

# Universidad Tecnológica de Pereira

## Ingeniería Física



Universidad  
Tecnológica  
de Pereira

Facultad de Ingenierías

Grupo de investigación GPLA (Plasma láser y  
aplicaciones)

---

Implementación de una interfaz gráfica para la  
simulación de plasmas electrostáticos y su  
aplicación en el estudio de ondas solitarias  
observadas en la región aural y otras regiones de  
la magnetosfera

---

Juan Camilo Osorio Tascón

Oscar Julián Gómez Tabares

Tesis de grado para optar al título de:  
Ingeniero Físico

Director de grado:

Ph.D. Henry Riascos Landázuri

**PEREIRA 2021**



# Índice general

<b>Índice general</b>	<b>3</b>
<b>Índice de tablas</b>	<b>5</b>
<b>Índice de figuras</b>	<b>6</b>
<b>Resumen y Agradecimientos</b>	<b>8</b>
<b>1. Introducción</b>	<b>9</b>
1.1. Objetivos . . . . .	10
1.1.1. Objetivos Generales . . . . .	10
1.1.2. Objetivos específicos . . . . .	10
<b>2. Modelo Teórico</b>	<b>11</b>
2.1. Magnetosfera . . . . .	11
2.2. Teoría Cinética del Plasma . . . . .	13
2.3. Longitud de Debye . . . . .	14
2.4. Frecuencia del Plasma . . . . .	14
2.5. Velocidad Térmica . . . . .	15
2.6. Amortiguamiento de Landau . . . . .	15
2.7. Inestabilidades en el Plasma . . . . .	16
2.7.1. Inestabilidad Two-Stream . . . . .	16
2.7.2. Inestabilidad Bump-of-Tail . . . . .	17
2.7.3. Inestabilidad Beam-Plasma . . . . .	17
2.8. Teoría de Fluido del Plasma . . . . .	18
2.8.1. Ecuación de Continuidad . . . . .	18
2.8.2. Ecuación de Momento . . . . .	18
2.8.3. Ecuación de estado . . . . .	19
2.9. Ondas Solitarias y la Ecuación Korteweg-de Vries . . . . .	19
<b>3. Modelo Computacional</b>	<b>25</b>
3.1. Método Particle-in-Cell . . . . .	25
3.1.1. Método de ponderación . . . . .	25
3.1.2. Integración de las ecuaciones de campo . . . . .	26
3.1.3. Integración de las ecuaciones de movimiento . . . . .	27
3.1.4. Condiciones de frontera . . . . .	28
3.1.5. Normalización . . . . .	28
3.1.6. Condiciones iniciales . . . . .	29

<b>4. Interfaz Gráfica PEWAV</b>	<b>30</b>
4.1. Librerías Usadas de Python . . . . .	32
4.1.1. Numpy . . . . .	32
4.1.2. Pandas . . . . .	32
4.1.3. Tkinter . . . . .	32
4.1.4. Matplotlib . . . . .	32
4.1.5. H5py . . . . .	33
4.2. Adaptación de los códigos OneGPLA . . . . .	33
4.3. Composición . . . . .	34
4.3.1. ROneGPLAfunctions . . . . .	34
4.3.2. ROneGPLA_AIO . . . . .	34
4.3.3. PEWAV . . . . .	37
4.4. Manual de Usuario . . . . .	37
4.4.1. Introducción . . . . .	37
4.4.2. Ejecución de la Interfaz: Introducción al Entorno . . . . .	37
4.4.3. Elementos de la Interfaz : Condiciones Iniciales . . . . .	38
4.4.4. Elementos de la Interfaz: Gráficas . . . . .	40
<b>5. Simulaciones de plasmas electrostáticos en PEWAV</b>	<b>42</b>
5.1. Parámetros . . . . .	42
<b>6. Resultados</b>	<b>45</b>
6.1. Dos especies de electrones . . . . .	45
6.1.1. Inestabilidad two-stream . . . . .	45
6.1.2. Inestabilidad bump-on-tail . . . . .	46
6.1.3. Inestabilidad warm two-stream . . . . .	46
6.1.4. Gráficas 2S . . . . .	48
6.2. Tres especies de electrones . . . . .	55
6.2.1. Inestabilidad two-stream . . . . .	55
6.2.2. Inestabilidad weak-beam . . . . .	55
6.2.3. Inestabilidad bump-on-tail . . . . .	56
6.2.4. Gráficas 3S . . . . .	57
<b>7. Conclusiones, Discusiones y Trabajo Futuro</b>	<b>69</b>
7.1. Conclusiones y Discusiones . . . . .	69
7.2. Trabajo Futuro . . . . .	70
<b>8. Anexos</b>	<b>72</b>
<b>Bibliografía</b>	<b>73</b>

# Índice de tablas

4.1. Resumen funciones de <i>ROneGPLAfunctions</i> . . . . .	35
4.2. Resumen atributos de la clase <i>seed</i> . . . . .	36
4.3. Resumen atributos de la clase <i>main</i> . . . . .	36
5.1. Parámetros sistema generales . . . . .	43
5.2. Parámetros sistema tres especies . . . . .	43
5.3. Parámetros velocidades dos especies . . . . .	43
5.4. Número partículas tres especies . . . . .	43
5.5. Parámetros sistema tres especies . . . . .	43
5.6. Parámetros velocidades tres especies . . . . .	44
5.7. Número partículas tres especies . . . . .	44

# Índice de figuras

2.1. Ilustración magnetosfera. . . . .	11
2.2. Esquema entorno terrestre de la magnetosfera. . . . .	12
2.3. Longitud de Debye . . . . .	14
2.4. Amortiguamiento de Landau en el espacio de fase de la inestabilidad Two-Stream (resultado propio obtenido en el capítulo 6). . . . .	15
2.5. Esquema ejemplo función de distribución two-stream. . . . .	16
2.6. Esquema ejemplo función de distribución two-stream. . . . .	17
2.7. Esquema Inestabilidad Bump-of-Tail. . . . .	17
2.8. Esquema ejemplo inestabilidad Beam-Plasma. . . . .	18
2.9. Estructura solitaria encontrada en la literatura [1]. . . . .	23
2.10. Onda electrostática simulada de tipo compresiva. . . . .	24
2.11. Onda electrostática simulada de tipo rarefactiva. . . . .	24
3.1. Ciclo PIC . . . . .	26
3.2. Ponderación unidimensional . . . . .	26
3.3. Esquema de integración del método leapfrog. [2] . . . . .	28
4.1. Interfaz gráfica anterior de OneGPLA [3]. . . . .	31
4.2. Ejemplo de arrays en numpy. . . . .	32
4.3. Diagrama de flujos simplificado de OneGPLA . . . . .	33
4.4. Datos Partículas . . . . .	38
4.5. Entorno interfaz gráfica . . . . .	39
4.6. Apartado: Sistemas-Condicionales Iniciales . . . . .	39
4.8. Gráficas. . . . .	41
4.9. Ejemplo del aspecto de la interfaz en funcionamiento después de haber cargado los datos simulados. . . . .	41
6.1. Función de distribución $f(v)$ y gráfica del espacio de fase para Run 1A. . . . .	48
6.2. Campo eléctrico $E$ , potencial eléctrico $\phi$ y densidad de carga $\rho$ para Run 1A. . . . .	49
6.3. Función de distribución $f(v)$ y gráfica del espacio de fase para Run 2A. . . . .	50
6.4. Campo eléctrico $E$ , potencial eléctrico $\phi$ y densidad de carga $\rho$ para Run 2A. . . . .	51
6.5. Función de distribución $f(v)$ y gráfica del espacio de fase para Run 3A. . . . .	52
6.6. Campo eléctrico $E$ , potencial eléctrico $\phi$ y densidad de carga $\rho$ para Run 3A. . . . .	53
6.7. Energía total $E_T$ , cinética $K$ y potencial $U$ de las simulaciones del circuito Run A. . . . .	54
6.8. Función de distribución $f(v)$ y gráfica del espacio de fase para Run 1B. . . . .	57
6.9. Campo eléctrico $E$ , potencial eléctrico $\phi$ y densidad de carga $\rho$ para Run 1B. . . . .	58
6.10. Función de distribución $f(v)$ y gráfica del espacio de fase para Run 2B. . . . .	59
6.11. Campo eléctrico $E$ , potencial eléctrico $\phi$ y densidad de carga $\rho$ para Run 2B. . . . .	60
6.12. Función de distribución $f(v)$ y gráfica del espacio de fase para Run 3B. . . . .	61
6.13. Campo eléctrico $E$ , potencial eléctrico $\phi$ y densidad de carga $\rho$ para Run 3B. . . . .	62
6.14. Función de distribución $f(v)$ y gráfica del espacio de fase para Run 4B. . . . .	63

6.15. Campo eléctrico $E$ , potencial eléctrico $\phi$ y densidad de carga $\rho$ para Run 4B. . .	64
6.16. Función de distribución $f(v)$ y gráfica del espacio de fase para Run 5B. . . . .	65
6.17. Campo eléctrico $E$ , potencial eléctrico $\phi$ y densidad de carga $\rho$ para Run 5B. . .	66
6.18. Energía total $E_T$ , cinética $K$ y potencial $U$ de las simulaciones del circuito Run 1-3B. . . . .	67
6.19. Continuación de la figura 6.18 para las energías de las simulaciones 4-5. . . . .	68

# Resumen y Agradecimientos

## Resumen

Para este trabajo se construyó una interfaz gráfica a partir del lenguaje de programación Python para simulación de plasmas electrostáticos tomando como base el código OneGPLA.py desarrollado anteriormente en el grupo de investigación GPLA. Esto se hizo con la finalidad de brindar una herramienta académica para el estudio de plasmas bajo condiciones que no son fácilmente reproducibles ya sea por la falta de los instrumentos o infraestructura necesaria para ello, como por su poca probabilidad de obtenerla aquí en la tierra, por ejemplo, el caso del plasma astrofísico. Es así que al implementar la interfaz gráfica Plasma Electrostatic Wave (PEWAV), el cual utiliza el algoritmo enfocado en la dinámica de múltiples partículas llamado Particle-in-Cell (PIC), se decidió aplicarlo en las inestabilidades electrónicas del plasma para corroborar que bajo ciertas condiciones dichas inestabilidades son el mecanismo principal de generación de las ondas solitarias electrostáticas (OSE) que han sido observadas en algunas regiones de la magnetosfera como la zona de media y alta altitud. También, para brindar un soporte teórico de que el plasma presente en estas regiones da lugar a las OSEs y para precisar una descripción apropiada del comportamiento de este plasma no magnetizado, se incluyó un modelo fisicomatemático que parte de las ecuaciones de fluido del plasma de un sistema electrostático compuesto de iones estáticos, electrones fríos, calientes y haz, tal que, mediante la técnica de perturbación, se obtuviera la ecuación Korteweg-de Vries (K-dV) cuyas soluciones son solitones. Por último, se realizaron las simulaciones usando PEWAV de las inestabilidades Two-Stream, Bump-on-Tail, Beam-plasma y Weak-Beam, para las cuales se variaron sus condiciones de temperatura (velocidad térmica), velocidad de deriva, razón de densidad y cantidad de especies utilizadas (dos o tres especies). Estas simulaciones se dividieron en dos circuitos A y B, el primero se compone de tres “run” para dos especies de partículas, mientras que el segundo para cinco “run” con tres especies de partículas. De estos circuitos se encontraron la presencia de estructuras solitarias en el potencial eléctrico que corresponden directamente a los vórtices visualizados en el espacio de fases.

## Agradecimientos

Agradecemos a nuestros familiares por su apoyo incondicional y a nuestro director de tesis, PhD. Henry Riascos Landázuri, por sus correcciones y observaciones en el desarrollo de este trabajo. Además, realizamos una especial mención a Cristian Rosero Arias por sus explicaciones sobre el método PIC y el código OneGPLA.



# Capítulo 1

## Introducción

Se denomina plasma al medio gaseoso que contiene un número apreciable de cargas libres. Un 99 % de la materia en el universo existe aparentemente en forma de plasma, tales como el medio estelar, interplanetario e interestelar, y altas atmósferas planetarias. Estos también pueden generarse en laboratorios para procesos industriales, tales como recubrimientos de superficies, realizar cortes de precisión en ambientes quirúrgicos, estudios enfocados a la ingeniería de materiales hasta motores y reactores de plasma [4]. Estos entornos requieren ser desarrollados de forma tanto teórica como experimental para entender los fenómenos físicos que para la comunidad científica aun no son claros y nos permiten mejorar nuestro entendimiento colectivo del tema. Para esta investigación se estudiará el mecanismo de formación de las ondas solitarias observadas en la región auroral y en diversas zonas de la magnetosfera, con la finalidad de comparar los resultados obtenidos a través de simulaciones con las que se encuentran en la literatura y los datos observados. En el estudio del plasma la simulación tiene un papel protagónico para el desarrollo de los avances físicos actuales, pues siguiendo el método de científico, cada proceso que involucre plasma en su ejecución requiere una simulación previa. Ejemplos claros pueden ser encontrados en la literatura [5], donde se explica la importancia de las simulaciones en diversos proyectos, como la misión BepiColombo (*misión conjunta de la Agencia espacial Europea y Agencia Japonesa de Exploración Aeroespacial*) para la exploración del planeta mercurio. Las simulaciones demostraron que el plasma producido por el propulsor no estaban dañando los sistemas de la nave espacial, dichos resultados fueron de gran ayuda para mejorar las mediciones a bordo de la nave y entender mejor las interacciones de plasma con materiales del vehículo espacial. El desarrollo de las simulaciones del plasma siempre ha requerido nuevos avances tecnológicos, donde se evidencia claramente con el ejemplo anterior, que los requerimientos son cada vez más grandes en capacidad computacional y entendimiento general de estos fenómenos.

El auge por las simulaciones en este área de la física comienza en 1985, con la publicación del libro “*Plasma Physics via computer simulation*” [6] por los autores Birdsall y Langdon. Material que sienta las bases para la construcción de códigos que permiten simular diversos fenómenos característicos del plasma, con explicaciones que permiten al lector entender la física detrás del desarrollo del código. Este material provee ejemplos desde, oscilaciones en el plasma hasta eventos más complejos como inestabilidades y amortiguamientos. Los avances en simulación presentaron códigos construidos en lenguajes de programación actuales como el creado por el físico Daniel Martin en su trabajo “*Electrostatic pic simulation of plasmas in one dimension*” [7], para inestabilidades two-stream en plasmas electroestáticos de especies frías. Hasta el trabajo desarrollado por Fox, Williams y Messina [8] para la universidad de Oxford en códigos de simulación en paralelo.

De esta forma con respecto a lo expuesto en los párrafos anteriores, se busca la optimización del código preexistente desarrollado por el grupo de investigación de plasma, láser y aplicaciones presentado por Rosero y Duque [3]. Así como la implementación de una interfaz gráfica de fácil manejo. Esta investigación contendrá simulaciones para estudiar y analizar las condiciones en las cuales las ondas solitarias se forman y el rol de las inestabilidades electrónicas del plasma en su generación, fenómeno que no ha sido estudiado a profundidad en Colombia.

## 1.1. Objetivos

### 1.1.1. Objetivos Generales

Estudiar a través de la teoría y el método de simulación Particle-in-Cell (PIC), las ondas solitarias observadas en la magnetosfera y las inestabilidades de haz de electrones como su mecanismo de generación.

### 1.1.2. Objetivos específicos

1. Implementar una interfaz gráfica en Python de fácil manejo para el código OneGPLA.py realizando una optimización del código y la documentación de las funciones de cada botón disponible en la interfaz.
2. Desarrollar el formalismo teórico detrás de la formación y el comportamiento de las ondas solitarias observadas en la magnetosfera.
3. Utilizar la interfaz gráfica para la simulación y el análisis de las inestabilidades tipo two-stream, bump of tail, beam plasma y weak beam como mecanismo de generación de las ondas solitarias electrostáticas.

# Capítulo 2

## Modelo Teórico

### 2.1. Magnetosfera

Para estudiar todos los procesos referentes al plasma en el entorno espacial, debemos tener en cuenta ciertos conceptos básicos sobre su formación y que eventos lo producen, sabiendo de antemano que aunque en la atmósfera terrestre podemos encontrar plasma generado de forma natural. En el espacio tenemos un 99 % de materia compuesta por plasma. Para estudiar entornos mas cercanos a la tierra es necesario describir como el viento solar produce dichos fenómenos donde partículas cargadas son liberadas con grandes velocidades desde las altas temperaturas de la superficie solar, generando la llamada corona solar compuesta en esencia de electrones y protones. Estas partículas al viajar a tan enormes velocidades chocan frecuentemente con el campo magnético de la tierra generando múltiples fenómenos. Independiente de la alta cantidad de energía que llevan dichas partículas, esta no es suficiente para penetrar dicho campo, por lo que sufre una deformación producida por la interacción entre el plasma (*partículas cargadas de la corona solar*) y el fuerte campo magnético que recubre la tierra ocasionando que la energía cinética de los protones y electrones se convierta en energía térmica aumentando la temperatura.

Estas partículas libres reaccionan en la atmósfera terrestre con las fuerzas magnéticas siendo direccionadas o atrapadas por estas, acelerándose o disminuyendo sus velocidades generando efectos como ondas de radio de todo tipo. Esta desviación del viento solar alrededor de la magnetosfera guía estas partículas a través de las líneas de campo, provocando que el plasma

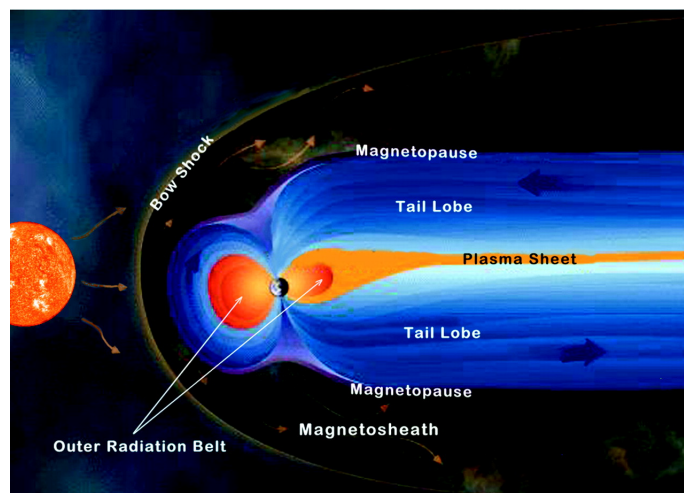


Figura 2.1: Ilustración magnetosfera.

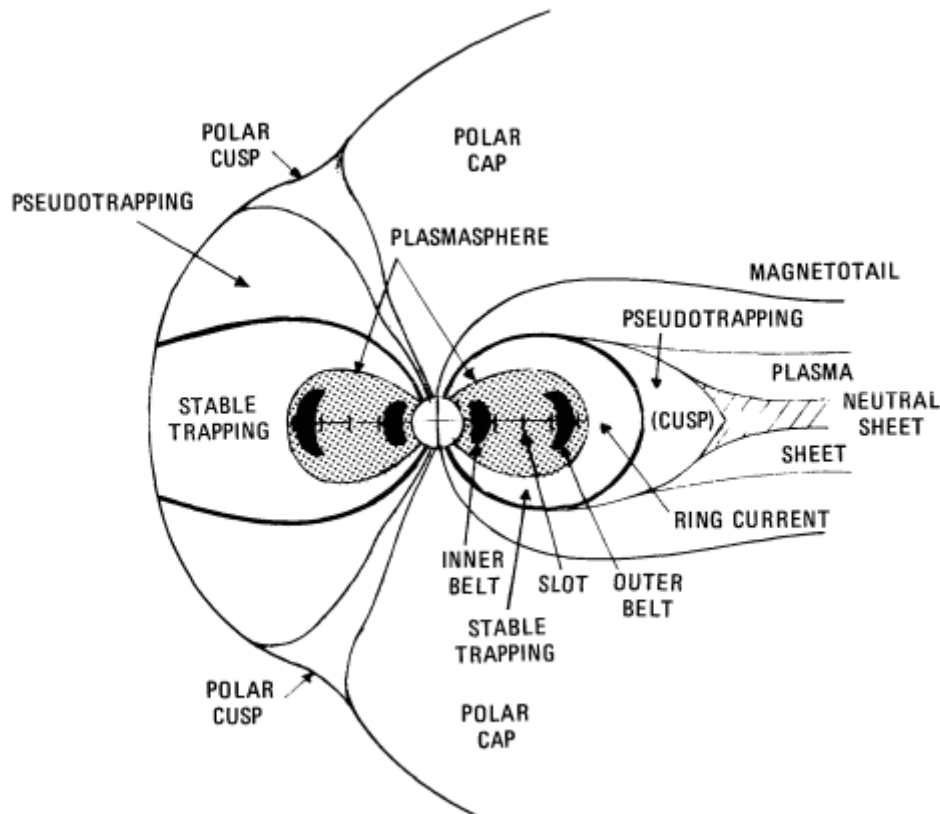


Figura 2.2: Esquema entorno terrestre de la magnetosfera.

no pueda moverse libremente como material superconductor, característica propia de este a temperaturas tan elevadas. Estas ondas de radio, generadas por la interacción del plasma espacial y las líneas de campo magnético terrestre, se propagan en todas direcciones dando lugar al plasma que podemos observar en la magnetosfera. Este plasma solar fluye con tanta velocidad, que supera ampliamente las velocidades de las ondas que se propagan en su interior. Por ende al momento del choque con el campo magnético se generan múltiples ondas en diversas rutas, ocasionando que grupos de ondas quieran moverse en dirección contraria, específicamente en dirección al sol. Finalmente estas partículas no pueden regresar porque son barridas en dirección del flujo [9], por tanto este plasma toma ciertas características como aumentos de temperatura, adicionalmente las familias de partículas que tratan de ir en contra del flujo se vuelven irregulares con temperaturas que varían entre frías y calientes, convirtiéndose este límite en la llamada magnetopausa.

Parte de la configuración de la magnetosfera es generada por las partículas que no son atrapadas por las líneas del campo magnético. En el lado diurno o región iluminada de la magnetosfera las partículas, de los vientos solares al colisionar con el campo magnético, son atrapadas y su energía cinética disminuye, generando en la parte más externa de la magnetosfera el denominado “arco de choque”. Estas partículas son redireccionadas hacia todas las demás secciones de la magnetosfera. Por otro lado, en el lado nocturno, la cola de la magnetosfera, las partículas provenientes del viento solar, son altamente energéticas. La alta velocidad de estas partículas dan origen a las auroras boreales, auroras polares, que consisten en una gran iluminación de la atmósfera terrestre. La cola de la magnetosfera está compuesta por líneas de campo paralelas que se extienden desde el norte del Ecuador incluyendo el polo norte en su recorrido, incluyendo otro conjunto de líneas de campo que van desde el sur con dirección a la parte externa de la

tierra incluyendo el polo sur. Estos lóbulos de cola en algún punto entre los 200 y 260 radios terrestres ya están combinados con el plasma del viento solar aunque el plasma no es muy denso, en su recorrido por la tierra, los lóbulos no contienen grandes cantidades de partículas cargadas. Como se menciona al principio del texto la magnetopausa funciona como un límite determinado por el campo magnético en el cual se relaciona su alcance fuera de la órbita terrestre. En el plano meridional de la misma donde se separan el lado nocturno y el iluminado se encuentra la cúspide solar [10], zona caracterizada por ser la altura a la cual se generan más fenómenos asociados a los comportamientos del plasma, siendo este ampliamente estudiado pues a estas altitudes se precipitan muchas partículas a las secciones como los lóbulos de colas para generar auroras boreales.

## 2.2. Teoría Cinética del Plasma

En el estudio del plasma se pueden realizar experimentos para tratar de responder o comprender todos los fenómenos que involucran este estado de la materia, dinámicas que, permiten estudiar a profundidad descripciones del plasma como fluido y como un conjunto teórico de partículas sin movimiento. Dentro de las descripciones cinéticas que puede tener el plasma tenemos las que consideran aspectos macroscópicos [11], sin tener en cuenta aspectos detallados en las distribuciones de velocidades microscópicas, dentro de estas descripciones encontramos características propias del plasma en cualquier entorno, como las densidades asociadas a la función de distribución, la frecuencia propia del plasma, la longitud de Debye, entre otras consideraciones necesarias para cada especie de partículas presentes en el plasma.

Estos fundamentos de descripción deben tener un método eficiente para desarrollar la posición y velocidad [12] de cada una de las partículas contenidas en el plasma, en este caso por la cantidad de partículas requeridas para las simulaciones, dichos análisis son virtualmente imposibles para la capacidad computacional requerida, pues se debería poder medir la evolución temporal de cada partícula sin resolver las ecuaciones de movimiento de Newton características de forma individual. Por tanto el método para resolver esta problemática es aplicar la función de distribución  $f=f(x,v,t)$  donde podemos definir la densidad de cada partícula para cada especie modelado dentro del plasma.

Los procesos matemáticos descritos en la referencia [12], consideran que, si no se tienen procesos de recombinación donde los iones positivos contenidos se adhieren a un electrón libre cargado energéticamente, y estos se combinan con iones negativos formando nuevos átomos neutros liberando grandes cantidades de calor [11], dado que este tipo de recombinación se genera una gran cantidad de partículas que no permite hacer un análisis acertado sobre las consideraciones cinéticas de las especies, donde se necesita tener un volumen establecido en un espacio limitado con velocidades característica  $v$ . Siendo esta la variable que puede modificarse con respecto a las partículas que cambian de velocidad levemente. Esta descripción se pueden expresar de la siguiente forma.

$$\frac{\partial f}{\partial t} = -v_x \cdot \frac{\partial f}{\partial x} - \frac{F}{m} \cdot \frac{\partial f}{\partial v_x} \quad (2.1)$$

Descripción cinética

$$N_s(t) = \int_{-\infty}^{\infty} f_s(x, v_x, t) dv dx \quad (2.2)$$

Función de distribución número de partículas de una especie  $N_s$

## 2.3. Longitud de Debye

En el estudio de la física del plasma la longitud de Debye se usa como una escala o rango en el cual las especies de partículas como electrones e iones generan un apantallamiento de los campos eléctricos, como ejemplo se puede pensar en una partícula de carga positiva que ingresa con cierta velocidad en un plasma compuesto de iones y electrones, en este caso los electrones serán atraídos y generaran una coraza o capa alrededor de esta partícula, estos electrones estarán a cierta distancia de la partícula interna quedando neutralizada, dicha formación conocida también como esfera de Debye evita que el campo eléctrico producido por esta formación afecte el entorno del plasma. Concluyendo entonces que la longitud de Debye [13] es una estimación de esta distancia, definida por la siguiente ecuación.

$$\lambda_{Ds} = \sqrt{\frac{\epsilon_0 k_B T_s}{n_s e^2}} \quad (2.3)$$

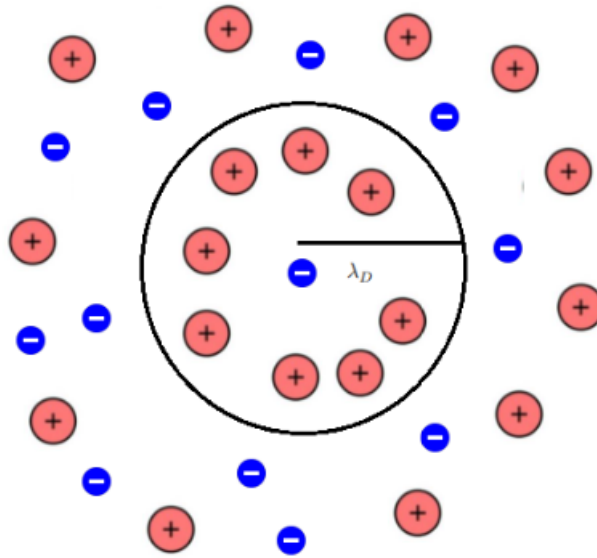


Figura 2.3: Longitud de Debye

Los términos que definen esta estimación están dados por,  $k_B$  como la constante de Boltzmann,  $T_s$  como la temperatura del electrón,  $\epsilon_0$  la permitividad eléctrica en el vacío,  $n_s$  la densidad de la especie y finalmente la carga elemental de las partículas.

## 2.4. Frecuencia del Plasma

En los modelos teóricos del plasma, la frecuencia del plasma es usado para describir los tiempos característicos, de igual forma que la longitud de Debye es una escala de medida de longitud, la frecuencia mide el tiempo en los que se presentan ciertos fenómenos, como ejemplo, se puede suponer un plasma que se encuentra en equilibrio térmico en el cual se introduce una perturbación que produce un desplazamiento en los electrones que componen el plasma, estos sienten la atracción de los iones que se mueven en dirección opuesta generando oscilaciones en torno al punto inicial de equilibrio, por ende la suma de todas estas oscilaciones es lo que conoce como frecuencia de plasma notada con la siguiente ecuación, donde  $m_{ps}$  indica la masa de partícula y  $n_{ps}$  la especie a la que pertenece.

$$\omega_{ps} = \sqrt{\frac{n_{ps} e^2}{m_{ps} \epsilon_0}} \quad (2.4)$$

## 2.5. Velocidad Térmica

El término de velocidad térmica es importante al momento de realizar un análisis general del movimiento de las especies de partículas que componen el plasma, generalmente no tienen la misma velocidad así comportan posiciones cercanas, estas están relacionadas bajo un principio importante donde a mayor temperatura mayor será la dispersión de velocidad coincidiendo con el equilibrio térmico de la distribución de *Maxwell Boltzmann*; para medir dicha dispersión denominada velocidad térmica se puede medir a partir de una velocidad cuadrática media directamente proporcional a la temperatura con la siguiente expresión, notando que en la teoría formal si el plasma no se encuentra en equilibrio termodinámico dicha relación no es posible.

$$v_{ds} = \left( \frac{k_B T_s}{m_s} \right)^{\frac{1}{2}} \quad (2.5)$$

## 2.6. Amortiguamiento de Landau

Partiendo desde la definición características de interacción onda partícula, en el plasma se pueden generar condiciones donde es posible un intercambio de energía. En el caso puntual de esta tesis, una de estas interacciones presentes en plasmas no colisionales es el amortiguamiento de Landau [12]. Donde se describe que una onda no puede perder energía al momento de la interacción con otras generando un movimiento donde las partículas tendrán temperaturas diferentes, suceso que si ocurre en otros tipos de análisis para fluidos viscosos conocido como movimiento aleatorio térmico de partículas. Estas interacciones se describen correctamente como una transferencia de energía de las ondas a las partículas partiendo de una condición física denominada resonancia de Landau [14]. Para las inestabilidades que serán tratadas en este caso se usará una descripción del sistema a través de las ecuaciones de Vlasov-Poisson llegando a una función de dispersión que describe las oscilaciones y las transferencias de energía que se presentan entre las ondas y partículas.

$$D(w, k) = 1 + \frac{e^2}{\epsilon_0 m_e k} \frac{1}{k} \frac{f_0}{w - ku} - \int_{-\infty}^{\infty} \frac{k f_0}{w - ku} du \quad (2.6)$$

Esta ecuación describe los modos de oscilación del plasma, donde las ondas que poseen la misma velocidad que la velocidad de fase [3], generan el intercambio energético referido al inicio del numeral, en el cual las partículas con altas velocidades cederán su energía a la onda perdiendo velocidad y el grupo con velocidad menor ganará energía de la onda.

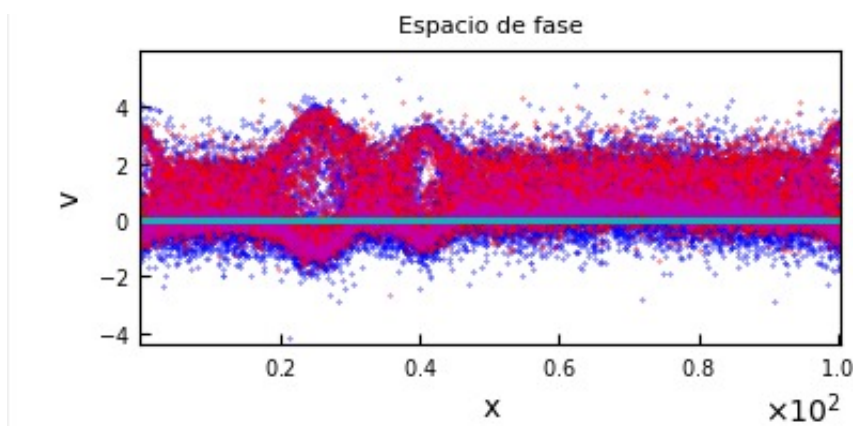


Figura 2.4: Amortiguamiento de Landau en el espacio de fase de la inestabilidad Two-Stream (resultado propio obtenido en el capítulo 6).

## 2.7. Inestabilidades en el Plasma

Las inestabilidades en el plasma son un tópico frecuente en las investigaciones actuales, por ejemplo, en el estudio de la fusión nuclear, las tormentas espaciales, erupciones solares, entre otros [15], siendo estas manifestaciones extremas de inestabilidades del plasma. Por nombrar análisis a gran a escala, aquí en la planeta tierra también podemos evidenciar múltiples inestabilidades generadas en la magnetosfera y sus diferentes partes producto de su interacción con los vientos solares, estas son regiones donde se producen turbulencias debido a cambios drásticos en la estructura del plasma, alteraciones repentinos de temperatura, de densidad, campos magnéticos o eléctricos, así como lo aduce su nombre son plasmas inestables en su morfología y las reconfiguraciones del sistema tanto a nivel microscópico como macroscópico, de igual forma, pueden evaluarse ciertos cambios en su evolución temporal para definir qué tipo de inestabilidad se está presentando.

### 2.7.1. Inestabilidad Two-Stream

La inestabilidad Two-Stream es la más conocida y estudiada dentro del amplio conjunto de inestabilidades conocidas en el plasma, principalmente por el hecho de no requerir un capacidad computacional extremadamente alta para visualizar dicho fenómeno. Esta inestabilidad consiste en la interacción de dos flujos de partículas cargadas, con velocidad de deriva y frecuencias características diferentes (caso puntual usado para este trabajo) que al momento [16] de interactuar generan vórtices evidenciados en el espacio de fase de estas, y comportamientos típicos como los vistos en las figuras 2.5-2.6 donde se generan picos de igual comportamiento apartir de la función de distribución. En este fenómeno las ondas son excitadas por la energía cinética de las partículas, lo cual se traduce en la generación del campo eléctrico de la onda [17]. Esta inestabilidad puede describirse bajo el modelo de la función de distribución de velocidades desde la forma estable de la distribución maxwelliana. Para efectos de la descripción teórica se planteara la ecuación para describir la dinámica de los electrones, donde  $v_t$  describe la velocidad térmica,  $v_d$ , la velocidad de deriva y  $n_0$ , la densidad de los electrones.

La función distribución maxwelliana que modela la inestabilidad de dos haces es la siguiente:

$$f(x, v_x) = \frac{n_0}{2} \frac{1}{\sqrt{2\pi}v_t} e^{-\frac{(v-v_{d1})^2}{2v_t^2}} + \frac{1}{\sqrt{2\pi}v_t} e^{-\frac{(v+v_{d2})^2}{2v_t^2}} \quad (2.7)$$

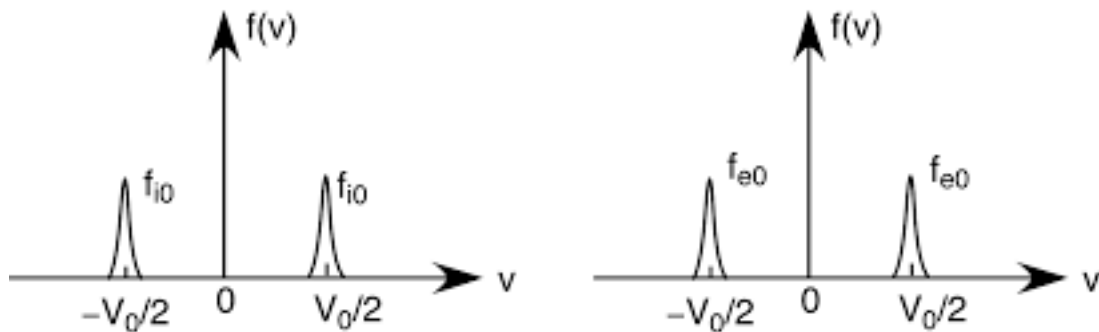


Figura 2.5: Esquema ejemplo función de distribución two-stream.



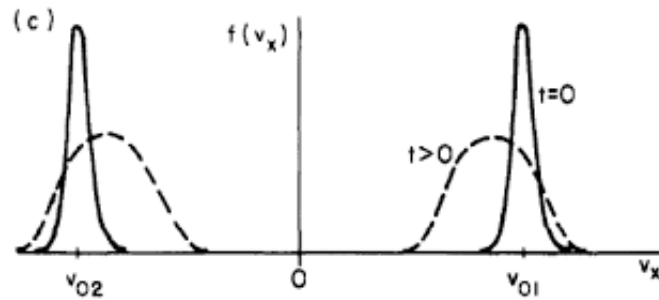


Figura 2.6: Esquema ejemplo función de distribución two-stream.

### 2.7.2. Inestabilidad Bump-of-Tail

Esta inestabilidad se caracteriza por generarse en ciertas regiones de la ionósfera. Dicho fenómeno se produce cuando un haz de electrones atraviesa la ionósfera entregando grandes cantidades de energía y provocando que los electrones se aceleren en la capa. Este haz de electrones puede ser descrito mediante una distribución de velocidades Gaussiana. En términos prácticos, esta distribución Gaussiana es la función de distribución maxwelliana, por lo tanto, la inestabilidad puede ser descrita similar a la inestabilidad Two-Stream [18] y cuyo aspecto es la visualizada en la figura 2.7, la cual tiene una joroba en la cola de la campana de Gauss.

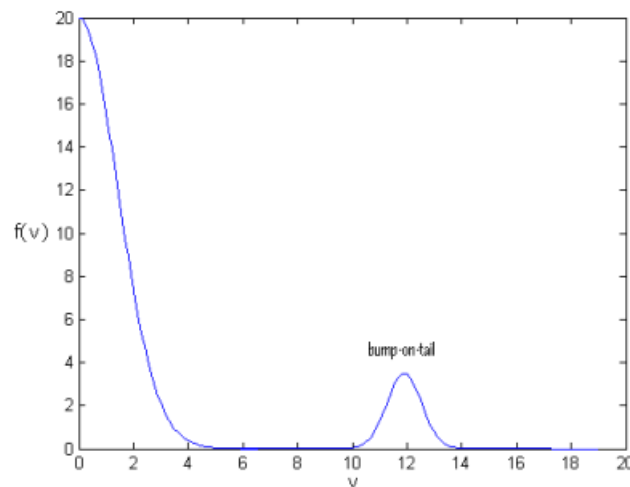


Figura 2.7: Esquema Inestabilidad Bump-of-Tail.

### 2.7.3. Inestabilidad Beam-Plasma

El modelo de inestabilidad Beam-Plasma indica una interacción entre un plasma estacionario bajo condiciones de equilibrio térmico y un haz de partículas. Esta inestabilidad puede variar su configuración con la función de distribución, por eso en algunos casos en simulación a pequeñas escalas es complicado ver la diferencia entre la inestabilidad Beam-Plasma y otras posibles inestabilidades presentes en el plasma, pues cuando se modifica la distribución de las partículas este fenómeno adquiere características muy similares a la Bump-on-Tail o la inestabilidad Warm-Beam. Esta similitud se puede identificar en la figura 2.8, la cual puede ser confundida con la inestabilidad Bump-on-Tail de la figura 2.7

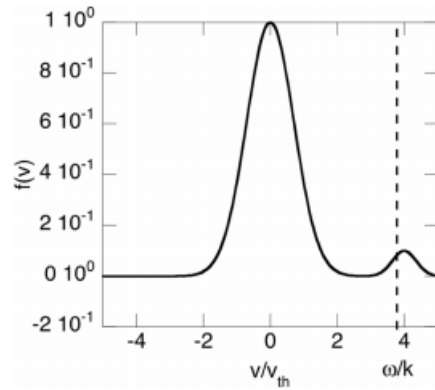


Figura 2.8: Esquema ejemplo inestabilidad Beam-Plasma.

## 2.8. Teoría de Fluido del Plasma

La perspectiva del plasma como un fluido es un gran logro, pues permite explicar una gran cantidad de fenómenos observados en experimentos. Este modelo no busca explicar el comportamiento de cada partícula que compone el plasma, sino que se enfoca en el movimiento de elementos de fluido. La ventaja de este acercamiento es su simplicidad, pero pierde efectos importantes como el amortiguamiento de Landau (Sec. 2.6).

### 2.8.1. Ecuación de Continuidad

Si se tiene un sistema donde las partículas no son generadas o aniquiladas dentro del fluido (ej. recombinación o ionización), entonces existe una conservación del número de partículas que puede ser expresado por la *ecuación de continuidad*

$$\frac{\partial n_s}{\partial t} + \nabla \cdot (n_s u_s) = 0 \quad (2.8)$$

Donde  $n_s$  es la densidad de partículas,  $u_s$  es el vector de la velocidad promedio para las partículas en un elemento de fluido y  $s$ , la especie de partícula.

### 2.8.2. Ecuación de Momento

La segunda ecuación de la teoría de fluido es la *ecuación de fuerza*, también conocida como la *ecuación de momento*. Al considerar tanto fuerzas internas como la suma de aquellas que surgen de partículas entrando y saliendo a un elemento de fluido, se obtiene que el transporte de momento para un sistema sin colisiones es:

$$n_s m_s \left[ \frac{\partial u_s}{\partial t} + (u_s \cdot \nabla) u_s \right] = n_s q_s E - \nabla p_s \quad (2.9)$$

Aquí  $m_s$  y  $p_s$  representan la masa y la presión cinética respectivamente.

Dado que en este trabajo se tiene un enfoque electrostático, las ecuaciones de fluido presentadas aquí no contienen campo magnético.

### 2.8.3. Ecuación de estado

La última relación necesaria para cerrar el sistema de ecuaciones proviene de la ecuación de estado adiabática que relaciona la presión con la densidad, cuya forma es  $p_s = C_s n_s^{\gamma_s}$ . La cantidad  $\gamma_s$  expresa qué tanto aumenta la temperatura cuando un plasma es comprimido, mientras que  $C_s$  es una constante. De esta relación llegamos a:

$$\nabla p_s = \gamma_s T_s \nabla n_s \quad (2.10)$$

Donde  $\gamma_s = (N + 2)/N$  y  $N$  representa el número de grados de libertad. Para el caso en el que la compresión es lenta comparada con la conducción térmica  $\gamma_s = 1$ , recibiendo el nombre de compresión isotérmica.

## 2.9. Ondas Solitarias y la Ecuación Korteweg-de Vries

El modelo teórico considerado para realizar una descripción apropiada del comportamiento del plasma no magnetizado, en las identificaciones de inestabilidades y detección de ondas solitarias es el presentando por el matemático neerlandés Diederik Korteweg y su discípulo Gustav de Vries en su estudio de 1985 ecuación K-dV (Korteweg-de Vries).

Esta ecuación a través de derivadas parciales modelan los efectos de la dispersión de partículas mediante un modelo especial para describir la propagación de ondas en el plasma con gran exactitud, para entornos tan dispersivos como este. Para el caso puntal de esta investigación se utilizó un plasma no magnetizado compuesto de iones estáticos, tres poblaciones de electrones Maxwellianos en la categoría de partículas frías y calientes con un tercer componente, un haz de electrones.

A continuación se describirá el proceso matemático del conjunto de ecuaciones fundamentales que describen el movimiento de los fluidos en el plasma para verificar a través de los resultados obtenidos en esta investigación si estos describen la propagación de ondas solitarias electrostáticas (OSEs).

$$\eta_0 = \eta_{h0} + \eta_{c0} + \eta_{b0} = \eta_{i0} \quad (2.11)$$

$$\frac{\partial \eta_s}{\partial t} + \frac{\partial}{\partial x}(\eta_s u_s) = 0 \quad (2.12)$$

$$\frac{\partial u_s}{\partial t} + u_s \frac{\partial u_s}{\partial x} = \frac{e}{m_e} \frac{\partial \varphi}{\partial x} - \frac{\gamma_s T_s}{m_e \eta_s} \frac{\partial \eta_s}{\partial x} \quad (2.13)$$

$$\frac{\epsilon_0}{e} \frac{\partial^2 \varphi}{\partial x^2} = \eta_b + \eta_c + \eta_h - \eta_i \quad (2.14)$$

Las ecuaciones 2.12 a 2.14 son las ecuaciones de fluido sin normalizar.

Aquí  $\eta_h$  sigue una distribución de Boltzman:

$$\eta_h = \eta_{h0} e^{e\varphi/k_B T_h} \quad (2.15)$$

Realizando el proceso de normalización:

$$\begin{aligned} t = \omega_{pe} t \quad ; \quad x = x/\lambda_{Dh} \quad ; \quad n_s = \frac{\eta_s}{\eta_0} \quad ; \quad u_s = \frac{u_s}{v_{th}} \quad ; \quad v_{th} = \frac{u_{th}}{\sqrt{2}} \\ u_{th}^2 = \frac{2k_B T_h}{m_e} \quad ; \quad \lambda_{Dh} = \frac{v_{th}}{\omega_{pe}} \quad ; \quad \omega_{pe}^2 = \frac{\eta_0 e^2}{\epsilon_0 m_e} \quad ; \quad \psi = \frac{e\varphi}{k_B T_h} \end{aligned} \quad (2.16)$$

$$\beta_h = \frac{\Gamma_h}{\Gamma_h} \quad (2.17)$$

$$\beta_c = \frac{\Gamma_c}{\Gamma_h} \quad (2.18)$$

$$\beta_b = \frac{\Gamma_b}{\Gamma_h} \quad (2.19)$$

$$\beta_s = \frac{\Gamma_s}{\Gamma_h} \quad (2.20)$$

Resolviendo:

$$\frac{\partial n_s}{\partial t} + \frac{\partial}{\partial x} (n_s u_s) = 0 \quad (2.21)$$

$$\frac{\partial u_s}{\partial t} + u_s \frac{\partial u_s}{\partial x} = \frac{\partial \psi}{\partial x} - \frac{\gamma_s \beta_s}{n_s} \frac{\partial n_s}{\partial x} \quad (2.22)$$

$$\frac{\partial^2 \psi}{\partial x^2} = n_b + n_c + n_h - 1 = n_b + n_c + n_{h0} e^\psi - 1 \quad (2.23)$$

$$\xi = \delta^{\frac{1}{2}} x - t \quad ; \quad \tau = \delta^{\frac{3}{2}} t \quad ; \quad \delta = M - 1 \quad (2.24)$$

$$M = \frac{v_0}{v_{th}} \quad (2.25)$$

Donde  $M$  equivale al número mach y  $v_0$  define la velocidad de la onda. Se usó el parámetro  $\delta \ll 1$  para realizar las perturbaciones.

Partiendo de la descripción de las ondas electrostáticas con amplitud finita, se usó el concepto fundamental de la técnica de perturbación de las partículas teóricas. Introduciendo ciertos cambios de variables arbitrarias para continuar con el proceso matemático. Las densidades dentro del proceso también están normalizadas; el resto de las variables se expanden a través de la serie trabajada desde la expresión 2.24 por tanto partimos de la siguiente definición:

$$n_s = n_{s0} + \delta n_{s1} + \delta^2 n_{s2} \quad (2.26)$$

$$u_s = u_{s0} + \delta u_{s1} + \delta^2 u_{s2} \quad (2.27)$$

$$\psi = \delta \psi_1 + \delta^2 \psi_2 \quad (2.28)$$

$$\delta \frac{\partial n_s}{\partial \tau} - \frac{\partial n_s}{\partial \xi} \frac{\partial n_s u_s}{\partial \xi} = 0 \quad (2.29)$$

$$\delta \frac{\partial u_s}{\partial \tau} - \frac{\partial u_s}{\partial \xi} + u_s \frac{\partial u_s}{\partial \xi} = \frac{\partial \psi}{\partial \xi} - \frac{\gamma_s \beta_s}{n_s} \frac{\partial n_s}{\partial \xi} \quad (2.30)$$

$$\delta \frac{\partial^2 \psi}{\partial \xi^2} = n_b + n_c + n_{h0} e^\psi - 1 \quad (2.31)$$

Continuando el proceso, La perturbación del primer orden para la velocidad y la densidad a partir de la ecuación 2.29 y 2.30 se obtienen las ecuaciones 2.32 y 2.33:

$$u_{s1} = \frac{-(1 - u_{s0})}{(1 - u_{s0})^2 - \gamma_s \beta_s} \psi_1 = a_{s1} \psi_1 \quad (2.32)$$

$$n_{s1} = \frac{-n_{s0}}{(1 - u_{s0})^2 - \gamma_s \beta_s} \psi_1 = a_{s2} \psi \quad (2.33)$$

Retomando, de las ecuación 2.31 obtenemos el conjunto de ecuaciones desde 2.34 hasta la 2.37.

$$0 = n_{b0} + \delta n_{b1} + n_{c0} + \delta n_{c1} + n_{h0} + \delta n_{h0} \psi_1 - 1 \quad (2.34)$$

$$n_{b0} + n_{c0} + n_{h0} = 1 \quad (2.35)$$

$$n_{b1} + n_{c1} + n_{h0} \psi_1 = 0 \quad (2.36)$$

$$[a_{b2} + a_{c2} + n_{h0}] \psi_1 = 0 \rightarrow n_{h0} = -(a_{b2} + a_{c2}) \quad (2.37)$$

Después del proceso anterior, se obtiene el factor  $n_{h0}$  que se usara durante el cálculo de forma implícita para obtener los coeficientes que componen la ecuación KdV.

$$n_{h0} = \frac{n_{b0}}{d_b} + \frac{n_{c0}}{d_c} \quad (2.38)$$

Usando para la sustitución el término  $d_s$

$$d_s = (1 - u_{s0})^2 - \gamma_s \beta_s \quad (2.39)$$

Para la perturbación de segundo orden de la velocidad se uso el mismo proceso que para el primer numeral.

$$\frac{\partial u_{s2}}{\partial \xi} = \frac{1}{(1 - u_{s0})n_{s0}} \left\{ n_{s0} \left[ \frac{u_{s1}}{\partial \tau} + u_{s1} \frac{\partial u_{s1}}{\partial \xi} - \frac{\partial \psi_2}{\partial \xi} \right] - n_{s1} \left[ (1 - u_{s0}) \frac{\partial u_{s1}}{\partial \xi} + \frac{\partial \psi_1}{\partial \xi} \right] + \gamma_s \beta_s \frac{n_{s2}}{\partial \xi} \right\} \quad (2.40)$$

Despejando el término  $u_{s2}$ , conseguimos la perturbación de segundo orden para la densidad:

$$\begin{aligned} \frac{\partial n_{s2}}{\partial \xi} = \frac{1}{d_s} \left\{ (1 - u_{s0}) \left[ \frac{n_{s1}}{\partial \tau} + \frac{\partial(n_{s1}u_{s1})}{\partial \xi} \right] - n_{s1} \left[ (1 - u_{s0}) \frac{u_{s1}}{\partial \xi} + \frac{\partial \psi_1}{\partial \xi} \right] + \right. \\ \left. + n_{s0} \left[ \frac{\partial u_{s1}}{\partial \tau} + u_{s1} \frac{\partial u_{s1}}{\partial \xi} + \frac{\partial \psi_2}{\partial \xi} \right] \right\} \quad (2.41) \end{aligned}$$

Obteniendo la perturbación de segundo orden para la ecuación 2.31

$$\frac{\partial^3 \psi_1}{\partial \xi^3} = \frac{\partial n_{b2}}{\partial \xi} + \frac{\partial n_{c2}}{\partial \xi} + \frac{n_{h0}}{2} \frac{\partial \psi_1}{\partial \xi} + n_{h0} \frac{\partial \psi_2}{\partial \xi} \quad (2.42)$$

Reemplazando en la ecuación anterior los resultados 2.32-2.41 y simplificando, encontramos la forma de la ecuación K-dV:

$$\frac{\partial \psi_1}{\partial \tau} + \frac{1}{2} \frac{B}{A} \frac{\partial \psi_1^2}{\partial \xi} + \frac{C}{A} \frac{\partial^3 \psi_1}{\partial \xi^3} = 0 \quad (2.43)$$

$$A = - \{ d_c [(1 - u_{b0})a_{b2} + n_{b0}a_{b1}] + d_b [(1 - u_{c0})a_{c2} + n_{c0}a_{c1}] \} \quad (2.44)$$

$$B = - \{ d_c [(1 - u_{b0})a_{b1}a_{b2} - a_{b2} + n_{b0}(1 + a_{b1}^2)] + d_b [(1 - u_{c0})a_{c1}a_{c2} - a_{c2} + n_{c0}(1 + a_{c1}^2)] \} \quad (2.45)$$

$$C = d_b d_c \quad (2.46)$$

Reescalamos a partir de la ecuación 2.43 para llegar a la solución esperada:

$$\xi = \left(\frac{C}{A}\right)^{\frac{1}{3}} \xi \quad (2.47)$$

$$\frac{\partial \psi_1}{\partial \tau} + \frac{1}{2} \frac{B}{(A^2 C)^{\frac{1}{3}}} \frac{\partial \psi_1^2}{\partial \xi} + \frac{\partial^3 \psi_1}{\partial \xi^3} = 0 \quad (2.48)$$

$$\psi_1 = \frac{(A^2 C)^{\frac{1}{3}}}{B} \psi_1 \quad (2.49)$$

$$\frac{\partial \psi_1}{\partial \tau} + \frac{1}{2} \frac{\partial \psi_1^2}{\partial \xi} + \frac{\partial^3 \psi_1}{\partial \xi^3} = 0 \quad (2.50)$$

Sustituyendo  $\zeta = \xi - D\tau$  tal que:

$$\frac{\partial}{\partial \xi} = \frac{d}{d\zeta} \quad ; \quad \frac{\partial}{\partial \tau} = -D \frac{d}{d\zeta} \quad (2.51)$$

$$-D \frac{d\psi_1}{d\zeta} + \frac{1}{2} \frac{d\psi_1^2}{d\zeta} + \frac{d^3 \psi_1}{d\zeta^3} = 0 \quad (2.52)$$

Integrando con respecto a una variable arbitraria  $\zeta_a$  propuesta para continuar con el proceso matemático:

$$-D \int_{\zeta}^{\infty} \frac{d\psi_1}{d\zeta_a} d\zeta_a + \frac{1}{2} \int_{\zeta}^{\infty} \frac{d\psi_1^2}{d\zeta_a} d\zeta_a + \int_{\zeta}^{\infty} \frac{d^3 \psi_1}{d\zeta_a^3} d\zeta_a = 0 \quad (2.53)$$

Si  $\psi_1(\zeta)$  y sus derivadas se desvanecen a largas distancias del solitón  $\zeta \rightarrow \infty$ , el resultado es:

$$\frac{D}{2} \psi_1^2 - \frac{1}{6} \frac{\partial \psi_1^3}{\partial \zeta} - \frac{1}{2} \frac{d}{d\zeta} \left( \frac{d\psi_1}{d\zeta} \right)^2 = 0 \quad (2.54)$$

Integrando cada término de la ecuación, después de un proceso de factorización y al desescalar, se llega finalmente a la solución de una onda solitaria:

$$\psi_1 = \frac{3D(A^2 C)^{\frac{1}{3}}}{B} \text{Sech}^2 \left[ \frac{\sqrt{D}}{2} \left[ \left( \frac{A}{C} \right)^{\frac{1}{3}} \zeta - D\tau \right] \right] \quad (2.55)$$

En el desarrollo de la ecuación vemos que cada componente de la onda se mueve con la misma velocidad de fase o con menos velocidad. En consecuencia el movimiento es no dispersivo, esa precisamente es la definición de una solución de un solitón. Una onda sin ningún tipo de alteración.

A partir del resultado anterior encontramos que la ecuación K-dV posee soluciones en forma de onda viajera (*teniendo en cuenta que la ecuación 2.55 nos indica que es una onda solitaria*), con diversas propiedades interesantes que nos permiten plasmar condiciones para entender el comportamiento de las ondas electrostáticas. Se puede analizar en función de los términos que componen dicha ecuación, primero podemos ver que amplitud de la onda es directamente proporcional a su velocidad. Esto sugiere que cuando más alta sea la onda obtenida más rápido se mueve, efecto visto en diversos medios donde se propagan ondas solitarias. La propiedad más importante de este modelo, es que la onda no tiene una dispersión, mantiene su forma al avanzar aún en el caso de sufrir interacciones no lineales con otras ondas [19]. Esta destacable estabilidad de las ondas solitarias, en la que se exhibe un comportamiento como de una partícula en mecánica clásica, llevó al físico estadounidense **Zabusky** y al matemático **Kruskal** a acuñar el término solitón.

Volviendo al plasma, los iones juegan un papel relevante en la escala dinámica, dada la alta frecuencia donde se puede suponer que los iones de carga positiva forman una capa uniforme distribuida en el plasma proporcionando neutralidad a la carga, pero sin cambiar de forma abrupta la dinámica de la generación de ondas electrostáticas [5]. En el caso del aporte de la temperatura, la velocidad térmica

de los electrones fríos proporciona la inercia relativa para mantener las oscilaciones en su estado, entendiéndose por inercia en este caso la resistencia de la OSE a modificar su estado inicial de movimiento. Mientras que la fuerza de restauración que termina de generar la onda proviene de las electrones calientes permitiendo que la estructura se mantenga y se cumpla el requisito para ser una onda electrostática.

Estas ondas se caracterizan por tener altas frecuencias asociadas, en comparación de las frecuencias que se encuentran en los plasmas compuestos de iones. Estas OSEs se propagan a velocidad establecidas entre las velocidades térmicas de las poblaciones frías y calientes. A continuación se adjuntan dos gráficas de estructuras solitarias generadas por electrones, una teórica (Fig. 2.9) con la finalidad de comparar con las simulaciones<sup>1</sup> hechas para la ecuación 2.55 del cual se pueden obtener estructuras solitarias compresivas (Fig. 2.10) y rarefactivas (Fig. 2.11). Se esperan obtener OSEs de esta naturaleza en el capítulo 6.

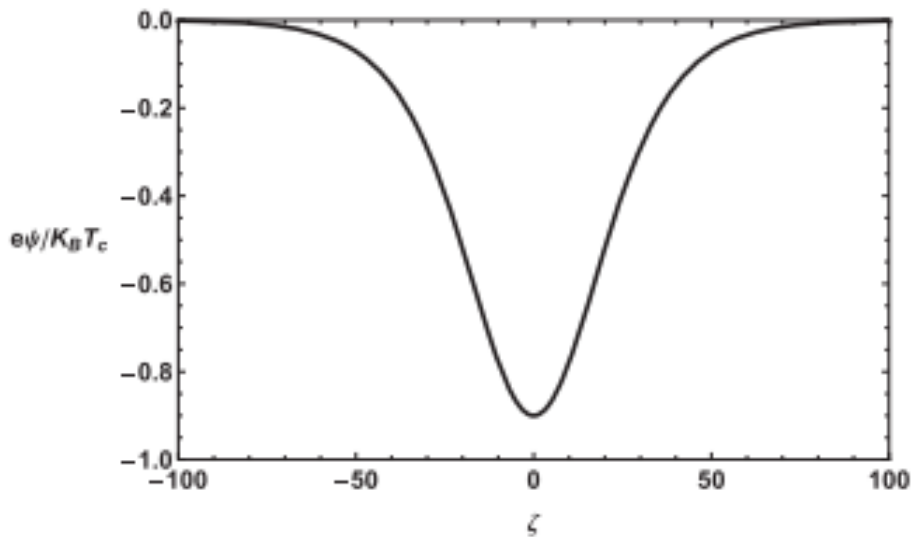


Figura 2.9: Estructura solitaria encontrada en la literatura [1].

<sup>1</sup>Estas simulaciones se hicieron con dos códigos independientes de *PEWAV*, llamados *SK\_compressive* y *SK\_rarefactive*. Estos códigos se pueden encontrar en Anexos 8.

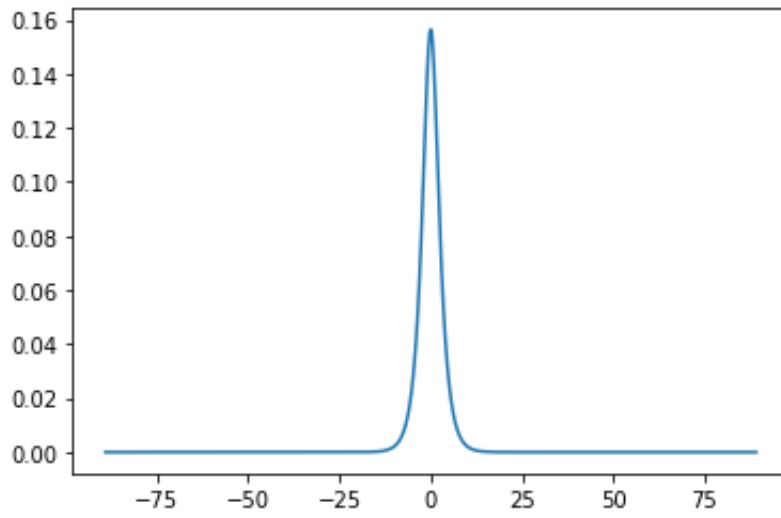


Figura 2.10: Onda eletrostática simulada de tipo compressiva.

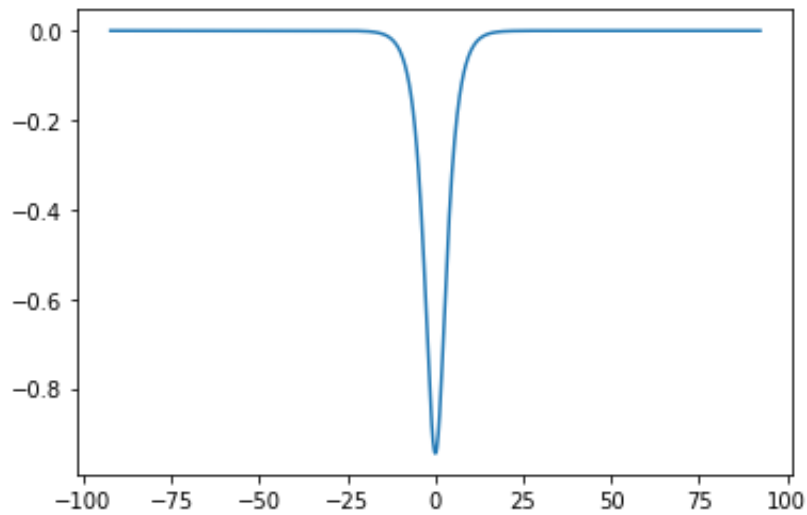


Figura 2.11: Onda eletrostática simulada de tipo rarefactiva.



# Capítulo 3

## Modelo Computacional

El uso de métodos computacionales tanto para simular como para predecir la naturaleza de nuestro universo o fragmentos de ella, ha sido de gran interés debido a que nos permite no solo a entenderla, sino aprovecharla en nuestro beneficio como ha sido la simulación del plasma en el área de la fusión nuclear por confinamiento inercial [20] o para el estudio del plasma astrofísico [21]. También cabe destacar logros sorprendentes como los del sistema de inteligencia artificial AlphaFold [22] que ha resuelto un problema de más de 50 años, el de cómo las proteínas se envuelven formando estructuras 3D. Este último a pesar de no formar parte del campo de la física de plasma, muestra el camino hacia un futuro en el cual las simulaciones por computador asistidas por inteligencia artificial serán necesarias para seguir generando conocimiento.

### 3.1. Método Particle-in-Cell

El método Particle-in-Cell (PIC) fue propuesto inicialmente para modelar fluidos compresibles [23], pero con el desarrollo de los algoritmos Cloud-in-Cell (CIC) en 1970 [24] y Material-Point-Method (MPM) en 1995 [25], se aplicaría para el modelamiento de plasmas cinéticos y en la mecánica de sólidos respectivamente. Es así que este método aplicado para plasmas cinéticos se conoce como CIC-PIC y consiste en que este último es representado mediante superpartículas (SP), las cuales contienen propiedades como la velocidad y la posición. Las SP son ubicadas en una malla espacial con el fin de ponderar la densidad en cada uno de sus puntos. Conociendo la distribución de densidad se obtiene el campo eléctrico para luego a través de una interpolación calcular la fuerza que están experimentando las partículas en un instante  $t$  y así continuar con el estado  $t + \Delta t$ . Este ciclo se repite un número de veces  $N_{IT}$  especificado por el usuario. En la figura 3.1 se resume los elementos básicos del método para el caso de un plasma electrostático unidimensional (1D).

#### 3.1.1. Método de ponderación

Dado que la densidad de partículas es asignada a puntos discretos en la malla espacial, se hace necesario utilizar un método de ponderación para así calcular el campo eléctrico y de allí, la fuerza experimentada por las partículas. La elección del método depende del ruido dispuesto a tolerar y de las prestaciones de nuestro equipo, ya que entre más refinado sea, mayor es el coste computacional.

Algunos de estos métodos según su orden son:

- Orden cero: punto de malla más cercano o también conocido como NGP (Nearest-Grid-Point).
- Primer orden: nube en celda o CIC (Cloud-in-Cell).
- Orden superior: splines cúbicos y cuadráticos.

El método utilizado en este trabajo fue el de CIC que a pesar de tener un coste computacional superior

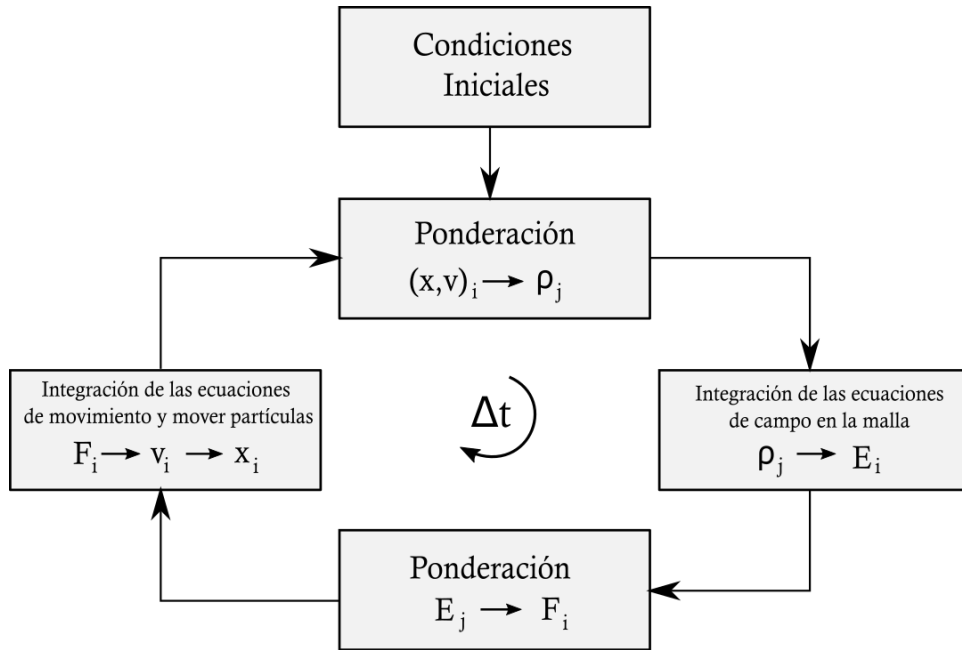


Figura 3.1: Un ciclo computacional realizado por PIC. El subíndice  $i$  y  $j$  corresponden a propiedades en las partículas y en los puntos de malla respectivamente.

a NGP, la densidad y el campo eléctrico resultante tendrán un ruido lo suficientemente tolerable para la mayoría de simulaciones. Aquí el peso de cada partícula es asociado a dos puntos de malla dependiendo de su proximidad como se muestra en la figura (3.2).

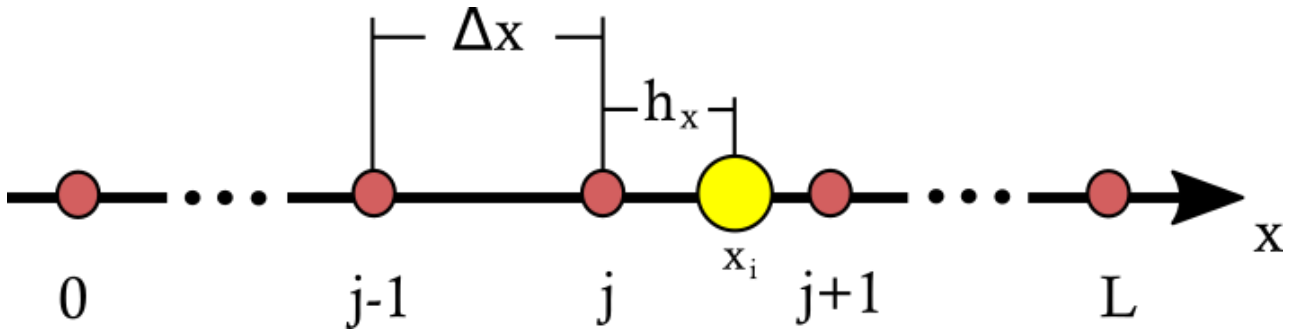


Figura 3.2: Ponderación en una malla unidimensional. Un punto rojo corresponde a un punto de malla ubicado a  $X_j = j\Delta x$ . El tamaño de la celda es  $\Delta x$ . Las distancias relativas. El punto amarillo es una superpartícula SP. La distancia de SP relativa al nodo  $j$  es  $h_x$ .

Los factores de ponderación correspondientes a los nodos  $j$  y  $j + 1$  son:

$$w_j = 1 - h_x \quad (3.1)$$

$$w_{j+1} = h_x \quad (3.2)$$

### 3.1.2. Integración de las ecuaciones de campo

Partiendo de las ecuaciones Maxwell para un problema electrostático en una sola dimensión, usaremos la densidad  $\rho$  para hallar tanto el campo como el potencial eléctrico.

$$\mathbf{E} = -\nabla\phi \quad (3.3)$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad (3.4)$$

$$\nabla \times \mathbf{E} = 0 \quad (3.5)$$

Y la ecuación de Poisson es:

$$\nabla^2\phi = -\frac{\rho}{\epsilon_0} \quad (3.6)$$

La ecuación anterior y la (3.4) pueden ser discretizadas en una malla como la de la figura (3.2) usando diferencias finitas de la siguiente forma:

$$E_j = \frac{\phi_{j-1} - \phi_{j+1}}{2\Delta x} \quad (3.7)$$

$$\frac{\phi_{j-1} - 2\phi_j + \phi_{j+1}}{(\Delta x)^2} = -\frac{\rho_j}{\epsilon_0} \quad (3.8)$$

Aplicando la transformada rápida de Fourier (FFT) a la ecuación (3.6) podemos hallar  $\phi$  o directamente a  $E$ , es decir, como  $\partial^2/\partial x^2 \rightarrow -k^2$ , entonces

$$\phi(k) = \frac{\rho(k)}{\epsilon_0 k^2} \quad (3.9)$$

Permitiendo la secuencia (3.10).

$$\begin{aligned} \text{FFT}[\rho(x)] &\rightarrow \rho(k) \\ \frac{\rho(k)}{\epsilon_0 k^2} &\rightarrow \phi(k) \\ \text{IFFT}[\phi(k)] &\rightarrow \phi(x) \\ \nabla\phi(x) &\rightarrow E(x) \end{aligned} \quad (3.10)$$

### 3.1.3. Integración de las ecuaciones de movimiento

El método de integración que se usa comúnmente es el de leap-frog (LF), el cual nos permite resolver ecuaciones diferenciales como la fuerza de Lorentz  $\mathbf{F}$ . Las primeras ecuaciones en ser integradas separadamente para cada partícula son

$$m \frac{d\mathbf{v}}{dt} = \mathbf{F} = q\mathbf{E} \quad (3.11)$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{v} \quad (3.12)$$

Estas ecuaciones son reemplazadas por ecuaciones en diferencias finitas como

$$m \frac{\mathbf{v}_{new} - \mathbf{v}_{old}}{\Delta t} = \mathbf{F}_{old} \quad (3.13)$$

$$\frac{\mathbf{x}_{new} - \mathbf{x}_{old}}{\Delta t} = \mathbf{v}_{new} \quad (3.14)$$

Dado que el método LF no calcula la velocidad y la posición al mismo tiempo, sino en intervalos de  $\Delta t/2$ , es necesario prestar especial atención en las condiciones iniciales, es decir, en  $t = t_0$ , de tal forma que siga el flujo presentado en la figura (3.3).

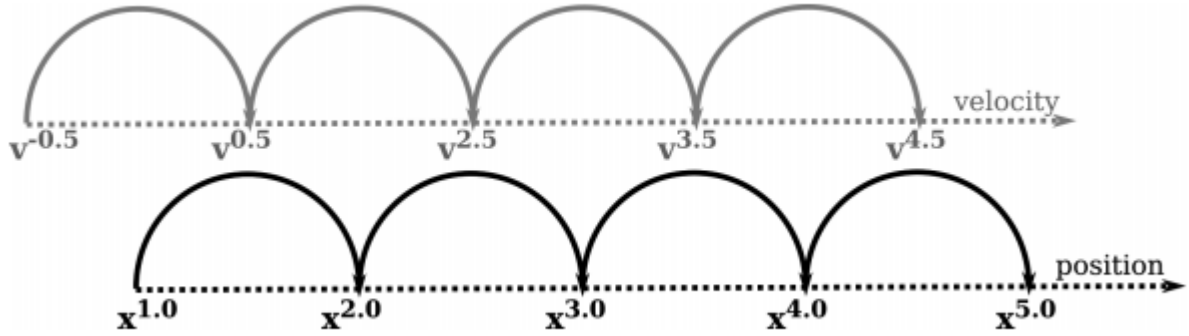


Figura 3.3: Esquema de integración del método leapfrog. [2]

A partir de este comportamiento se pueden obtener los tiempos simulados  $t_k$  y  $t_{k-1/2}$  en la que son calculadas la posición y la velocidad respectivamente, los cuales se expresan como

$$t_k = t_0 + k\Delta t \quad (3.15)$$

$$t_{k-1/2} = t_0 + \left(k - \frac{1}{2}\right) \Delta t \quad (3.16)$$

Donde  $k = 0, 1, 2, \dots$ , es una variable ficticia.

Como consecuencia de lo anterior, las ecuaciones son centradas en el tiempo de tal forma que son fáciles de entender, requieren mínimo espacio de almacenamiento y lo suficientemente precisas. Así que las ecuaciones recursivas de movimiento [26] son

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_{k+1/2}\Delta t \quad (3.17)$$

$$\mathbf{v}_{k+1/2} = \mathbf{v}_k + \frac{q}{m}E\Delta t \quad (3.18)$$

### 3.1.4. Condiciones de frontera

Es necesario revisar el comportamiento de las fronteras con el fin de recalculas las posiciones de las partículas que hayan chocado con un objeto o sobrepasado los límites definidos computacionalmente. Entre los tipos de condiciones de frontera que se pueden encontrar están los de *Dirichlet*, *Neumann*, *mixtas* y *periódicas*. Siendo la última, la implementada en el código y consiste en que las velocidades de las partículas se conserven si la posición está entre  $x < 0$  y  $x > L$  donde  $L$  es la longitud del plasma o bien, el tamaño del sistema. La posición es reasignada ya sea a  $x = 0$  o  $x = L$  dependiendo cuál frontera se haya pasado. Cabe mencionar que al implementarse el código para plasmas electrostáticos unidimensionales no colisionales, solo se aplica condición de frontera en  $x$ , de tal forma que las partículas podrían llegar a superponerse o pasarse unas encima de las otras sin afectar su trayectoria.

### 3.1.5. Normalización

La normalización tiene como objetivo de reducir el número de operaciones en las ecuaciones diferenciales. Usualmente se utilizan los parámetros  $\Delta x$  y  $\Delta t$  para normalizar el resto de cantidades físicas como, por ejemplo, el implementado por el simulador Kempo 1 [27]. A pesar de que esto podría facilitar el análisis de unidades físicas, el código utilizado en este trabajo no cuenta con una normalización de este tipo, sino un sistema unitario tal que  $\epsilon_0 = 1$ . La masa  $m_s$  y la carga  $q_s$  son calculadas a través de la frecuencia  $\omega_{ps}$  y la razón carga-masa  $(QM)_s$  (Ecs. 3.19-3.20). A su vez la frecuencia  $\omega_{ps}$  se encuentra normalizada por  $\omega_0$  que generalmente es la frecuencia predominante entre las partículas. Aquí el subíndice  $s$  diferencia el tipo de población de partícula.

$$q_s = \frac{\epsilon_0 \omega_{ps}^2 L}{(QM)_s n_{ps}} \quad (3.19)$$

$$m_s = \frac{q_s}{(QM)_s} \quad (3.20)$$

Por lo tanto, las unidades arrojadas por el simulador serán consideradas como arbitrarias a no ser que se mencione lo contrario.

### 3.1.6. Condiciones iniciales

Debido a que en PIC se trabaja con pasos temporales y espaciales finitos, sus valores deben encontrarse por debajo de cierto límite tal que el error inducido por los métodos numéricos durante la simulación sea aceptable. Inestabilidades no físicas pueden tener una tasa de crecimiento considerable si estos valores no son adecuados [28], lo cual podría darnos un diagnóstico incorrecto del plasma o bien destruir el fenómeno que se busca estudiar. Es por esto que frecuentemente [6, 29] se utilizan las siguientes condiciones

$$\omega_0 \Delta t \leq 0.2 \quad (3.21)$$

$$\lambda_D \geq \frac{\Delta x}{2} \quad (3.22)$$

Donde  $\omega_0$  es la frecuencia de normalización en radianes y  $\lambda_D$  la longitud de Debye.

Otra de las estrategias utilizadas para evitar el ruido numérico es aumentar el número de partículas dado que reducen las fluctuaciones en la densidad [2], pues éstas provocan regiones cargadas en la que se presentan campos eléctricos finitos cuando en realidad no deberían existir. Por lo tanto, cualquier partícula estacionaria en esta región será acelerada introduciendo energía cinética no física al sistema. Este comportamiento de forma colectiva es conocido como *calentamiento numérico* o *autocalentamiento*.

# Capítulo 4

## Interfaz Gráfica PEWAV

Anteriormente en el grupo de investigación GPLA se desarrollaron una serie de códigos escritos en Python para la simulación de ondas electrostáticas uni- y bidimensionales [3], **OneGPLA** y **TwoGPLA** respectivamente. Además, se adaptaron a una interfaz gráfica con la intención de tener una herramienta interactiva (Fig. 4.1). A pesar de que éste permite al usuario ingresar una cantidad importante de parámetros como lo son el tamaño del sistema, la frecuencia del plasma o el paso temporal (Fig. 4.1a), sigue necesitando la manipulación en el código interno (Fig. 4.3) por parte del usuario para ajustar el número de partículas. Otras limitaciones presentes son:

1. En el diagnóstico solo se cuentan con tres gráficas (Fig. 4.1b) que si bien podrían ayudar a identificar el fenómeno esperado, no brindan un análisis completo de lo que sucede en el plasma.
2. Falta de etiquetas en la gráfica que ayuden a identificar cuál parámetro se está observando.
3. Como no se guardan los datos en ningún archivo, no se puede realizar un post-análisis.
4. Susceptible a bugs si se desea pausar la simulación.

Con el fin de corregir estas limitaciones y de alejar al usuario de la programación, se construyó una nueva interfaz desde cero para el código OneGPLA, el cual recibe el nombre de **PEWAV** (Plasma Electrostatic Waves).

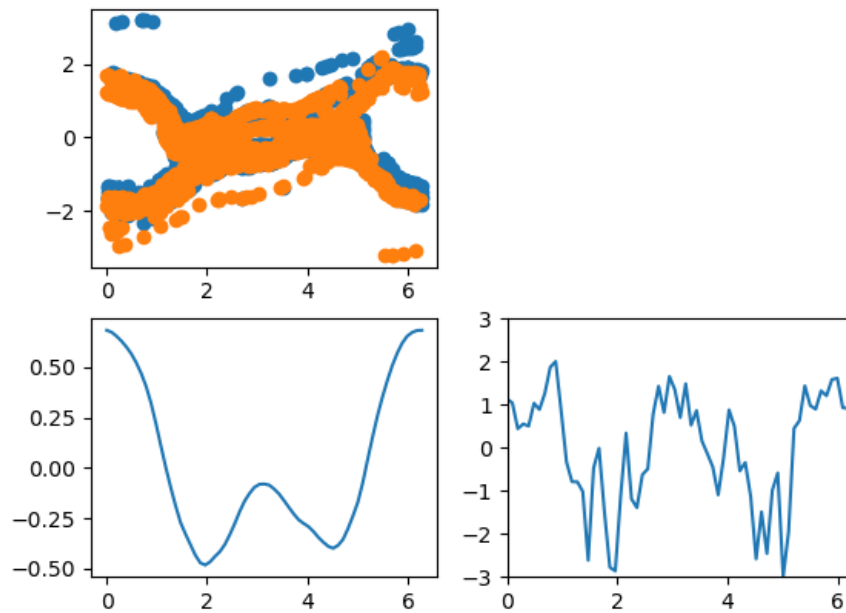
Anteriormente se contaba con tres especies de partículas, una para iones y dos para electrones, pero se decidió aumentar una extra para estos últimos. Esto se hizo para simular condiciones del plasma donde no solo existan haces o haz y una especie de electrones. Un ejemplo claro es la situación presente en la magnetósfera interna (Sec. 2.9) donde podemos encontrar tanto electrones fríos y calientes [30] como en forma de haz [1]. Aún así en el interfaz gráfica está la opción de tener dos, tres o cuatro especies. Los detalles de *PEWAV* se darán en las secciones subsiguientes.

Welcome to LikeGeeks app

Dominio:	L/pi:	<input type="text" value="2"/>	Nodos:	<input type="text" value="64"/>	Wp:	<input type="text" value="1"/>
Carga:	npCell:	<input type="text" value="20"/>	Modo:	<input type="text" value="1"/>	dx0:	<input type="text" value="0.0001"/>
Especies:	qm1:	<input type="text" value="-1"/>	v1:	<input type="text" value="1"/>	qm2:	<input type="text" value="-1"/>
Solucionador:	dt:	<input type="text" value="0.1"/>	Tolerancia:	<input type="text" value="0.000001"/>	fps:	<input type="text" value="10"/>
					Sub it:	<input type="text" value="1"/>

Gráficos:  xvFase:  Rho:  Phi:

(a) Parámetros de entrada.



(b) Gráficas del espacio de fase, potencial eléctrico y densidad de carga.

Figura 4.1: Interfaz gráfica anterior de OneGPLA [3].

## 4.1. Librerías Usadas de Python

Primero es necesario remarcar que en esta sección no aparecen todas las librerías usadas, sino las más importantes, ya sea porque se usa nomenclatura y conceptos provenientes de éstas o por el rol que jugaron en la creación y optimización de los códigos.

### 4.1.1. Numpy

Esta librería propia del lenguaje Python “Numerical Python” librería principal para la informática científica, se especializa en el cálculo y análisis numérico a gran escala, esta se usa para analizar grandes cantidades de datos, por tanto indispensable para integrar un código enfocado en simulación de plasma donde el volumen de datos entorno a las partículas es enorme, este incorpora objetos conocidos como arrays que permiten guardar datos que tengan las mismas características por tanto la manipulación de datos se hace muy eficiente por su implementación de matrices y matrices multidimensionales.

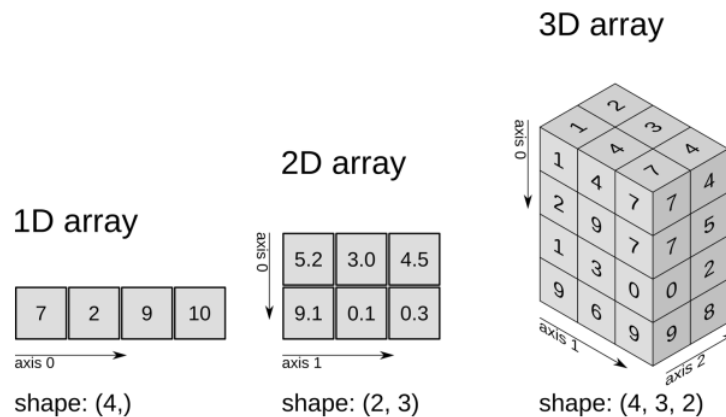


Figura 4.2: Ejemplo de arrays en numpy.

### 4.1.2. Pandas

La librería Pandas pertenece al lenguaje Python, este comando es generado para el análisis de estructura de datos, estas estructuras son más flexibles que otras librerías por tanto permiten trabajar de forma más eficiente. Esta se caracteriza principalmente por definir estructuras de datos basadas en los arreglos matriciales generados por la librería NumPy agregando nuevas funciones, tales como: leer ficheros en formato CSV o bases de datos SQL, métodos de reordenamiento o combinación de conjuntos de datos ya establecidos y finalmente trabajar con series temporales (estructuras de una dimensión, DataFrame para dos dimensión tipo tablas y Panel de tres dimensiones tipo cubos).

### 4.1.3. Tkinter

Esta librería fue usada para el modelo de la interfaz grafica, Tkinter es un binding de la biblioteca gráfica Tcl, esta librería es de las más versátiles para usar al momento de crean interfaces su programación no es muy compleja e interactuar de forma fácil con el usuario para el ingreso de datos, esta proporciona un conjunto de herramientas para administrar las ventanas disponibles. Las ventajas de usar módulos de tkinter son la velocidad de reacción y que generalmente se suministra de forma nativa con Python lenguaje en el cual se construyo el código base y el interfaz.

### 4.1.4. Matplotlib

Esta librería se usa para generar graficas a partir de datos guardados en listas, carpetas o vectores enlazados en el código usando lenguaje Python en su extensión de análisis de datos Numpy. En el caso particular de este trabajo a través de la ruta de datos se pueden seleccionar los datos preestablecidos en



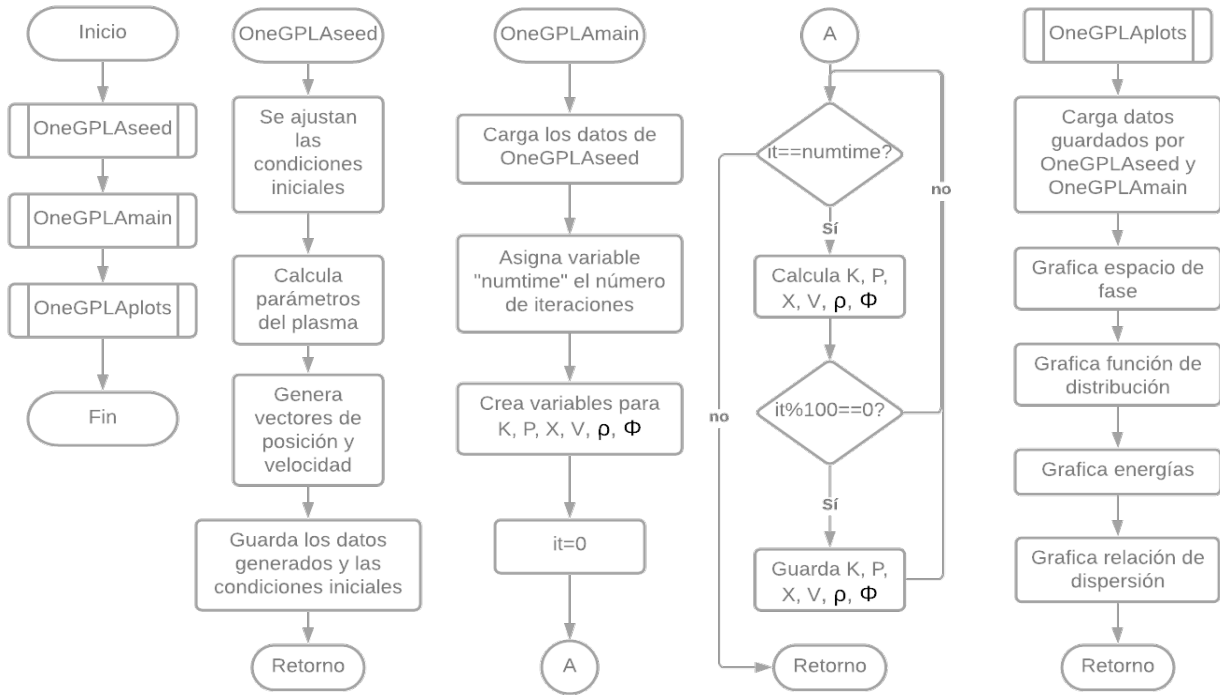


Figura 4.3: Diagrama de flujos simplificado de los códigos OneGPLA. *OneGPLAfunctions* no se muestra dado que contiene funciones dentro de sí que son llamadas por los demás módulos y su representación podría prestarse para confusiones.

la sección de configuraciones del sistema (proceso detallado en el manual de usuario) para generarlas graficas con diferentes tipos de líneas, colores, marcadores, mallas, textos en los ejes y demás agregados para terminar guardando en la ruta requerida.

#### 4.1.5. H5py

Esta librería permite guardar grandes cantidades de datos numéricos que se pueden manipular fácilmente desde la extensión matemática Numpy. Con esta función se pueden dividir conjuntos grandes de datos incluso teniendo capacidad para dividir Tera Bytes, el uso de esta librería se hace indispensable por las grandes cantidades de datos que se guardan por las miles de partículas que se simulan y los datos que se guardan; Por tanto se puede concluir que la característica principal es almacenar miles de conjuntos de datos en un único archivo y etiquetarlo como se requiera.

## 4.2. Adaptación de los códigos OneGPLA

Los códigos que conforman el programa OneGPLA son cuatro módulos principales, *OneGPLAseed*, *OneGPLAmain*, *OneGPLAplots* y *OneGPLAfunctions* que en conjunto cumplen con el diagrama de flujos mostrado en la figura 4.3. En realidad, cada módulo a excepción de *OneGPLAfunctions* es ejecutado por separado, pero secuencialmente. Primero, se ejecuta *OneGPLAseed* que genera las condiciones iniciales y guardarlas en archivos .np. Luego éstos son leídos por *OneGPLAmain* quien calcula y guarda las siguientes magnitudes físicas: el campo eléctrico  $E$ , la energía cinética  $\mathcal{K}$  y potencial  $P$ , el potencial eléctrico  $\phi$ , la posición  $X$ , velocidad  $V$  y la densidad de carga  $\rho$ . Por último, *OneGPLAplots* carga estas magnitudes físicas y las grafica. ¿Y en todo esto dónde quedó el último módulo? Bueno, él ha estado siendo llamado en cada uno de los anteriores, ya que contiene tanto las funciones de PIC (Sec. 3.1) como las necesarias para graficar la función de distribución  $f(v)$  y la relación de dispersión  $\omega - k$ .

Como se mencionó anteriormente, los códigos son ejecutados por separado y secuencialmente de forma manual, pero esto no es viable si se desea una interfaz gráfica. Por lo que la tarea estuvo en adaptar los módulos a nuestras necesidades y de allí se obtuvieron tres códigos, dos “remasterizados” y uno nuevo.

El primer código “remasterizado” fue el de *OneGPLAfunctions* que pasa a llamarse *ROneGPLAfunctions*, pues se eliminaron y ajustaron funciones (subrutinas), por lo que su composición se mantuvo cercana a la original. El segundo corresponde a los módulos *OneGPLAseed* y *OneGPLAmain* que ahora se fusionan en uno solo, *ROneGPLA\_AIO*.

El nuevo código es *PEWAV* quien contiene toda la estructura necesaria tanto para iniciar la simulación del plasma como para guardar y graficar ya sean resultados nuevos o anteriores. Este código se ejecuta en forma de interfaz gráfica y por tanto considerado el principal.

En las secciones posteriores se explicará de forma general el contenido de cada uno y el papel que tienen.

## 4.3. Composición

### 4.3.1. ROneGPLAfunctions

Se puede considerar que este código es el corazón del simulador, dado que contiene todas las funciones necesarias no solo para llevar a cabo el método PIC, sino también para calcular la función de distribución y la relación de dispersión. En la tabla 4.1 se presenta un resumen de las funciones que están en su interior.

Este módulo es cargado por *ROneGPLA\_AIO*, el cual le pasa todos los parámetros necesarios para su funcionamiento.

### 4.3.2. ROneGPLA\_AIO

Aquí se reúnen las dos partes que se encargan de aprovechar las funciones descritas en el apartado anterior y además de usar los parámetros ingresados mediante la interfaz gráfica para la simulación del plasma. Este módulo se compone de las clases *seed* y *main* que a su vez contiene atributos y métodos. A continuación se da una explicación más detallada de cada uno.

- Clase *seed*

Tiene como objetivo crear, guardar y pasar la “semilla” de los datos. Las dos primeras acciones se deben a que cada proyecto de simulación contará con posiciones y velocidades diferentes a pesar de que los parámetros del plasma sean los mismos. Por último, se devuelve la semilla al programa principal. Aunque por el momento no se pueda cargar el archivo guardado para pasarla directamente a la clase *main*, se podría utilizar para realizar un análisis apartado sobre las condiciones iniciales del espacio de fase.

Cuando se crea una instancia de la clase y se asigna el objeto a una variable local, se requiere pasar los siguientes parámetros:

- *datos\_sistema* → es un *DataFrame* constituido por valores “técnicos” del programa. Éstos son  $N_{IT}$ ,  $L$ ,  $\Delta t$ ,  $\epsilon_0$ , *iones\_dinamicos*<sup>1</sup>,  $P_2^2$ ,  $P_3^3$ , *modo*<sup>4</sup>, la perturbación inicial  $\delta x_0$  y el número de nodos de velocidad  $N_{pvel}$ <sup>5</sup>.
- *datos\_particulas* → es un *DataFrame* que contiene  $n_p$ ,  $v_d$ ,  $v_t$ ,  $(QM)$  y  $\omega$  para cada población de partículas.

<b>Función</b>	<b>Resumen</b>
<b>Maxwell</b>	Retorna valor de la parte exponencial de la función de distribución de velocidades (ec. 5.1) para un punto $v$ dada la velocidad de deriva $v_{ds}$ y la térmica $v_{ts}$ .
<b>CargarRandomPosicion</b>	Genera un vector de posiciones aleatorias (usando una distribución uniforme) dado $L$ y el número de partículas $n_{ps}$ .
<b>CondicionesDeFrontera</b>	Se usa en los vectores de posición $X_s$ de las SP tal que siempre estén en el intervalo $[0, L]$ .
<b>CargarVelocidadMaxwell</b>	Genera un vector de velocidades $V_s$ dada $n_{ps}$ , $v_{ds}$ , $v_{ts}$ y la densidad lineal de partículas $n_{ls}$ . Para lograrlo llama a <b>Maxwell</b> .
<b>densidad</b>	Calcula la densidad de carga $\rho$ dado el paso temporal $\Delta x$ , los nodos de malla $N_g$ y $parts$ .
<b>potentialFourier</b>	Calcula el potencial eléctrico $\phi$ a través de la FFT dado $N_g$ , $\Delta x$ , $\rho$ y $\epsilon_0$ .
<b>campo</b>	Calcula el campo eléctrico $E$ dado $N_g$ , $\phi$ y $\Delta x$ .
<b>movimiento</b>	Utiliza la ponderación usada para $E$ y así mover las partículas utilizando el método LF.
<b>energias</b>	Calcula la energía cinética y potencial dado $N_g$ , $\rho$ , $\phi$ y $parts$ .
<b>stepion</b>	Utiliza las funciones anteriores y devuelve los vectores de $\rho$ , $\phi$ , $E$ y $parts$ . Tiene como parámetros de entrada $N_g$ , $\Delta t$ , $\epsilon_0$ , $L$ , $parts$ , $partsi$ y la iteración actual $it$ .
<b>stepmoveion</b>	Similar a <b>stepion</b> , pero esta vez se mueven los iones.
<b>nextpow2</b>	Calcula la potencia más cercana de dos para $L$ .
<b>fdv</b>	Calcula los valores de la función de distribución al ver en qué posición se encuentra la velocidad del nodo más cercano, luego realiza una suma de porcentajes en estos puntos.
<b>relacion_dispersion</b>	Calcula la relación de dispersión dado $N_g$ , $\Delta t$ , $E$ , $L$ y el número total de iteraciones $N_{IT}$ .

Tabla 4.1: Resumen funciones de *ROneGPLAfunctions*.

- *folders* → es un valor entero del número de carpetas existentes de los resultados arrojados anteriormente por el simulador.

Una vez hecho esto, se ejecuta el método `__init__` que desempaqueta los parámetros y asigna a variables de instancia. Luego, manualmente se utiliza el atributo `generate_seed` que retornará la semilla en forma de diccionario. Un resumen de los atributos pertenecientes a esta clase, se presenta en la tabla 4.2.

<sup>1</sup>Indica si se tienen iones estáticos o no.

<sup>2</sup>Indica si se usa población extra de electrones.

<sup>3</sup>Cumple la misma función que  $P_2$ .

<sup>4</sup>Modo de la perturbación.

<sup>5</sup>Cantidad relacionada con los puntos graficados horizontalmente para  $f(v)$ .

Atributo	Resumen
<b>precalculations</b>	Calcula $\Delta x$ , $n_{ls}$ , $N_g$ , $m_s$ , $q_s$ , $(QM)_s$ , el número total de partículas $N_T$ , el número total de electrones $N_e$ .
<b>generate_seed</b>	Utiliza <b>precalculations</b> para crear los vectores de posición, velocidad, carga, masa y carga-masa. Si hay más de una población de electrones, sus vectores se concatenan en uno solo. Por último, utiliza <b>save_seed</b> y retorna un diccionario (llamado <i>all_parts</i> ) con los vectores anteriores, además de $N_e$ y el número de iones $n_{pi}$ .
<b>save_seed</b>	Guarda los vectores creados en <b>generate_seed</b> en un archivo .hdf5 con el nombre de “semilla”.

Tabla 4.2: Resumen atributos de la clase *seed*.

- Clase *main*

En esta clase se encuentra el ciclo de  $N_{IT}$  iteraciones donde se calculan y guardan los datos que necesitamos para realizar el diagnóstico del plasma.

Al igual que con *seed*, es necesario iniciar una instancia de *main* para poder utilizar sus atributos (Tabla 4.3).

Atributo	Resumen
<b>timer</b>	Guarda en un archivo .txt el tiempo (en segundos) en terminar la simulación. También escribe el valor de $N_{IT}$ .
<b>terminate</b>	Detiene la simulación.
<b>create_var</b>	Crea <i>grupos</i> y <i>datasets</i> en un archivo .hdf5 donde se almacenarán los datos.
<b>initiate</b>	Usa los anteriores atributos. Inicia la simulación. Calcula y guarda los siguientes datos: $X$ , $V$ , $\rho$ , $\phi$ , $E$ , $\mathcal{K}$ , $P$ , $k$ , $\omega$ , $f(v)$ . Necesita como parámetros de entrada <i>datos_sistema</i> , <i>all_parts</i> y <i>folders</i> .

Tabla 4.3: Resumen atributos de la clase *main*.

En el código mostrado abajo se presenta de forma simplificada su funcionamiento dentro de *PEWAV*. Debido a como fue estructurado *ROneGPLA\_AIO*, éste puede usarse sin la interfaz gráfica siempre y cuando se le suministren los parámetros con el formato adecuado.

```

1 import ROneGPLA_AIO as RAI0
2
3 # Suponiendo que ya contamos con ‘‘datos_sistema’’,
4 # ‘‘datos_particulas’’ y ‘‘folders’’.
5
6 # Crea instancia para seed
7 semilla=RAI0.seed(datos_sistema,datos_particulas, folders)
8 # Genera la semilla
9 all_parts=semilla.generate_seed()
10 # Crea instancia para main
11 main_pic=RAI0.main()
12 # Inicia la simulacion
13 main_pic.initiate(datos_sistema,all_parts,folders)

```

### 4.3.3. PEWAV

Este código contiene los elementos de la interfaz gráfica y usa los anteriores como módulos. Es por esta razón que se considera el código principal y el único a ejecutar.

Debido a que *PEWAV* es en su mayoría programación de la interfaz gráfica (widgets, eventos y excepciones), solo se menciona su estructura general, la cual se divide en tres clases, la primera contiene a las dos secundarias y éstas a su vez, el código de las pestañas “Configuraciones” y “Gráficas” (Fig. 4.5).

- Clase *Config\_page*

El propósito de esta clase es controlar todas las funciones dentro de la pestaña “Configuraciones” de las cuales destacamos aquellas que involucran a *ROneGPLA\_AIO*.

Cuando el usuario ha ingresado las condiciones del plasma y presiona el botón “Cargar condiciones iniciales”, internamente se crean y guardan *DataFrames* con estos datos. Luego, al presionar el botón “Iniciar”, los *DataFrames* anteriores son cargados mediante *ROneGPLA\_AIO* quien comenzará la simulación en un hilo aparte para evitar que la interfaz gráfica se congele.

- Clase *Results\_page*

Esta clase se encarga de controlar todas las funciones dentro de la pestaña “Gráficas”. Se destaca la función del botón “Cargar datos”, el cual utiliza el archivo .hdf5 generado en la simulación, para mostrar gráficamente los resultados.

- Clase *Application*

Contiene la ventana principal en donde son agregadas las pestañas, por lo tanto, usa las dos clases anteriores.

## 4.4. Manual de Usuario

### 4.4.1. Introducción

Parte importante del trabajo realizado, fue el desarrollo de una interfaz gráfica como mecanismo interactivo, que permita realizar simulaciones de plasma para su estudio y análisis con un fin académico. Unas de las problemáticas identificadas para el desarrollo de esta tesis fue el no contar con una interfaz de usuario que fuera fácil de usar donde la interacción fuera lo más intuitiva posible. Como solución se creó una interfaz gráfica con todos los elementos posibles del software. Esta ventana entre usuario y código contiene los elementos típicos, una barra de ayuda donde podrá ingresar directamente desde el ambiente a un documento soporte que solucionara las dudas sobre cómo usar apropiadamente el ambiente, ventanas de fácil lectura donde se podrán agregar los datos más relevantes para el proceso de simulación de plasma, elementos que al momento de la ejecución permitirán guardar y analizar los resultados gráficos tal como se verá en el desarrollo de este numeral.

### 4.4.2. Ejecución de la Interfaz: Introducción al Entorno

Al ejecutar el código *PEWAV*, se abrirá la interfaz gráfica mostrada en la figura 4.5. Primero hablaremos de la funcionalidad de cada uno de sus botones y el proceso que realiza al ejecutarse.

- *Ruta de datos*

En esta opción se puede elegir la carpeta que contendrá el programa, también permite definir el espacio donde se guardaran los resultados obtenidos a través de las simulaciones, de forma que permita al usuario tener un orden concreto de donde se encuentra la información. Definir una adecuada ruta de datos le brindará al usuario facilidad para el análisis posterior de datos.

- *Entorno*

Al momento de ejecutar el programa, el entorno se divide en diferentes secciones. Primero tenemos el esquema izquierdo de **configuraciones** que cumple la función de contener todas las opciones. El apartado **Sistema** contiene el conjunto inicial de opciones disponibles, primero **número de iteraciones** cuyo valor es arbitrario, pero que influye en el número de ciclos del plasma. Para el **tamaño de sistema** se indica la longitud de plasma, en el **número de celdas** se elige la cantidad de celdas que componen la malla computacional. En el **paso de tiempo** se puede ingresar el tiempo en el cual ocurrirán las iteraciones seleccionadas, se recomienda un valor menor a 0.2 s para tener resultados lo más exactos posibles. En la **perturbación inicial** tenemos un proceso esencial para simulaciones de plasma, pues siempre se necesita modificar la posición inicial de las partículas ubicadas de forma sinusoidal, teóricamente podría interpretarse como la amplitud de la onda generada. Seguidamente se ingresa la **permitividad eléctrica** que es un valor constante y finalmente los **nodos de velocidad** indican un cálculo del número de celdas para definir la función de distribución que debe tener un tamaño de puntos definidos, esta velocidad en la que se distribuyen los puntos permite que entre más puntos tenga más definida será la función, se recomienda usar la misma cantidad de número de celdas.

#### 4.4.3. Elementos de la Interfaz : Condiciones Iniciales

En el apartado de las condiciones iniciales tenemos las opciones de las **configuraciones de iones** (donde también está la opción de elegir iones dinámicos) y configuraciones de partículas en general, haciendo especial énfasis en que la interfaz permite ampliar dicha opción seleccionando “**Habilitar partículas 2**” o “**Habilitar partículas 3**”. Cuando se selecciona la configuración se habilita una ventana de cinco opciones, en el primero se ingresa el número de partículas opción vital para simular entornos del plasma, velocidad de deriva, velocidad térmica y finalmente los valores estándar de carga-masa y la frecuencia características de la partícula.

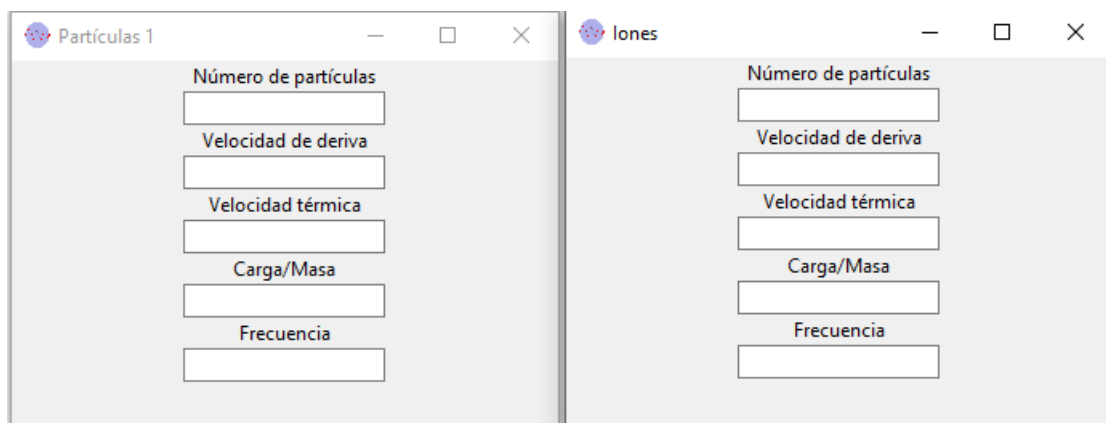


Figura 4.4: Datos Partículas

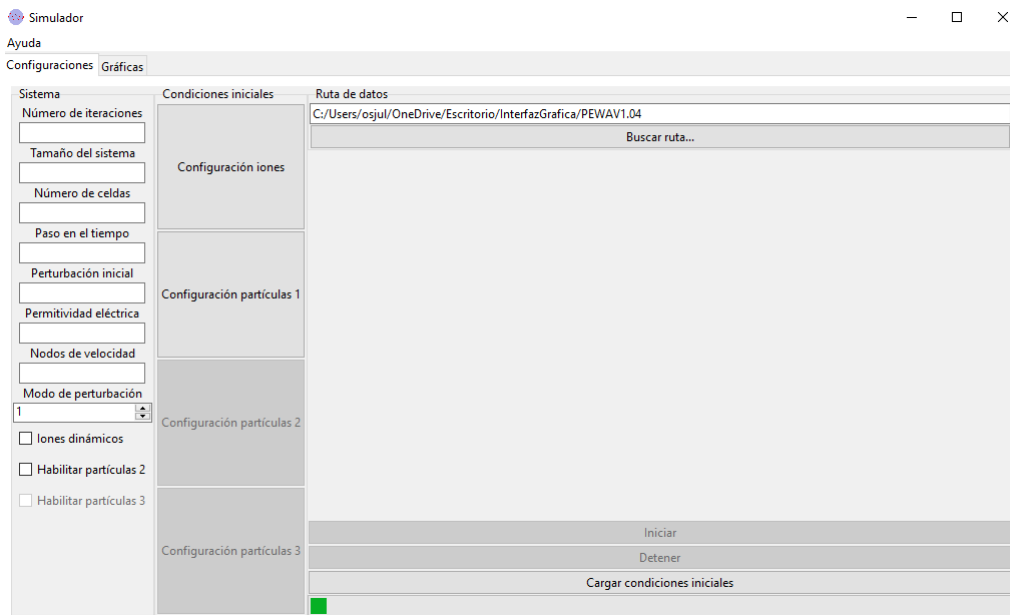


Figura 4.5: Entorno interfaz gráfica

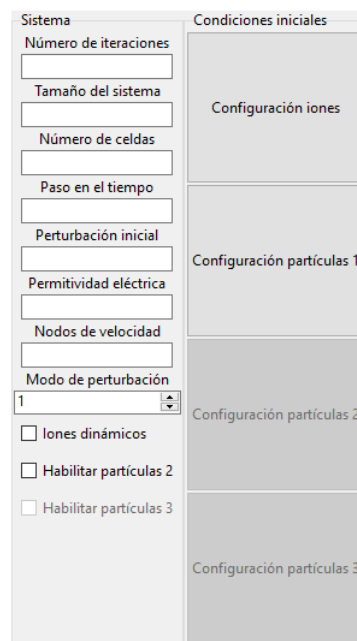


Figura 4.6: Apartado: Sistemas-Condicionales Iniciales

#### 4.4.4. Elementos de la Interfaz: Gráficas

En la segunda opción de “Gráficas” se cuenta con cinco apartados gráficos correspondientes para: **Espacio de fase, Campo eléctrico, Potencial eléctrico, Función de distribución y Densidad** (Fig. 4.8). Cada gráfica cuenta con sus ejes numerados y con la escala de magnitud apropiada. En la parte superior se tienen cuatro botones, uno se usa para cargar los datos, en este punto se puede acceder a cualquier carpeta de datos lo importante es que se cuente con las carpetas y archivos necesarios para ejecutar el programa. Dependiendo de la cantidad de datos ingresados el sistema tardara un tiempo para mostrar los resultados gráficos, entre mayor cantidad de datos hayan mas tardara en cargar.

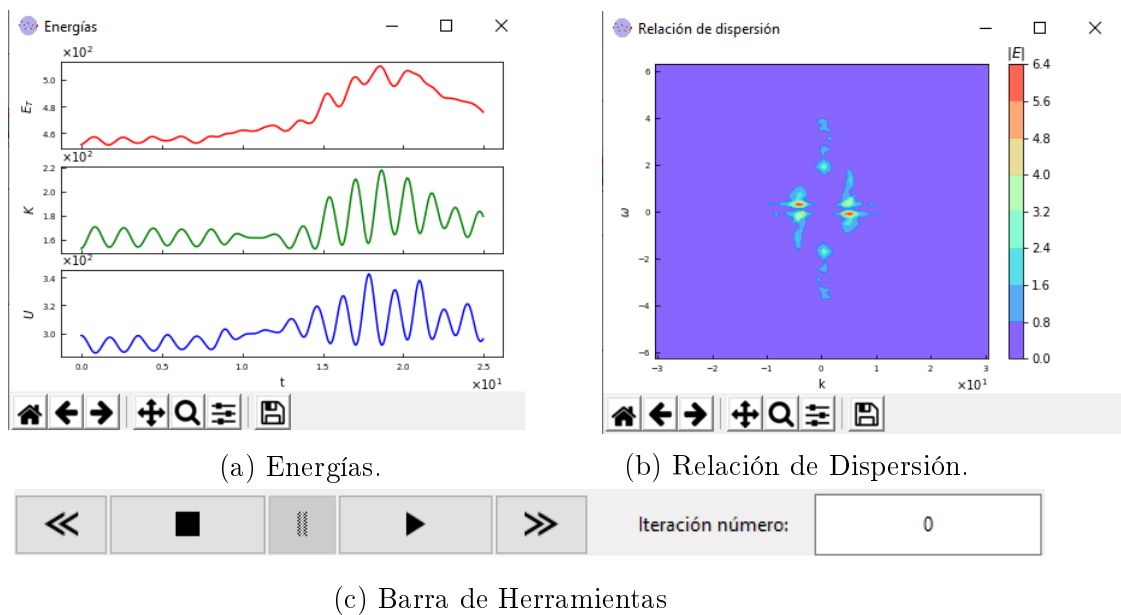
- *Funciones: Energía y Relacion de dispersión*

Las pestañas de energías (Fig. 4.7a) y de la relación de dispersión (Fig. 4.7b) cuenta con una barra de herramientas predeterminada para realizar varias funciones, la primera función guardara la imagen producida, las demás pueden realizar zoom en las áreas de importancia, configurar los colores y la definición de la misma y finalmente resetear la imagen a su estad inicial.

- *Barra de herramientas*

La interfaz cuenta con una barra de herramientas ubicada en la parte inferior derecha del entorno (Fig. 4.8), donde se podrán manipular iteraciones de los datos cargados para adelantar o retroceder la iteración. Un botón para pausar o activar el proceso de simulación gráfica, y una opción para digitar el número de iteración exacta que se quiere analizar, incluyendo el valor de cero para reiniciar el proceso (Fig. 4.7c).

Por último, se muestra en la figura 4.9 un ejemplo de como se ven las gráficas al cargar los datos.





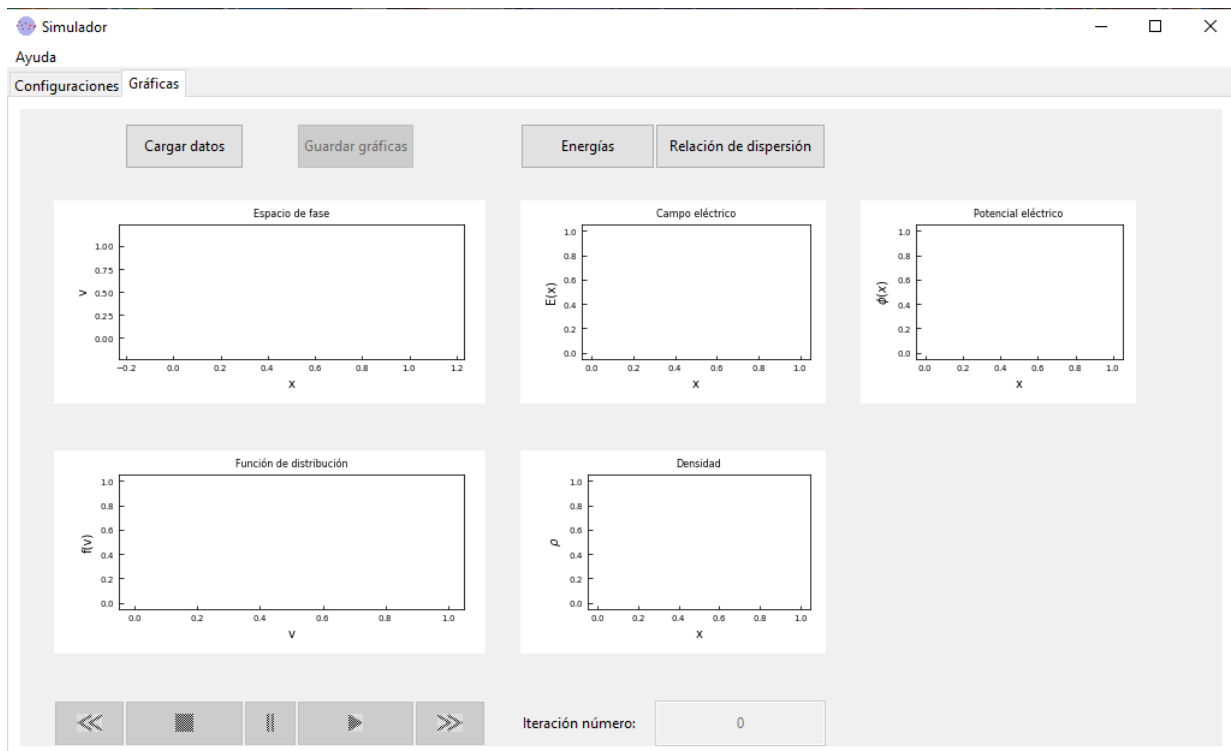


Figura 4.8: Gráficas.

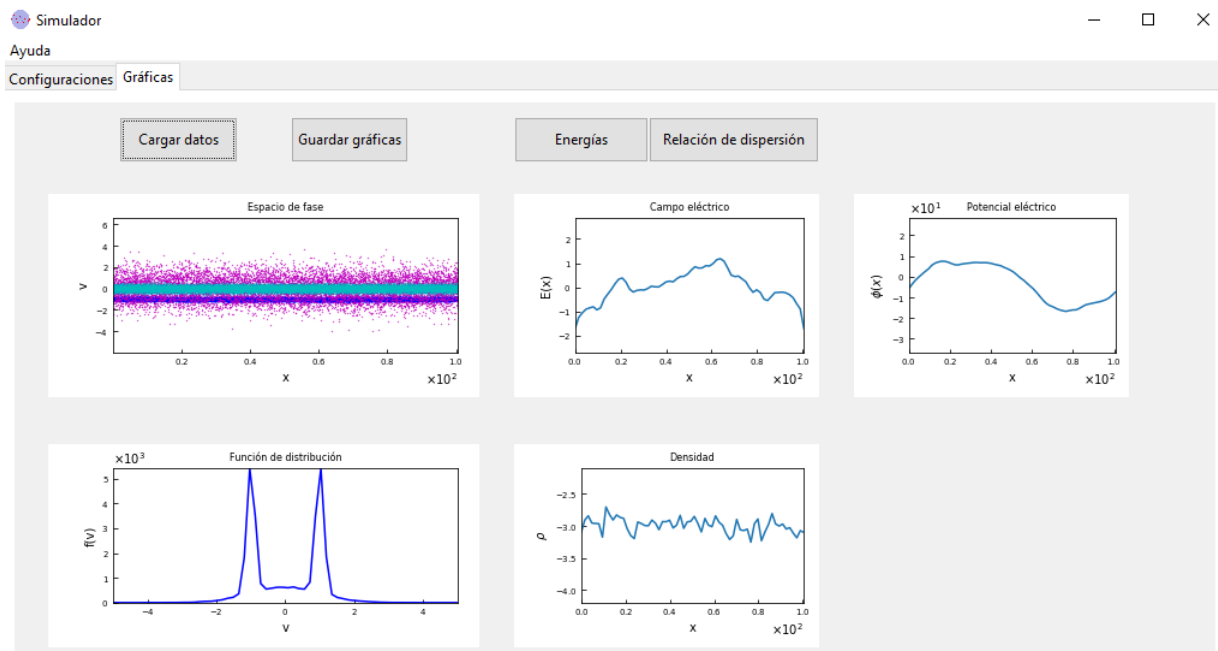


Figura 4.9: Ejemplo del aspecto de la interfaz en funcionamiento después de haber cargado los datos simulados.

# Capítulo 5

## Simulaciones de plasmas electrostáticos en PEWAV

### 5.1. Parámetros

Se simularon las inestabilidades vistas en la sección 2.7 con el fin de establecer si se producían las ondas solitarias electrostáticas (OSE) observadas en regiones de la magnetósfera como en la zona aural de media y alta altitud [31, 32]. Esto es posible, debido a que este tipo de ondas electrostáticas son casi paralelas al campo magnético ambiente, sugiriendo inestabilidades estrictamente electrostáticas a lo largo del campo [21]. La principal función de estas simulaciones es acercarse lo más posible a las condiciones reales del entorno de la magnetósfera donde coexisten diferentes especies de partículas, las cuales poseen distintas características tales como su velocidad, temperatura, carga, entre otras.

Se debe tener en cuenta que las aproximaciones con respecto a las condiciones presentes en la magnetosfera se hacen entorno a la adición de especies bajo diversos valores de temperatura (velocidad térmica). Todo esto con la finalidad de simular a baja escala dichos fenómenos con fines académicos, de igual manera creando los ambientes propicios para generar las inestabilidades que se analizaran durante el trabajo.

Dado que existen al menos dos poblaciones de electrones (2S) en la magnetósfera interna, frías y calientes, se simularon estas condiciones con parámetros anteriormente usados en el GPLA [3], pero esta vez con menor cantidad de partículas. Luego, se usaron tres especies de electrones (3S) con la finalidad de describir el comportamiento de este entorno con la presencia de otra población, además de comprobar si se producen OSE similares a las obtenidas en la sección 2.9.

El marco de referencia es respecto a los iones, por lo tanto, se consideran estáticos. Los iones, al igual que los electrones, siguen una distribución de velocidades maxwelliana como la mostrada en la ecuación (5.1).

$$f_s(v) = \frac{n_{ps}}{\sqrt{2\pi}v_{ts}} \exp \left[ -\frac{(v - v_{ds})^2}{2v_{ts}^2} \right] \quad (5.1)$$

Los parámetros del sistema que se mantienen constantes en todas las simulaciones son mostradas en la tabla 5.1. Para el caso 2S y 3S, el valor de  $\Delta t$  cambia y se encuentran en las tablas 5.2 y 5.5 respectivamente.

En el circuito Run A, se realizaron tres simulaciones para 2S donde se tienen las inestabilidades two-stream, bump-on-tail y warm two-stream. En la tabla 5.3 se encuentran los valores usados tanto para  $v_{dl}$  como para  $v_{tl}$  y en la tabla 5.4, el número de partículas. Para este caso se define la razón de densidad como  $R \equiv n_2 / (n_1 + n_2)$ .

Parámetro	Símbolo	Valor
Número de iteraciones	$N_{IT}$	5000
Número de celdas	$N_{cell}$	1063
Nodos de malla	$N_g$	$N_{cell} + 1$
Longitud del plasma	$L$	$\sim 32\pi$
Tamaño celda	$\Delta h$	$L/N_g$
Frecuencia electrónica	$\Pi_e$	1
Relación de masa	$m_i/m_e$	100
Velocidad de deriva iones	$v_{di}$	0

Tabla 5.1: Parámetros del sistema generales usados en las simulaciones.

Parámetro	Símbolo	Valor
Paso temporal	$\Delta t$	0.25

Tabla 5.2: Paso temporal para la simulación de dos especies de electrones.

Run A	$v_{d1}$	$v_{d2}$	$v_{t1}$	$v_{t2}$	$v_{ti}$	Inestabilidad
1	0	2	0.1	0.1	0.2	Two-stream
2	0	2	1	0.1	0.01	Bump-on-tail
3	0	2	0.6	0.6	0.01	Warm two-stream

Tabla 5.3: Velocidades de deriva  $v_{dl}$  y térmicas  $v_{tl}$  donde  $l = 1, 2, i$  correspondientes a las dos poblaciones de electrones y a los iones para el circuito Run 1-3A.

Run A	R	$n_i$	$n_1$	$n_2$	$N_e$	$N_T$
1-2	0.50	20971	26650	26650	53300	74271
3	$\sim 0.05$	20971	50690	2620	53310	74271

Tabla 5.4: Número de partículas usadas para un sistema de dos especies de electrones.  $R$  es la razón de densidad,  $N_e$  es el número total de electrones y  $N_T$  es el número total de partículas.

En el circuito Run B, tenemos cinco simulaciones correspondientes a 3S. En las primeras tres se trabaja con una inestabilidad Two-Stream, pero se disminuye la cantidad de partículas usadas en uno de sus haces. En Run 4B, tenemos una inestabilidad weak-beam. La última simulación corresponde a una inestabilidad bump-on-tail. En las tablas 5.6 y 5.7 se resumen los parámetros utilizados para las velocidades y el número de partículas.

A diferencia del caso 2S, aquí se usa  $R = n_2 / (n_1 + n_2 + n_3)$  como la razón de densidad.

Parámetro	Símbolo	Valor
Paso temporal	$\Delta t$	0.05

Tabla 5.5: Paso temporal para la simulación de tres especies de electrones.

Run B	$v_{d1}$	$v_{d2}$	$v_{d3}$	$v_{t1}$	$v_{t2}$	$v_{t3}$	$v_{ti}$	Inestabilidad
1								
2	-1	1	0	0.1	0.1	1	0.2	Two-stream
3								
4	0	2	0.1	0.1	0.1	0.5	0.01	Weak-beam
5	0	2	0.1	1	0.1	0.5	0.01	Bump-on-tail

Tabla 5.6: Velocidades de deriva  $v_{dj}$  y térmicas  $v_{tj}$  donde  $j = 1, 2, 3$ ,  $i$  correspondientes a las tres poblaciones de electrones y a los iones para el circuito Run 1-5B.

Run B	R	$n_i$	$n_1$	$n_2$	$n_3$	$N_e$	$N_T$
1	$\sim 0.33$	39000	10000	10000	10000	30000	69000
2	0.50	52000	10000	20000	10000	40000	92000
3	$\sim 0.71$	91000	10000	50000	10000	70000	161000
4-5	0.30	55000	30000	15000	5000	50000	105000

Tabla 5.7: Número de partículas usadas para un sistema de tres especies de electrones.  $R$  es la razón de densidad,  $N_e$  es el número total de electrones y  $N_T$  es el número total de partículas.

# Capítulo 6

## Resultados

En este capítulo se muestran y analizan los resultados gráficos obtenidos del simulador *PEWAV*. Se cuentan con dos secciones principales correspondientes a las simulaciones hechas con dos y tres especies de electrones (2S y 3S) que a su vez se dividen según las inestabilidades. A través del análisis de las gráficas del espacio de fase, la función de distribución, el campo eléctrico, el potencial eléctrico y la densidad de carga, se identificaron los mecanismos de generación de las OSE.

En cuanto a las gráficas, se eligieron en cuatro instantes de tiempo diferentes para los cuales se tuvieron dos criterios. El primero es en base al comportamiento de la energía como máximos y mínimos. El otro es más subjetivo, debido a que se eligieron según el movimiento de los electrones en el espacio de fase.

Para distinguir las poblaciones de partículas presentes en el espacio de fase, se utilizaron diferentes colores. Éstos son:

- Cian  $\rightarrow$  Iones
- Azul  $\rightarrow$  Electrones # 1
- Rojo  $\rightarrow$  Electrones # 2
- Magenta  $\rightarrow$  Electrones # 3

Además de este código de colores, se les asignará nombres como “fríos” o “calientes” para resaltar su naturaleza.

### 6.1. Dos especies de electrones

#### 6.1.1. Inestabilidad two-stream

En esta simulación, contamos con dos haces de electrones fríos (azul y rojo) y una población de iones tibios. Se denotan partículas tibias, a las especies que no tiene una razón de temperatura apreciablemente alta con respecto al otro grupo de partículas.

En la figura 6.1a vemos que la función de distribución corresponde a la inestabilidad two-stream teórica (Fig. 2.6). También, se observa que la densidad de carga es negativa y no nula, debido a que la cantidad de iones y de electrones no son aproximadamente iguales. Desde  $t = 0$  hasta  $t = 4999\Delta t$ , se visualiza el proceso de formación de los vórtices o huecos de electrones en el espacio de fase que conllevan a la generación de las estructuras solitarias. Para  $t = 4999\Delta t$ , la función de distribución pierde su forma inicial debido a la termalización de los electrones quienes buscan el equilibrio térmico. Ésto se observa gráficamente a la izquierda de la figura 6.1 donde los picos se han fusionado y achatado. Los electrones atrapados forman colas de alta energía en  $f(v)$ . En este mismo instante de tiempo, se encuentran tres vórtices en el espacio de fase en las posiciones  $x \approx 30$ ,  $x \approx 40$  y  $x \approx 95$ . En la figura 6.2d, se

encuentra que el campo eléctrico tiene un comportamiento bipolar en las mismas posiciones que los huecos de electrones. A su vez este tipo de comportamiento corresponde a OSE compresivas visibles en el potencial eléctrico. Estas OSE son llamadas ondas portadoras dado que conservan su forma al propagarse. Como consecuencia, hay una disminución de densidad de carga en estos puntos por lo que es posible identificarlos geoméricamente como unos arcos con un mínimo de  $\sim -3$  hasta un máximo de  $\sim -1$ .

En la figura 6.7a, encontramos un período de crecimiento lineal en la energía potencial correspondiente a  $t \approx 0-10$ , producido por el intercambio energético entre las partículas que tienen velocidades mayores a la velocidad de fase de la onda y, la onda misma, por lo que son desaceleradas. A partir de  $t > 10$ , se presenta un decrecimiento no lineal hasta un punto de quasi-estabilidad donde las partículas entran en resonancia con la onda. Todo este proceso es conocido como el amortiguamiento de Landau visto en la sección 2.6.

### 6.1.2. Inestabilidad bump-on-tail

La inestabilidad bump-on-tail se caracteriza por su configuración en la función de distribución, donde se genera un pico abultado en la cola en la que se encuentra el haz con menor cantidad de partículas. La gráfica 2.8 comparte las características más importantes de la forma teoría de dicha inestabilidad. Es así que para su simulación contamos con una población mayoritaria de electrones estacionarios (azul) y, una minoritaria, de haz de electrones fríos (rojo), por lo que  $n_h \gg n_c$ . A diferencia de la Run 1A, aquí los iones son considerados fríos por su baja velocidad térmica.

Las condiciones iniciales mencionadas anteriormente se observan en la figura 6.3a. También, se hace necesario notar que en la evolución de  $f(v)$  y de la densidad (Fig. 6.4) se presenta ruido térmico debido a los electrones calientes. En  $t = 399 \Delta t$ , el pico correspondiente al haz frío en  $f(v)$  es absorbido por el proceso de termalización como se puede observar en el espacio de fase donde sus electrones (rojo) van perdiendo velocidad (Fig. 6.4b) En este mismo instante, se presentan unos picos prominentes en la densidad en la derecha de la figura 6.4b que corresponden a abultamientos de los electrones fríos en ciertas regiones del espacio de fase.

Al final de la simulación, en  $t = 4999 \Delta t$ , no se formaron las estructuras solitarias, pero se observan ciertos atrapamientos de partículas en el espacio de fase (Fig. 6.3d). Comparado con los resultados de otros autores [3, 21], la función de distribución no adopta una forma aplanada donde se encontraba el haz y tampoco se obtienen OSEs, lo cual puede ser debido a que el haz no contiene la cantidad suficiente de partículas, a pesar de contar con el mismo valor de  $R$  (Tab. 5.4).

En la figura 6.7b no se evidencia ningún período de crecimiento lineal en la energía, a diferencia del resultado obtenido en la Run 1A. Posiblemente debido a que la componente imaginaria de la frecuencia no es lo suficientemente grande que permita la evolución de la inestabilidad.

### 6.1.3. Inestabilidad warm two-stream

En la función de distribución de velocidades (Fig. 6.5a), podemos observar una variante de la inestabilidad two-stream, producida tanto por un haz (rojo) como por un plasma estacionario (azul). Ambas poblaciones con una velocidad térmica intermedia, reciben el nombre de electrones tibios. Esta inestabilidad se caracteriza por tener los picos típicos de la two-stream, pero con tendencia a fusionarse.

En la evolución del sistema se puede observar como ambos plasmas se compenentran en la interfaz, obteniendo una forma peculiar en el espacio de fase semejante a las olas (Fig. 6.5b). Luego, en  $t = 1599 \Delta t$ , estas “olas” colapsan en lo que aparentemente corresponde a vórtices que atrapan a los electrones, lo cual es un comportamiento típico de la generación de las OSE.

A la izquierda de la figura 6.6c, se encuentra una estructura bipolar en el campo eléctrico en  $x \approx 10$  que usualmente está ligado a una estructura solitaria en el potencial [33]. En esta misma posición, pero en la gráfica del potencial eléctrico (a la derecha de la figura 6.6c), es posible identificar una onda electrostática rarefactiva, estructura señalada en la literatura por otros autores [1, 34] y que también se obtuvo como solución de la ecuación K-dV vista en la sección 2.9. Independientemente de ser candidata a OSE, durante los siguientes instantes de la simulación, esta estructura no mantiene ni su forma ni su velocidad, por lo cual no cumple con las propiedades de las ondas solitarias.

El comportamiento de la energía en este caso es muy similar al obtenido en la Run 2A, es decir, no hay crecimiento lineal significativo en la energía total, sino un estado de equilibrio marginal, por lo que no se desarrollan estructuras solitarias.

## 6.1.4. Gráficas 2S

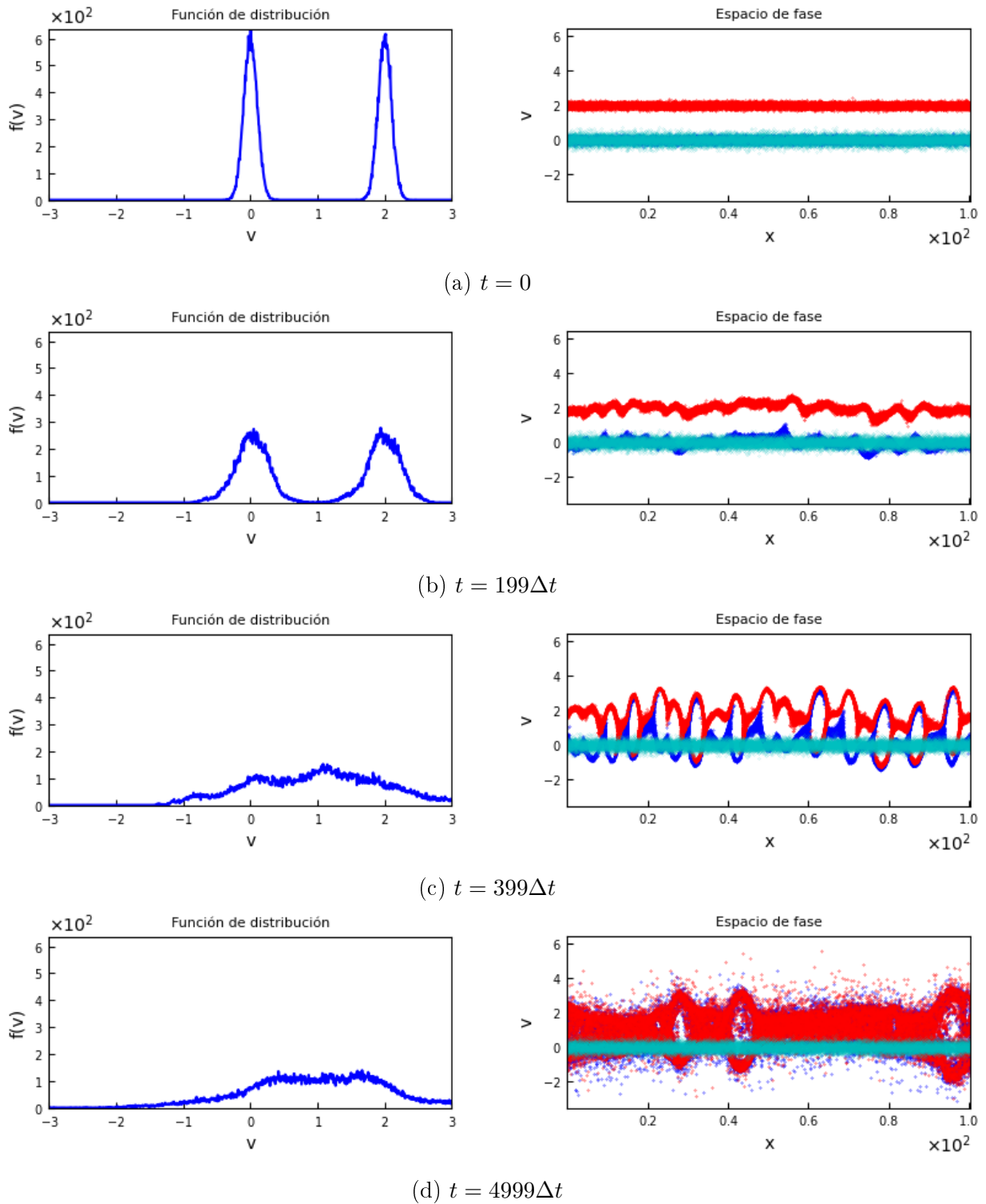


Figura 6.1: Función de distribución  $f(v)$  y gráfica del espacio de fase para Run 1A.



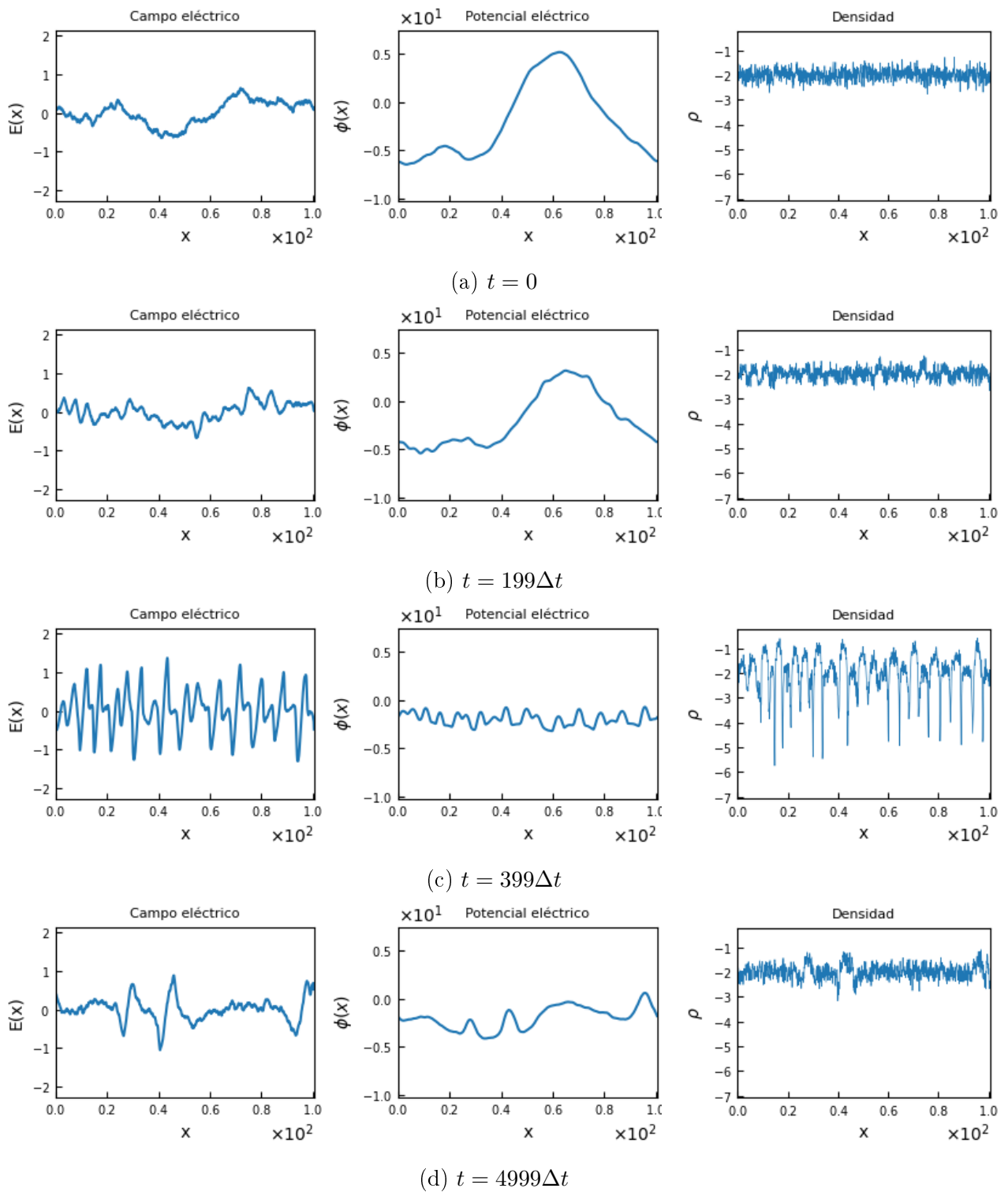
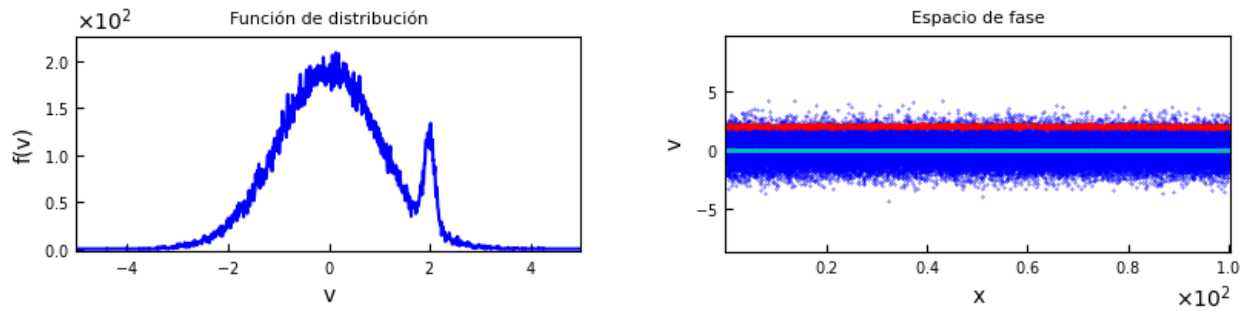
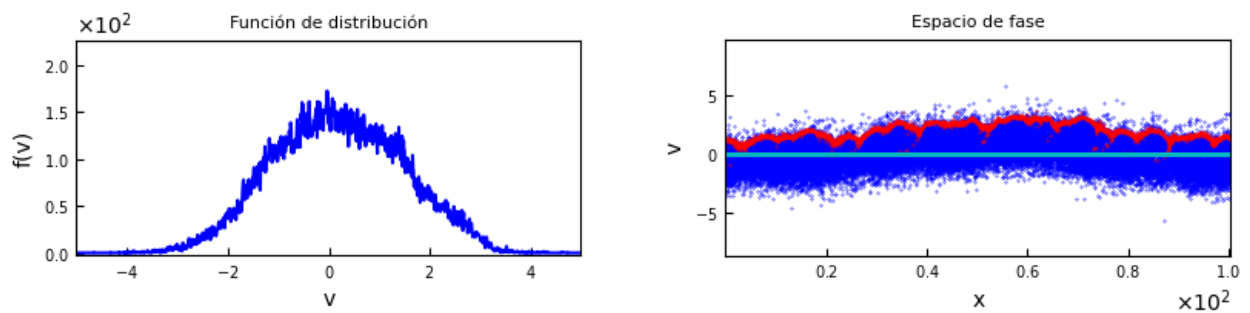
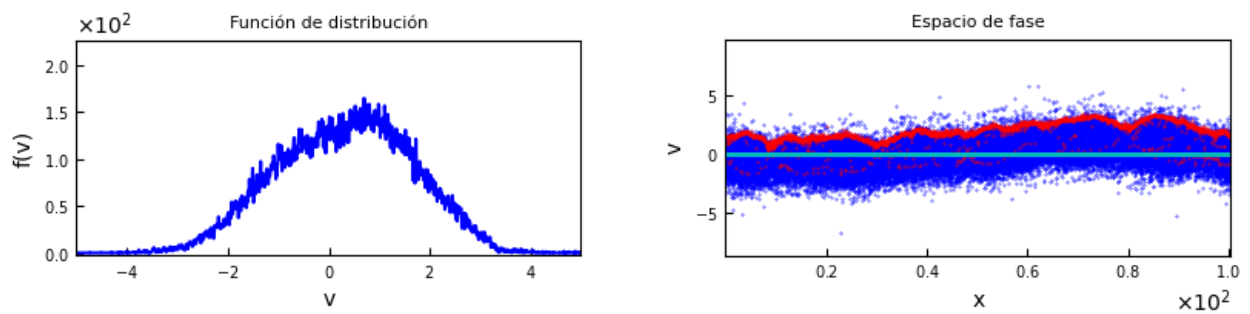
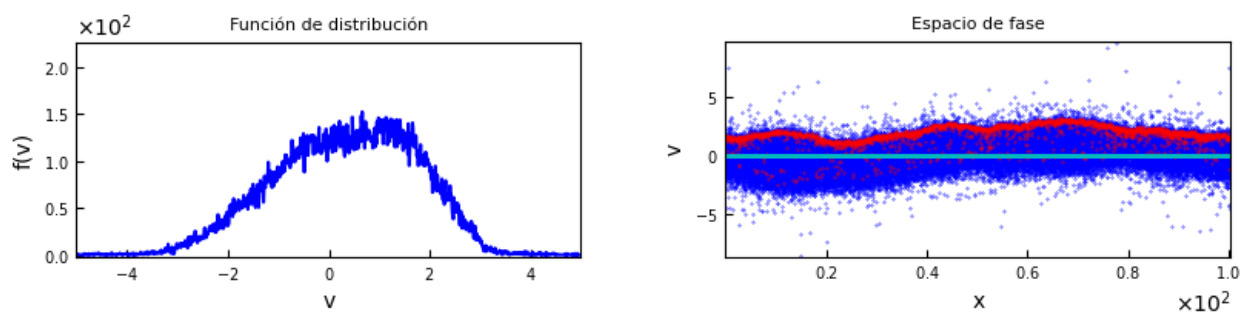


Figura 6.2: Campo eléctrico  $E$ , potencial eléctrico  $\phi$  y densidad de carga  $\rho$  para Run 1A.

(a)  $t = 0$ (b)  $t = 399\Delta t$ (c)  $t = 1110\Delta t$ (d)  $t = 4999\Delta t$ Figura 6.3: Función de distribución  $f(v)$  y gráfica del espacio de fase para Run 2A.

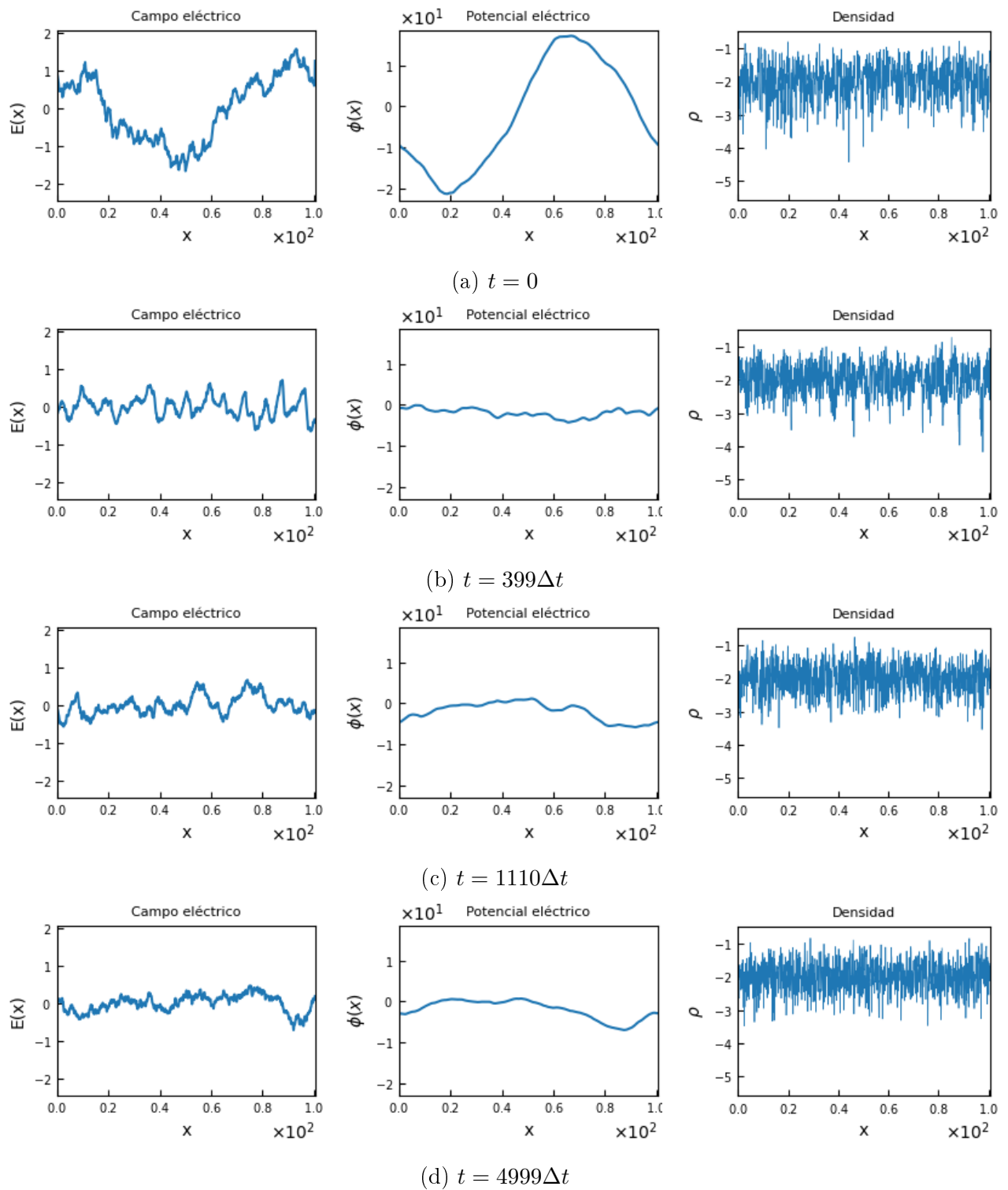
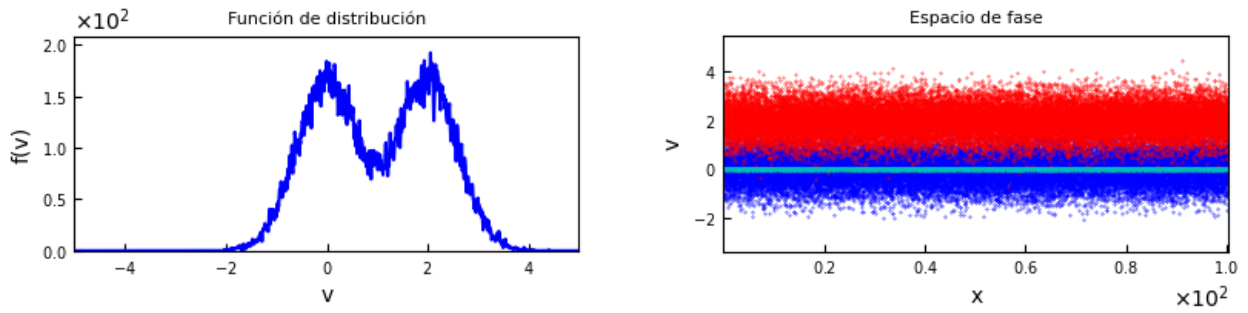
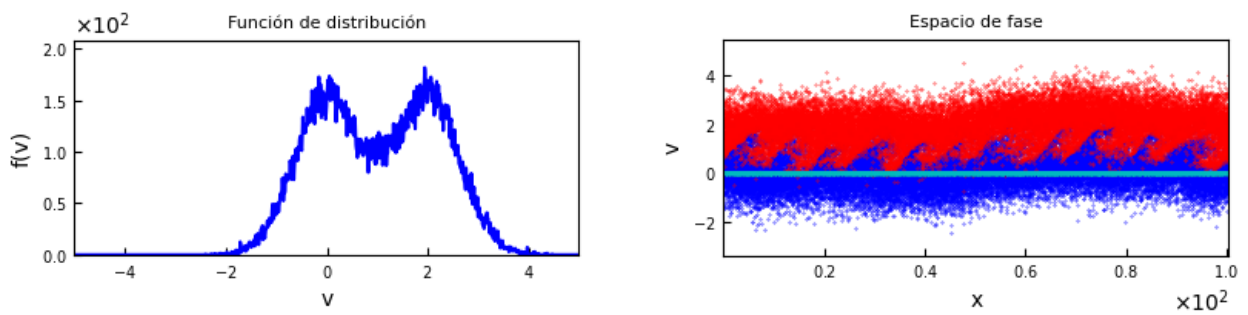
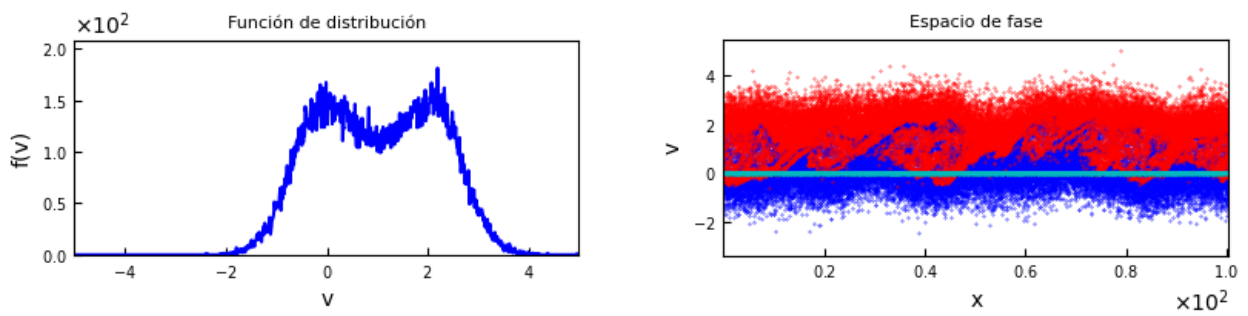
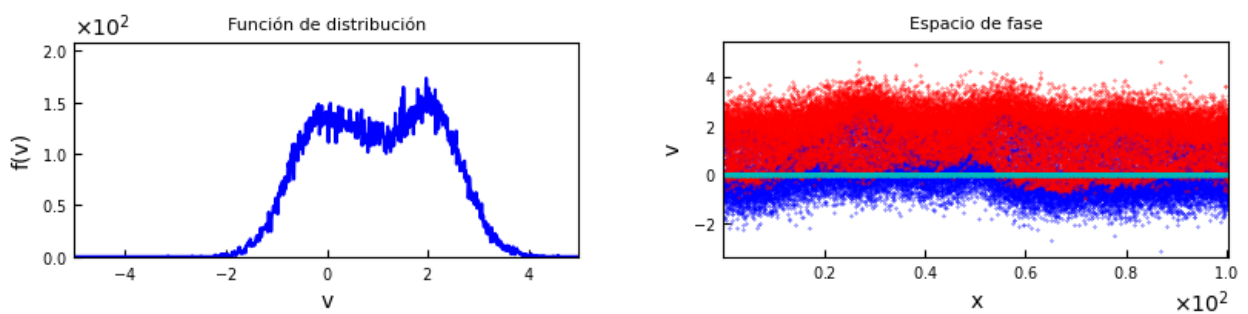


Figura 6.4: Campo eléctrico  $E$ , potencial eléctrico  $\phi$  y densidad de carga  $\rho$  para Run 2A.

(a)  $t = 0$ (b)  $t = 500\Delta t$ (c)  $t = 1599\Delta t$ (d)  $t = 4999\Delta t$ Figura 6.5: Función de distribución  $f(v)$  y gráfica del espacio de fase para Run 3A.

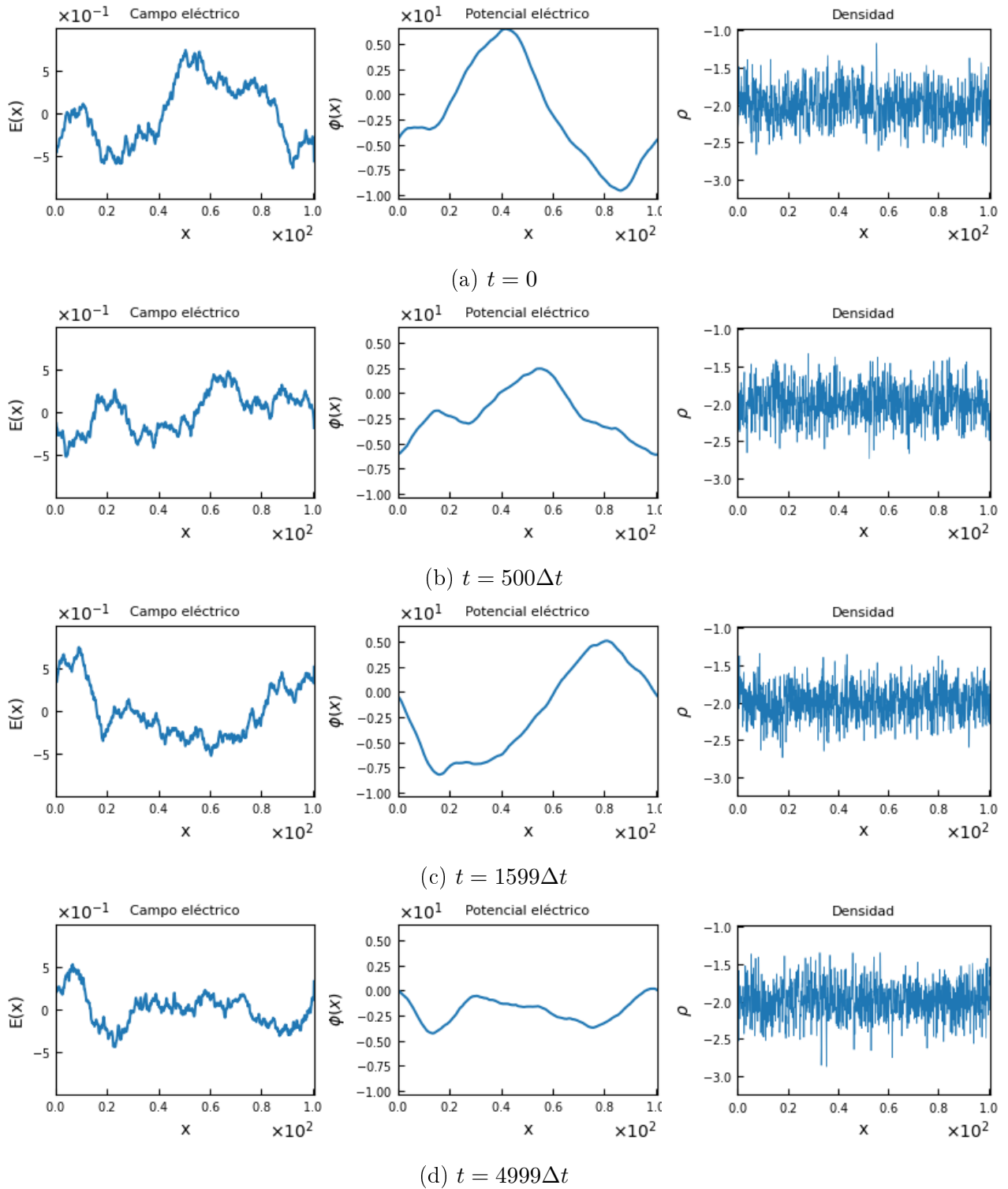
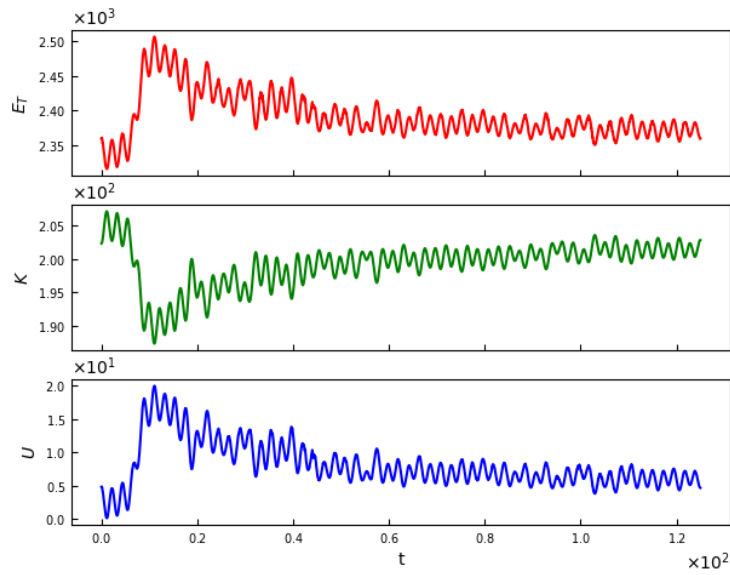
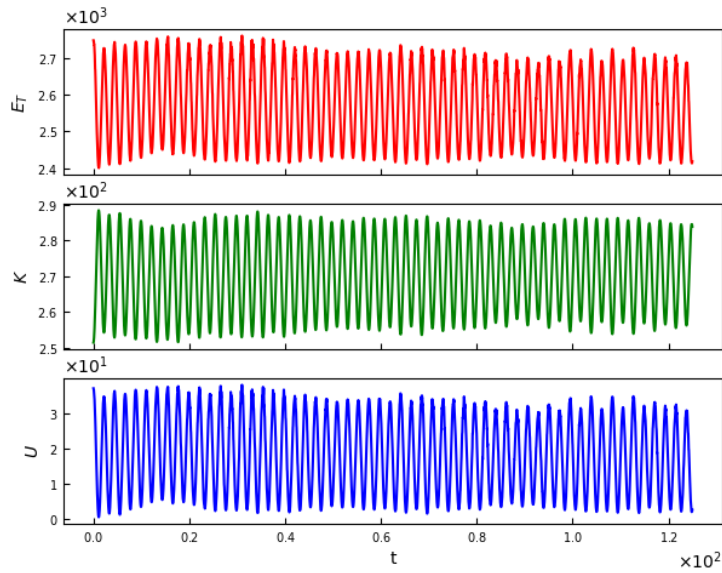


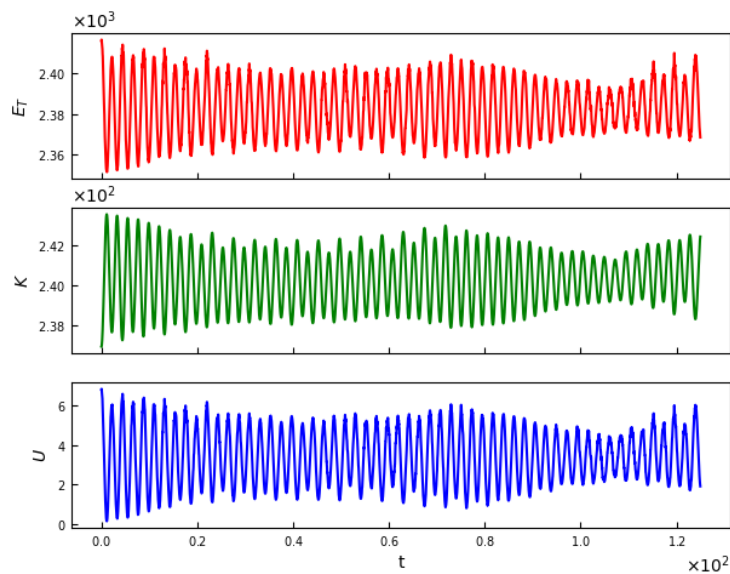
Figura 6.6: Campo eléctrico  $E$ , potencial eléctrico  $\phi$  y densidad de carga  $\rho$  para Run 3A.



(a) Run 1.



(b) Run 2.



(c) Run 3.

Figura 6.7: Energía total  $E_T$ , cinética  $K$  y potencial  $U$  de las simulaciones del circuito Run A.

## 6.2. Tres especies de electrones

### 6.2.1. Inestabilidad two-stream

Se ha estudiado la influencia de  $R$  en la formación de las estructuras solitarias en la inestabilidad two-stream, pero solo para 2S [35]. Por lo que en el caso de 3S, se varió  $R$  para así observar si existe un cambio sustancial en la formación de las OSE. Teniendo en cuenta esta razón, se realiza un análisis en conjunto de las Run 1-3B, en donde se tienen dos haces fríos y una población de electrones calientes (magenta). El haz al que se le altera la densidad es el correspondiente al color rojo en el espacio de fase. En la Run 1A se presenta el caso en donde ambos tienen la misma cantidad de partículas, pero en las simulaciones subsecuentes, se disminuyen. Esto se ve reflejado en la función de distribución de velocidades donde uno de los picos es más reducido que el otro (Figs. 6.8, 6.10, 6.12).

El impacto más notorio al variar  $R$  es en el número de vórtices presentes en  $t = 4999\Delta t$ . En la Run 1B, se tienen tres vórtices en el espacio de fase (Fig. 6.8d) que corresponden a las posiciones  $x \approx 35$ ,  $x \approx 50$  y  $x \approx 90$  y a su vez se presentan las OSEs en el potencial eléctrico (Fig. 6.9d). Por otro lado, realizando la misma observación sobre la Run 2B, se tienen solo dos vórtices en  $x \approx 10$  y  $x \approx 40$ . De igual forma, en estas posiciones hay estructuras solitarias presentes en la figura 6.10d. En la Run 3B, al igual que en la simulación anterior, hubo una disminución en el número vórtices a uno solo (Fig. 6.12d) en  $x \approx 80$ . Este comportamiento puede deberse al hecho que, al disminuir la cantidad de electrones de uno de los haces, los vórtices se fusionan entre sí a mayor velocidad. De esta forma, se demuestra la dependencia del parámetro  $R$  en la formación de las OSEs para el caso en donde se tienen tres poblaciones de partículas, dos haces fríos y una población caliente estacionaria.

Para estas tres simulaciones, se presenta el amortiguamiento de Landau en la energía que puede ser visualizado en las figura 6.18.

### 6.2.2. Inestabilidad weak-beam

La inestabilidad weak-beam se presenta cuando existe un sistema compuesto por mínimo un haz de electrones y un plasma estacionario, donde el primero tiene menor densidad que el segundo. En esta simulación se utilizan tres especies de electrones: un haz de electrones fríos (rojo), un haz leve<sup>1</sup> de electrones tibios (magenta) y un plasma estacionario de electrones fríos (azul). La función de distribución para este sistema se presenta en la figura 6.14a, de la cual cabe resaltar que los electrones calientes forman ese pequeño offset debajo de la gaussiana ubicada en el origen.

En la literatura se han realizado simulaciones con las características necesarias para generar la inestabilidad weak-beam, con la finalidad de probar que bajo ciertas condiciones ( $R = 0.05$  [21]) dicha inestabilidad no tiene rastro alguno de estructuras solitarias por la difusión de los potenciales electrostáticos. Principalmente porque el plasma más denso no es afectado por el haz de electrones. Es por este motivo que se ha decidido utilizar una razón de densidad  $R = 0.3$ , ya que para el caso 2S se ha comprobado que utilizando  $R \geq 0.3$  se producen las estructuras solitarias. A pesar de que se cuenta con una definición diferente a la empleada por Yoshiharu Omura [35], se obtuvieron OSEs en  $x \approx 20$ ,  $x \approx 40$  y  $x \approx 100$ , como se puede observar en las figuras 6.14d y 6.15d.

En  $t = 4999\Delta t$ , vemos el estado “final” del sistema, donde la función de distribución adopta una cola de alta energía correspondiente a los electrones atrapados (Fig. 6.14d), los cuales tienen una velocidad máxima  $v_{max} \approx 2v_{d2}$ . También, se puede observar en los resultados gráficos que los electrones fríos, inicialmente estacionarios, parecen ser más veloces que las demás especies. Efecto reflejado en la dispersión de las partículas sobre el movimiento de los vortices (las partículas de color azul parecen recubrir a las demás especies en movimiento). Existe una gran posibilidad de que esto sea debido a la

---

<sup>1</sup>con una velocidad de deriva igual a 0.1

gran cantidad de partículas correspondientes a esta población ( $n_1 = 30000$ ). Por otro lado, desde un punto de vista estadístico, hay más probabilidades de que las partículas ocupen altas velocidades.

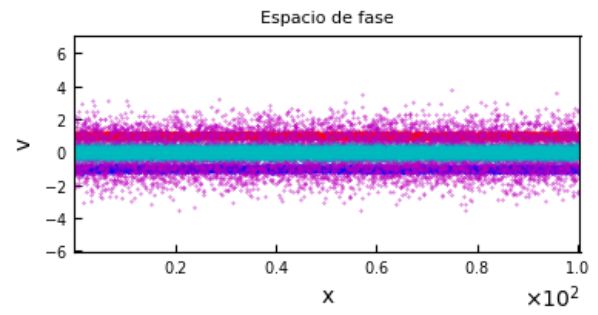
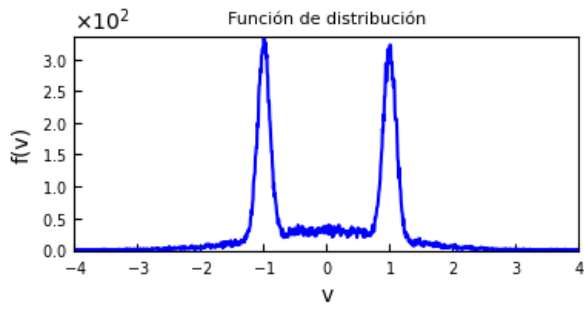
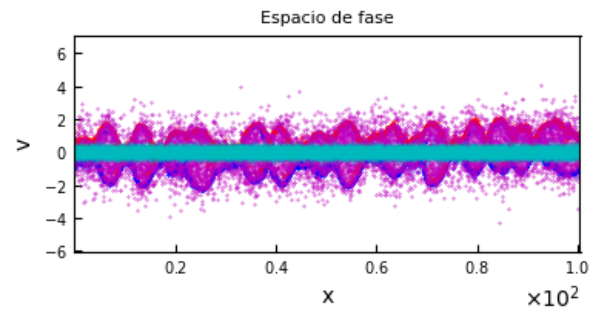
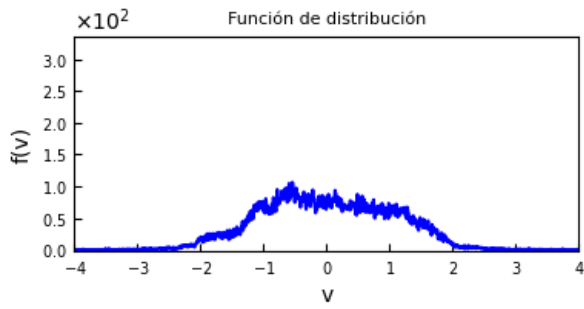
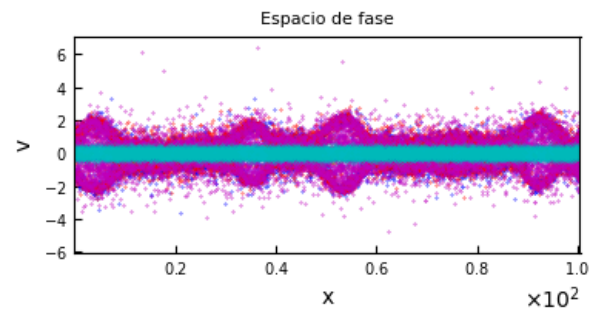
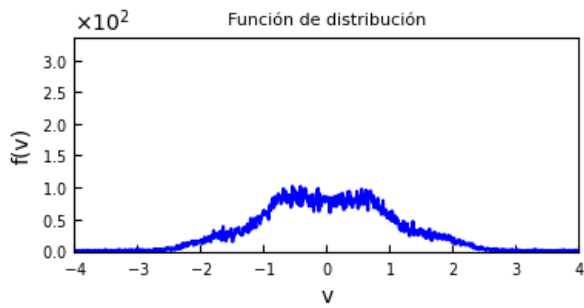
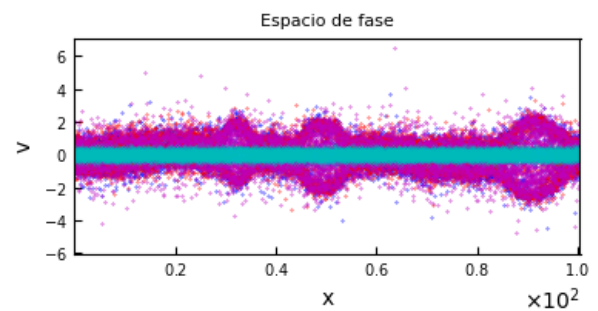
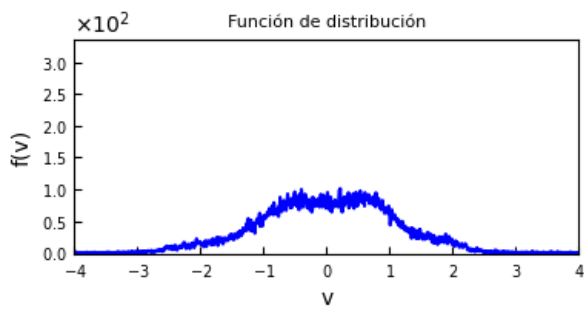
### 6.2.3. Inestabilidad bump-on-tail

El primer análisis que puede realizarse es comparar el resultado obtenido con la Run 2A, donde se generaron las condiciones para obtener la misma inestabilidad, pero con un cambio sustancial en la cantidad de partículas que la componen. A primera vista el cambio más llamativo es el tamaño del segundo pico que compone la función de distribución de velocidades, donde la elongación es mayor en la Run 5B, teniendo la cresta en el límite del espacio graficado (Fig. 6.16a). Los rangos de partículas del haz que conforma este pico son de 2.500 partículas para la Run 2A y 15.000 para este caso. Además, cabe mencionar que en esta simulación se utiliza una especie extra que corresponde a un haz leve de electrones tibios. Los colores de las poblaciones utilizadas fueron las siguientes: azul para el plasma estacionario de electrones calientes, rojo para el haz de electrones fríos y magenta para el haz leve de electrones tibios.

En la figura 6.16d se visualiza el atrapamiento de los electrones que corresponde a la estructuras solitarias ubicadas en  $x \approx 45$ ,  $x \approx 60$  y  $x \approx 80$  que también se ven reflejadas en el campo y potencial eléctrico, así como en la densidad (Fig. 6.17d). Notamos que los electrones calientes no son atrapados del todo en el espacio de fase por los vórtices, a pesar de que sí se presentan OSEs al igual que un aumento en la densidad (picos) debido al abultamiento de las partículas. Esto puede ser causado porque los electrones calientes son la especie con mayor cantidad de partículas.



## 6.2.4. Gráficas 3S

(a)  $t = 0$ (b)  $t = 199\Delta t$ (c)  $t = 2999\Delta t$ (d)  $t = 4999\Delta t$ Figura 6.8: Función de distribución  $f(v)$  y gráfica del espacio de fase para Run 1B.

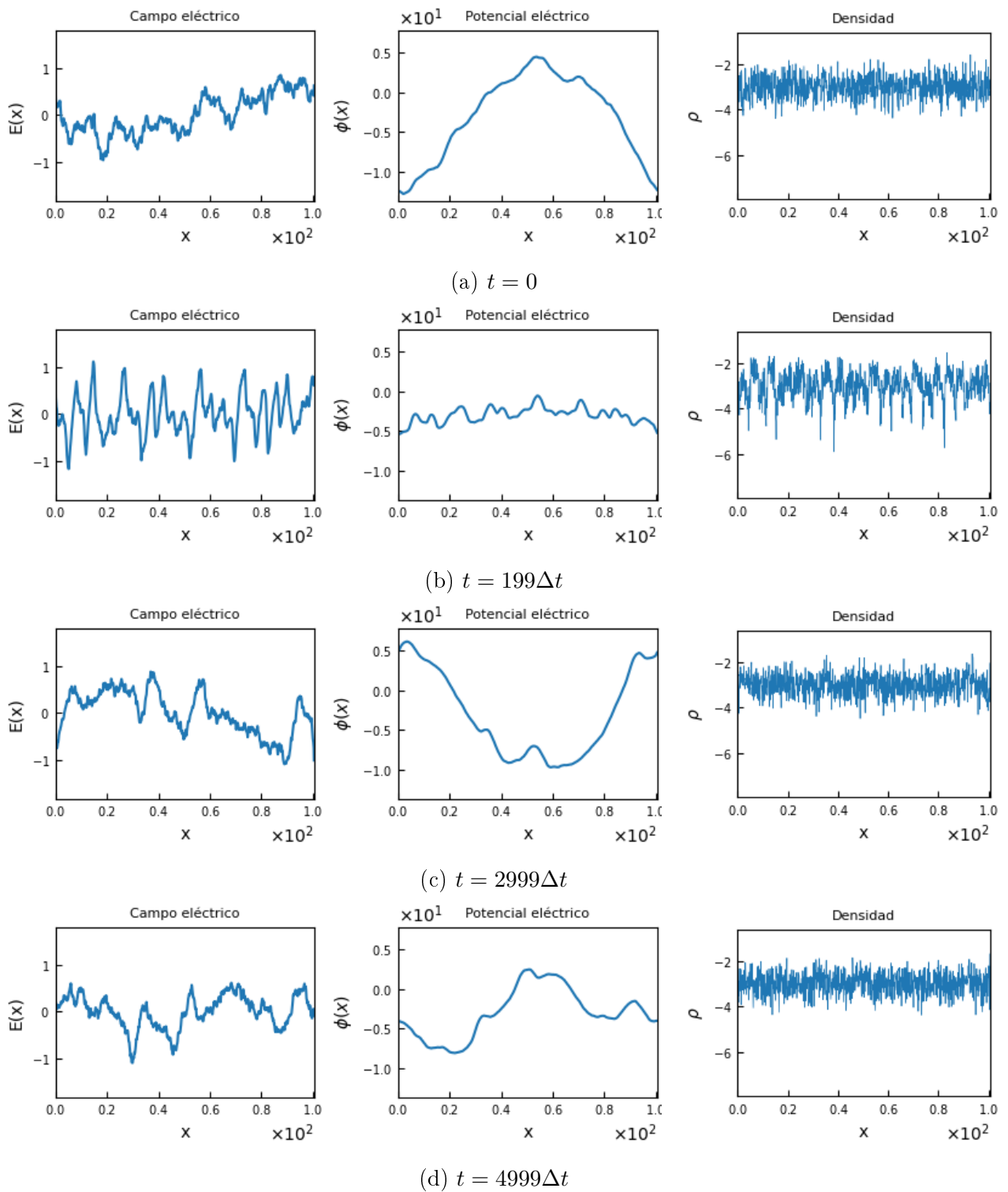
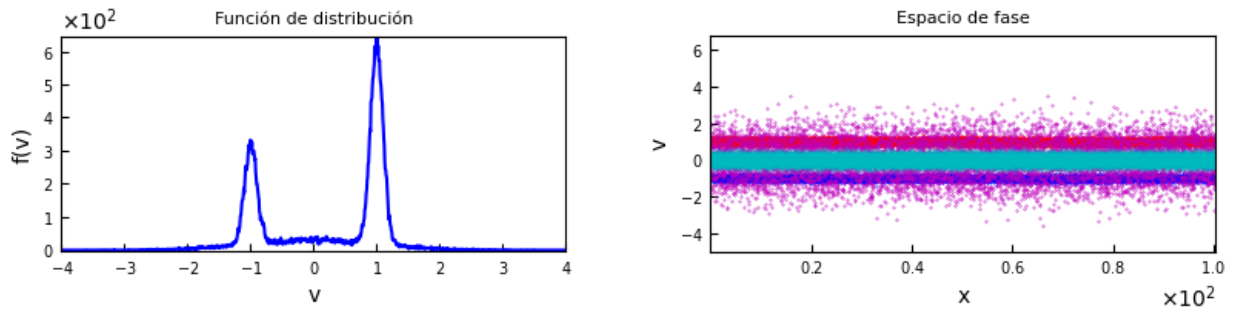
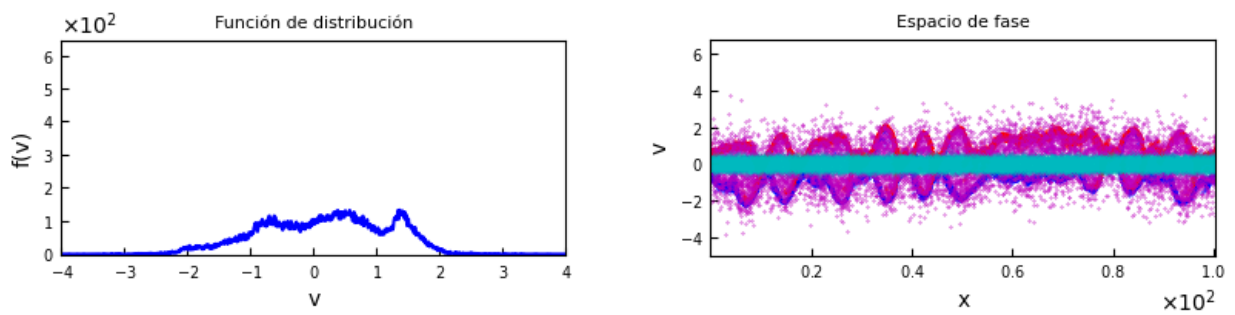
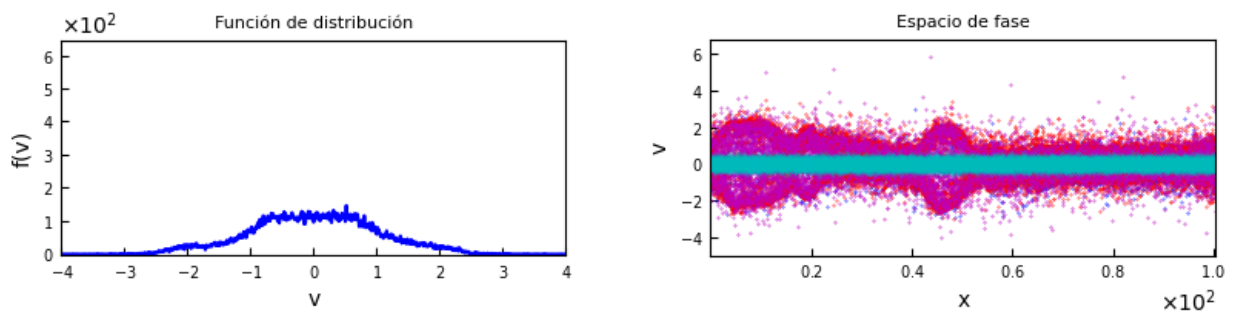
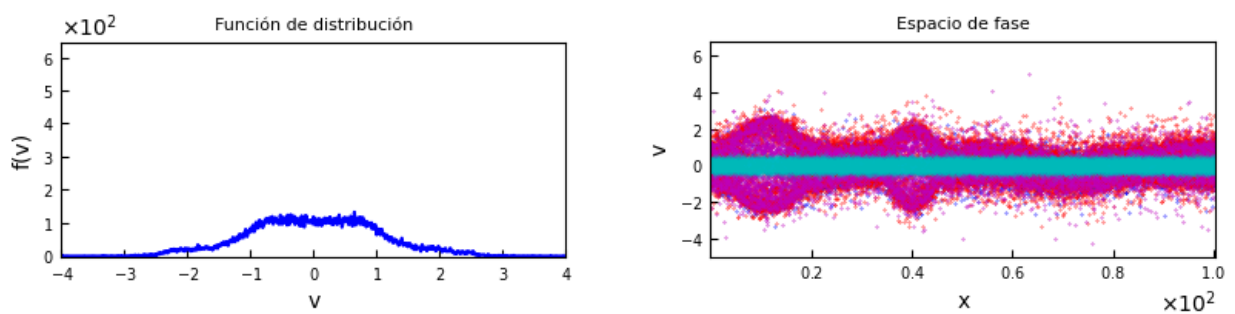


Figura 6.9: Campo eléctrico  $E$ , potencial eléctrico  $\phi$  y densidad de carga  $\rho$  para Run 1B.

(a)  $t = 0$ (b)  $t = 199\Delta t$ (c)  $t = 2999\Delta t$ (d)  $t = 4999\Delta t$ Figura 6.10: Función de distribución  $f(v)$  y gráfica del espacio de fase para Run 2B.

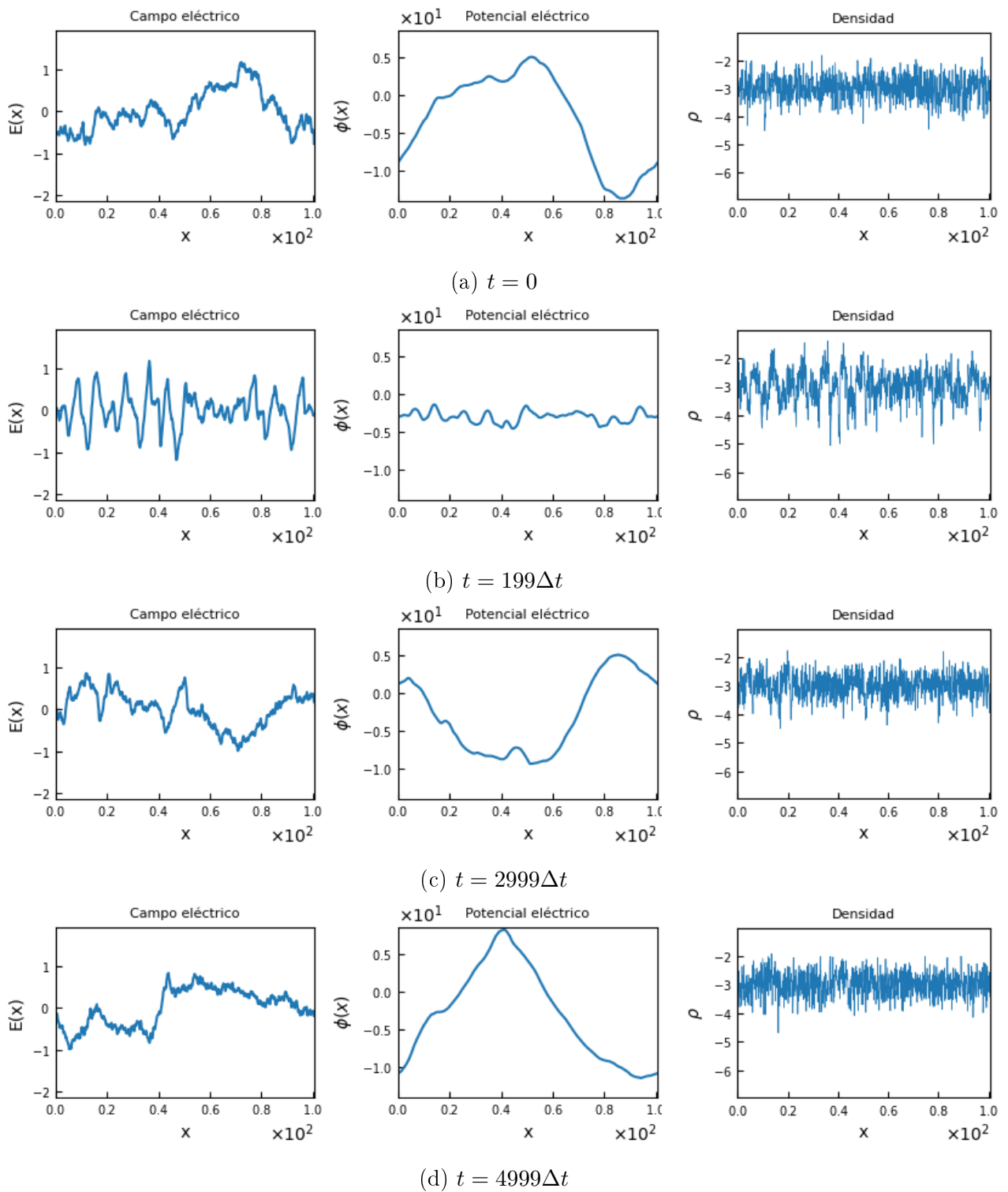
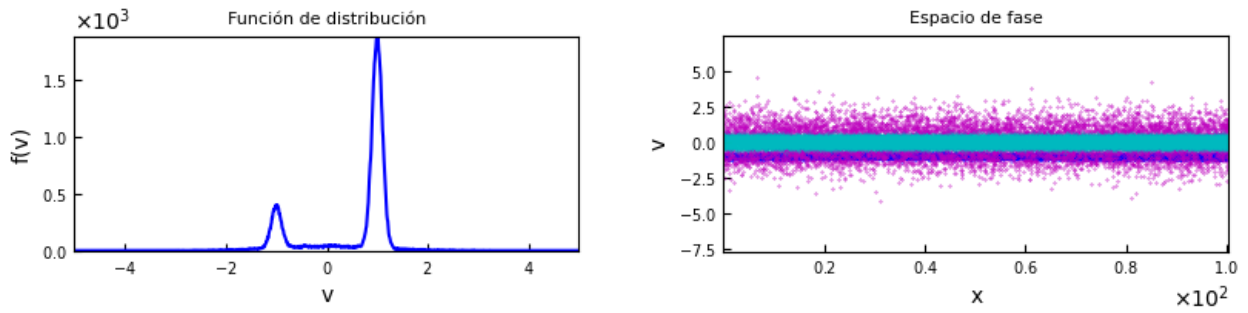
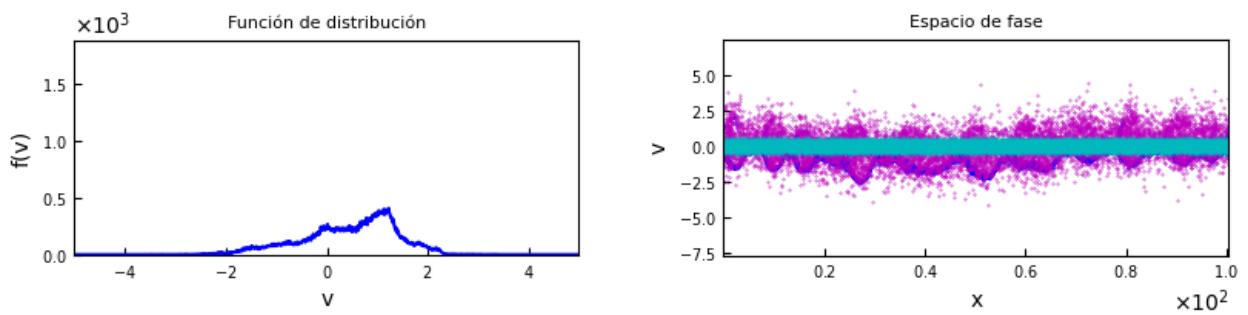
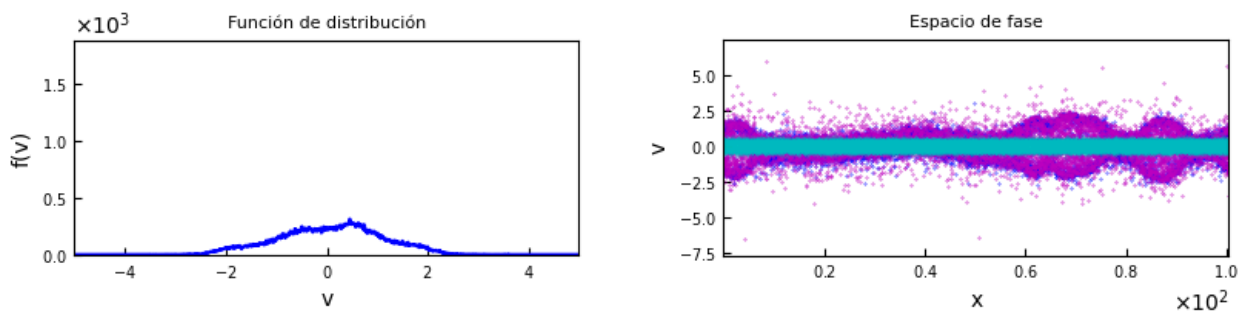
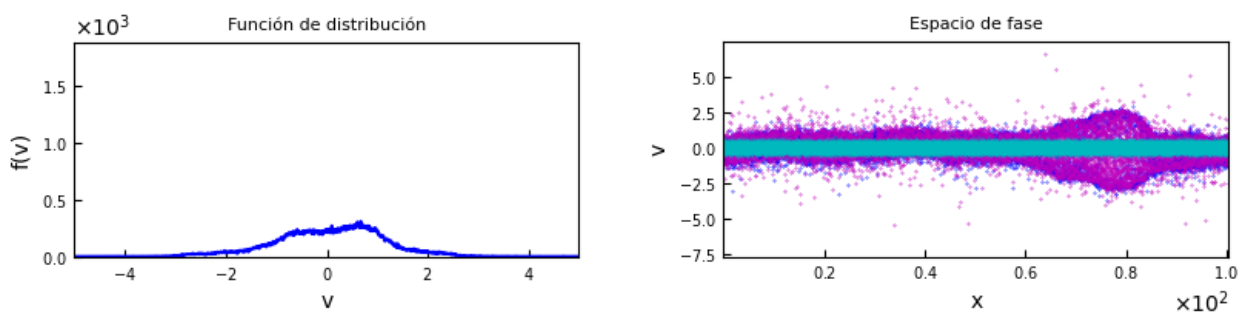


Figura 6.11: Campo eléctrico  $E$ , potencial eléctrico  $\phi$  y densidad de carga  $\rho$  para Run 2B.

(a)  $t = 0$ (b)  $t = 199\Delta t$ (c)  $t = 2999\Delta t$ (d)  $t = 4999\Delta t$ Figura 6.12: Función de distribución  $f(v)$  y gráfica del espacio de fase para Run 3B.

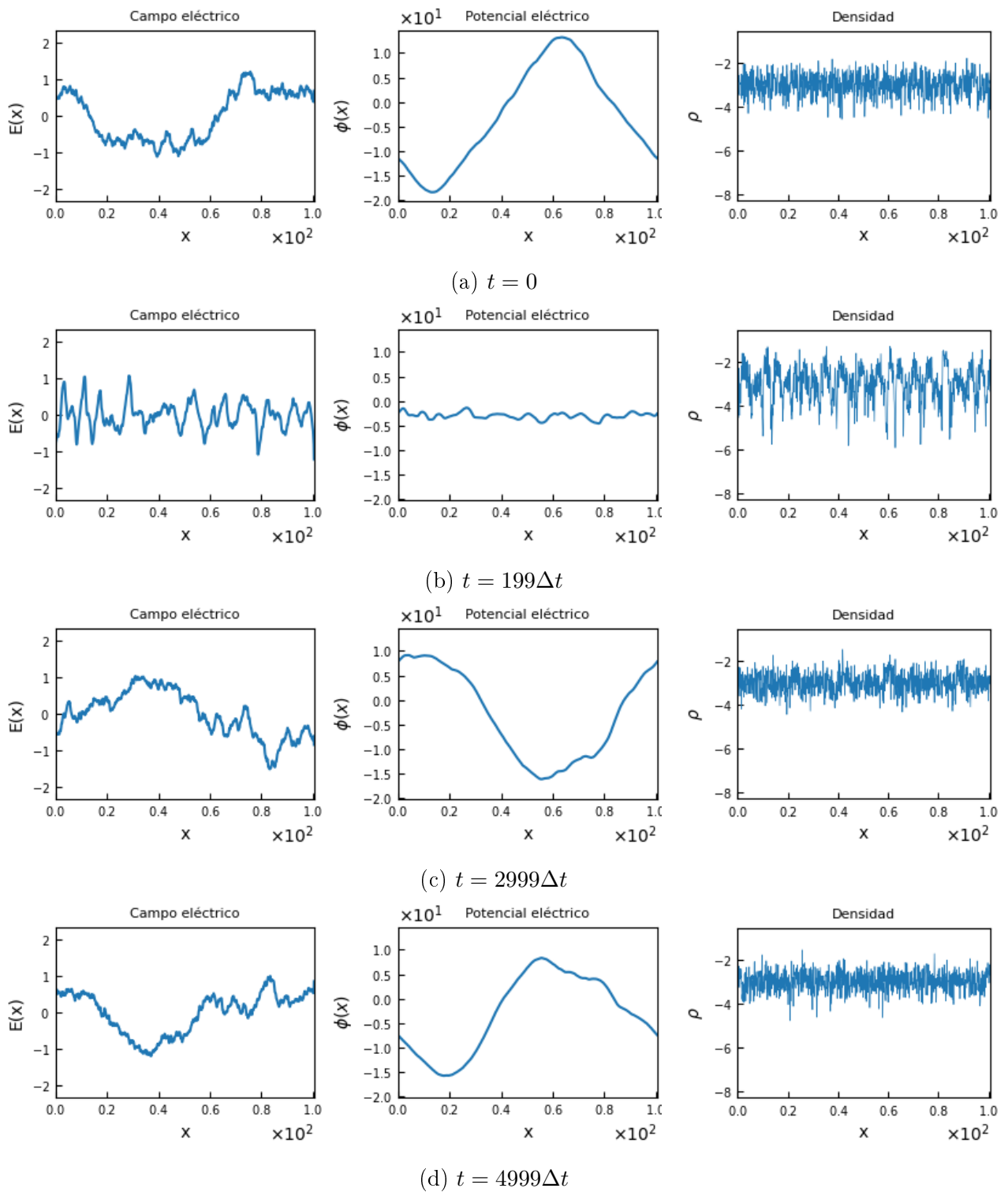
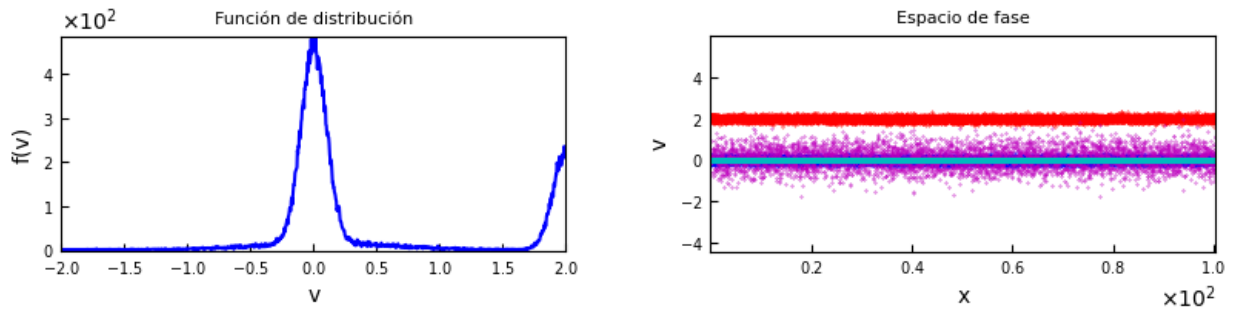
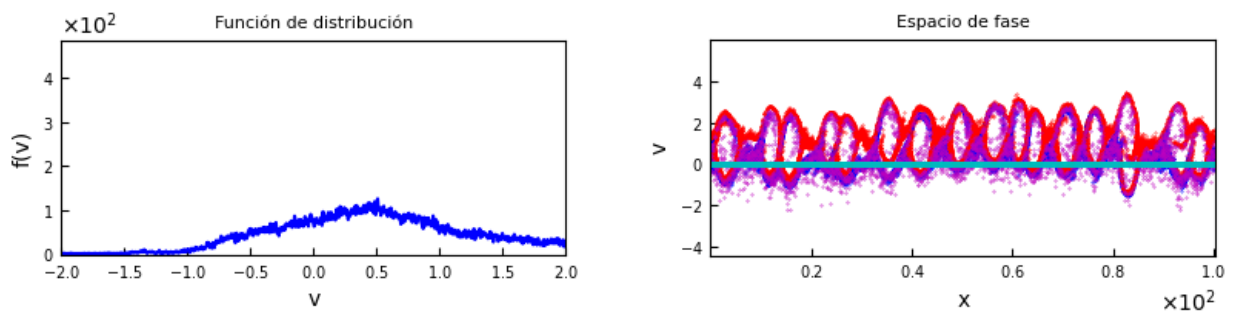
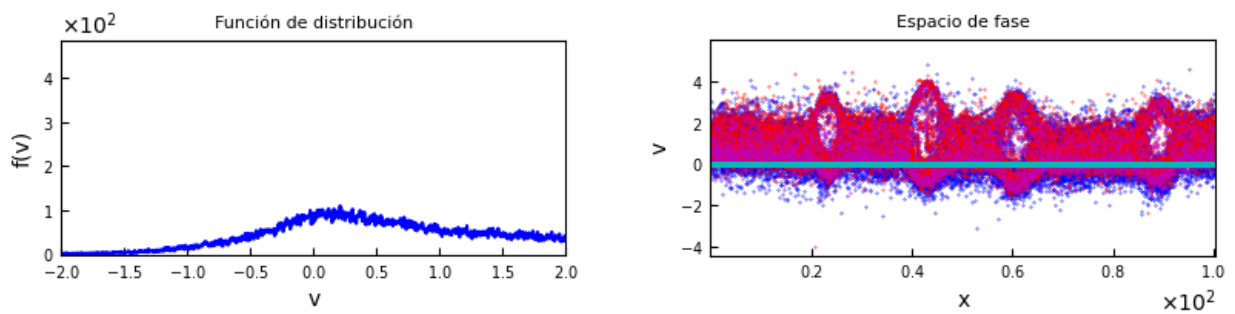
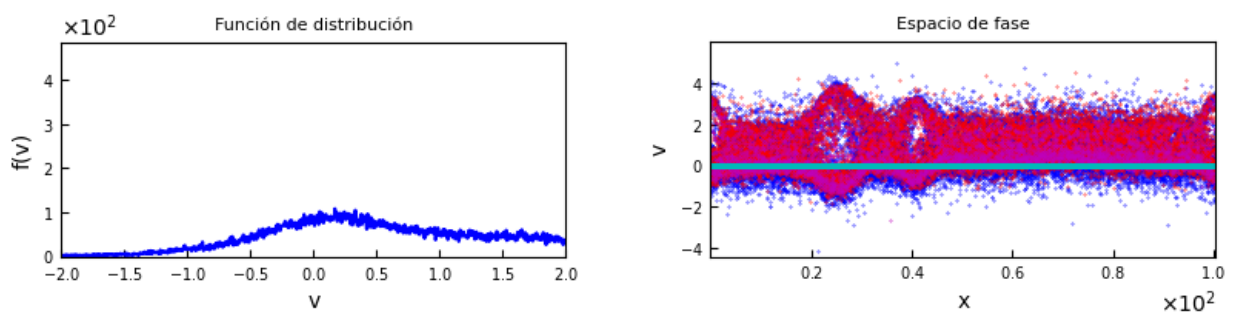


Figura 6.13: Campo eléctrico  $E$ , potencial eléctrico  $\phi$  y densidad de carga  $\rho$  para Run 3B.

(a)  $t = 0$ (b)  $t = 199\Delta t$ (c)  $t = 2999\Delta t$ (d)  $t = 4999\Delta t$ Figura 6.14: Función de distribución  $f(v)$  y gráfica del espacio de fase para Run 4B.

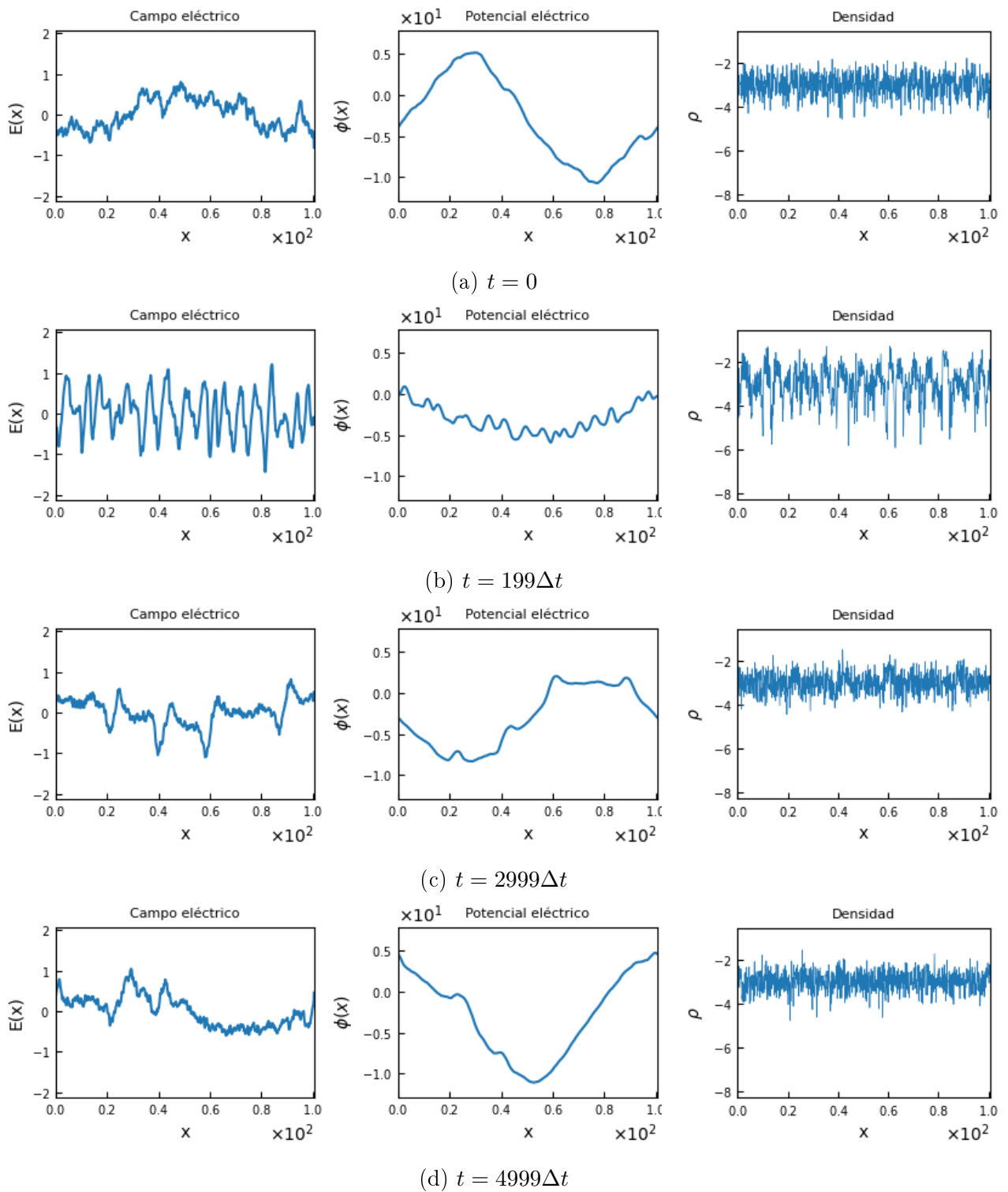
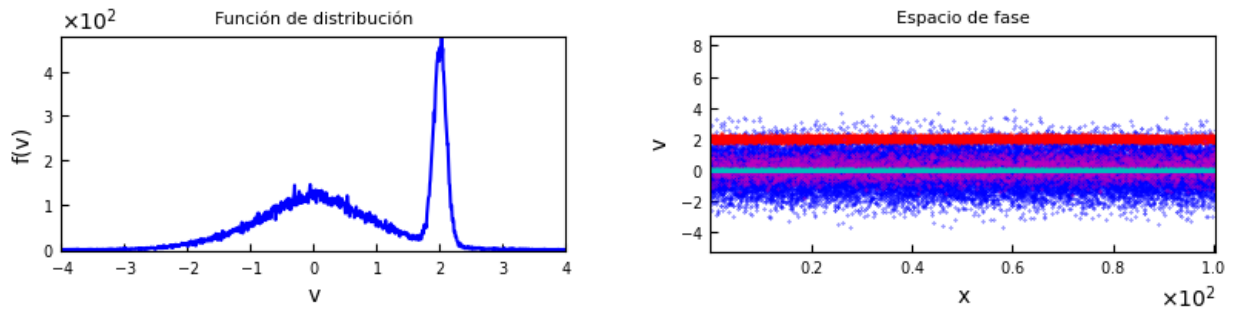
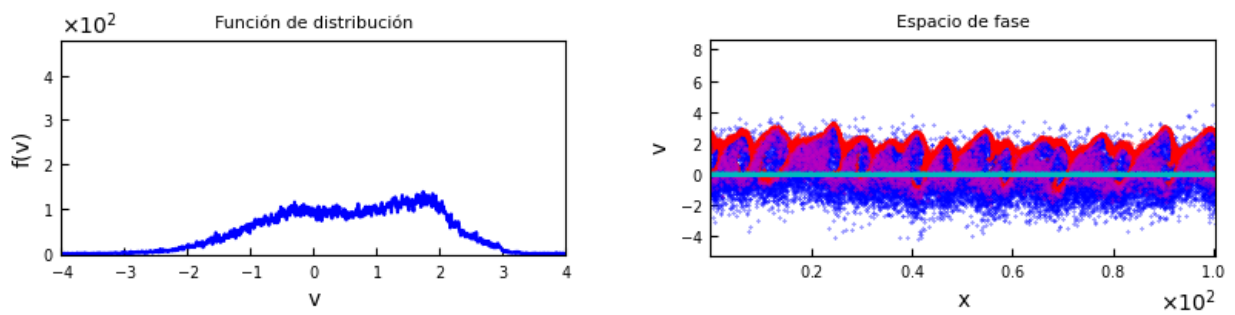
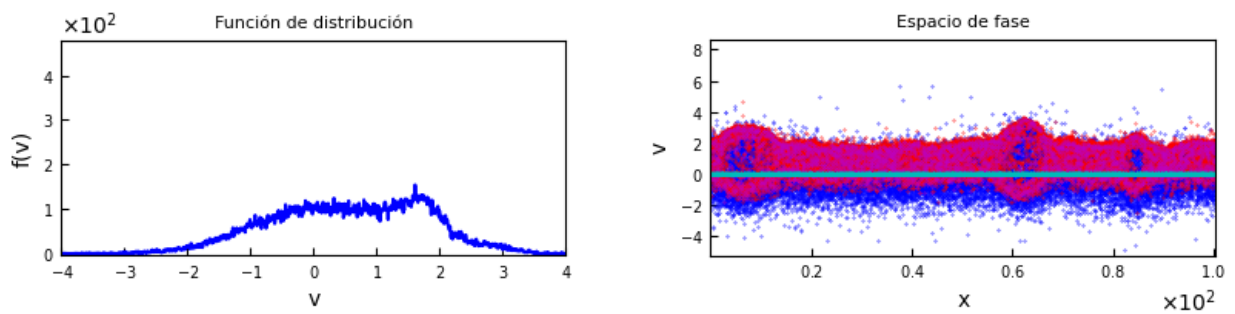
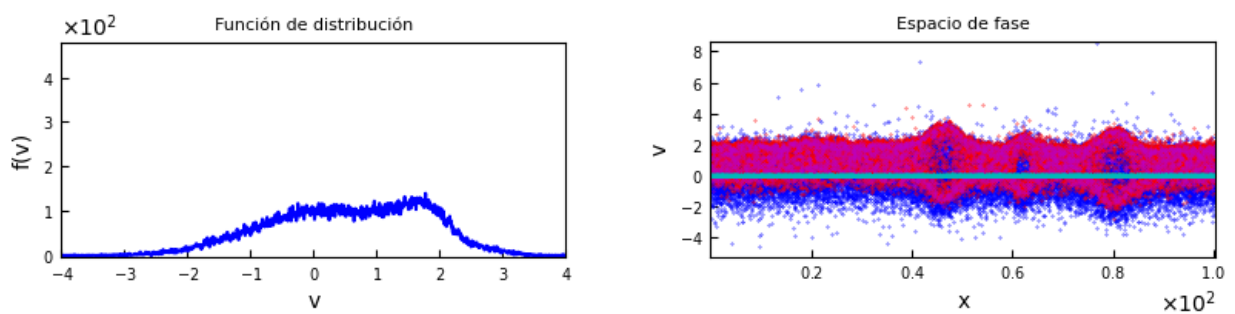


Figura 6.15: Campo eléctrico  $E$ , potencial eléctrico  $\phi$  y densidad de carga  $\rho$  para Run 4B.



(a)  $t = 0$ (b)  $t = 199\Delta t$ (c)  $t = 2999\Delta t$ (d)  $t = 4999\Delta t$ Figura 6.16: Función de distribución  $f(v)$  y gráfica del espacio de fase para Run 5B.

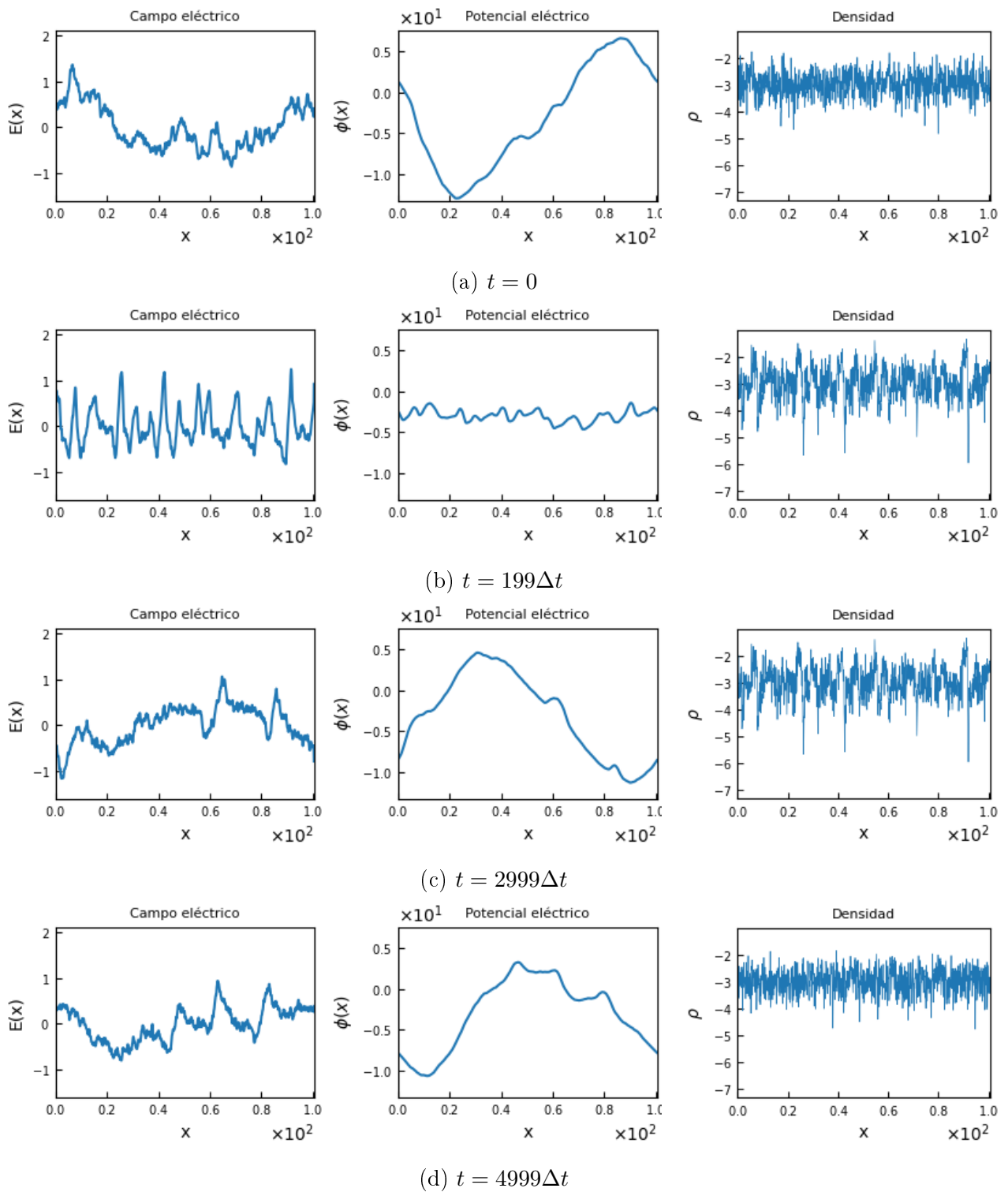
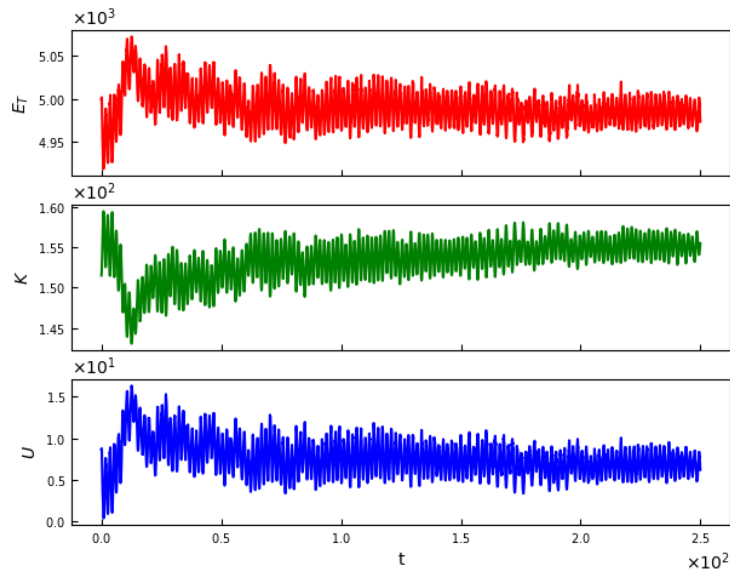
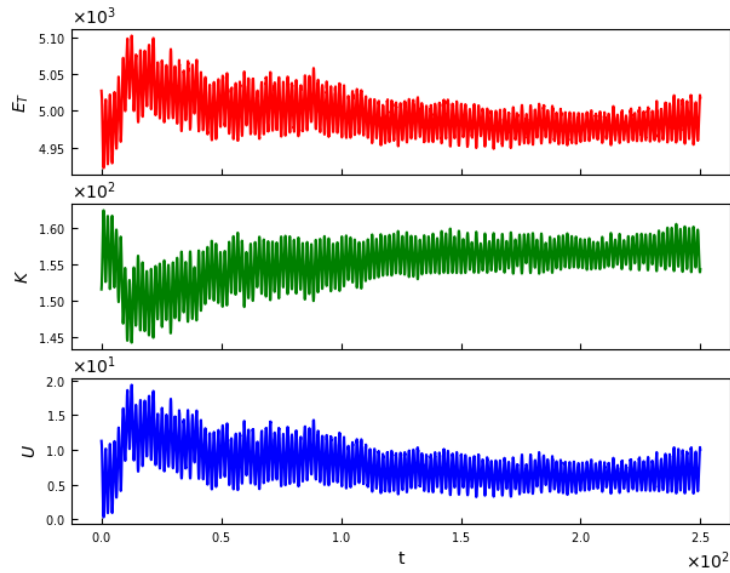


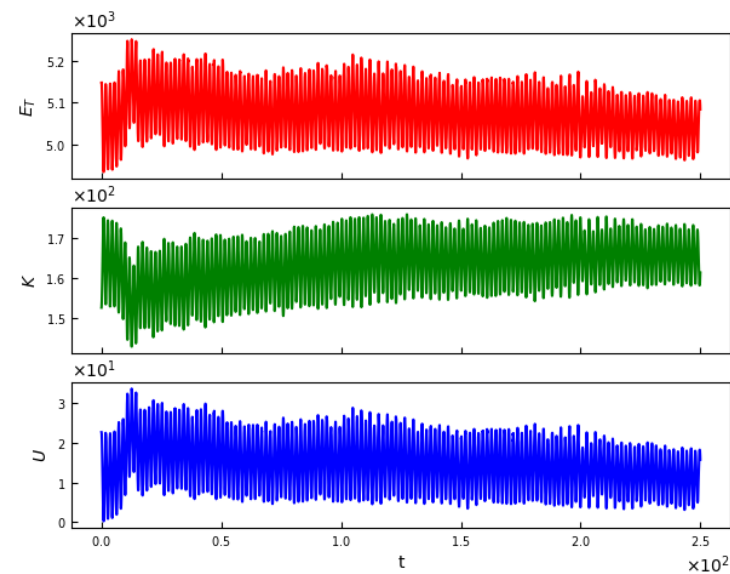
Figura 6.17: Campo eléctrico  $E$ , potencial eléctrico  $\phi$  y densidad de carga  $\rho$  para Run 5B.



(a) Run 1.

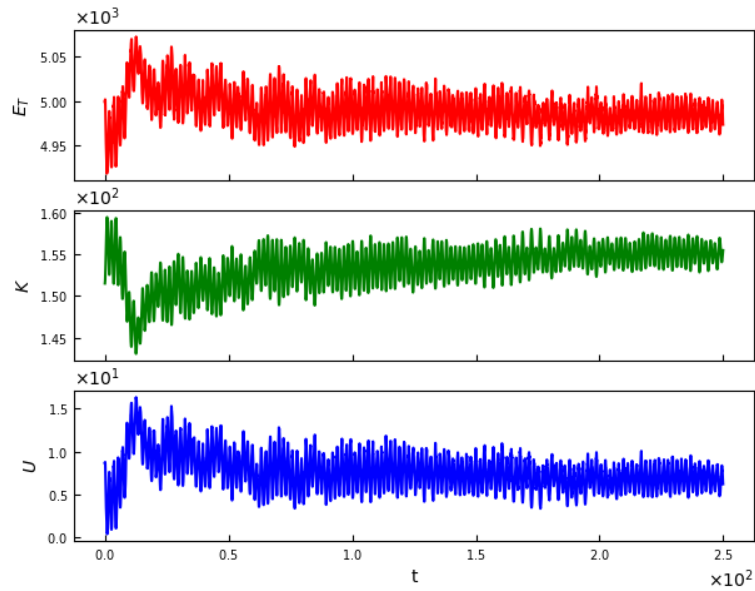


(b) Run 2.

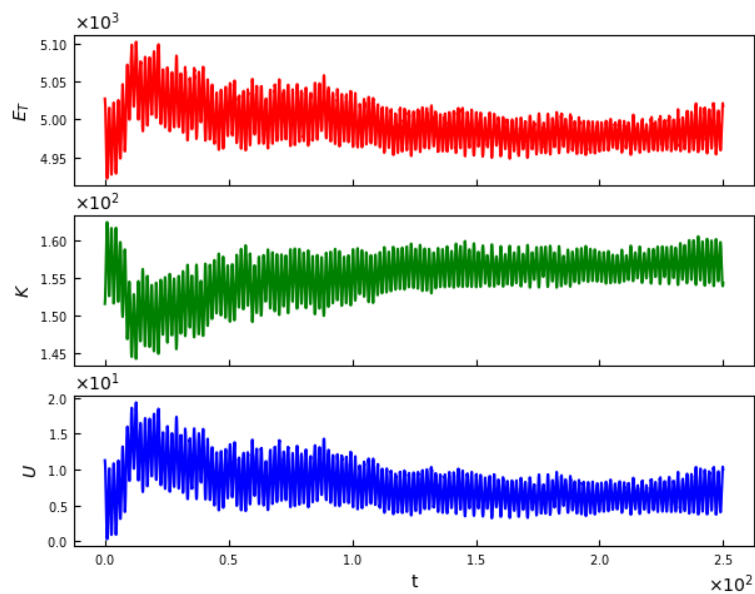


(c) Run 3.

Figura 6.18: Energía total  $E_T$ , cinética  $K$  y potencial  $U$  de las simulaciones del circuito Run 1-3B.



(a) Run 4.



(b) Run 5.

Figura 6.19: Continuación de la figura 6.18 para las energías de las simulaciones 4-5.

# Capítulo 7

## Conclusiones, Discusiones y Trabajo Futuro

### 7.1. Conclusiones y Discusiones

- A partir del código optimizado, se implementó una interfaz gráfica intuitiva para el usuario de fácil manejo con un entorno detallado a través de un manual de usuario que se encarga de enfatizar el uso de cada herramienta disponible. Esto permite que la interfaz PEWAV pueda ser utilizada para futuros trabajos o progresos en el estudio del plasma simulado, sea en el grupo de investigación GPLA (Grupos de plasma, láser óptica y aplicaciones) o grupos externos que requieran de dicha herramienta.
- A través de la interfaz gráfica PEWAV se simularon las inestabilidades de haz de electrones con sistemas compuestos por dos (2S) y tres (3S) especies de electrones, también se utilizaron sus herramientas para analizar el comportamiento y la evolución del plasma. De allí, se obtuvieron resultados diferentes entre el circuito de simulaciones A y B que corresponden a los casos 2S y 3S respectivamente. En el primero solo se desarrollaron OSEs en la inestabilidad two-stream, mientras que en el segundo para todas se consiguieron las estructuras solitarias. Estos resultados muestran que es muy probable que el mecanismo de generación de las ondas OSEs observadas en la magnetosfera, en la región aural de media y alta altitud [31, 32], correspondan a inestabilidades de haz de electrones como la two-stream y la bump-on-tail que también ha sido señalado por otros autores [36, 21], además de la weak-beam con la presencia de una tercera especie de electrones calientes.
- Se utilizaron las ecuaciones de fluido para modelar un sistema de plasma compuesto por iones inmóviles y tres especies maxwellianas de electrones que corresponden a las condiciones presentadas en la magnetosfera. Así que a través de la técnica de perturbación se obtuvo una ecuación diferencial similar a la K-dV, por lo cual se puede concluir que en este sistema se producen ondas solitarias electrostáticas. Estas ondas son compresivas o rarefactivas dependiendo de la temperatura del haz de electrones, así como el porcentaje de cada población de electrones.
- El ruido electrostático de banda ancha conocido como (BEN) por sus siglas en inglés, se puede observar generalmente en diferentes regiones de la magnetosfera de la Tierra, por ejemplo, en la región auroral integrando la zona de media y alta altitud, la capa límite de la hoja de plasma, la magnetopausa, entre otras. La frecuencia de estos ruidos electrostáticos se encuentran en el rango que va desde las bajas frecuencias hasta la frecuencia del plasma de los electrones locales donde se ven frecuencias más altas. Las observaciones de diferentes naves espaciales sugieren que las partes de baja y alta frecuencia del ruido BEN surgen a través de dos modos de onda diferentes que están ligadas a la frecuencia. Existe una teoría bien establecida para la parte de alta frecuencia que puede explicarse por ondas electrostáticas solitarias producidas en los plasmas que componen la magnetosfera. En referencia a la teoría, como se trató en esta investigación, es posible generar OSEs en presencia de dos y tres especies sustentando el fenómeno BEN en altas

frecuencias, mientras que para bajas frecuencias se ha demostrado por S.V. Singh et al. [37] que la BEN se produce a partir de plasmas magnetizados con cuatro especies a diversas temperaturas, resolviendo la incógnita tanto analítica como numéricamente. Finalmente el conocer el porqué ocurre el BEN en la magnetosfera nos permite mejorar la comprensión de la comunidad científica y estudiantil del plasma de esta región que afecta el clima espacial cercano al ambiente terrestre.

- Con el fin de tener una situación similar a la magnetosfera se agregó una especie de partículas extra a la simulación que, a través de parámetros adecuados, se comprueba que aún en el caso de tres poblaciones de partículas, se excitan las estructuras solitarias que pueden contribuir a los datos observados, por ejemplo, en la BEN.
- En la detección de OSEs, se identificó que la forma más acertada de ubicar dichas ondas, es en el espacio de fase donde se presentan vórtices en el que son atrapados los electrones. También se pueden señalar los intervalos de espacio donde estos ocurren con el fin de ubicarlos en las gráficas dispuestas para el campo eléctrico y el potencial eléctrico. En la gráfica del campo eléctrico se visualizaron estructuras bipolares, mientras que en el potencial, estructuras compresivas. Al paso de las iteraciones, dichas estructuras no pierden su forma al moverse en el espacio, característica propia de los solitones.
- Es de esperarse que los modos de oscilación de las simulaciones que contienen poblaciones de electrones fríos y calientes, sean electrón-acústicos, debido que se cumplen con las condiciones para su formación [38, 39, 40], es decir,  $T_h/T_c \geq 10$ ,  $0 < n_c < 0.8N_e$  y  $T_i \neq T_c$ . Aún así es necesario analizar las gráficas de relación de dispersión para poder concluir si las ondas excitadas son modos de Langmuir, iónico-acústicas, electrón-acústicas o forzadas por haz, como ha sido notado en la literatura [41, 21]. En este trabajo no se realizó este análisis a pesar de que la interfaz gráfica PEWAV cuenta con la relación de dispersión, debido a que consideramos que para esto es necesario normalizar las variables de una forma adecuada para que el resultado tenga mayor sentido.
- Se demuestra la dependencia del parámetro R en la formación de OSEs en el caso de tres especies de electrones, pues entre mayor sea su valor, más vórtices se presentan en el espacio de fase en el último tiempo simulado. En el caso de la inestabilidad two-stream, esto se observa a través de la disminución del ritmo de fusión de las estructuras solitarias.

## 7.2. Trabajo Futuro

- El análisis de las ondas solitarias se realizó utilizando el plasma como un fluido, pero esto también se puede realizar desde un punto de vista cinético, pues se ha demostrado que existen ciertas diferencias como en el ancho del solitón [42].
- Establecer los parámetros de simulación necesarios en PIC para obtener ondas solitarias electrostáticas rarefactivas.
- Utilizar la ecuación del solitón obtenido mediante el modelo de fluido (ec. 2.55) para encontrar las regiones teóricas en que las ondas son compresivas o rarefactivas. Se espera una dependencia en  $T_h/T_b$ , así como en la velocidad inicial del haz  $u_{b0}$  y en el porcentaje de cada población presente.
- A la hora de obtener la ecuación 2.55, se asumieron poblaciones de partículas maxwellianas, pero se propone realizar el análisis utilizando la función de distribución Kappa o la función generalizada  $(r, q)$ , ya que se han observado en la magnetosfera, distribuciones electrónicas no maxwellianas [30].
- Con respecto al código utilizado para las simulaciones, este puede mejorarse en múltiples aspectos. Previamente se optimizaron diversas funciones, pero de igual forma se pueden encontrar

otros métodos de interpolación o paralelizar el código con más equipos para mejorar su velocidad de cálculo. En específico se resalta como actividad futura encontrar un método más efectivo para calcular los máximos y mínimos de las funciones a graficar, pues este proceso consume una cantidad de tiempo considerable en las simulaciones.

- Para futuras versiones de la interfaz gráfica PEWAV, se plantea el habilitar una opción disponible en el entorno para considerar los campos magnéticos producidos por la interacción de los electrones y los iones, permitiendo estudiar plasmas magnetizados. De igual forma otra herramienta quedaría habilitada para considerar que el plasma simulado sea colisional o no colisional. Esta posibilidad de simular la magnetización del plasma y las colisiones de las partículas, acercará aún más las simulaciones al campo real.
- Actualmente la interfaz cuenta con múltiples funciones que mejoran la interacción con el usuario, pero a futuro, se plantea el agregar la opción de no solamente poder tener tres especies de electrones disponibles, sino de contar con más de una especie de iones, ya que a pesar de que en la interfaz PEWAV se utilicen cargas positivas en las poblaciones extras de partículas, dentro del código se interpreta como si fueran electrones.
- Por último, se plantea realizar la normalización de las variables dentro del código.

# Capítulo 8

## Anexos

Los códigos desarrollados en este trabajo se encuentran en el siguiente repositorio de Github <https://gitfront.io/r/user-1533735/bc0c57d4d1f428ba39910dfde17342a16c0dd00c/PEWAV/>



# Bibliografía

- [1] A. A. Abid, Q. Lu, M. N. Qureshi, X. L. Gao, H. Chen, K. H. Shah, and S. Wang, “1-D particle-in-cell simulations of electron acoustic solitary structures in an electron beam-plasma,” *AIP Advances*, vol. 9, no. 2, 2019.
- [2] L. Brida, *Plasma Simulations by Example*. 2019.
- [3] L. D. Herrera and C. R. Arias, “Simulación de ondas generadas en la inestabilidad two-stream para plasmas electrostáticos usando particle-in-cell simulación de ondas generadas en la inestabilidad two-stream para plasmas electrostáticos usando particle-in-cell,” 2019.
- [4] R. Schulz and J. Benkhoff, “Bepicolombo: Payload and mission updates,” *Advances in Space Research*, vol. 38, no. 4, pp. 572–577, 2006. Mercury, Mars and Saturn.
- [5] A. Danehkar, N. S. Saini, M. A. Hellberg, and I. Kourakis, “Electron-acoustic solitary waves in the presence of a suprathermal electron component,” *Physics of Plasmas*, vol. 18, no. 7, p. 072902, 2011.
- [6] C. K. Birdsall and A. B. Langdon, *Plasma physics via computer simulation*. New York: Taylor and Francis, 2005.
- [7] D. Martin, “Electrostatic pic simulation of plasmas in one dimension,” 2007.
- [8] M. Paprzycki, “Parallel computing works!,” *IEEE Parallel Distributed Technology: Systems Applications*, vol. 4, no. 4, pp. 82–83, 1996.
- [9] D. Baker, S. Elkington, X. Li, and M. Wiltberger, *Particle Acceleration in the Inner Magnetosphere*, pp. 73–85. American Geophysical Union (AGU), 2005.
- [10] W. Baumjohann and R. A. Treumann, *Basic space plasma physics*. 1996.
- [11] F. F. Chen, *Introduction to plasma physics*. New York: Plenum Press, 1974.
- [12] M. S. Arenada, J. and A. Patricio, 2011.
- [13] P. Debye and E. Hückel *Phys. Zeit.*, vol. 24, pp. 185–206, 1923.
- [14] L. D. Landau, “On the vibrations of the electronic plasma,” *Yad. Fiz.*, vol. 10, p. 25, 1946.
- [15] H. Koskinen, *Physics of Space Storms*, pp. 163–190. 01 2011.
- [16] J. Humphries, Stanley, *Charged particle beams*. 1990.
- [17] M. Ghorbanalilu, E. Abdollahzadeh, and S. E. Rahbari, “Particle-in-cell simulation of two stream instability in the non-extensive statistics,” *Laser and Particle Beams*, vol. 32, no. 3, p. 399–407, 2014.
- [18] P. Abarca and N. Accel, “Simulación fluido-cinética de los efectos ionosféricos producidos por la inyección de partículas desde el espacio.”
- [19] P. G. Drazin and R. S. Johnson, *Solitons: An Introduction*. Cambridge Texts in Applied Mathematics, Cambridge University Press, 2 ed., 1989.

- [20] C. Orban, M. Fatenejad, S. Chawla, S. C. Wilks, and D. Q. Lamb, “A Radiation-Hydrodynamics Code Comparison for Laser-Produced Plasmas: FLASH versus HYDRA and the Results of Validation Experiments,” vol. 2, no. 1, pp. 1–21, 2013.
- [21] Y. Omura, H. Matsumoto, T. Miyake, and H. Kojima, “Electron beam instabilities as generation mechanism of electrostatic solitary waves in the magnetotail,” vol. 101, no. 95, pp. 2685–2697, 1996.
- [22] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, and A. e. a. Bridgland, “Improved protein structure prediction using potentials from deep learning,” *Nature*, vol. 577, no. 7792, pp. 706–710, 2020.
- [23] F. H. Harlow, “The particle-in-cell method for numerical solution of problems in fluid dynamics,”
- [24] A. B. Langdon and C. K. Birdsall, “Theory of plasma simulation using finite-size particles,” *Physics of Fluids*, vol. 13, no. 8, pp. 2115–2122, 1970.
- [25] D. Sulsky, S.-J. Zhou, and H. L. Schreyer, “Application of a particle-in-cell method to solid mechanics,” *Computer Physics Communications*, vol. 87, no. 1, pp. 236–252, 1995. Particle Simulation Methods.
- [26] D. Tskhakaya, K. Matyash, R. Schneider, and F. Taccogna, “The particle-in-cell method,” *Contributions to Plasma Physics*, vol. 47, no. 8-9, pp. 563–594, 2007.
- [27] O. Y. Matsumoto, H. and I. S. for Space Simulations, “Computer space plasma physics: Simulation techniques and software,” 1993.
- [28] H. Okuda, “Nonphysical noises and instabilities in plasma simulation due to a spatial grid,” *Journal of Computational Physics*, vol. 10, no. 3, pp. 475–486, 1972.
- [29] E. L. Lindman, “Dispersion relation for computer-simulated plasmas,” *Journal of Computational Physics*, vol. 5, no. 1, pp. 13–22, 1970.
- [30] C. S. Dillard, I. Y. Vasko, F. S. Mozer, O. V. Agapitov, and J. W. Bonnell, “Electron-acoustic solitary waves in the Earth’s inner magnetosphere,” *Physics of Plasmas*, vol. 25, no. 2, 2018.
- [31] X. Deng, R. Tang, H. Matsumoto, J. Pickett, A. Fazakerley, H. Kojima, W. Baumjohann, A. Coates, R. Nakamura, D. Gurnett, and Z. Liu, “Observations of electrostatic solitary waves associated with reconnection by geotail and cluster,” *Advances in Space Research*, vol. 37, no. 7, pp. 1373–1381, 2006. Reconnection at Sun and in Magnetospheres.
- [32] R. E. Ergun, C. W. Carlson, J. P. McFadden, F. S. Mozer, G. T. Delory, W. Peria, C. C. Chaston, M. Temerin, I. Roth, L. Muschietti, R. Elphic, R. Strangeway, R. Pfaff, C. A. Cattell, D. Klumpar, E. Shelley, W. Peterson, E. Moebius, and L. Kistler, “Fast satellite observations of large-amplitude solitary structures,” *Geophysical Research Letters*, vol. 25, no. 12, pp. 2041–2044, 1998.
- [33] I. Y. Vasko, O. V. Agapitov, F. S. Mozer, J. W. Bonnell, A. V. Artemyev, V. V. Krasnoselskikh, G. Reeves, and G. Hospodarsky, “Electron-acoustic solitons and double layers in the inner magnetosphere,” *Geophysical Research Letters*, vol. 44, no. 10, pp. 4575–4583, 2017.
- [34] J. C. Holmes, R. E. Ergun, D. L. Newman, F. D. Wilder, A. P. Sturmer, K. A. Goodrich, R. B. Torbert, B. L. Giles, R. J. Strangeway, and J. L. Burch, “Negative potential solitary structures in the magnetosheath with large parallel width,” *Journal of Geophysical Research: Space Physics*, vol. 123, no. 1, pp. 132–145, 2018.
- [35] Y. Omura, H. Kojima, and H. Matsumoto, “Computer simulation of electrostatic solitary waves: A nonlinear model of broadband electrostatic noise,” *Geophysical Research Letters*, vol. 21, no. 25, pp. 2923–2926, 1994.

- [36] T. Miyake, Y. Omura, and H. Matsumoto, “Electrostatic particle simulations of solitary waves in the auroral region,” *Journal of Geophysical Research: Space Physics*, vol. 105, no. A10, pp. 23239–23249, 2000.
- [37] S. Singh, R. Reddy, and G. Lakhina, “Broadband electrostatic noise and low-frequency waves in the earth’s magnetosphere,” *Advances in Space Research*, vol. 43, no. 12, pp. 1940–1944, 2009.
- [38] S. P. Gary and R. L. Tokar, “The electron-acoustic mode,” *The Physics of Fluids*, vol. 28, no. 8, pp. 2439–2441, 1985.
- [39] R. L. Tokar and S. P. Gary, “Electrostatic hiss and the beam driven electron acoustic instability in the dayside polar cusp,” *Geophysical Research Letters*, vol. 11, no. 12, pp. 1180–1183, 1984.
- [40] K. Watanabe and T. Taniuti, “Electron-acoustic mode in a plasma of two-temperature electrons,” *Journal of the Physical Society of Japan*, vol. 43, no. 5, pp. 1819–1820, 1977.
- [41] Q. Lu, S. Wang, and X. Dou, “Electrostatic waves in an electron-beam plasma system,” *Physics of Plasmas*, vol. 12, no. 7, p. 072903, 2005.
- [42] M. Zulfiqar Ahmad and A. Qamar, “Kinetic treatment of nonlinear ion-acoustic waves in multi-ion plasma,” *Physics of Plasmas*, vol. 24, no. 9, p. 092304, 2017.