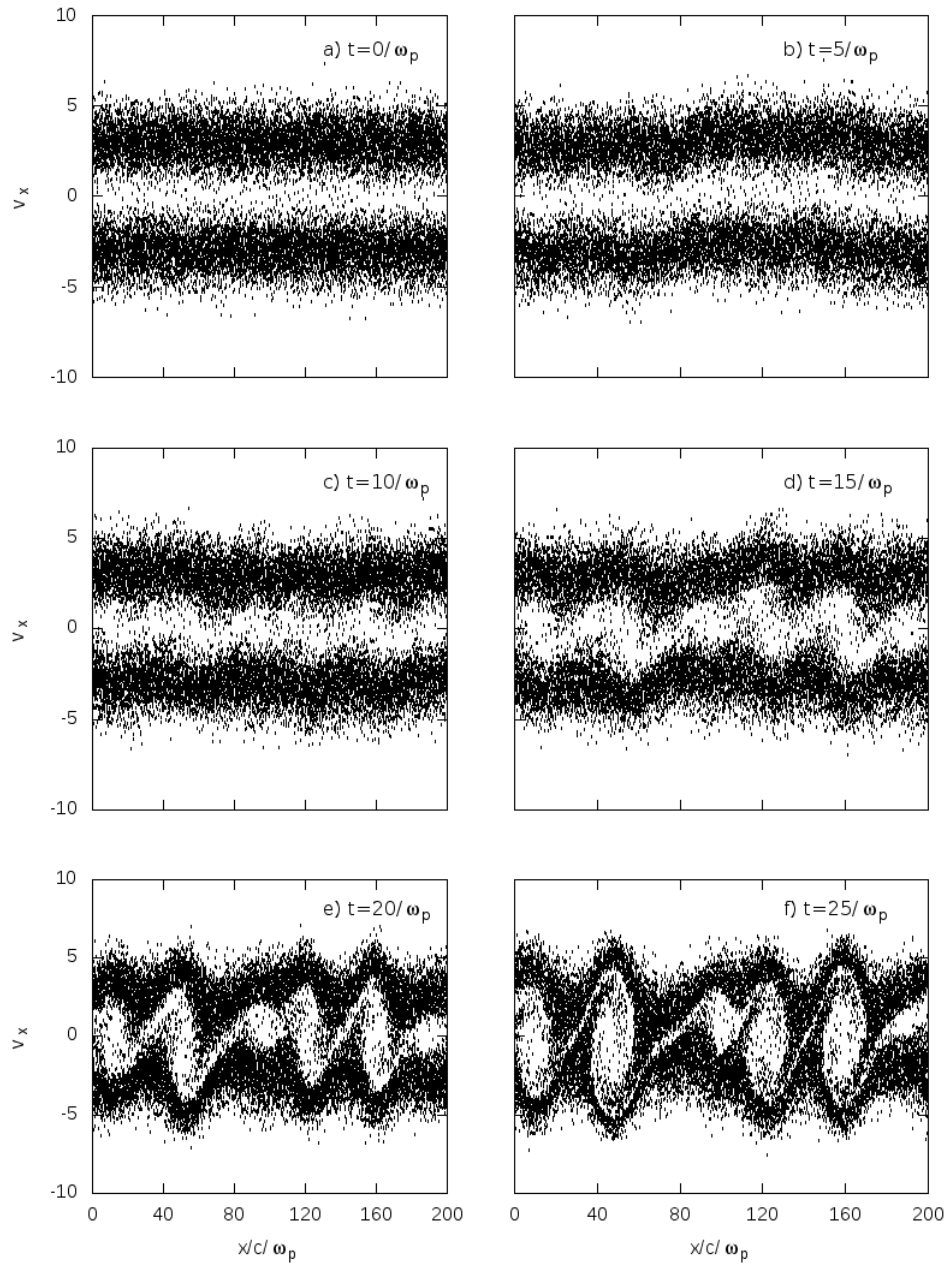


Figura 3.1: Esta gráfica presenta uno de los ejemplos más ampliamente usados, para verificar un algoritmo PIC. Podemos observar como dos corrientes con igual magnitud de velocidad pero con sentido contrario, evolucionan a inestabilidades que forman vórtices, por la formación de potenciales periódicos.

### 3.2. Inestabilidad de dos corrientes: caso relativista

Probamos el algoritmo PIC, para el caso relativista. Consideramos dos corrientes de electrones Figura 3.2. Una corriente es lenta con  $\Gamma \approx 0$ , mientras que una segunda

con mayor velocidad fluye con un factor  $\Gamma \approx 6$ . Se puede observar un efecto que la capa mas rápida empieza a arrastrar a las partículas de la capa mas lenta, formando vórtices con características similares al caso clásico comparando las Figuras 3.1 y 3.2.

En este caso se usaron 40000 partículas repartidas de la siguiente forma: 30000 para la capa mas rápida y 10000 para la capa mas lenta. Se realizan 5000 ciclos con pasos de tiempo  $\Delta t = 0,1\omega_p$ , obteniendo un tiempo total  $t = 500/\omega_p$ . El espacio tiene una longitud de  $500 c/\omega_p$  y es discretizado  $\Delta x = 1c/\omega_p$  [25].

### 3.3. Shock relativista netamente eléctrico

En esta sección presentamos la simulación de un shock relativista netamente eléctrico. Como hemos mencionado es bastante aceptado que estos shocks generen radiación electromagnética de alta energía, cuando las partículas, son aceleradas por las fluctuaciones de campos electromagnéticos. Puesto que nuestro objetivo es determinar estas fluctuaciones. En este primer caso vamos a calcular las fluctuaciones de la magnitud del campo eléctrico, generado por el shock de dos capas electrones moviéndose libremente entre protones. Consideramos en este shock, que solo hay la componente de campo eléctrico  $E_x$  y es iniciada mediante la ecuación de Poisson 2.1.6. No consideramos la existencia del campo magnético externo o auto generado. Por lo cual el problema es unidimensional, e intervienen únicamente las componentes en  $x$ . La componente  $p_x$  del momento tendrá una distribución de Maxwell relativista, de acuerdo al factor  $\Gamma$ , y dirección que lleve la capa de electrones.

En esta simulación. Las capas de electrones inicialmente va a su encuentro con  $\Gamma$  de 5 (dirección positiva) y 10 (dirección negativa), en el espacio de fases. El número de partículas fue de 40000, repartidas igualmente en cada capa. El número de ciclos de 5000 con pasos temporales de  $\Delta t = 0,01$  resultando en un tiempo total de  $t = 50/\omega$ . El espacio de longitud  $600c/\omega$ , discretizado  $\Delta x = 1c/\omega_p$ . El tiempo y el dominio computacional elegido aquí corresponde al orden de formación de los shocks [29]. Se consideran condiciones de borde periódicas. Como una medida para que ambos frentes de cada capa del shock, entren en interacción, Figura 1.4.

Empezamos exponiendo el shock en el espacio de fase Figura 3.3. La capas, al entrar en interacción, panel *a*). Estas empiezan a frenarse, panel *b*) con un leve incremento en la intensidad del campo eléctrico. Mientras la interacción continua entre las capas,

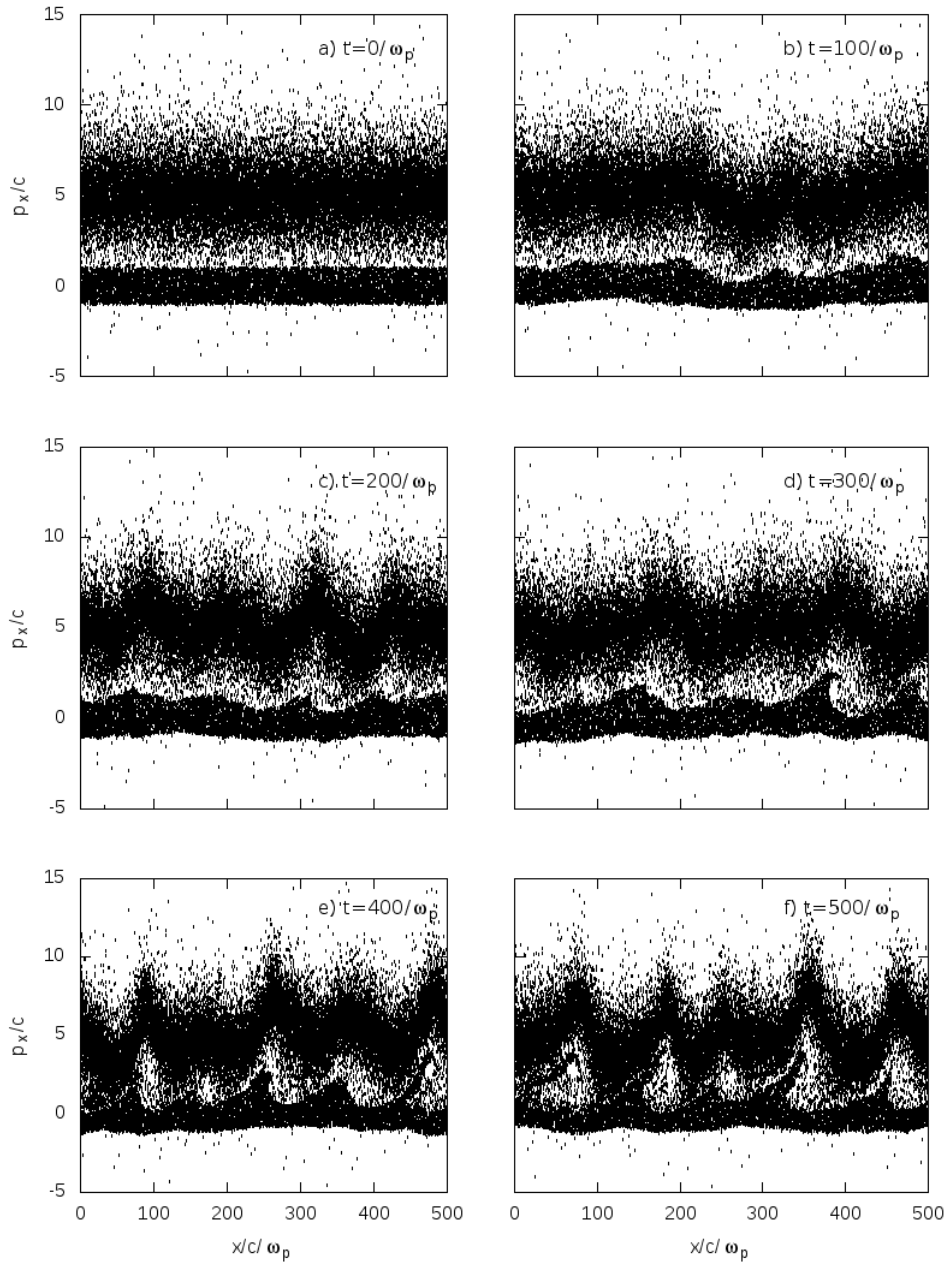


Figura 3.2: Inestabilidad de dos corrientes relativistas. La corriente superior se mueve hacia la derecha con  $\Gamma \approx 6$ , sobre una capa lenta. Se observa la evolución de estas dos corrientes casi uniformes (panel *a*), hasta que ambas generan de inestabilidades parecidas a vórtices (panel *c*). Estos vórtices son a causa de la generación de potenciales periódicos.

la mayor parte de sus partículas adquieren un momento cercano a cero, la mayoría de las partículas se frenan. Mientras que en los frentes de interacción las partículas pueden mantener o aumentar su momento, paneles *c*) y *d*). Con el incremento del campo eléctrico, hará que las capas cambien el sentido de su desplazamiento. Haciendo que estas retrocedan, como se muestra en los paneles *e*) y *f*).

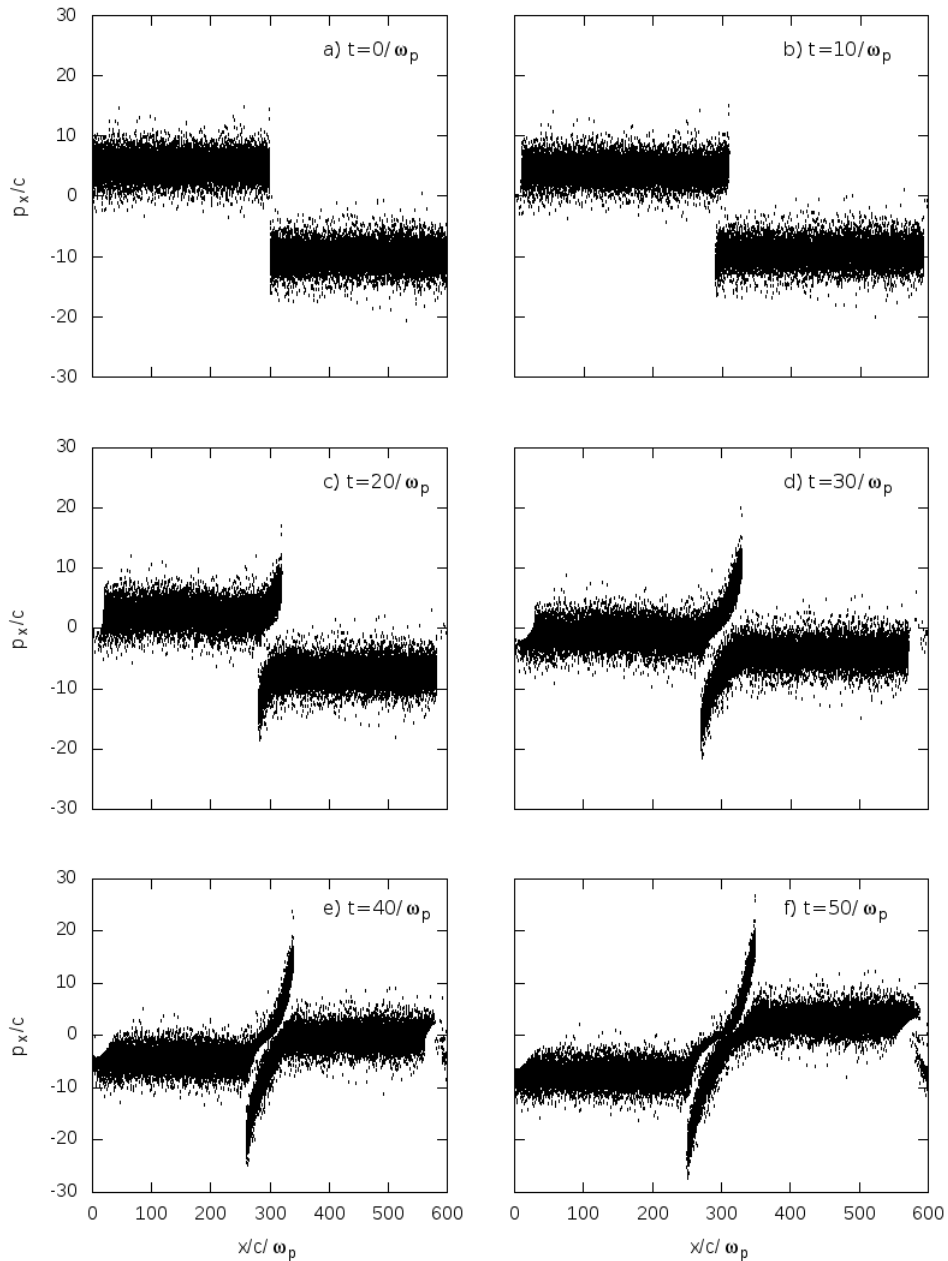


Figura 3.3: Dos capas interactúan con factores  $\Gamma$  de 5 a la izquierda (dirección positiva) y 10 a la derecha (dirección negativa), panel *a*). A continuación se muestra como las capas se frenan, aunque ciertas partículas pueden aumentar o mantener su momento, paneles *b*), *c*) y *d*). Finalmente las capas retroceden, paneles *e*) y *f*).

La intensidad del campo eléctrico sufre un cambio brusco en las proximidades de los frentes de las capas (ver Figura 3.4), es muy parecido a una discontinuidad. Pasando de valores positivos a negativos. Debido a las condiciones de borde periódicas un nuevo shock se forma entre los frentes opuestos al primero, comenzando un proceso similar en los bordes como se observa en la Figura 3.4. Mientras el campo se sigue incrementando, las capas pierden energía cinética llegando a frenarlas. El campo generado retrocede a las capas, formando un estado oscilatorio, que homogenizará el momento de las partículas presentes en el shock. Esto no lo permite ver la simulación ya que únicamente simula los primeros instantes del shock, de dos capas con condiciones de borde periódicas.

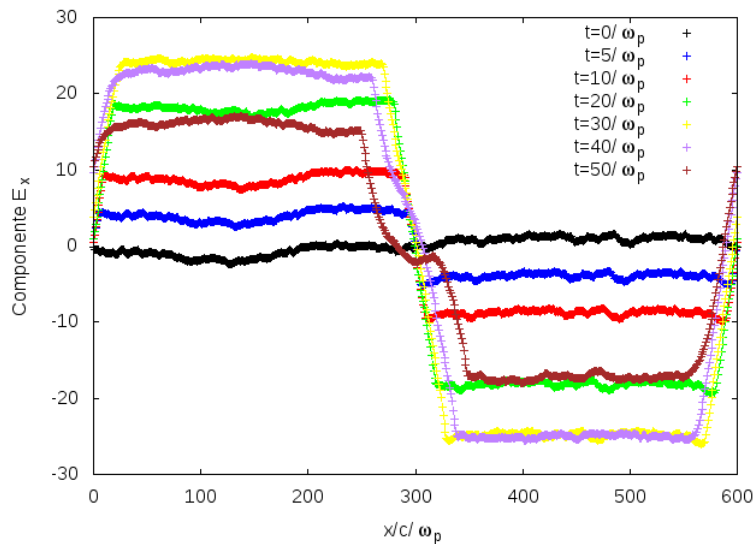


Figura 3.4: Variación de la intensidad del campo eléctrico durante el shock. En centro de esta Figura, se observa como cambia en el tiempo, de positivo a negativo el campo eléctrico, en frente de las capas del shock. Cercano a los bordes se puede observar el mismo efecto, esto causado por las condiciones de borde periódicas, a la que es impuesto el campo eléctrico, donde se genera el otro frente de interacción en las capas.

La siguiente Figura 3.5 muestra la conservación de energía del sistema. La línea roja muestra la energía total del shock, que en un principio es totalmente cinética. La línea azul muestra la evolución de la energía cinética. Se puede observar la disminución de energía cinética de las partículas debido claramente al aumento de la energía del campo eléctrico (línea verde). A su vez este incremento de la energía del campo eléctrico en frente del shock, empezará a empujar las capas haciendo que retrocedan y aceleren las partículas ganando otra vez energía cinética, en el frente del shock.



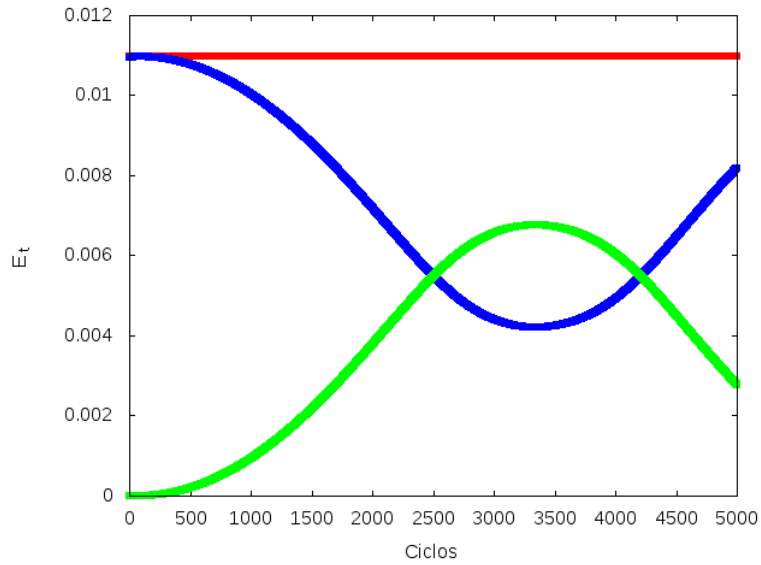


Figura 3.5: Conservación de la energía en el shock eléctrico para 5000 ciclos durante la simulación. La línea roja, muestra la energía total del sistema. Línea azul, la variación de la energía cinética. Línea verde, la energía eléctrica.

Finalmente, mostramos el espectro de las partículas en función de  $\gamma$ , para el shock eléctrico. Inicialmente las capas de los shock tienen distribución espectral Gaussiana, con  $\Gamma$  de 5 y 10 respectivamente, como se muestran en la Figura 3.6, panel *a*). Durante la interacción de las capas, cuando se frenan, por aumento en la intensidad del campo eléctrico y llevan a la distribución a valores cercanos a cero, paneles desde *b*) hasta *e*). Mientras que en el panel *f*) la distribución de partículas se aleja del cero hacia valores mas altos. Se forma una vez mas picos en la distribución, ya que las capas adquieren velocidad porque son repelidas por el campo eléctrico. Si bien es leve, también se puede observar una cierta cantidad de partículas que adquiere valores mas altos de  $\gamma$ .

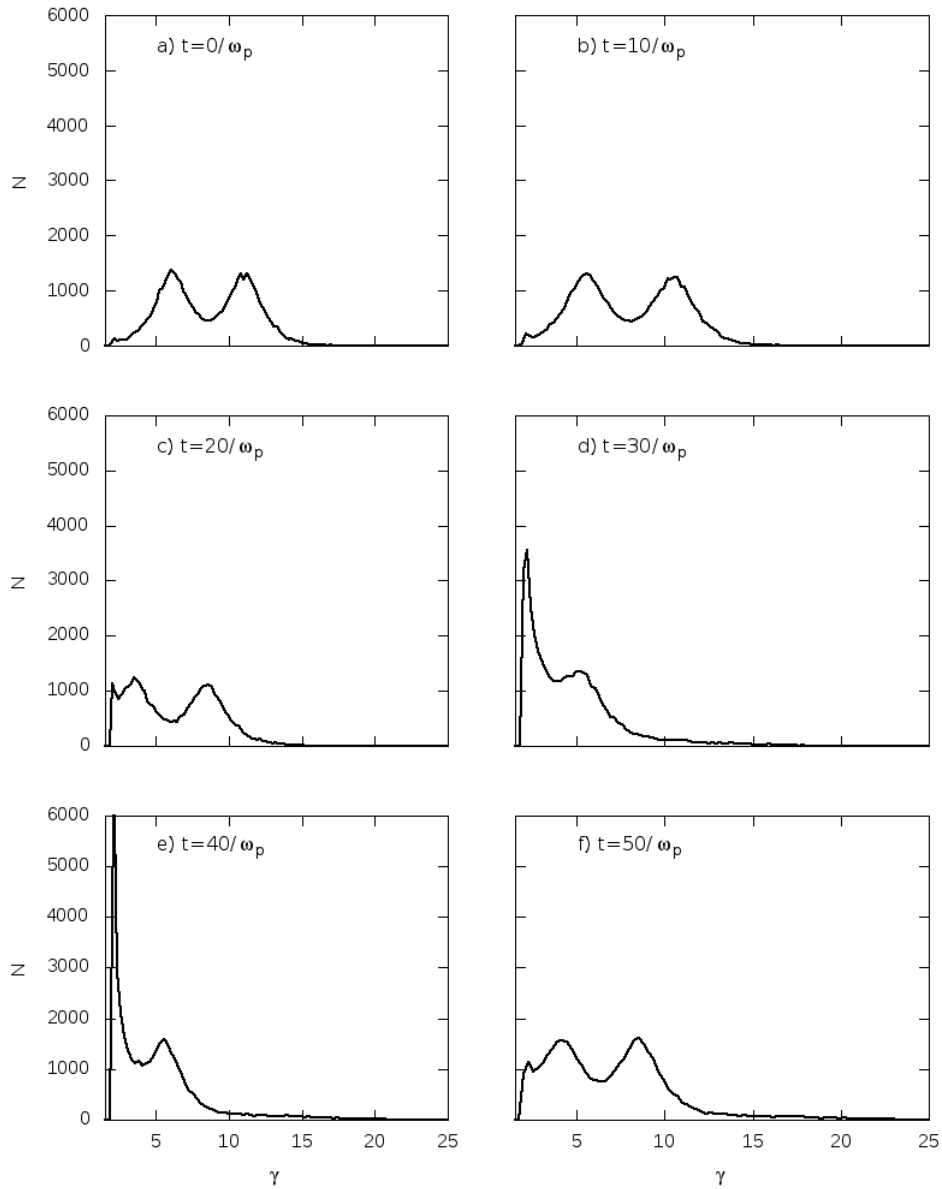


Figura 3.6: Evolución de la distribución espectral de las partículas en función del  $\Gamma$ . En el panel *a*) las capas en el shock tiene una distribución con  $\Gamma$  con 5 y 10, respectivamente. El incremento en la intensidad hace que las capas disminuyan su velocidad, acercándose a valores cercanos de cero, paneles *b*), *c*), *d*) y *e*). En el panel *f*), observamos que después de la interacción de las capas, el espectro de distribución de las partículas se aleja del cero.

### 3.4. Shock relativista electromagnético

En este caso preparamos un shock relativista inicialmente no magnetizado. Con diferencia al caso anterior, las variaciones del campo eléctrico inducen campo magnético de acuerdo a las ecuaciones de Maxwell. Como estamos tratando con un shock

unidimensional, pero con componentes tridimensionales para  $\mathbf{p}$ ,  $\mathbf{E}$  y  $\mathbf{B}$ . Básicamente daremos una distribución Maxwelliana relativista a la componente  $p_x$  de momento, de acuerdo al factor  $\Gamma$ , y dirección que lleve la capa o haz de partículas. Mientras que las componentes  $p_y$  y  $p_z$  del momento tendrán una distribución de Maxwell de acuerdo a la velocidad térmica de las partículas. La componente  $E_x$  del campo eléctrico es iniciado por la ecuación de Poisson, para el potencial periódico 2.1.6. Las componentes  $E_y$  y  $E_z$  inicialmente son cero. Con la condición inicial de que el shock no es magnetizado. Tenemos que la componente  $B_x = 0$  de acuerdo a la ecuación 1.23, transformándose en una constante del problema. Por lo cual nos centramos en la inducción de campo magnético perpendicular al eje  $x$ , es decir  $B_y$  y  $B_z$  [18].

Las capas inicialmente tiene distribución de Maxwell relativista con  $\Gamma \approx 16$ , para la capa izquierda que se desplaza hacia la derecha, mientras que  $\Gamma \approx 5$  para la capa derecha que se mueve hacia la izquierda. Las componentes térmicas  $p_y$  y  $p_z$  es elegida como  $p_{th} = 0,5c$ . Las capas forman frente en el centro del dominio computacional (ver Figura 3.7, panel *a*). El número de partículas usadas es de 40000, repartidas igualmente en cada capa. El número de ciclos es de 5000 con pasos temporales  $t = 0,01/\omega_p$ , resultando en un tiempo de  $t = 50/\omega_p$ . La longitud del dominio espacial es  $600c/\omega_p$ , discretizado a  $\Delta x = 1c/\omega_p$  [29]. Una vez mas consideramos condiciones de borde periódicas.

En la Figura 3.7 revela un comportamiento del shock electromagnético similar al presentado al shock netamente eléctrico. Las capas se encuentran en el centro del dominio computacional y dirigiéndose en direcciones contrarias. Las capas se frenan por el incremento del campo electromagnético, aunque las partículas cercanas al frente de interacción pueden aumentar o mantener su momento, paneles *b*), *c*), *d* y *e*). El campo electromagnético realizar un trabajo de retroceder a las capas, paneles *f*.

Damos paso al análisis del campo Electromagnético. En la Figura 3.8, en el panel *a*) podemos ver como varía la componente  $E_x$  del campo eléctrico en el tiempo de forma muy similar al caso puramente eléctrico. En el panel *b*), muestra la intensidad de las componentes  $E_y$ ,  $E_z$ , rojo y verde respectivamente, a lo largo del dominio computacional, al tiempo  $t = 50/\omega_p$ . Puesto que  $B_x = 0$  por ser una constante del problema. En el panel *c*), observamos las componentes del campo magnético  $B_y$ ,  $B_z$  azul y negro, al tiempo  $t = 50/\omega_p$ . Esto deja claro que las componentes perpendiculares al eje  $x$  del campo electromagnético, adquieren valores distintos de cero durante el shock.

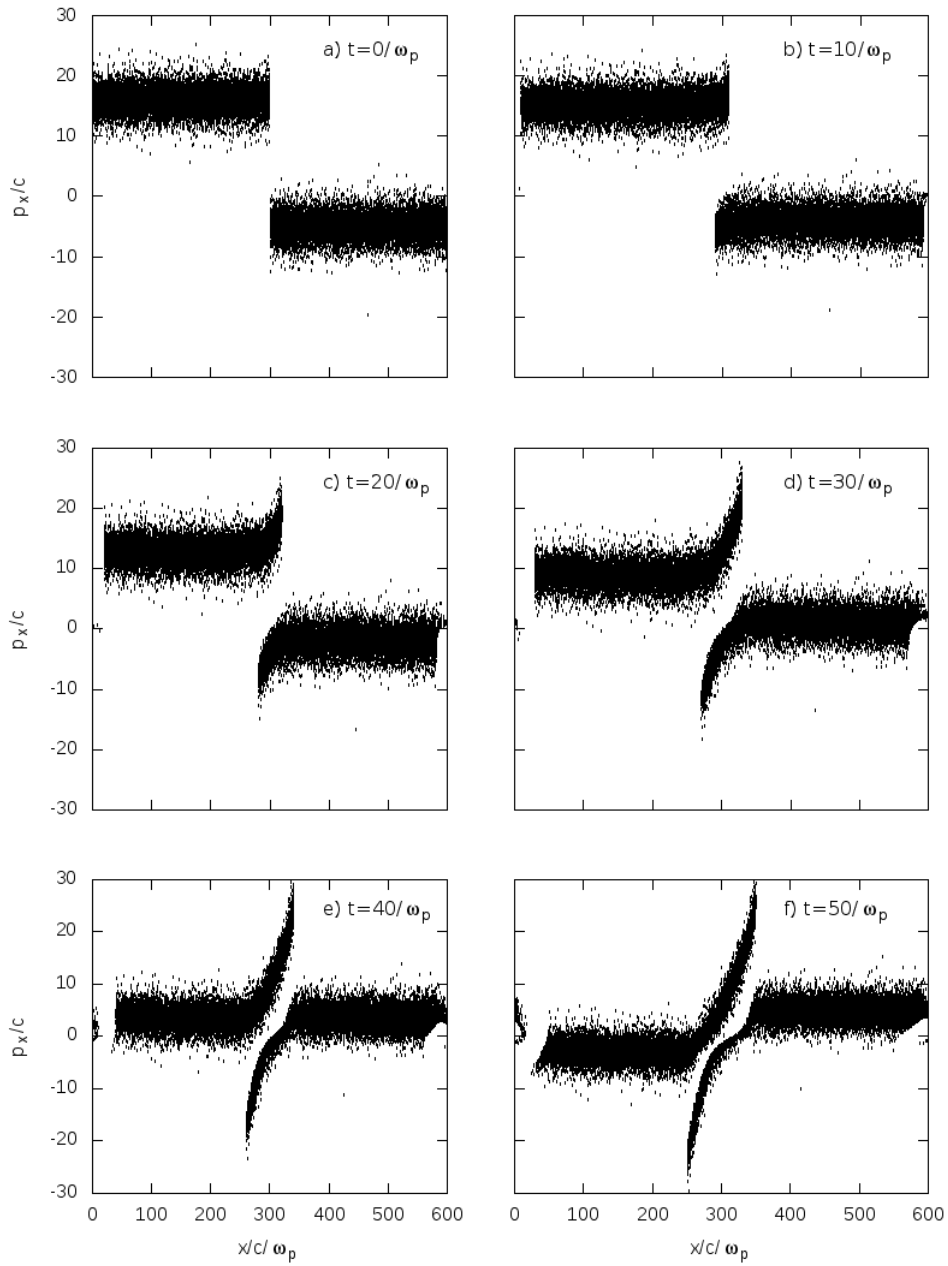


Figura 3.7: La estructura del shock, en este caso es muy similar al caso mostrado netamente eléctrico ver Figura 3.3. Inicialmente vemos a las capas interactuando con valores  $\Gamma \approx 16$  capa izquierda con dirección positiva y  $\Gamma \approx 5$  capa derecha con dirección negativa, panel *a*). Las capas se frenan, aunque ciertas partículas cercanas al frente de dirección mantienen o aumentan sus momentos, paneles *b*), *c*), *d*), y *e*). Finalmente se repelan retrocediendo, panel *f*).

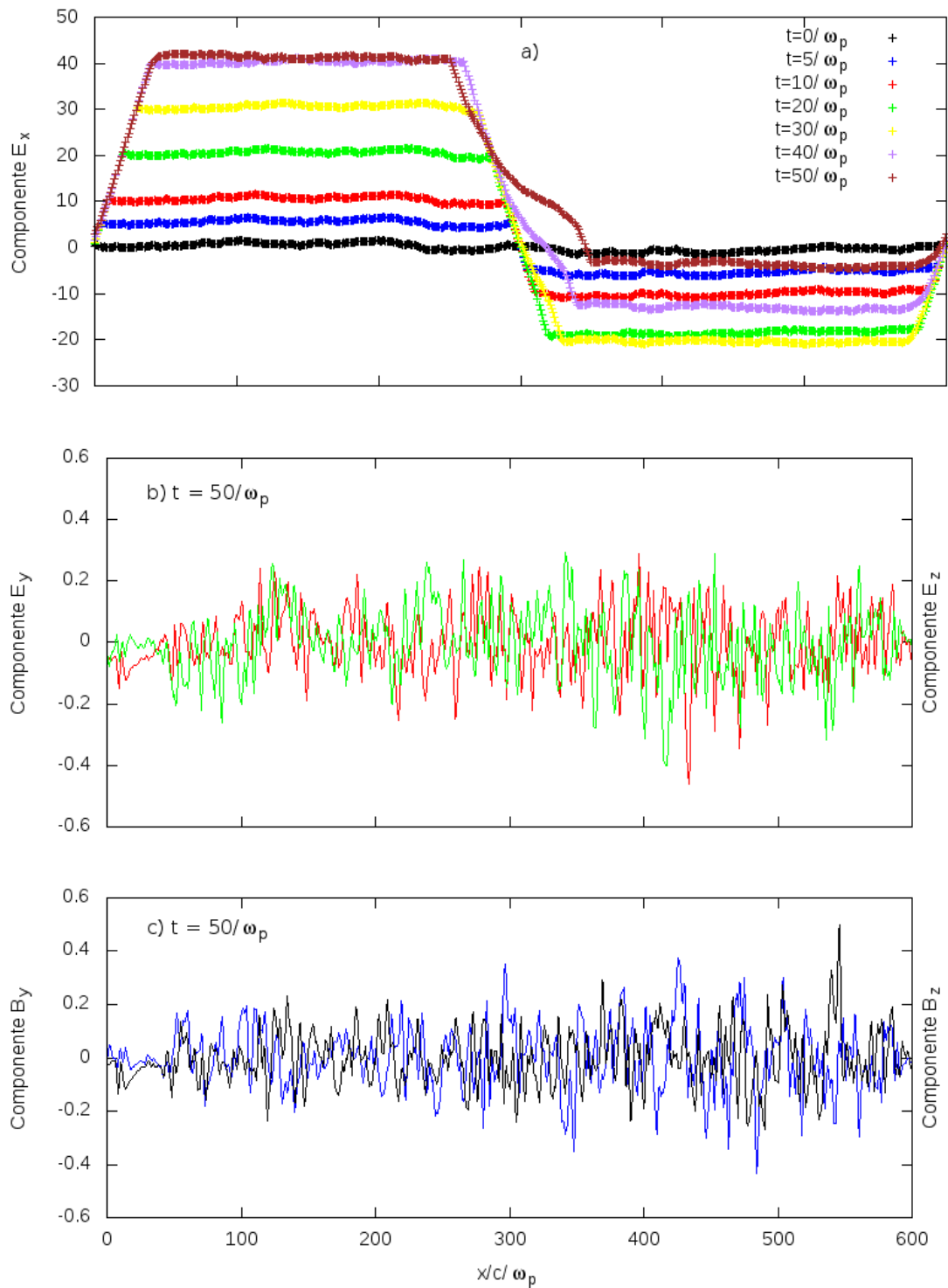


Figura 3.8: Variación del campo electromagnético en el shock electromagnético. En el panel a), muestra la variación de la componente  $E_x$  en el tiempo. En el panel b) las componentes  $E_y$  y  $E_z$ , rojo y verde respectivamente, al tiempo  $t = 50/\omega_p$ . En el panel c) las componentes  $B_y$  y  $B_z$ , azul y negro respectivamente.

Dirigimos nuestra atención a la conservación de energía para ver que cambio se

puede observar, con la presencia del campo magnético. La figura 3.9 muestra que el shock genera muy poca energía de campo magnético. Por lo cual no influye mucho en la aceleración de las partículas. En cambio el campo eléctrico sigue siendo el mayor dominante en la interacción de los shocks. Ya que el incremento de energía eléctrica, conlleva al decremento de energía cinética, frenando a las capas de partículas. De igual manera el campo eléctrico repele y retrocede a las capas.

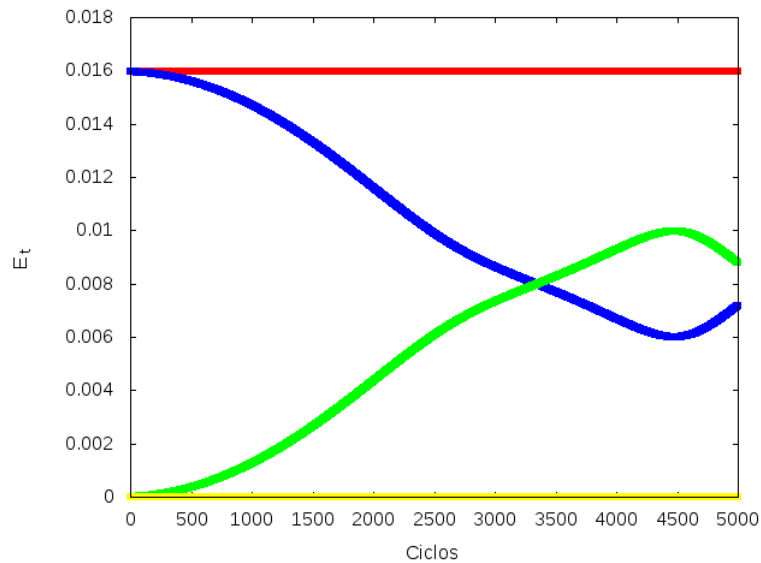


Figura 3.9: Conservación de la energía en un shock electromagnético, para 5000 ciclos durante la simulación. La línea roja muestra la energía total del shock. La azul la energía cinética. La verde, la energía eléctrica y la amarilla, la energía del campo magnético.

Finalmente, en la Figura 3.10, muestra la variación del espectro de las partículas, similar al caso del shock netamente eléctrico. Dos capas distribuidas con  $\Gamma = 16$  y  $\Gamma = 5$ . Esta gráfica muestra la forma de la dinámica de las partículas siendo frenadas. Desplazando su distribución espectral a valores cercanos de cero en  $\gamma$ , por un aumento en la intensidad del campo electromagnético (sobre todo la componente  $E_x$ ). Con el decaimiento del campo electromagnético desplazan la distribución de las partículas a valores alejados de cero en  $\gamma$ .

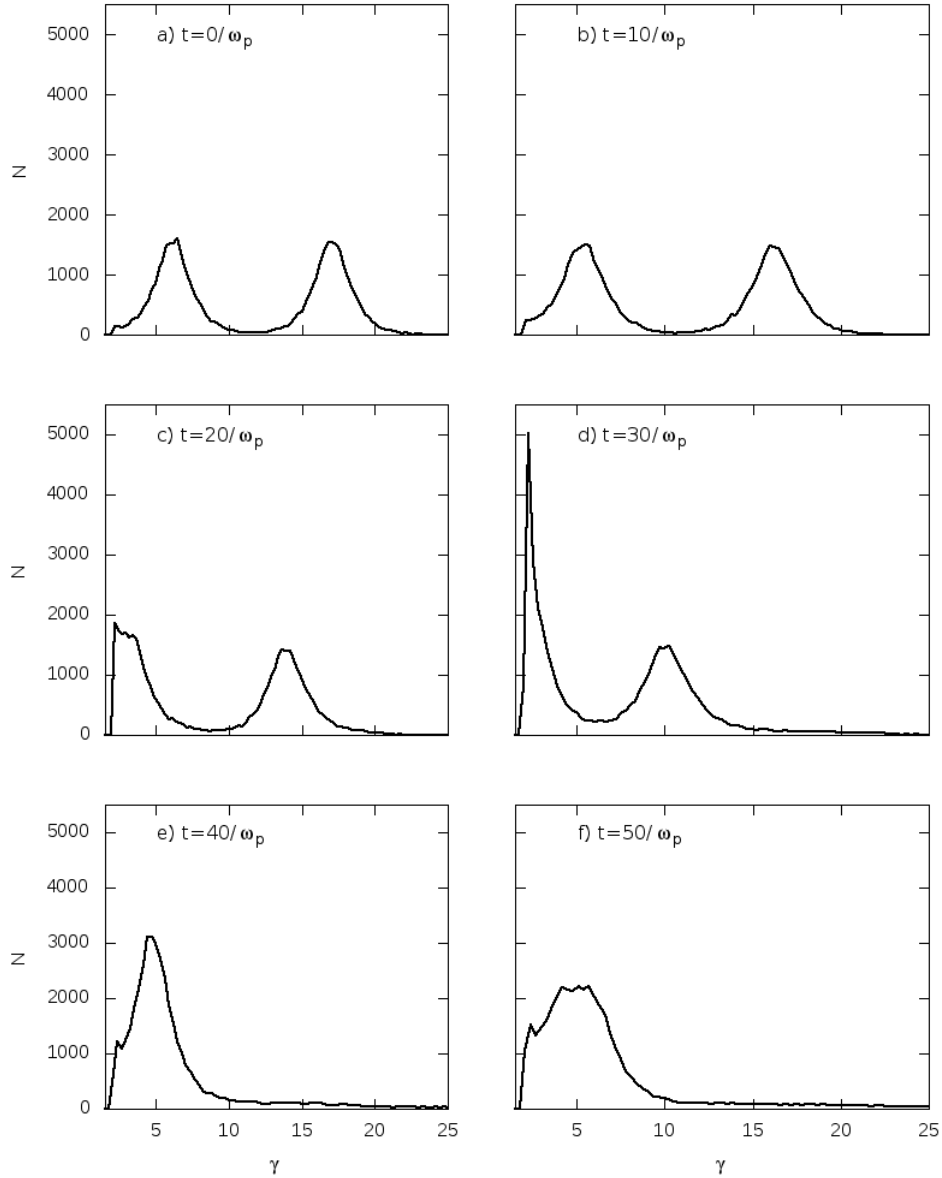


Figura 3.10: Variación del espectro de las partículas, durante el shock electromagnético. Muestra dos capas con  $\Gamma$  aproximadamente de 16 y 5 respectivamente (panel *a*). Las partículas se frenan, por un incremento del campo eléctrico. Desplazando su distribución a valores de  $\gamma$ , cercanos a cero. Cuando el campo eléctrico alcanza su máximo valor. Las partículas son repelidas en retroceso, aumentando su velocidad y alejándose de los valores de cero en  $\gamma$ .

# Capítulo 4

## Conclusiones

El modelo de fireball nos permite, describir la propagación de un material eyectado a velocidades relativistas, lejos su origen, como un plasma no colisional. Esto nos permite que sea posible la implementación de métodos computacionales, para estudiar el comportamiento de estos plasmas. Es por eso que hemos usado un código PIC relativista, basado en KEMPO1.

En este trabajo hemos logrado, mediante una simulación de este código PIC 1D3P, ver el comportamiento colectivo que tienen las partículas cargadas en un shock relativista. Para generar campos electromagnéticos entre capas que interactúan, y que se desplazan con distintos valores de  $\Gamma$ . Los resultados expuestos en las gráficas, muestran que el campo eléctrico es muy influyente en la presencia de shock relativistas. Pero no como esperábamos con el campo magnético generado, puesto que es débil. En los GRBs la presencia de campo magnético extremadamente fuerte es necesaria, para que las partículas aceleradas emitan radiación por sincrotrón o procesos de aceleración de Fermi I y II.

El considerar inicialmente un shock como un plasma no magnetizado, no generará a tan corto tiempo  $t \approx 1/\omega_p$ , las condiciones para que un plasma gane un campo magnético de gran magnitud, para que las partículas se aceleren y generen radiación sincrotrón. Entonces es necesario empezar el problema con plasma altamente magnetizado [2], para que las partículas adquieran grandes frecuencias de giro  $\omega > 10^{19} s^{-1}$ . Este sistema de estudio lleva consigo simulaciones tipo PIC con reconexión magnética y en más dimensiones espaciales.

A pesar de las ventajas mostradas con el algoritmo PIC utilizado en este trabajo, se demostró que las partículas pueden ser aceleradas únicamente con las fluctuaciones



del campo eléctrico, intercambiando energía cinética con las partículas.

# Apéndice A

## Código PIC en C++

particula.h este archivo de cabecera tiene en principio describir a la partícula como una estructura de clase en C++. Contiene la subrutina del algoritmo de Boris que permite integrar la ecuación de movimiento de las partículas.

```
#ifndef PARTICULA_H_INCLUDED
#define PARTICULA_H_INCLUDED

#include <cmath>
#include <cstdlib>

class particula
{
public:
    int j; // indice de posicion en la malla
    double q; // carga elemental de la macropartícula
    double l; // tamaño de la macropartícula
    double x; // posicion
    double px, py, pz; // momento de la partícula
    double gamma;

    particula();
    void valores(int, double, double);
    void iniciarPosicion(double, double);
    void iniciarMomentoPx(double);
    void iniciarMomentoPy(double);
    void iniciarMomentoPz(double);
    void posicion(double, double);
    void momento(double, double, double, double, double, double, double);
    int get_j(){return j = floor(x/l);}
    double get_q(){return q;}
    double get_l(){return l;}
    double get_x(){return x;}
};
```

```

double get_px(){return px;}
double get_py(){return py;}
double get_pz(){return pz;}
double Winterpolacion(double, double);
double get_gamma(){return gamma = sqrt(1.0 + px*px + py*py + pz*pz);}
};

//*****
//
//          I M P L E M E N T A C I O N
//
//*****

//constructor, que designa todos los valores iguales a cero en el momento de
//declara el tipo de particula.

particula::particula()
{
    j=0;
    q=l=x=pz=px=py=0.0;
}

//ingresa valores iniciales de indice de malla ip, la carga de la particula, y
//su tamanio.

void particula::valores(int ip, double e, double lo)
{
    j = ip;    //indice de la particula en la malla
    q = e;    //carga de la particula
    l = lo;    //tamanio de la particula
}

//inicia la posicion de una particula en el eje de las x aleatoriamente dentro
//del rango x_min hasta x_max

void particula::iniciarPosicion(double x_min, double x_max)
{
    x = x_min + (x_max - x_min)*double(rand())/double(RAND_MAX);
}

//inicia el momento lineal a partir de valores externo preobtenidos

void particula::iniciarMomentoPx(double pa)
{
    px = pa;
}

```

```

void partícula::iniciarMomentoPy(double pb)
{
    py = pb;
}

void partícula::iniciarMomentoPz(double pc)
{
    pz = pc;
}

//actualiza la posición de la partícula en el dominio [0 , L]

void partícula::posición(double dt, double L)
{
    x = x + px*dt/get_gamma();

    if(x < 0.0) x = x + L;
    if(x > L) x = x - L;
}

//Calcula el momento de una partícula al desplazarse por el campo electromagnético ,
//para ello se usa el algoritmo de Boris

void partícula::momento(double Ex, double Ey, double Ez, double Bx, double By,
                        double Bz, double dt)
{
    double tx, ty, tz, pxt, pyt, pzt, s;
    //para determinar los momentos usamos el algoritmo de Boris
    //obtenemos p-
    px = px + 0.5*q*Ex*dt;
    py = py + 0.5*q*Ey*dt;
    pz = pz + 0.5*q*Ez*dt;

    //obtenemos p'
    tx = 0.5*q*Bx*dt/get_gamma(); //get_gamma() da el de gamma^n
    ty = 0.5*q*By*dt/get_gamma();
    tz = 0.5*q*Bz*dt/get_gamma();
    pxt = px + py*tz - pz*ty;
    pyt = py + pz*tx - px*tz;
    pzt = pz + px*ty - py*tx;

    //obtenemos p+
    s = 2.0/(1.0 + tx*tx + ty*ty + tz*tz);
    px = px + s*(pyt*tz - pzt*ty);
    py = py + s*(pzt*tx - pxt*tz);
    pz = pz + s*(pxt*ty - py*tx);
}

```

```

    //obtenemos p(n+1/2)
    px = px + 0.5*q*Ex*dt;
    py = py + 0.5*q*Ey*dt;
    pz = pz + 0.5*q*Ez*dt;
}

//Esta funcion miembro devuelve la fraccion de la particula a la que pertenece cuando
//esta se divide entre dos celdas

double particula::Winterpolacion(double xi, double xp)
{
    double W;
    W = (xi-xp)/l;
    if (-1.0 < W && W < 0.0)
        return 1.0 + W;
    else if (0.0 <= W && W < 1.0)
        return 1.0 - W;
    else
        return 0.0;
}

#endif // PARTICULA_H_INCLUDED

```

procedimiento.h es un archivo de cabecera que posee las distintas rutinas que permiten calcular: las funciones de distribución de las partículas, la integración de los campos eléctrico y magnético, el cálculo de la densidad de carga, densidad de corriente, campo eléctrico promedio sobre las partículas, la densidad de energía eléctrica y magnética, y la energía total.

```

#ifndef PROCEDIMIENTOS_H_INCLUDED
#define PROCEDIMIENTOS_H_INCLUDED

#include <vector>
#include <fftw.h>
#include "particula.h"

using namespace std;

extern vector<particula> P;
extern vector<double> Ex, phi, rho, Ey, Ez, By, Bz, jx, jy, jz, E2, B2, K,
                    Ny, xgamma;

extern int N, I;
extern double L, dx, dt, Bx, no, q;
extern double Et, Kt, Uet, Umt;

```

```

// Procedimientos para calcular las variables del campo electromagnetico

void directa_fft(vector<double>, vector<double>&, vector<double>&, int);
void inversa_fft(vector<double>, vector<double>, vector<double> &, int);

void densidad();
void poisson1D(vector<double> &, vector<double>, double, int);

//*****
//
//                               F U N C I O N E S
//
//*****

//esta funcion realiza una distribucion de velocidades para una componente
//de acuerdo al factor gamma de la capa.
//p_s es la cantidad de movimiento mas probables de las particulas en la shell, para
//la distribucion Gaussiana relativista:
//
//                                $f(p) = 0.5 * \exp(-\sqrt{1.0 - (p - p_s)^2})$  .
//

double maxwell_distributionP1(double p_s)
{
    double fmax, p, pmax, pmin, f, frand;

    fmax = 0.5*( exp(- 1.0) ); //valor de la funcion para el momento mas probable
    pmax = 5.0*p_s;
    pmin = -5.0*p_s;
    p = pmax - (pmax-pmin)*(double(rand())/double(RAND_MAX) );

    //Aceptamos o rechazamos valores de p
    f = 0.5*exp( - sqrt(1.0 + (p - p_s)*(p - p_s) ) );
    frand = fmax*double(rand())/double(RAND_MAX);

    if (frand > f) return maxwell_distributionP1(p_s);
    else return p;
}

//esta funcion realiza una distribucion de velocidades en un direccion
//de acuerdo al factor gamma del jet.
double maxwell_distributionP2(double p_s)
{
    double fmax, p, pmax, pmin, f, frand;

    fmax = 0.5*( exp(-sqrt(1)) + exp(-sqrt(4.0*p_s*p_s) ) );
    pmax = 5.0*p_s;
    pmin = -5.0*p_s;

```

```

    p = pmin + (pmax - pmin)*(double(rand())/double(RAND_MAX) );

    //Aceptamos o rechazamos valores de p
    f = 0.5*(exp(-sqrt(1.0 + (p - p_s)*(p - p_s))/2.0)
            +exp(-sqrt(1.0 + (p + p_s)*(p + p_s))/2.0));
    frand = fmax*double(rand())/double(RAND_MAX);

    if (frand > f) return maxwell_distributionP2(p_s);
    else return p;
}

// Calcula la transformada rapida de Fourier de los coeficientes de f en
// los arreglos Fr (coeficientes reales) and Fi (coeficientes imaginarios)

void directa_fft(vector<double>f, vector<double> &Fr, vector<double> &Fi,
                int M)
{
    fftw_complex ff[M], FF[M];

    // cargando datos
    for (int i = 0; i < M; i++)
    {
        c_re (ff[i]) = f[i];
        c_im (ff[i]) = 0.0;
    }

    // calculando coeficientes
    fftw_plan p = fftw_create_plan(M, FFTW_FORWARD, FFTW_ESTIMATE);
    fftw_one(p, ff, FF);
    fftw_destroy_plan(p);

    // cargando datos
    for (int i = 0; i < M; i++)
    {
        Fr[i] = c_re (FF[i]);
        Fi[i] = c_im (FF[i]);
    }

    // normalizando datos
    for (int i = 0; i < M; i++)
    {
        Fr[i] = Fr[i]/double (M);
        Fi[i] = Fi[i]/double (M);
    }
}

// Calcula la transformada inversa de Fourier de los coeficientes de f a

```

```

// partir de los arreglos Fr and Fi.
void inversa_fft(vector<double> Fr, vector<double> Fi, vector<double> &f, int M)
{
    fftw_complex ff[M], FF[M];

    //cargando datos
    for(int i = 0; i < M; i++)
    {
        c_re(FF[i]) = Fr[i];
        c_im(FF[i]) = Fi[i];
    }

    // calculando coeficientes
    fftw_plan p = fftw_create_plan(M, FFTW_BACKWARD, FFTW_ESTIMATE);
    fftw_one(p, FF, ff);
    fftw_destroy_plan(p);

    // sobrecargando datos
    for(int i = 0; i < M; i++)
    {
        f[i] = c_re(ff[i]);
    }
}

// Densidad de la carga electrica presente en el sistema
void densidad()
{
    for (int i = 0; i < I+1; i++)
        rho[i] = 0.0;

    for (int j = 0; j < N; j++)
    {
        int i = P[j].get_j();           //indice de la particula
        double q, xj, xi, xil;
        q = P[j].get_q();               //carga de la particula
        xj = P[j].get_x();              //posicion de la particula
        xi = double(i)*dx;
        xil = double(i+1)*dx;

        rho[i] = rho[i] + q*P[j].Winterpolacion(xi, xj)/dx;
        if (i+1 == I)
            rho[0] = rho[0] + q*P[j].Winterpolacion(xil-L, xj-L)/dx;
        else
            rho[i+1] = rho[i+1] + q*P[j].Winterpolacion(xil, xj)/dx;
    }
    rho[I] = rho[0];                   //condicion de borde

    // normaliza la carga
    for(int i = 0; i <= I; i++)

```



```

    {
        rho[i] = rho[i]/(q*no) - 1;
    }
}

// Calcula el potencial electrico 1D a travez de la ecuacion de Poisson
// a partir de la rapida transformada de fourier.
//          Grad^2 phi = rho
void poisson1D(vector<double> &fi, vector<double> den, double L, int I)
{
    double kappa;
    kappa = 2*M_PI/L;
    vector<double> den_R, den_I, fi_R, fi_I;
    den_R.resize(I+1); den_I.resize(I+1);
    fi_R.resize(I+1); fi_I.resize(I+1);

    // El calculo del potencial phi se lo hace desde 0:I-1
    //*****
    // Se aplica la transformada de Fourier para la fuente
    directa_fft(den,den_R,den_I,I);

    // calculamos la transformada de fourier para fi

    fi_R[0] = fi_I[0] = 0.0;    //condiciones de borde

    for(int i = 1; i <= I/2; i++)
    {
        fi_R[i] = -den_R[i]/(double(i*i)*kappa*kappa);
        fi_I[i] = -den_I[i]/(double(i*i)*kappa*kappa);
    }

    // por el algoritmo de bits inverso
    for(int i = I/2; i < I; i++)
    {
        fi_R[i] = fi_R[I-i];
        fi_I[i] = -fi_I[I-i];    //-aqui
    }

    // se aplica la trasformada inversa de fourier
    inversa_fft(fi_R,fi_I,fi,I);
}

// Calcula el campo electrico en 1D para el PIC electrico
//Esta rutina calcula el campo en la mitad de la celdas de la malla
void componenteEx()
{
    for(int i = 0; i < I-1; i++)
    {
        Ex[i] = (phi[i] - phi[i+1])/dx;    //componente Ex(i+1/2)
    }
}

```

```

    }
    Ex[I-1] = (phi[I-1] - phi[0])/dx;    //componente Ex(N-1/2)
    Ex[I] = Ex[0];
}

// Realiza la conservacion de la carga y corrige el campo electrico
// en caso de ser solamente PIC electrico determina el campo elec-
// trico
void corregir_E()
{
    vector<double> rho_c, phi_c;
    rho_c.resize(I+1); phi_c.resize(I+1);

    // correccion de la carga
    for(int i = 1; i < I ; i++)
    {
        rho_c[i] = rho[i] + (Ex[i] - Ex[i-1])/dx;
    }
    rho_c[0] = rho[0] + (Ex[0] - Ex[I-1])/dx;
    rho_c[I] = rho[0];

    //resolviendo la ecuacion de Poisson
    poisson1D(phi_c,rho_c,L,I);

    for(int i = 0; i < I-1; i++)
    {
        Ex[i] = Ex[i] + (phi_c[i] - phi_c[i+1])/dx;
    }
    Ex[I-1] = Ex[I-1] + (phi_c[I-1] - phi_c[0])/dx;
    Ex[I] = Ex[0]; // condicion de borde
}

// Determina las nuevas posiciones
void nuevasPosiciones(double Dt)
{
    for(int j = 0; j < N; j++)
    {
        P[j].posicion(Dt, L);    //Dt es la longitud del paso del tiempo
    }
}

// Procedimientos para PIC EM
//
//*****

// Calcula el campo magnetico en el tiempo
void campo_magnetico(double Dt)
{
    for(int i = 0; i < I; i++)

```

```

{
    By[i+1] = By[i+1] + Dt*(Ez[i+1] - Ez[i])/dx;
    Bz[i] = Bz[i] - Dt*(Ey[i+1] - Ey[i])/dx;
}
By[0] = By[I];
Bz[I] = Bz[0];
}

// Calcula el campo electrico en el tiempo
void campo_Electrico()
{
    for(int i = 0; i < I; i++)
    {
        Ex[i] = Ex[i] - dt*jx[i];
        Ey[i+1] = Ey[i+1] - dt*(Bz[i+1] - Bz[i])/dx - dt*jy[i+1];
        Ez[i] = Ez[i] + dt*(By[i+1] - By[i])/dx - dt*jz[i];
    }
    Ex[I] = Ex[0]; //condiciones de borde en el tiempo
    Ez[I] = Ez[0];
    Ey[0] = Ey[I];
}

// Calcula las corrientes electricas en el plasma
void corrientes()
{
    for(int i = 0; i < I+1; i++)
    {
        jx[i] = 0.0;
        jy[i] = 0.0;
        jz[i] = 0.0;
    }

    for(int j = 0; j < N; j++)
    {
        int i = P[j].get_j();
        double q, xjn, xjn1, xi, xi1;
        q = P[j].get_q();
        xjn1 = P[j].get_x(); //x^(n+1)
        xjn = xjn1; // - P[j].get_px()*0.5*dt/P[j].get_gamma(); //x^(n+1/2)
        //xjn = xjn - P[j].get_px()*0.5*dt/P[j].get_gamma(); //x^n
        xi = i*dx; //xi
        xi1 = (i+1)*dx; //xi+1

        jx[i] = jx[i] + q*P[j].get_px()*P[j].Winterpolacion(xi, xjn)/P[j].get_gamma()/
            dx;
        jy[i] = jy[i] + q*P[j].get_py()*P[j].Winterpolacion(xi, xjn)/P[j].get_gamma()/
            dx;
        jz[i] = jz[i] + q*P[j].get_pz()*P[j].Winterpolacion(xi, xjn)/P[j].get_gamma()/

```

```

dx;

if(i+1 == I)
{
    jx[0] = jx[0] + q*P[j].get_px()*P[j].Winterpolacion(0.0,xjn-L)/P[j].
        get_gamma()/dx;
    jy[0] = jy[0] + q*P[j].get_py()*P[j].Winterpolacion(0.0,xjn-L)/P[j].
        get_gamma()/dx;
    jz[0] = jz[0] + q*P[j].get_pz()*P[j].Winterpolacion(0.0,xjn-L)/P[j].
        get_gamma()/dx;
}
else
{
    jx[i+1] = jx[i+1] + q*P[j].get_px()*P[j].Winterpolacion(xi1,xjn)/P[j].
        get_gamma()/dx;
    jy[i+1] = jy[i+1] + q*P[j].get_py()*P[j].Winterpolacion(xi1,xjn)/P[j].
        get_gamma()/dx;
    jz[i+1] = jz[i+1] + q*P[j].get_pz()*P[j].Winterpolacion(xi1,xjn)/P[j].
        get_gamma()/dx;
}
}
jx[I] = jx[0];
jy[I] = jy[0];
jz[I] = jz[0];

// se reubican las corrientes jx y jz en el centro de las celdas para evitar
// auto-fuerzas
for(int i = 0; i < I; i++)
{
    jx[i] = 0.5*(jx[i] + jx[i+1]);
    jy[i] = 0.5*(jy[i] + jy[i+1]); //prueba
    jz[i] = 0.5*(jz[i] + jz[i+1]);
}
jx[I] = jx[0];
jy[I] = jy[0]; //prueba
jz[I] = jz[0];

/*
//se cancela la componente uniforme:
double jxm, jym, jzm;
jxm=0.0; jym=0.0; jzm=0.0;
for(int i=1; i<I; i++)
{
    jxm = jxm + jx[i];
    //jym = jym + jy[i];
    //jzm = jzm + jz[i];
}

```

```

jxm = jxm/double(I);
//jym = jym/double(I);
//jzm = jzm/double(I);
for(int i=1; i < I+1; i++)
{
    jx[i] = jx[i]-jxm;
    //jy[i] = jy[i]-jym;
    //jz[i] = jz[i]-jzm;
}*/
}

void nuevosMomentums()
{
    double ex, ey, ez, by, bz;
    vector<double> Exm, Bym;
    Exm.resize(I+1); Bym.resize(I+1);

    for(int i = 1; i < I+1; i++) // se ubica el campo en los nodos
    {
        Exm[i] = 0.5*(Ex[i-1] + Ex[i]);
    }
    Exm[0] = Exm[I];

    for(int i = 0; i < I; i++) // se ubica el campo en el centro de
    { // de la celdas
        Bym[i] = 0.5*(By[i] + By[i+1]);
    }
    Bym[I] = Bym[0];

    for(int j = 0; j < N; j++)
    {
        int i = P[j].get_j();
        double xj, xi, xi1;
        xj = P[j].get_x();
        xi = double(i)*dx;
        xi1 = double(i+1)*dx;

        ex = Exm[i]*P[j].Winterpolacion(xi, xj)
            + Exm[i+1]*P[j].Winterpolacion(xi1, xj);
        ey = Ey[i]*P[j].Winterpolacion(xi, xj)
            + Ey[i+1]*P[j].Winterpolacion(xi1, xj);
        ez = Ez[i]*P[j].Winterpolacion(xi, xj)
            + Ez[i+1]*P[j].Winterpolacion(xi1, xj);
        by = Bym[i]*P[j].Winterpolacion(xi, xj)
            + Bym[i+1]*P[j].Winterpolacion(xi1, xj);
        bz = Bz[i]*P[j].Winterpolacion(xi, xj)
            + Bz[i+1]*P[j].Winterpolacion(xi1, xj);
    }
}

```

```

        P[j].momento(ex , ey , ez , Bx, by , bz , dt );
    }
}

void densidadE2B2()
{
    for(int i = 1; i < I+1; i++)
    {
        E2[i] = 0.5*(Ex[i]*Ex[i] + Ey[i]*Ey[i] + Ez[i]*Ez[i]);
        B2[i] = 0.5*(Bx*Bx + By[i]*By[i] + Bz[i]*Bz[i]);
    }
}

void energiaTotal()
{
    //Energia cinetica de las particulas
    Et=0.0;
    Kt = 0.0; Uet = 0.0; Umt = 0.0;
    for(int p = 1; p < N; p++)
    {
        K[p] = (P[p].get_gamma() - 1);
        Kt = Kt + K[p];
    }
    //densidad de energia cinetica
    Kt = Kt/N/L;

    for(int i = 0; i < I ; i++)
    {
        Uet = Uet + E2[i]*dx;
        Umt = Umt + B2[i]*dx;
    }
    //densidad de energia electromagnetica
    Uet = Uet/N/L;
    Umt = Umt/N/L;

    Et = Kt + Uet + Umt;
}

void espectro()
{
    int N, n;
    double dgamma, gamma0;
    vector<double> GAMMA;
    N = P.size();
    n = 200;
    gamma0 = 0.0;
    GAMMA.resize(N);

    for(int p = 0; p < N; p++)
    {

```

```

        GAMMA[p] = P[p].get_gamma();
    }
    //ordeno los valores de gamma
    sort(GAMMA.begin(), GAMMA.end());

    dgamma = (GAMMA[N-1] - gamma0)/double(100); //cout<<dgamma<<endl;

    Ny.resize(n); xgamma.resize(n);

    for(int i = 0; i < n; i++)
    {
        Ny[i] = 0;
        xgamma[i] = 0.0;
    }
    for(int i = 0; i < n; i++)
    {
        for(int p = 0; p < N; p++)
        {
            if((GAMMA[p] >= gamma0 + double(i)*dgamma) &&
                (GAMMA[p] < gamma0 + double((i+1)*dgamma))
            )
            {
                Ny[i] = Ny[i]+1;
            }
        }
        xgamma[i] = GAMMA[0] + (double(i)+0.5)*dgamma;
    }
}

#endif // PROCEDIMIENTOS_H_INCLUDED

```

Esta extensión de código inout.h posee los valores iniciales para los distintos casos de estudio en las simulaciones. Además, los resultados para los distintos casos en la simulaciones. Estos resultados son almacenados en un conjunto de archivos ".dat".

```

#ifndef INOUT_H_INCLUDED
#define INOUT_H_INCLUDED

#include <iostream>
#include <vector>
#include <cstdio>
#include <cstdlib>
#include <algorithm>
#include <cmath>
#include <fstream>
#include "particula.h"

```

```

vector<particula> P;
vector<double> phi, Ex, Ey, Ez, By, Bz; //variables del campo EM
vector<double> rho, jx, jy, jz; //densidad y corriente
vector<double> K, E2, B2, Ny, xgamma; //variables de medicion
int N, I, Tmax, tn; //# de particulas y # de nodos
//de la malla

double qm, q, L, dx, dt, wp, no, Bx, xmin, xmax, GAMMA, p_s;
double Et, Kt, Uet, Umt;
const char *fase[12], *campoE[12], *campoM[12], *phirho[12], *E2B2[12],
*spectro[12];

fstream f;

//*****
// inestabilidad de corrientes
//*****

void parametrosE1()
{
// constantes fisicas del sistema
// c = 1.0; velocidad de la luz
// me = 1.0; masa del electron mp = me
// qe = 1.0; carga del eletron
qm = -1.0; // relacion carga masa para el electron (positron)
wp = 1.0; // frecuencia del plasma

// varaiables de inicio
I = 500;
N = 40000;
Tmax = 5000;
tn = 500;
dx = 1.0; // consideramos un plasma ne- = ne+
dt = 0.1;
L = double(I)*dx; // dominio computacional
no = double(N)/L; // densidad relativa del sistema
q = wp*wp/no/qm; // la carga de la celda computacional
Bx = 0.0;
GAMMA = 10.0; //factor de Lorentz
xmin = 0.0;
xmax = L;
p_s = sqrt(GAMMA*GAMMA - 1.0);
// se redimensionan las variables de campo
P.resize(N); // particulas
K.resize(N);
phi.resize(I+1); rho.resize(I+1); //potencial y densidad
Ex.resize(I+1); Ey.resize(I+1); Ez.resize(I+1); //campo electrico
By.resize(I+1); Bz.resize(I+1); //campo magnetico
E2.resize(I+1); B2.resize(I+1); //energia del campo electromagnetico
}

```



```

void valores_inicialesE1()
{
    // iniciamos dos corrientes que fluyen con igual factor de Lorentz pero
    // con orientacion contraria

    for(int j = 0; j < 30000; j++)
    {
        P[j].valores(0,q,dx);
        P[j].iniciarPosicion(xmin,xmax);
        P[j].iniciarMomentoPx(maxwell_distributionP1(5));
    }

    for(int j = 30000; j < N; j++)
    {
        P[j].valores(0,q,dx);
        P[j].iniciarPosicion(xmin,xmax);
        P[j].iniciarMomentoPx(maxwell_distributionP1(-0.2));
    }

    for (int i = 0; i < I+1; i++)
    {
        phi[i] = 0.0; Ex[i] = 0.0;
    }
}

//*****
//                               shock con campo electrico
//*****

void parametrosE2()
{
    // constantes fisicas del sistema
    // c = 1.0;           velocidad de la luz
    // me = 1.0;         masa del electron   mp = me
    // qe = 1.0;         carga del eletron
    qm = -1.0;          // relacion carga masa para el electron (positron)
    wp = 1.0;          // frecuencia del plasma

    // variables de inicio
    I = 600;
    N = 40000;
    Tmax = 5000;
    tn = 500;
    dx = 1.0;          // consideramos un plasma ne- = ne+
    dt = 0.01;
    L = double(I)*dx; // dominio computacional
    no = double(N)/L; // densidad relativa del sistema
    q = wp*wp/no/qm;  // la carga de la celda computacional
}

```

```

    Bx = 0.0;
    GAMMA = 10.0;          //factor de Lorentz
    p_s = sqrt(GAMMA*GAMMA - 1.0);
    // se redimensionan las variables de campo
    P.resize(N);          // particulas
    K.resize(N);
    phi.resize(I+1); rho.resize(I+1);          //potencial y densidad
    Ex.resize(I+1); Ey.resize(I+1); Ez.resize(I+1);          //campo electrico
    By.resize(I+1); Bz.resize(I+1);          //campo magnetico
    E2.resize(I+1); B2.resize(I+1);          //energia del campo electromagnetico
}

void valores_inicialesE2()
{
    // iniciamos dos corrientes que fluyen con igual factor de Lorentz pero
    // con orientacion contraria

    for(int j = 0; j < N/2; j++)
    {
        P[j].valores(0,q,dx);
        P[j].iniciarPosicion(0.0,0.5*L);
        P[j].iniciarMomentoPx(maxwell_distributionP1(5));
    }

    for(int j = N/2; j < N; j++)
    {
        P[j].valores(0,q,dx);
        P[j].iniciarPosicion(0.5*L,L);
        P[j].iniciarMomentoPx(maxwell_distributionP1(-10));
    }

    for (int i = 0; i < I+1; i++)
    {
        phi[i] = 0.0; Ex[i] = 0.0;
    }
}

//*****
//
//          PIC EM 1
//*****

// Parametros de inicio para el sistema
void parametrosEM1()
{
    // constantes fisicas del sistema
    // c = 1.0;          velocidad de la luz
    // me = 1.0;          masa del electron    mp = me
    // qe = 1.0;          carga del eletron
    qm = -1.0;          // relacion carga masa para el electron (positron)
}

```

```

wp = 1.0;          // frecuencia del plasma

// variables de inicio
I = 600;
N = 40000;
Tmax = 5000;
tn = 500;
dx = 1.0;        // consideramos un plasma ne- = ne+
dt = 0.01;
L = double(I)*dx; // dominio computacional
no = double(N)/L; // densidad relativa del sistema
q = wp*wp/(no*qm); // la carga de la celda computacional
Bx = 0.0;
GAMMA = 5.0;     //factor de Lorentz
xmin = 0.0;
xmax = L;
p_s = sqrt(GAMMA*GAMMA - 1.0);

// se redimensionan las variables de campo
P.resize(N);          // particulas
K.resize(N);
phi.resize(I+1); rho.resize(I+1); //potencial y densidad
Ex.resize(I+1); Ey.resize(I+1); Ez.resize(I+1); //campo electrico
By.resize(I+1); Bz.resize(I+1); //campo magnetico
jx.resize(I+1); jy.resize(I+1); jz.resize(I+1); //corriente electrica
E2.resize(I+1); B2.resize(I+1); //densidad de energia
}

// Iniciar variables del campo
void valores_inicialesEM1()
{

// iniciamos dos corrientes que fluyen con igual factor de Lorentz pero
// con orientacion contraria

for(int j = 0; j < N/2; j++)
{
P[j].valores(0,q,dx);
P[j].iniciarPosicion(0,L/2.0);
P[j].iniciarMomentoPx(maxwell_distributionP1(16));
P[j].iniciarMomentoPy(0.5*cos(2.0*M_PI*double(rand())/double(RAND_MAX)) );
P[j].iniciarMomentoPz(0.5*sin(2.0*M_PI*double(rand())/double(RAND_MAX)) );
}

for(int j = N/2; j < N; j++)
{
P[j].valores(0,q,dx);
P[j].iniciarPosicion(L/2,L);
P[j].iniciarMomentoPx(maxwell_distributionP1(-5));
}
}

```

```

        P[j].iniciarMomentoPy(0.5*cos(2.0*M_PI*double(rand())/double(RAND_MAX)) );
        P[j].iniciarMomentoPz(0.5*sin(2.0*M_PI*double(rand())/double(RAND_MAX)) );
    }

    for (int i = 0; i < I+1; i++)
    {
        Ex[i] = 0.0; Ey[i] = 0.0; Ez[i] = 0.0; By[i] = 0.0; Bz[i] = 0.0;
        E2[i] = 0.0; B2[i] = 0.0;
    }
}

void nombre_archivos()
{
    fase[0] = "fase0.dat"; fase[1] = "fase1.dat";
    fase[2] = "fase2.dat"; fase[3] = "fase3.dat";
    fase[4] = "fase4.dat"; fase[5] = "fase5.dat";
    fase[6] = "fase6.dat"; fase[7] = "fase7.dat";
    fase[8] = "fase8.dat"; fase[9] = "fase9.dat";
    fase[10] = "fase10.dat";

    campoE[0] = "campoE0.dat"; campoE[1] = "campoE1.dat";
    campoE[2] = "campoE2.dat"; campoE[3] = "campoE3.dat";
    campoE[4] = "campoE4.dat"; campoE[5] = "campoE5.dat";
    campoE[6] = "campoE6.dat"; campoE[7] = "campoE7.dat";
    campoE[8] = "campoE8.dat"; campoE[9] = "campoE9.dat";
    campoE[10] = "campoE10.dat";

    campoM[0] = "campoM0.dat"; campoM[1] = "campoM1.dat";
    campoM[2] = "campoM2.dat"; campoM[3] = "campoM3.dat";
    campoM[4] = "campoM4.dat"; campoM[5] = "campoM5.dat";
    campoM[6] = "campoM6.dat"; campoM[7] = "campoM7.dat";
    campoM[8] = "campoM8.dat"; campoM[9] = "campoM9.dat";
    campoM[10] = "campoM10.dat";

    phirho[0] = "phirho0.dat"; phirho[1] = "phirho1.dat";
    phirho[2] = "phirho2.dat"; phirho[3] = "phirho3.dat";
    phirho[4] = "phirho4.dat"; phirho[5] = "phirho5.dat";
    phirho[6] = "phirho6.dat"; phirho[7] = "phirho7.dat";
    phirho[8] = "phirho8.dat"; phirho[9] = "phirho9.dat";
    phirho[10] = "phirho10.dat";

    E2B2[0] = "E2B20.dat"; E2B2[1] = "E2B21.dat";
    E2B2[2] = "E2B22.dat"; E2B2[3] = "E2B23.dat";
    E2B2[4] = "E2B24.dat"; E2B2[5] = "E2B25.dat";
    E2B2[6] = "E2B26.dat"; E2B2[7] = "E2B27.dat";
    E2B2[8] = "E2B28.dat"; E2B2[9] = "E2B29.dat";
    E2B2[10] = "E2B210.dat";

    espectro[0] = "espectro0.dat"; espectro[1] = "espectro1.dat";
    espectro[2] = "espectro2.dat"; espectro[3] = "espectro3.dat";
}

```

```

    spectro[4] = "spectro4.dat"; spectro[5] = "spectro5.dat";
    spectro[6] = "spectro6.dat"; spectro[7] = "spectro7.dat";
    spectro[8] = "spectro8.dat"; spectro[9] = "spectro9.dat";
    spectro[10] = "spectro10.dat";
}

void escribir_fase(const char archivo[])
{
    ofstream f;
    f.open(archivo, ios::out);
    f<<"# " <<"x" <<" " <<"px" <<" " <<"py" <<" " <<"pz" <<endl;
    for (int j = 0; j < N; j++)
    {
        f<<" " <<P[j].get_x() <<" " <<P[j].get_px() <<" " <<P[j].get_py()
            <<" " <<P[j].get_pz() <<endl;
    }
    f.close();
}

void escribir_campoE(const char archivo[])
{
    ofstream f;
    f.open(archivo, ios::out);
    f<<"# " <<"x" <<" " <<"Ex" <<" " <<"Ey" <<" " <<"Ez" <<endl;
    for (int i = 0; i < I+1; i++)
    {
        f<<" " <<i*dx <<" " <<Ex[i] <<" " <<Ey[i] <<" " <<Ez[i] <<endl;
    }
    f.close();
}

void escribir_campoM(const char archivo[])
{
    ofstream f;
    f.open(archivo, ios::out);
    f<<"# " <<"x" <<" " <<"By" <<" " <<"Bz" <<endl;
    for (int i = 0; i < I+1; i++)
    {
        f<<" " <<i*dx <<" " <<By[i] <<" " <<Bz[i] <<endl;
    }
    f.close();
}

void escribir_PhiRho(const char archivo[])
{
    ofstream f;
    f.open(archivo, ios::out);
    f<<"# " <<"x" <<" " <<"phi" <<" " <<"rho" <<endl;
    for (int i = 0; i < I+1; i++)

```

```

    {
        f<<"\n"<<i*dx<<"\n"<<phi [ i]<<"\n"<<rho [ i]<<endl;
    }
    f.close ();
}

void escribir_E2B2(const char archivo [])
{
    ofstream f;
    f.open(archivo , ios::out);
    f<<"#\n"<<"x"<<"\n"<<"E2"<<"\n"<<"B2"<<endl;
    for (int i = 0; i < I+1; i++)
    {
        f<<"\n"<<i*dx<<"\n"<<E2 [ i]<<"\n"<<B2 [ i]<<endl;
    }
    f.close ();
}

void escribir_espectro(const char archivo [])
{
    ofstream f;
    int n=xgamma.size ();
    f.open(archivo , ios::out);
    f<<"#\n"<<"xgamma"<<"\n"<<"Ny"<<endl;
    for(int i = 0; i < n; i++)
    {
        f<<"\n"<<xgamma [ i]<<"\n"<<Ny [ i]<<endl;
    }
    f.close ();
}

void escribir_Et(int n)
{
    f<< n <<"\n"<<Et<<"\n"<<Kt<<"\n"<<Uet<<"\n"<<Umt<<endl;
    if (n==Tmax)
        f.close ();
}
#endif // PROCEDIMIENTOS_H_INCLUDED

```

La siguiente rutina contiene la función principal "main" del código y con ellos realiza los distintos llamados de las subrutinas de acuerdo a los pasos del los Pseudo códigos (ver sección 2.2).

```

#include <iostream>
#include <vector>
#include <cstdio>
#include <cstdlib>

```

```

#include <algorithm>
#include <cmath>
#include <fstream>
#include "particula.h"
#include "procedimientos.h"
#include "inout.h"

using namespace std;

int main()
{
    srand(time(NULL));
    int input1, input2;
    f.open("energia.total", ios::out);

    cout<<"codigo_PIC:_\n";
    cout<<"1_PIC_Electrico_\n";
    cout<<"otros_PIC_Electromagnetico_\n";
    cout<<"Elija:_"; cin>>input1;

    switch (input1) {
        case 1:
            {
                cout<<"1_Inestabilidad_de_dos_corrientes._\n";
                cout<<"Otros_shock_con_campo_electrico._\n";
                cout<<"Elija:_"; cin>>input2;

                nombre_archivos();
                switch (input2) {
                    case 1:
                        {
                            parametrosE1();
                            valores_inicialesE1();
                        }
                        break;
                    default:
                        {
                            parametrosE2();
                            valores_inicialesE2();
                        }
                        break;
                }
            }
        int i = 0;
        int n = 1;
        densidad();
        poisson1D(phi, rho, L, I);
        componenteEx();
        //corregir_E();
        espectro();
    }
}

```

```

densidadE2B2();
energiaTotal();
escribir_fase(fase[i]);
escribir_PhiRho(phirho[i]);
escribir_campoE(campoE[i]);
escribir_espectro(spectro[i]);
escribir_E2B2(E2B2[i]);
escribir_Et(n);
while(n <= Tmax)
{
    nuevosMomentums();
    nuevasPosiciones(0.5*dt);
    nuevasPosiciones(0.5*dt);
    densidad();
    poisson1D(phi,rho,L,I);
    //coregir_E();
    componenteEx();
    densidadE2B2();
    energiaTotal();
    if(n % tn == 0)
    {

        i++; cout<<n<<endl;
        espectro();
        escribir_fase(fase[i]);
        escribir_PhiRho(phirho[i]);
        escribir_campoE(campoE[i]);
        escribir_espectro(spectro[i]);
        escribir_E2B2(E2B2[i]);

    }
    escribir_Et(n);
    n++;
}
}
break;
default:
{
    cout<<"shock_EM_frontal.\n";
    nombre_archivos();
    parametrosEM1();
    valores_inicialesEM1();
    int n = 1;
    int i = 0;
    densidad();
    corrientes();
    campo_Electrico();
    corregir_E();
    densidadE2B2();
    energiaTotal();
}
}

```



```

espectro ();
escribir_fase ( fase [ i ] );
escribir_campoE ( campoE [ i ] );
escribir_campoM ( campoM [ i ] );
escribir_PhiRho ( phirho [ i ] );
escribir_espectro ( espectro [ i ] );
escribir_E2B2 ( E2B2 [ i ] );
escribir_Et ( n );
while ( n <= Tmax )
{
    campo_magnetico ( 0.5 * dt );
    nuevosMomentums ();
    nuevasPosiciones ( 0.5 * dt );
    corrientes ();
    nuevasPosiciones ( 0.5 * dt );
    campo_magnetico ( 0.5 * dt );
    campo_Electrico ();

    corregir_E ();

    densidad ();
    densidadE2B2 ();
    energiaTotal ();
    if ( n % tn == 0 )
    {
        i++; cout << n << endl;
        densidadE2B2 ();
        espectro ();
        escribir_fase ( fase [ i ] );
        escribir_campoE ( campoE [ i ] );
        escribir_campoM ( campoM [ i ] );
        escribir_PhiRho ( phirho [ i ] );
        escribir_E2B2 ( E2B2 [ i ] );
        escribir_espectro ( espectro [ i ] );
    }
    escribir_Et ( n );
    n++;
}
break;
}
}

```

# Bibliografía

- [1] ABBOTT, B. P., ABBOTT, R., ABBOTT, T. D., ACERNESE, F., ACKLEY, K., ADAMS, C., ... & AFFELDT, C. (2017). *Gravitational waves and gamma-rays from a binary neutron star merger: GW170817 and GRB 170817A*. *The Astrophysical Journal Letters*, 848(2), L13.
- [2] BENIAMINI, P., & GRANOT, J. (2016). *Properties of GRB light curves from magnetic reconnection*. *Monthly Notices of the Royal Astronomical Society*, 459(4), 3635-3658.
- [3] BIALYNICKI-BIRULA, I., & BIALYNICKA-BIRULA, Z. (1985). *Solutions of Vlasov-Maxwell equations for a magnetically confined relativistic cold plasma*. *Physica A: Statistical Mechanics and its Applications*, 133(1-2), 228-246.
- [4] BIRDSALL, C. K., & LANGDON, A. B. (2004). *Plasma physics via computer simulation*. CRC press.
- [5] BLOOM, J. S. (2011). *What are gamma-ray bursts?* Princeton University Press.
- [6] BOYD, T. J. M., & SANDERSON, J. J. (2003). *The physics of plasmas*. Cambridge University Press.
- [7] FITZPATRICK, R. (2006). *Computational physics*. Lecture notes, University of Texas at Austin.
- [8] GERMASCHEWSKI, K., FOX, W., ABBOTT, S., AHMADI, N., MAYNARD, K., WANG, L., & BHATTACHARJEE, A. (2013). *The Plasma Simulation Code: A modern particle-in-cell code with load-balancing and GPU support*. arXiv preprint arXiv:1310.7866.

- [9] GERMASCHEWSKI, K., FOX, W., ABBOTT, S., AHMADI, N., MAYNARD, K., WANG, L., & BHATTACHARJEE, A. (2016). *The Plasma Simulation Code: A modern particle-in-cell code with patch-based load-balancing*. Journal of Computational Physics, 318, 305-326.
- [10] GOODMAN, J. (1986). *Are gamma-ray bursts optically thick?*. The Astrophysical Journal, 308, L47-L50.
- [11] KUMAR, P., & ZHANG, B. (2015). *The physics of gamma-ray bursts and relativistic jets*. Physics Reports, 561, 1-109.
- [12] LEVAN, A., CROWTHER, P., DE GRIJS, R., LANGER, N., XU, D., & YOON, S. C. (2016). *Gamma-ray burst progenitors*. Space Science Reviews, 202(1-4), 33-78.
- [13] LAPENTA, G., BRACKBILL, J. U., & RICCI, P. (2006). *Kinetic approach to microscopic-macroscopic coupling in space and laboratory plasmas a. Physics of plasmas*, 13(5), 055904.
- [14] MÉSZÁROS, P. (2006). *Gamma-ray bursts*. Reports on Progress in Physics, 69(8), 2259.
- [15] MESZAROS, P. (2010). *The high energy Universe*. Ultra-High Energy Events in Astrophysics and Cosmology.
- [16] NISHIKAWA, K. I., NIEMIEC, J., SOL, H., MEDVEDEV, M., ZHANG, B., NORDLUND, A., FISHMAN, G. J. (2008, December). *New Relativistic Particle-In-Cell Simulation Studies of Prompt and Early Afterglows from GRBs*. In AIP Conference Proceedings (Vol. 1085, No. 1, pp. 589-593). AIP.
- [17] NOGUCHI, K., TRONCI, C., ZUCCARO, G., & LAPENTA, G. *Fully relativistic implicit particle-in-cell simulation scheme*.
- [18] OMURA, Y. (2007). *One-dimensional electromagnetic particle code: KEMPO1*. Advanced Methods for Space Simulations, edited by H. Usui and Y. Omura (Terra Sci., Tokyo, 2007), 1-21.
- [19] PACZYNSKI, B. (1986). *Gamma-ray bursters at cosmological distances*. The Astrophysical Journal, 308, L43-L46.

- [20] PEÉR, A. (2013). *Radiative Mechanisms in GRB prompt emission*. European Astronomical Society Publications Series, 61, 105-113.
- [21] PIRAN, T. (2005). *The physics of gamma-ray bursts*. Reviews of Modern Physics, 76(4), 1143.
- [22] REES, M. J., & MÉSZÁROS, P. (1992). *Relativistic fireballs: energy conversion and time-scales*. Monthly Notices of the Royal Astronomical Society, 258(1), 41P-43P.
- [23] REES, M. J., & MÉSZÁROS, P. (1994). *Unsteady outflow models for cosmological gamma-ray bursts*. arXiv preprint astro-ph/9404038.
- [24] SHEBALIN, J. V. *A Spectral Algorithm for Solving the Relativistic Vlasov-Maxwell Equations*. Lyndon B. Johnson Space Center, S-872.
- [25] SIMOES, F. J. R., ALVES, M. V., & GAELZER, R. (2010). *One-dimensional electromagnetic simulation of multiple electron beams propagating in space plasmas*. Journal of Geophysical Research: Space Physics, 115(A6).
- [26] UHM, Z. L., & BELOBORODOV, A. M. (2007). *On the mechanism of gamma-ray burst afterglows*. The Astrophysical Journal Letters, 665(2), L93.
- [27] VALENTINI, F., TRÁVNÍČEK, P., CALIFANO, F., HELLINGER, P., & MANGENEY, A. (2007). *A hybrid-Vlasov model based on the current advance method for the simulation of collisionless magnetized plasma*. Journal of Computational Physics, 225(1), 753-770.
- [28] VEDRENNE, G., & ATTEIA, J. L. (2009). *Gamma-ray bursts: The brightest explosions in the universe*. Springer Science & Business Media.
- [29] WAXMAN, E. (2006). *Gamma-ray bursts and collisionless shocks*. Plasma Physics and Controlled Fusion, 48(12B), B137.