



**UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO**

UNIVERSITÀ DEGLI STUDI DI BARI
Dipartimento Interateneo di Fisica

Corso di Laurea Triennale in Fisica

IL METODO PARTICLE-IN-CELL PER SIMULAZIONI DI PLASMA

Laureando:
Flavio D'ippolito

Relatore:
prof. Savino Longo

Anno Accademico 2015/2016

Indice

Introduzione	iii
1 Il metodo Particle-in-Cell	1
1.1 Cenni storici	1
1.2 Il modello elettrostatico monodimensionale ES1	2
1.3 Integrazione delle equazioni del moto	5
1.4 Integrazione delle equazioni sui campi	8
1.5 <i>Weighting</i> di particelle e forze.	10
1.6 Note sulla scelta dei valori iniziali	13
1.7 Output	13
2 Un'introduzione allo stato di plasma	15
2.1 Schermo di Debye	15
2.2 Parametro di plasma	17
2.3 Frequenza di plasma	18
2.4 Oscillazioni di plasma	20
3 L'instabilità di due stream	23
3.1 Instabilità in un plasma	23
3.2 Instabilità di due stream. Generalità.	24
3.3 Analisi lineare di due stream con velocità uguali ed opposte	28
3.4 L'instabilità di due stream: una simulazione col metodo Particle-in-Cell	31
3.5 Il codice	33
3.6 Output del programma	40
4 Conclusioni	43
A Soluzione numerica dell'equazione di Poisson con il metodo della matrice tridiagonale.	44

B Codice completo per la simulazione dell'instabilità di due stream	47
---	----

Introduzione

Il metodo *Particle-in-cell* è una tecnica comunemente utilizzata per simulare il moto di particelle cariche o plasmi. Per basse densità, il plasma si comporta più come una collezione di particelle discrete che un singolo fluido continuo; questa proprietà lo rende adatto a simulazioni discrete di tipo Particle-In-Cell. I plasmi ad alta densità sono simulati usando invece un'estensione della fluidodinamica per l'elettromagnetismo chiamato *magnetoidrodinamica*. Il metodo PIC è applicabile allo studio di vasti fenomeni tra cui la propagazione del vento solare o simulazioni per creste di propulsori al plasma. Nella tesi verranno esposte le basi di questo metodo più una semplice applicazione: lo studio dell'*instabilità di due stream*.

La simulazione di un plasma è complicata dalla presenza di campi elettromagnetici esterni e autoindotti dalle cariche in moto, dalla presenza di corpi solidi, oltre alla presenza di una differente scala di tempi per la propagazione degli ioni e degli elettroni. Per cercare di massimizzare la velocità di computazione, verranno fatte significative approssimazioni:

Assumeremo che la corrente generata dalle particelle di plasma sia trascurabile, in modo tale da non considerare eventuali campi magnetici autoindotti (condizione di plasma non magnetizzato). Questa è una valida assunzione per la classe di problemi con cui si avrà a che fare. Il set di equazioni di Maxwell per l'elettromagnetismo verrà ridotto quindi alla mera componente elettrostatica. Studieremo infatti un metodo Particle-In-Cell puramente elettrostatico. In aggiunta, nel capitolo 3 assumeremo che gli elettroni si distribuiscano secondo le relazioni di Boltzmann e che la densità del gas sia abbastanza bassa da trascurare le collisioni tra particelle.

Capitolo 1

Il metodo Particle-in-Cell

1.1 Cenni storici

Lo sviluppo e la diffusione di simulazioni per plasmi privi di collisioni costituisce uno dei primi e principali successi della fisica e dell'ingegneria computazionale moderna. I primi di questi modelli apparvero già nella seconda metà degli anni 50, di pari passo con lo sviluppo di più performanti computer digitali. Questi modelli erano accomunati dalle seguenti caratteristiche:

- Carica, velocità e accelerazione monodimensionali.
- Modello elettrostatico (calcolo di $E_x(x)$ dalla legge di Gauss o di $\varphi(x)$ dall'equazione di Poisson).
- Assenza di collisioni.
- Particelle modellizzate con dischi o strati bidimensionali di carica.
- Assenza di griglia matematica utilizzata per le interpolazioni della densità di carica elettrica dalla posizione delle particelle simulate.

Da questi primi tentativi monodimensionali di simulazione, mirati a catturare l'essenza di alcuni comportamenti collettivi da parte delle particelle cariche del plasma, si passò ai primi modelli *1-D 3v* (che consideravano in aggiunta tre componenti per la velocità) con 10^4 particelle simulate, e passando per modelli *2-D 3v* e *3-D 3v* con 10^6 e più particelle simulate da supercomputer, si arriva infine ai risultati odierni ottenuti da sistemi informatici di calcolo parallelo.

Modelli di simulazione basati sull'introduzione di una griglia spaziale entro cui interpolare la densità di carica in base alla posizione degli elettroni furono

introdotti per la prima volta nel 1963 presso l'università di Stanford da Buneman e Hockney. L'equazione di Poisson veniva risolta tramite differenze finite e la forza agente sulle singole particelle veniva anch'essa interpolata dalla griglia. Tali metodi offrivano notevoli vantaggi in termini di costo computazionale. Essi si presentavano infatti come notevolmente più efficienti rispetto alla mera somma di N^2 interazioni coulombiane sulle N particelle. Inoltre Buneman e Hockney realizzarono che certi problemi fisici e computazionali associati al carattere divergente del campo di Coulomb potevano essere facilmente bypassati con queste metodologie; certi picchi ottenuti dalle interazioni su distanze brevi tra particelle venivano smussati, riducendo gli angoli di scattering per collisioni binarie. Queste collisioni sono di moderato interesse nello studio di plasmi caldi, ma nelle simulazioni di questo tipo assumono un certo risalto per via dell'esiguo numero di particelle simulate rispetto a quelle di un plasma di laboratorio. Oltre ai suddetti vantaggi computazionali, la descrizione complessiva del sistema risultava semplificata dall'introduzione della nozione del campo elettrico E (somma degli effetti di ogni singola particella) derivabile dalla densità di carica ρ distribuita nello spazio. Inoltre, la fisica che i due studiosi si proponevano di ottenere dallo studio di un plasma necessitava di informazioni che rientravano in una scala di lunghezze per cui la distribuzione di carica poteva essere considerata continua.

In un plasma composto da molte particelle entro una lunghezza caratteristica chiamata lunghezza di Debye (vedi capitolo 2), le collisioni sono molto rare, quindi gli effetti macroscopici più importanti derivano in gran parte da effetti di tipo cumulativo da parte delle cariche che lo compongono.

1.2 Il modello elettrostatico monodimensionale ES1

Lo scheletro principale di un ES1 simula il movimento di particelle cariche nel campo elettrico e magnetico generato da loro stesse. La fisica di questo modello è principalmente orientata alla determinazione dei campi prodotti dalle cariche e del moto prodotto dalle forze che questi ultimi generano. I campi sono calcolati a partire dalle equazioni di Maxwell conoscendo tutte le posizioni di tutte le particelle e le loro velocità; le forze agenti sulle particelle sono invece dedotte dalle equazioni di Newton e Lorentz. La differenza fondamentale tra la simulazione e un plasma prodotto in laboratorio sono da ritrovarsi in una discontinuità temporale *step by step* tipica di una simulazione digitale, oltre che nell'utilizzo di una griglia spaziale entro la quale sono

approssimati i campi elettrici e magnetici in gioco. Nonostante ciò, l'utilizzo di adeguati metodi di approssimazione numerica, nonché la possibilità di disporre di griglie spaziali e intervalli temporali sempre più fitti, permettono di ottenere una sufficiente stabilità e predittività per certi fenomeni macroscopici tipici dei plasmi.

Specifichiamo che i sistemi di cui si sta simulando il comportamento sono composti da un numero estremamente elevato di particelle, dobbiamo quindi intendere le particelle usate nella simulazione come *Super-particelle*, ovvero come pacchetti singoli che stanno a rappresentare molteplici particelle reali. Esse seguiranno le stesse traiettorie che seguirebbero le particelle reali.

Modelli che prevedono interazioni tra particelle solo attraverso i campi medi prodotti da loro stesse sono chiamati **PM** (particle-mesh), quelli che includono interazioni dirette particella particella vengono indicati con **PP** (particle-particle). Esistono infine modelli che integrano entrambe le possibilità (modelli **PP-PM**).

In (fig 1.2) è mostrato un ciclo tipico per questo tipo di modelli. In sintesi, il ciclo prevede i seguenti passi:

1. **Calcolo della densità di carica:** le particelle vengono posizionate sulla griglia. Dalla loro posizione si interpola la densità di carica.
2. **Calcolo del potenziale elettrico:** si risolve l'equazione di Poisson alle differenze finite.
3. **Calcolo del campo elettrico:** dal gradiente del potenziale si ottiene il campo elettrico.
4. **Calcolo delle forze sulle particelle:** si determinano le forze sulle singole particelle dal campo elettrico da esse sentito.
5. **Movimento delle particelle:** si aggiornano le posizioni e le velocità delle particelle dalle forze centrate su di esse.
6. **Generazione di nuove particelle:** alcune proprietà fisiche potrebbero generare nuove particelle da includere nell'algoritmo (es. Ionizzazione).
7. **Output:** il salvataggio delle informazioni sullo stato della simulazione ed eventualmente ne mostra i risultati su schermo.
8. **Ripetizione:** il ciclo viene iterato un numero di volte pari agli step temporali in cui si è diviso il tempo totale di simulazione.

Per ogni step di tempo Δt , il programma calcola i campi dalla distribuzione delle particelle e successivamente muove le particelle. Possono rendersi necessari decine di step per unità di tempo normalizzato alla frequenza di plasma (vedi cap.2), e questo può implicare centinaia o migliaia di cicli per l'esecuzione dell'intero programma. L'esecuzione comincia con $t = 0$, con appropriate condizioni iniziali sulle velocità e le posizioni delle particelle. Tali quantità possono venir rappresentate in un opportuno spazio delle fasi. L'indice i distingue ciascuna particella dalle altre (es. x_i e v_i). I moduli dei campi magnetico ed elettrico sono invece riferiti alla griglia spaziale, i cui punti discreti verranno indicati con l'indice j . Ci riferiremo quindi ai campi E_j e B_j .

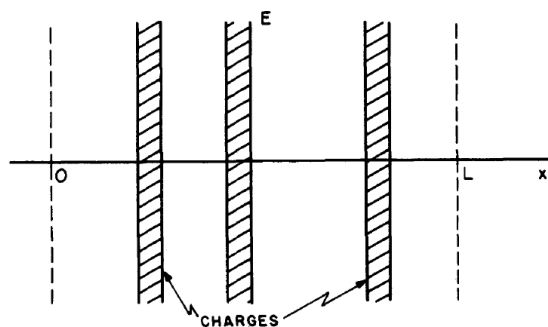


Figura 1.1:

Nel modello monodimensionale viene considerata la variazione lungo l'asse x di strati di carica, col relativo campo elettrico da essi generato e ad essi applicato.

Dalle posizioni delle particelle si determinano le densità di carica e correnti rispettivamente riferite alla griglia spaziale ρ_j e J_j . Questo passo del ciclo presuppone il calcolo di una densità di elettroni per ogni cella della griglia: il programma conta infatti il numero di elettroni che cadono nell'intervallo spaziale predefinito e ne deriva una densità. Una volta stabiliti i valori di ρ_j e J_j , si derivano tramite metodi di approssimazione numerica i campi E_j e B_j da cui si dedurranno le forze F_i centrate sulle singole particelle.

L'unica informazione sulle particelle che è necessario immagazzinare in memoria sono la velocità e la posizione "presente" di ciascuna particella, quindi le coordinate x_i e v_i del ciclo in questione. Trattandosi di sistemi composti da una sola specie di particelle (gli elettroni) i valori della carica q_i e della massa m_i , non dipenderanno dalla singola particella e non varieranno al variare dei cicli, potranno quindi essere immagazzinate in locazioni di memoria apposite. I campi invece potranno essere allocati direttamente in vettori di dimensione pari al numero totale di divisioni della griglia spaziale.

Ai fini di un'adeguata simulazione, è essenziale che il numero N di particelle

comutate sia maggiore del numero totale di divisioni della griglia spaziale; Il costo in memoria di questa condizione non è affatto eccessivo, dato che ogni particella verrà integrata indipendentemente dalle altre.

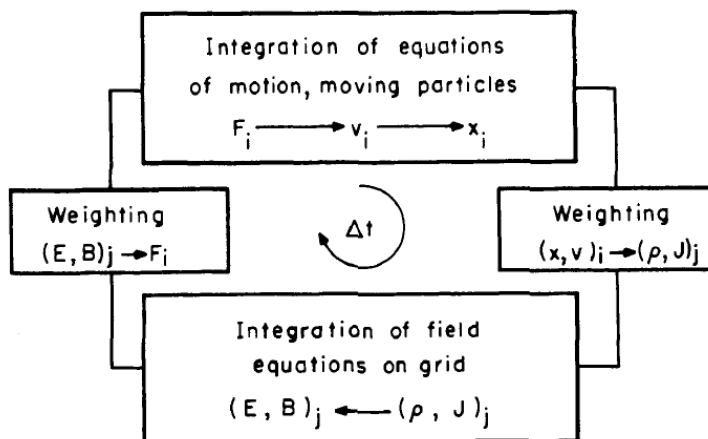


Figura 1.2:

Un ciclo tipico del programma di simulazione ES1.

1.3 Integrazione delle equazioni del moto

Dato che ci si potrebbe trovare a processare un numero di particelle dell'ordine di 10^6 in 1000 step temporali (questo significherebbe integrare le equazioni del moto 10^9 volte in totale nell'esecuzione del programma), è necessario un metodo computazionale per l'integrazione del moto che sia quanto più veloce e accurato possibile.

Se vogliamo computare la traiettoria di una particella, dovremmo tener conto nella nostra simulazione della posizione e della velocità di quest'ultima anche per gli step precedenti al ciclo considerato. L'informazione minima necessaria per l'integrazione consta quindi di due coordinate per particella, usando invece metodi di approssimazione numerica di più alto ordine (es. Runge-Kutta) le operazioni per singola particella saranno ancora maggiori. Per tali motivi cerchiamo un metodo quanto più veloce e meno dispendioso possibile in termini di memoria.

In uno dei metodi di integrazione numerica più usati in questi casi, detto *Leapfrog method*, vengono risolte separatamente due equazioni differenziali

per ogni particella:

$$m \frac{d\mathbf{v}}{dt} = \mathbf{F}, \quad \frac{d\mathbf{x}}{dt} = \mathbf{v}, \quad (1.1)$$

con \mathbf{F} la forza sentita dalla particella. le (1.1) vengono rimpiazzate da equazioni alle differenze finite:

$$m \frac{\mathbf{v}_{new} - \mathbf{v}_{old}}{\Delta t} = \mathbf{F}_{old}, \quad \frac{\mathbf{x}_{new} - \mathbf{x}_{old}}{\Delta t} = \mathbf{v}_{new}. \quad (1.2)$$

L'integrazione *leapfrog* consiste nell'aggiornare le posizioni e le velocità (e

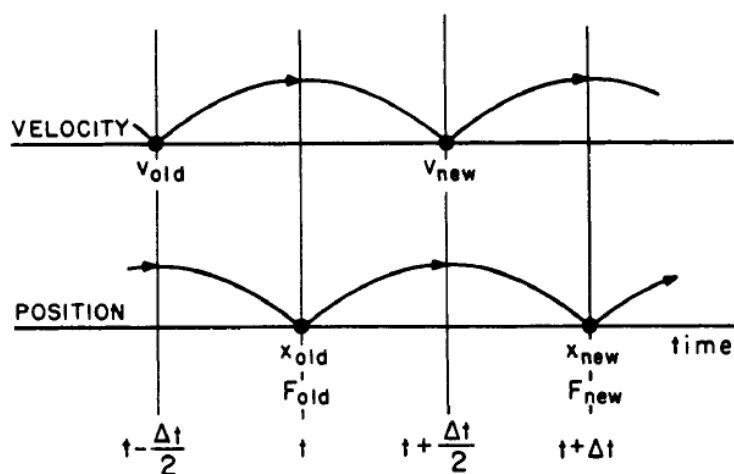


Figura 1.3:

Questa rappresentazione del metodo di integrazione Leapfrog mostra nel primo sketch l'evoluzione della forza \mathbf{F} nel tempo rispetto all'avanzamento della velocità \mathbf{v} e nel secondo sketch la velocità \mathbf{v} rispetto all'avanzamento della posizione \mathbf{x} .

di conseguenza le forze) in punti temporali "interfogliati", sfasati in modo tale che essi si scalvalchino reciprocamente durante l'esecuzione dell'algoritmo. Il programma riuscirà a calcolare $\mathbf{v}_{t+\Delta t}$ (velocità al tempo $t + \Delta t$) e $\mathbf{x}_{t+\Delta t}$ (posizione al tempo $t + \Delta t$) a partire da \mathbf{x}_t e \mathbf{v}_t anche se \mathbf{v} e \mathbf{x} non sono conosciuti nello stesso momento. Il metodo *Leapfrog* richiede che si faccia attenzione all'inserimento delle condizioni iniziali: se infatti si vuole computare il valore della velocità relativa alla forza \mathbf{F} a $t = 0$, essa deve essere riferita al tempo $-\Delta t/2$. Inoltre, nel calcolo dell'energia cinetica, bisogna fare attenzione nell'utilizzare i valori di posizione e velocità riferiti allo stesso istante di tempo.

Il metodo *leapfrog*, essendo un'approssimazione numerica, presenta ovviamente un errore che tende a zero per $\Delta t \rightarrow 0$, ma la sua semplicità e accuratezza sono perfettamente adatti allo scopo.

Analizziamo ora la forza applicata alla particella i per ogni ciclo. Essa è composta da due parti:

$$\mathbf{F} = \mathbf{F}_{\text{elettrico}} + \mathbf{F}_{\text{magnetico}} = q\mathbf{E} + q(\mathbf{v} \times \mathbf{B}), \quad (1.3)$$

in cui il campo elettrico e magnetico sono puntati sulla particella. Prima di calcolare le relative forze dobbiamo quindi interpolare \mathbf{B} e \mathbf{E} dai valori relativi alla griglia spaziale a quelli per ogni particella con lo stesso metodo usato nella determinazione della densità di carica (come vedremo in sez. 1.6). Consideriamo ora una disposizione delle particelle lungo x , le velocità v_x e v_y e un campo magnetico costante \mathbf{B}_0 diretto lungo z . La forza $q(\mathbf{v} \times \mathbf{B})$ (fig 1.5) è rappresentabile come una rotazione di \mathbf{v} (\mathbf{v} non cambia in modulo). Nemmeno la forza $q\mathbf{E} = qE_x\hat{x}$ altera il modulo di \mathbf{v} ($E_y = 0$). Uno schema fisicamente accettabile può quindi essere:

la prima metà accelerazione:

$$\begin{aligned} v_x(t') &= v_x\left(t - \frac{\Delta t}{2}\right) + \frac{q}{m}E_x(t)\left(\frac{\Delta t}{2}\right), \\ v_y(t') &= v_y\left(t - \frac{\Delta t}{2}\right). \end{aligned} \quad (1.4)$$

La rotazione:

$$\begin{pmatrix} v_x(t'') \\ v_y(t'') \end{pmatrix} = \begin{pmatrix} \cos\omega_c\Delta t & \sin\omega_c\Delta t \\ -\sin\omega_c\Delta t & \cos\omega_c\Delta t \end{pmatrix} \begin{pmatrix} v_x(t') \\ v_y(t') \end{pmatrix}, \quad (1.5)$$

l'altra metà accelerazione:

$$\begin{aligned} v_x\left(t - \frac{\Delta t}{2}\right) &= v_x(t'') + \frac{q}{m}E_x(t)\left(\frac{\Delta t}{2}\right), \\ v_y\left(t - \frac{\Delta t}{2}\right) &= v_y(t''), \end{aligned} \quad (1.6)$$

con t' e t'' nuove variabili per cui vale:

$$t - \frac{\Delta t}{2} < t' < t'' < t + \frac{\Delta t}{2}. \quad (1.7)$$

L'angolo di rotazione è dato da:

$$\Delta\theta = -\omega_c\Delta t, \quad \frac{d\theta}{dt} = -\omega_c, \quad (1.8)$$

con:

$$\omega_c \equiv \left(\frac{q}{m}\right) \mathbf{B}_0, \quad (1.9)$$

detta *frequenza di ciclotrone*.

L'unica complicazione si verifica a $t = 0$, quando le condizioni iniziali $\mathbf{x}(0)$ e $\mathbf{v}(0)$ sono definite nello stesso istante. Il ciclo principale parte con \mathbf{x} che precede \mathbf{v} di $\Delta t/2$, quindi, alla partenza del ciclo, $\mathbf{v}(0)$ deve essere spostato indietro a $\mathbf{v}(-\Delta t/2)$ facendo dapprima ruotare $\mathbf{v}(0)$ con l'angolo $\Delta\theta = \omega_c \Delta t/2$ e poi applicando la prima metà accelerazione usando $-\Delta t/2$ basata su $\mathbf{E}(0)$ ottenuto da $\mathbf{x}(0)$.

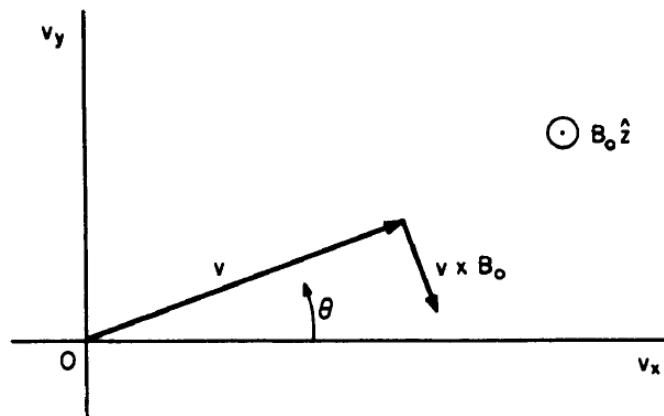


Figura 1.4:

Nel piano $v_y - v_x$ è mostrata la forza $(\mathbf{v} \times \mathbf{B})$ normale a \mathbf{v} , con risultato la rotazione di \mathbf{v} senza nessuna variazione del modulo.

1.4 Integrazione delle equazioni sui campi

Supponendo che la corrente e la densità di carica siano già definite per ogni punto sulla griglia, si può ottenere il campo magnetico ed elettrico usando le equazioni di Maxwell nell'unica dimensione x . Ricordiamo che, supponendo un problema di tipo elettrostatico, abbiamo che $\nabla \times \mathbf{E} = \partial \mathbf{B} / \partial t \approx 0$ e $\mathbf{E} = -\nabla \varphi$.

Le equazioni differenziali da risolvere sono:

$$\mathbf{E} = -\nabla \varphi \quad \left(E_x = -\frac{\partial \varphi}{\partial x} \right), \quad (1.10)$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad \left(\frac{\partial E_x}{\partial x} = \frac{\rho}{\epsilon_0} \right). \quad (1.11)$$

Combinandole entrambe otteniamo l'equazione di Poisson:

$$\nabla^2 \varphi = -\frac{\rho}{\epsilon_0} \quad \left(\frac{\partial^2 \varphi}{\partial x^2} = -\frac{\rho}{\epsilon_0} \right). \quad (1.12)$$

Un possibile approccio sarebbe quello di risolvere le equazioni alle differenze finite (1.10) e (1.12) usando la griglia mostrata in (fig. 1.5):

$$E_j = \frac{\varphi_{j-1} - \varphi_{j+1}}{2\Delta x}, \quad (1.13)$$

$$\frac{\varphi_{j-1} + \varphi_{j+1} - 2\varphi_j}{\Delta x^2} = -\frac{\rho}{\epsilon_0}. \quad (1.14)$$

Usiamo quindi le ρ_j ricavate dalle x_i per calcolare le φ_j e quindi le E_j con j che va da 0 a $L/\Delta x$, in cui L è la lunghezza totale della griglia spaziale.

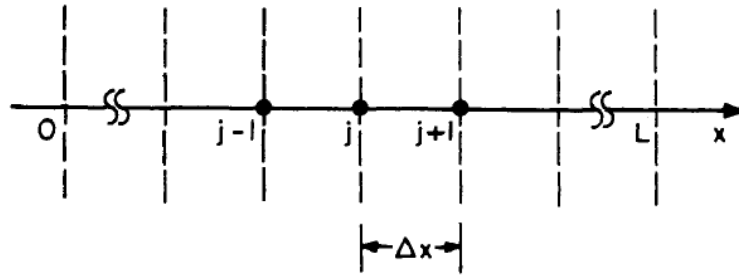


Figura 1.5:

Rappresentazione della griglia monodimensionale. Le linee verticali rappresentano i piani equidistanti $X_j = j\Delta x$ in cui è divisa la griglia.

Un approccio conveniente per sistemi periodici consiste nell'uso delle cosiddette *fast fourier transform* FFT per tutte le quantità computabili dalla griglia. Nel problema che vogliamo risolvere infatti, le quantità $\rho(\mathbf{x})$ e $\varphi(\mathbf{x})$ hanno rispettivamente delle trasformate di Fourier $\rho(\mathbf{k})$ e $\varphi(\mathbf{k})$ in cui k è il vettore d'onda del kernel della trasformata di Fourier. Possiamo così ottenere $\varphi(\mathbf{x})$ da $\rho(\mathbf{x})$ direttamente dall'equazione differenziale di Poisson in una dimensione:

$$\varphi(k) = \frac{\rho(k)}{\epsilon_0 k^2}, \quad (1.15)$$

in cui $\partial/\partial x^2$ è stato sostituito da $-k^2$. Usando la trasformata di Fourier inversa su $\varphi(k)$ otteniamo $\phi(x)$ e quindi $E(x)$ usando la (1.11).

1.5 *Weighting* di particelle e forze.

L'algoritmo presentato in precedenza dovrà calcolare la densità di carica a partire dalle posizioni delle particelle e di seguito calcolare la forza sentita da ciascuna particella a partire dai campi sulla griglia. Questo tipo di calcolo viene detto *Weighting* o *pesatura*, e implica una qualche forma di interpolazione sulle celle della griglia più vicine alla particella in questione. È opportuno usare lo stesso tipo di interpolazione sia per la densità di carica sia per le forze applicate alle particelle, così da evitare situazioni di *self-force*, cioè particelle che accelerano se stesse.

In un algoritmo di *Weighting* di ordine zero, si contano le particelle all'interno dell'intervallo $[\Delta x/2, -\Delta x/2]$ sulla j^{th} cella della griglia e si assegna quel numero ($N(j)$) a quella cella. Quindi $n(j) = N(j)/\Delta x$ può essere considerata la densità di carica in quella cella di griglia (fig. 1.7). Questo algoritmo di *Weighting* ha nome *Nearest-Grid-Point* o NGP. Il campo elettrico usato per calcolare la forza è quello nella X_j cella della griglia per tutte le j^{th} particelle all'interno della griglia.

Non appena una particella entra cella j^{th} (dai bordi $x = X_j \pm \Delta x/2$), la densità di quella cella aumenta, e quando la particella la attraversa ($x > X_j + \Delta x/2$ o $x < X_j - \Delta x/2$) la densità si riabbassa bruscamente.

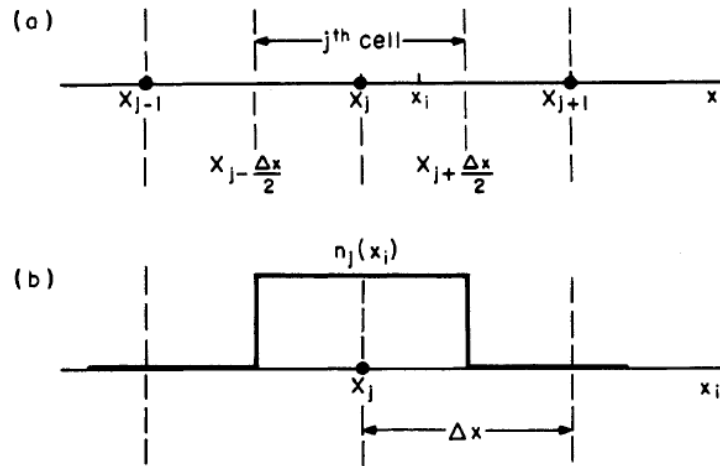


Figura 1.6:

- (a) *Weighting* di ordine zero per particelle e campi, o NGP. le particelle nella j^{th} cella, cioè quelle con le posizioni $x_i \in [\Delta x/2, -\Delta x/2]$ sono assegnate a X_j per ottenere $n(X_j)$. Su tutte queste particelle agirà il campo $E(X_j)$.
- (b) La densità $n_j(X_j)$ nel punto X_j dovuta alla particella x_j che si muove attraverso la cella centrata in X_j

Possiamo osservare due effetti. Per quanto riguarda il primo, la particella sembra possedere una forma rettangolare (vedi fig. 1.7 (b)) di lunghezza Δx . Questo ci porta a considerare che la fisica che stiamo simulando sia prodotta da particelle di questo tipo piuttosto che da particelle puntiformi. Ma poichè le collisioni o le interazioni su brevi distanze tra particelle in un plasma sono rare, questo effetto non altera eccessivamente gli effetti di plasma simulati. Una seconda considerazione verte sulla quantità di rumore che i continui passaggi delle particelle sui bordi delle celle produrrebbero nel tempo e nello spazio, questo tipo di rumore potrebbe risultare incompatibile con alcune simulazioni di plasma. Introduciamo quindi un metodo più efficace. Un algoritmo di *Weighting* di primo ordine smussa le fluttuazioni di campo e densità di carica, riducendo quindi il rumore, ma incrementa la spesa in termini di computazione usando due celle invece che una per ogni calcolo. In quest'algoritmo, le particelle vengono considerate come "nuvole di carica" di dimensione finita che possono liberamente attraversarsi l'un l'altra. Questo modello è chiamato *Cloud-In-Cell* o CIC. Se consideriamo ogni nuvola di

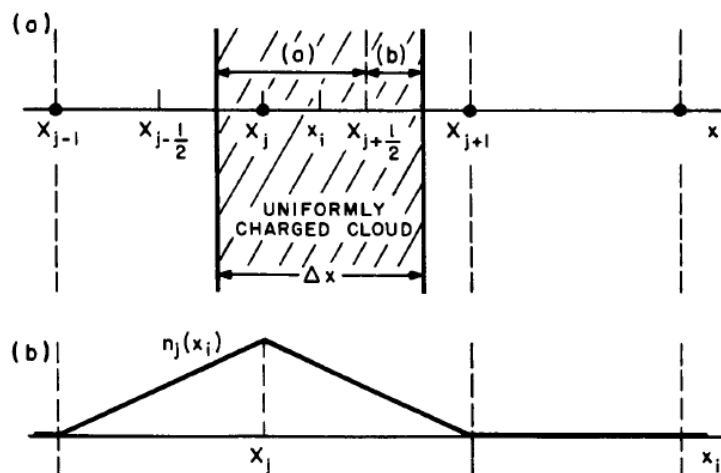


Figura 1.7:

(a) *Weighting* di primo ordine. La particella di grandezza finita, o "nuvola", ha la lunghezza di una cella di centro x_i . Questo algoritmo inserisce nella parte della nuvola che cade all'interno della j^{th} cella, la frazione (a) della carica, e la frazione (b) della carica nella parte della nuvola che cade all'interno della $(j+1)^{\text{th}}$ cella.

Questo processo di *Weighting* porta agli stessi risultati dell'applicazione dell'algoritmo NGP a tutte le singole parti di cui è composta la carica.

In (b) la densità $n_j(x_i)$ nel punto x_i quando la particella si muove presso X_j .

L'immagine mostra anche l'effettiva forma $S(x)$ della particella.

carica come avente una densità di carica uniforme e una lunghezza Δx , l'assegnamento di griglia può essere computato usando il metodo NGP (fig 1.7 (a)). Per quanto riguarda la quantità totale di carica q_c , la parte assegnata a j sarà:

$$q_j = q_c \left(\frac{\Delta x - (x_i - X_j)}{\Delta x} \right) = q_c \frac{X_{j+1} - x_i}{\Delta x} \quad (1.16)$$

e la parte assegnata a $j + 1$ sarà:

$$q_{j+1} = q_c \left(\frac{x_i - X_j}{\Delta x} \right) \quad (1.17)$$

L'effetto dell'algoritmo è quello di produrre una forma della particella $S(x)$ pari a quella di un triangolo di base $2\Delta x$. Si può notare che l'interpolazione lineare di una carica puntiforme x_i rispetto ai punti rispettivamente più vicini alla griglia, produce gli stessi risultati. Questo approccio al modello CIC è noto come *Particle-In-Cell* o PIC, ed esso sarà l'algoritmo utilizzato nei prossimi due capitoli.

Accenniamo ora all'algoritmo per la distribuzione di carica di un PIC bidimensionale. In figura (1.8) è mostrata una frazione della griglia spaziale

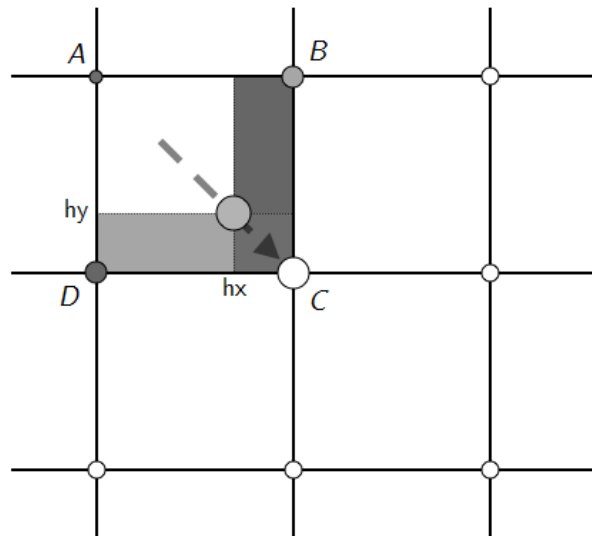


Figura 1.8:

Raffigurazione schematica di un weighting bidimensionale: la carica della particella è distribuita tra i nodi in base alla prossimità di questi ultimi rispetto alla stessa particella.

bidimensionale e della particella disposta all'interno del quadrato di estremi

ABCD. La carica della particella è distribuita tra i quattro nodi della cella in cui si trova. Il nodo C riceve la frazione più grande di carica per via della breve distanza dalla particella in questione. Il nodo A riceve invece la frazione minore di carica. Il weighting viene calcolato in proporzione alle frazioni di area evidenziate dai rettangoli:

$$\begin{aligned} w_A &= (1 - h_x)h_y, \\ w_B &= h_x h_y, \\ w_C &= h_x(1 - h_y), \\ w_D &= (1 - h_x)(1 - h_y), \end{aligned} \tag{1.18}$$

in cui, supponendo unitario il lato della cella, h_x è la distanza della particella dall'origine della cella nella direzione x, mentre h_y è la distanza rispetto all'asse y. Fanno eccezione le celle situate sul bordo della griglia, per cui contribuisce solo un quarto o metà della carica totale della particella.

1.6 Note sulla scelta dei valori iniziali

Una volta ultimato il codice desiderato, i valori iniziali da introdurre nel modello saranno:

- Il numero di particelle in ogni griglia.
- il parametri per il *Weighting* (numero di celle della griglia, lunghezza della griglia, ampiezza delle celle).
- La funzione di distribuzione iniziale di particelle $f(\mathbf{x}, \mathbf{v}, t = 0)$ incluse eventuali perturbazioni iniziali del sistema.

La disposizione iniziale più semplice è quella di un plasma freddo, uniforme e periodico composto da elettroni e ioni immobili ($m_i/m_e \rightarrow \infty$). Gli elettroni saranno posizionati in maniera uniforme, uno per uno, in ogni cella.

In caso di plasma caldo, gli elettroni invece saranno distribuiti con determinate velocità iniziali. Nel cap. 3 sarà implementato un semplice algoritmo per approssimare distribuzioni Maxwelliane di velocità.

1.7 Output

Tra le possibili scelte dell'output fornito dal programma ci potrebbe essere la distribuzione spaziale di certi parametri di plasma come il potenziale, la temperatura del gas elettronico, la densità di carica o il campo elettrico.

Oppure si potrebbe scegliere di rappresentare quantità associate alle singole particelle, come velocità e posizioni in uno spazio delle fasi. Inoltre, l'utilizzo di output dipendenti dal tempo di quantità come l'energia cinetica totale o l'energia potenziale del sistema, possono aiutare nel diagnosticare eventuali problemi nelle performance del codice.

Capitolo 2

Un'introduzione allo stato di plasma

In questo capitolo si introdurranno alcune nozioni essenziali e grandezze utili nello studio dei plasmi. Queste grandezze avranno un ruolo principale tra i parametri per la simulazione dell'*instabilità di due stream*. Alla fine del capitolo introdurremo un breve studio sulle oscillazioni di plasma; quest'analisi introduttiva sarà la base da cui partire per imbastire un approccio teorico sulla suddetta instabilità.

2.1 Schermo di Debye

Il termine *plasma* fu introdotto per la prima volta dai fisici Irving Langmuir e Lewi Tonks nello studio di gas ionizzati. Il termine "ionizzato" indica che una frazione significativamente grande di elettroni è stata strappata dai rispettivi atomi, sebbene il gas risulti globalmente neutro. Il comportamento dinamico di un plasma è qualitativamente diverso da quello di un gas neutro perché le forze intermolecolari a breve range che determinano la dinamica di un gas neutro sono sostituite dalle forze elettromagnetiche che si esercitano a lungo range tra cariche e correnti. Ad oggi, il termine viene esteso ad una gran varietà di sostanze neutre contenenti una certa percentuale di elettroni liberi e atomi o molecole ionizzate che esibiscono un comportamento collettivo.

Consideriamo una particella test di carica $q_T > 0$ di massa infinita, localizzata nell'origine di un sistema di riferimento tridimensionale contenente un plasma uniforme ed infinitamente esteso. La particella test, attraendo gli elettroni vicini, ne aumenta la densità n_e decrementando di conseguenza la densità degli altri ioni. In questo modo, lo ione test crea intorno a se un'area

di schermatura (Shielding cloud) in cui la sua carica risulta neutralizzata. Consideriamo l'equazione di Poisson relativa al potenziale elettrico φ di densità ρ dovuta ad elettroni, ioni e la particella test:

$$\nabla^2\varphi = -4\pi\rho = -4\pi e(n_e - n_i) - 4\pi q_t\delta(\mathbf{r}), \quad (2.1)$$

in cui $\delta(\mathbf{r}) = \delta(x)\delta(y)\delta(z)$ è il prodotto di tre Delta di Dirac relative alla posizione della particella test. Successivamente all'introduzione di quest'ultima, dopo un tempo t , gli elettroni raggiungeranno l'equilibrio termico alla temperatura T_e e gli ioni alla temperatura T_i con $T_i \neq T_e$, e usando alcuni risultati di meccanica statistica, si potrà scrivere:

$$n_e = n_0 \exp\left(\frac{\varphi e}{T_e}\right), \quad n_i = n_0 \exp\left(\frac{-e\varphi}{T_i}\right), \quad (2.2)$$

in cui le rispettive temperature sono valutate in elettronvolt (eV). Le densità n_e e n_i tenderanno al valore n_0 all'aumentare della distanza dalla particella test, data la progressiva diminuzione del potenziale φ .

Per $e\varphi/T_e \ll 1$ e $e\varphi/T_i \ll 1$ possiamo sviluppare gli esponenziali in serie di Taylor:

$$n_e = n_0 \left[\frac{\varphi e}{T_e} + \frac{1}{2} \left(\frac{\varphi e}{T_e} \right)^2 + \dots \right], \quad n_i = n_0 \left[-\frac{\varphi e}{T_i} + \frac{1}{2} \left(-\frac{\varphi e}{T_i} \right)^2 + \dots \right], \quad (2.3)$$

e considerando solo i termini di primo ordine dello sviluppo e coordinate sferiche, possiamo scrivere la (2.1), lontano dal punto $r = 0$ come:

$$\nabla^2\varphi = \frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{d\varphi}{dr} \right) = 4\pi n_0 e^2 \left(\frac{1}{T_e} + \frac{1}{T_i} \right) \varphi. \quad (2.4)$$

Se ora definiamo la *lunghezza di Debye* come:

$$\lambda_{e,i} \equiv \sqrt{\frac{T_{e,i}}{4\pi n_0 e^2}}, \quad (2.5)$$

rispettivamente per elettroni e ioni, e la lunghezza di Debye totale:

$$\lambda_D^{-2} = \lambda_e^{-2} + \lambda_i^{-2}. \quad (2.6)$$

L'equazione (1.3) può essere scritta come:

$$\frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{d\varphi}{dr} \right) = \lambda_D^{-2} \varphi. \quad (2.7)$$

con soluzione generale per $r \neq 0$ data da:

$$\varphi = \frac{q_r}{r} \exp\left(\frac{-r}{\lambda_D}\right). \quad (2.8)$$

Il potenziale dovuto alla particella test in un plasma decresce più rapidamente che nel vuoto. Questo fenomeno è conosciuto come *schermo di Debye*, ed è uno dei primi esempi del comportamento collettivo da parte delle particelle di un plasma. Per distanze molto maggiori dalla lunghezza di Debye, lo schermo di Debye neutralizza la carica della particella test q_t . La lunghezza di Debye, essendo una misura delle interazione columbiana nel plasma, definisce una minima lunghezza per il moto indipendente di elettroni e ioni: in una sfera di raggio λ_d (sfera di Debye) possono avvenire processi di particella singola. Al di fuori della sfera, il comportamento di ioni ed elettroni è determinato dalla parte a lungo raggio del potenziale elettrostatico. In sostanza, elettroni e ioni si muovono gli uni rispetto agli altri come se fossero un corpo unico; questi moti collettivi di cariche caratterizzano il plasma e ne rendono peculiare e diversificato il comportamento. La componente del plasma che più influisce sulla lunghezza di plasma è la densità di elettroni, questi ultimi infatti, a causa di una minore inerzia, hanno in generale una mobilità molto superiore a quella degli ioni. Numericamente, la lunghezza di Debye di un plasma è approssimativamente uguale a $740[T_s(eV)/n(cm^{-3})]^{1/2}$.

2.2 Parametro di plasma

In un plasma in cui ciascuna specie chimica s ha densità n_0 , la distanza media tra particelle è approssimativamente di $n_0^{-1/3}$. L'energia potenziale media sentita da una particella a causa delle particelle ad essa vicine è, in valore assoluto:

$$|\phi| \sim \frac{e^2}{r} \sim n_0^{1/3} e^2. \quad (2.9)$$

In un plasma, questa energia potenziale è molto minore dell'energia cinetica posseduta dalle particelle alla temperatura T_s :

$$\frac{1}{2} m_s \langle v^2 \rangle = \frac{3}{2} T_s = \frac{3}{2} m_s v_s^2, \quad (2.10)$$

in cui m_s è la massa della specie s e $\langle \rangle$ è la media sulla velocità di tutte le particelle in un dato punto dello spazio. Abbiamo anche usato la *velocità termica* v_s data da:

$$v_s \equiv \sqrt{\frac{T_s}{m_s}}, \quad (2.11)$$

che, per gli elettroni è $v_s \approx 4 \times 10^7 T_e^{1/2}$ (eV) in cm/s. La nostra condizione sull'energia potenziale richiede quindi che:

$$n_0^{1/3} e^2 \ll T_s, \quad (2.12)$$

oppure

$$n_0^{2/3} \left(\frac{T_s}{n_0 e^2} \right) \gg 1. \quad (2.13)$$

Elevando alla potenza di 3/2 entrambi i membri dell'ultima disequaglianza e richiamando la (2.5), abbiamo che (trascurando il fattore 4π), otteniamo:

$$\Lambda_s \equiv n_0 \lambda_s^3 \gg 1, \quad (2.14)$$

in cui Λ_s è chiamato *parametro di plasma della specie s*. Il parametro di plasma rappresenta il numero di particelle di specie s in un cubo di volume λ_s^3 . L'equazione (2.14) ci suggerisce che possiamo definire plasma un gas ionizzato che ha Λ_s particelle in un cubo di lato λ_s . È ovviamente possibile trovare in natura situazioni fisiche in cui lo stato di plasma coesiste con altri stati della materia. Per esempio, nella bassa ionosfera, ci sono regioni in cui il 99% degli atomi sono neutri e solo l'1% è ionizzato. In tali plasmi *parzialmente ionizzati* la componente ionizzata può legittimamente essere considerata plasma dalla (2.14) calcolando Λ esclusivamente dai parametri di quella stessa componente.

2.3 Frequenza di plasma

Un altro importante fenomeno collettivo è costituito dalle oscillazioni di plasma. Consideriamo un'ipotetica sezione di plasma di lato L e consideriamo ioni con massa infinita, carica opposta ma uguale densità n_0 rispetto agli elettroni, i quali sono rigidamente ancorati alle loro posizioni reciproche, ma passibili di muoversi liberamente attraverso gli ioni. Supponiamo inoltre che la sezione di plasma composta da elettroni sia ad una distanza δ da quella degli ioni e sia libera di muoversi attraverso quest'ultima (fig. 2.1). L'instaurarsi di un campo elettrico tra le due sezioni causerà un movimento collettivo degli elettroni verso gli ioni. Alla sovrapposizione delle due sezioni, la forza diverrà zero, ma gli elettroni avranno acquisito una velocità tale da oltrepassare la sezione ionica. Si instaurerà quindi un'oscillazione armonica tra le due sezioni. La frequenza di tale oscillazione è chiamata *frequenza di plasma*. Essa dipende dalla densità elettronica, dalla carica elettronica e dalla massa di questi ultimi. Vediamo in dettaglio. Per il campo elettrico in una

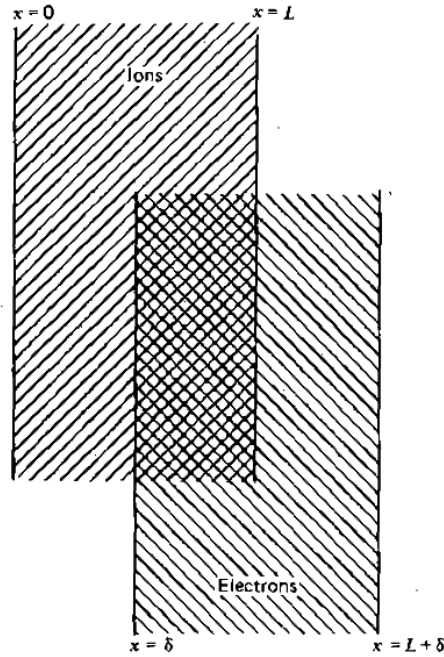


Figura 2.1:

modello a sezioni per calcolare la frequenza di plasma

dimensione:

$$\frac{\partial E}{\partial x} = \frac{\rho}{\varepsilon_0}, \quad (2.15)$$

con E campo elettrico. Prendiamo come condizione al contorno $E(x=0) = 0$ e assumiamo $\delta \ll L$ per tutto lo spazio. Dalla (2.15), il campo elettrico per gran parte della sezione sarà $-(n_0 e \delta) / \varepsilon_0$ e la forza per unità di area che agisce sulla sezione di elettroni sarà (campo elettrico \times carica per unità di area) quindi $-(n_0^2 e^2 \delta) / \varepsilon_0 L^2$. Dalla seconda legge di Newton (forza per unità di area = massa per unità di area \times accelerazione), scriviamo:

$$-\frac{n_0^2 e^2 \delta}{\varepsilon_0 L^2} = \left(\frac{n_0 m_e}{L^2} \right) \frac{\partial^2 \delta}{\partial t^2}, \quad (2.16)$$

che può essere riscritta nella forma:

$$\frac{\partial^2 \delta}{\partial t^2} + \left(\frac{n_0 e^2}{\varepsilon_0 m_e} \right) \delta = 0, \quad (2.17)$$

che è la forma standard dell'equazione di un oscillatore armonico di frequenza:

$$\omega_e \equiv \sqrt{\frac{n_e e^2}{\varepsilon_0 m_e}}, \quad (2.18)$$

detta *frequenza di plasma*. Analogamente, possiamo definire una frequenza di plasma per ioni di carica Ze e di densità n_i :

$$\omega_i \equiv \sqrt{\frac{n_i Z^2 e^2}{\varepsilon_0 m_i}}, \quad (2.19)$$

e di conseguenza, la frequenza di plasma totale ω_p^2 per due componenti sarà:

$$\omega_p^2 \equiv \omega_e^2 + \omega_i^2. \quad (2.20)$$

Per molti plasmi in natura $\omega_e \gg \omega_i$ e quindi possiamo considerare $\omega_p^2 \approx \omega_e^2$ per gli intenti di questa tesi.

Si deduce pertanto che il campo elettrico dovuto ad eventuali disomogeneità di carica in un plasma viene suddiviso in una parte a corto raggio (le diffusioni libere nella sfera di Debye), e in una parte a lungo raggio (fenomeni collettivi come l'oscillazione alla frequenza di plasma). Se siamo tuttavia interessati a fenomeni che avvengono su scale spaziali più grandi della sfera di Debye e su scale temporali più lente della frequenza di plasma, il plasma può essere trattato come un fluido neutro in cui i campi elettrici (spontanei) sono nulli.

2.4 Oscillazioni di plasma

Nella fisica dei plasmi, intendiamo per *electrostatic electron plasma wave (EPPW)* un sistema interconnesso di elettroni e campi elettrici che si propagano con proprietà periodiche in un plasma, escludendo oscillazioni di campi magnetici. Supponiamo di avere un EPPW, piana, di alta frequenza, propagantesi lungo la l'asse x in un plasma omogeneo. A causa dell'alta frequenza di oscillazione, gli ioni di cui è composto il plasma possono essere considerati come un background neutro e stazionario. Le equazioni che determinano le interazioni tra l'onda e gli elettroni di plasma sono le equazioni del moto e di continuità per il fluido elettrico, assieme all'equazione di Poisson, che provvede ad un accoppiamento autoconsistente tra l'onda e il moto degli elettroni. Poiché la velocità degli elettroni, il campo elettrico e il vettore d'onda giacciono sull'asse x , il problema è unidimensionale e le equazioni che governano il fenomeno sono:

$$\begin{aligned} \frac{\partial v}{\partial t} + v \frac{\partial v}{\partial x} &= -\frac{e}{m} E, \\ \frac{\partial n}{\partial t} + \frac{\partial(nv)}{\partial x} &= 0, \\ \frac{\partial E}{\partial x} &= -\frac{e}{\varepsilon_0} (n - n_0), \end{aligned} \quad (2.21)$$

con v la velocità e n la densità del fluido elettrico, E il campo elettrico, ε_0 costante dielettrica nel vuoto, con n_0 indichiamo la densità del background di ioni e con m ed e , rispettivamente la massa e la carica dell'elettrone.

Al fine di analizzare le caratteristiche della propagazione di EEPW, consideriamo l'evoluzione di piccole perturbazioni nella densità, velocità e campo elettrico in un background stazionario e omogeneo:

$$v = v_0 + v_1, \quad n = n_0 + n_1, \quad E = E_0 + E_1, \quad (2.22)$$

in cui il pedice 0 indica lo stato stazionario e omogeneo della quantità considerata e 1 indica la perturbazione su quella quantità. Considereremo una situazione in cui il campo elettrico in equilibrio sarà $E_0 = 0$, considereremo anche gli elettroni nello stato di equilibrio come stazionari rispetto agli ioni nello stato di equilibrio ($v_0 = 0$). Inserendo le (2.22) in (2.21), e prendendo solo i termini del primo ordine, avremo, per le (2.21):

$$\begin{aligned} \frac{\partial v_1}{\partial t} &= -\frac{e}{m} E_1, \\ \frac{\partial n_1}{\partial t} + n_0 \frac{\partial v_1}{\partial x} &= 0, \\ \frac{\partial E_1}{\partial x} &= -\frac{e}{\varepsilon_0} n_1. \end{aligned} \quad (2.23)$$

Stiamo cercando come soluzioni delle onde piane proporzionali a $\exp[i(kx - \omega t)]$ di ampiezza costante. k e ω sono rispettivamente la frequenza e il numero d'onda dell'EEPW. Possiamo quindi riscrivere le equazioni (2.23) come:

$$\begin{aligned} -i\omega v_1 &= -\frac{e}{m} E_1, \\ -i\omega n_1 + in_0 k v_1 &= 0, \\ ik E_1 &= -\frac{e}{\varepsilon_0} n_1. \end{aligned} \quad (2.24)$$

Eliminando v_1 ed n_1 dalle prime due equazioni, l'equazione di Poisson si riduce a:

$$\left(1 - \frac{\omega_p^2}{\omega^2}\right) E_1 = 0; \quad (2.25)$$

con ω_p frequenza di plasma. Escludendo quindi soluzioni banali della (2.25) otteniamo che $\omega = \pm\omega_p$; la frequenza della EEPW è uguale alla frequenza di plasma. Consideriamo ora una situazione in cui la parte stazionaria del problema corrisponde ad un fluido elettrico che scorre attraverso un background

stazionario di ioni. In questo caso abbiamo che $v_0 \neq 0$ e quindi le equazioni (2.23) divengono:

$$\begin{aligned}\frac{\partial v_1}{\partial t} + v_0 \frac{\partial v_1}{\partial x} &= -\frac{e}{m} E_1, \\ \frac{\partial n_1}{\partial t} + n_0 \frac{\partial v_1}{\partial x} + v_0 \frac{\partial v_1}{\partial x} &= 0, \\ \frac{\partial E_1}{\partial x} &= -\frac{e}{\varepsilon_0} n_1,\end{aligned}\tag{2.26}$$

e quindi, al posto delle (2.24), abbiamo:

$$\begin{aligned}-i(\omega - kv_0)v_1 &= -\frac{e}{m} E_1, \\ -i(\omega - kv_0)n_1 + in_0kv_1 &= 0, \\ ikE_1 &= -\frac{e}{\varepsilon_0} n_1,\end{aligned}\tag{2.27}$$

Ee in definitiva:

$$\left[1 - \frac{\omega_p^2}{(\omega - kv_0)^2}\right] E_1 = 0.\tag{2.28}$$

Quest'ultima avrà soluzioni non banali del tipo:

$$\omega = \omega_D \pm \omega_p.\tag{2.29}$$

La frequenza caratteristica trovata precedentemente è quindi spostata di una quantità $\omega_D = kv_0$ (detta *Doppler frequency*) a causa della velocità relativa del flusso di elettroni nel background di ioni.

Capitolo 3

L'instabilità di due stream

3.1 Instabilità in un plasma

Nella trattazione sulle oscillazioni di plasma si assume che il plasma si trovi in uno stato imperturbato, in perfetto equilibrio termodinamico. Le particelle cariche avranno distribuzioni di velocità di tipo Maxwelliano e la densità sarà uniforme. Gli stati di plasma considerato non sono perfettamente in equilibrio termodinamico, sebbene essi possano essere considerati in equilibrio nel senso in cui tutte le forze sono bilanciate ed è possibile scrivere una soluzione indipendente dal tempo per il sistema.

In questo stato di entropia massima, non esistono forme di energia libera capaci di innescare e sostenere oscillazioni di plasma, si suppone quindi che esse si alimentino traendo energia da fonti esterne al sistema. Quest'energia disponibile per il sistema a causa di fattori esterni, può far sì che le oscillazioni si autoalimentino, l'equilibrio del sistema diventa quindi instabile. Un'instabilità tende sempre ad abbassare il livello di energia libera disponibile per il sistema, portando il plasma vicino ad un vero equilibrio termodinamico. Queste instabilità possono essere classificate in base alla tipologia di energia libera fornita dall'esterno. Nel caso che studieremo, le *instabilità di streaming* sono dovute a fasci di particelle energetiche che attraversano il plasma o correnti formate da due specie diverse che si attraversano l'un l'altra con determinate velocità di deriva. L'energia associata a queste velocità di deriva è usata per innescare le oscillazioni e alimentarle a spese di quella stessa velocità di deriva, come vedremo più avanti.

3.2 Instabilità di due stream. Generalità.

L'instabilità di due stream di cariche è un fenomeno molto comune nella fisica dei plasmi. Essa è generata da intense correnti di particelle iniettate in un plasma oppure da due stream di particelle cariche che si attraversano l'un l'altra. Mentre un analogo problema idrodinamico fu già studiato da *H.Hertz*, i primi modelli per simulare instabilità di particelle cariche furono introdotti da *Haeff* (1949) e *Pierce* (1948), mentre uno studio dettagliato del suo comportamento non lineare fu effettuato con simulazioni compiute da *Dawson*. Per studiare quest'instabilità generalizziamo i risultati esposti nella sezione 2.4 combinando i due casi già esaminati. Consideriamo un fluido elettrico composto da due specie diverse, una a riposo rispetto al background stazionario degli ioni, l'altra che si muove con velocità v_0 . Indichiamo le densità dei due fluidi con n_{01} e n_{02} rispettivamente. La condizione di plasma neutrale impone quindi che $n_0 = n_{01} + n_{02}$. La dinamica di un EEPW propagantesi in questo tipo di mezzo è determinata separatamente da equazioni del moto (2.21a) e (2.24a) e di continuità (2.21b) e (2.24b) per i due rispettivi fluidi. L'equazione di Poisson combina invece la dinamica dei due fluidi e sarà, dalla (2.24c):

$$ikE_1 = -\frac{e}{\epsilon_0}n_{11} + n_{21}, \quad (3.1)$$

con n_{11} e n_{21} denotano piccole perturbazioni sulle densità dei fluidi di elettroni a causa del campo elettrostatico. Risolvendo la (3.1) per n_{11} ed n_{21} ed usando altri risultati della sezione 2.4, l'equazione di Poisson può essere scritta come:

$$E_1 = \left[\frac{n_{01}e^2}{m\epsilon_0} \frac{1}{\omega^2} + \frac{n_{02}e^2}{m\epsilon_0} \frac{1}{(\omega - \omega_D)^2} \right] E_1, \quad (3.2)$$

che porta alla relazione di dispersione:

$$1 = \frac{\omega_{p1}^2}{\omega^2} + \frac{\omega_{p2}^2}{(\omega - \omega_D)^2}, \quad (3.3)$$

dove ω_{p1} e ω_{p2} sono, rispettivamente, le frequenze di plasma del primo fluido elettronico e del secondo. È possibile trovare analiticamente la soluzione generale dell'equazione (3.3) (un'equazione di quarto grado in ω), ma, data la sua complessità, utilizzeremo altri metodi di soluzione. Alcune proprietà di questa soluzione possono infatti essere trovate considerando la funzione $F(\omega)$:

$$F(\omega) \equiv \frac{\omega_{p1}^2}{\omega^2} + \frac{\omega_{p2}^2}{(\omega - \omega_D)^2}. \quad (3.4)$$

La funzione $F(\omega)$ ha due asintoti verticali, uno ad $\omega = 0$ e un altro per $\omega = \omega_D$.

Se ω_D è sufficientemente grande, la funzione $F(\omega)$ appare qualitativamente

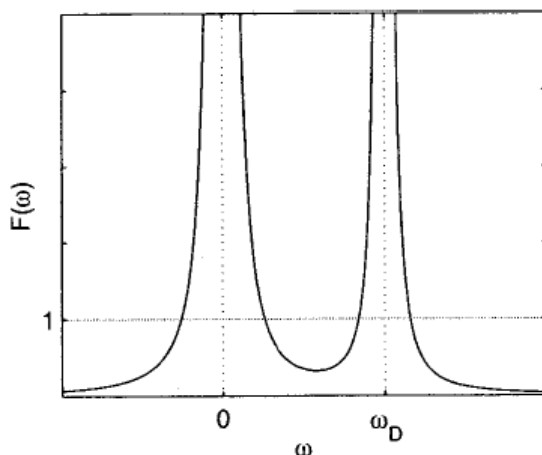


Figura 3.1:

Grafico qualitativo della funzione $F(\omega)$ in caso di grande valore di ω_D per $F_{min} < 1$.

come in fig 3.1, di conseguenza, le soluzioni dell'equazione (3.3) sono ben separate e si può scrivere che:

$$\frac{\omega_{p1}^2}{\omega^2} \approx 1, \quad (3.5)$$

$$\frac{\omega_{p2}^2}{(\omega - \omega_D)^2} \approx 1, \quad (3.6)$$

riottenendo le frequenze di oscillazione per i singoli fluidi (2.25) e (2.28). Se invece ω_D è equiparabile a ω_{p1} , la situazione si complica. Il comportamento qualitativo della soluzione dell'equazione (3.3) può essere dedotta focalizzando sul punto in cui la funzione $F(\omega)$ ha un minimo. Questo minimo sarà nel punto:

$$\omega = \omega_{min} = \frac{\omega_D}{1 + (\omega_{p2}/\omega_{p1})^{2/3}} \equiv \frac{\omega_D}{1 + \alpha^{2/3}}, \quad (3.7)$$

, in cui abbiamo indicato con $\alpha = \omega_{p1}/\omega_{p2}$. Il minimo valore di F sarà:

$$F_{min} = F(\omega_n) = \frac{\omega_{p1}^2}{\omega_D^2} (1 + \alpha^{2/3})^3. \quad (3.8)$$

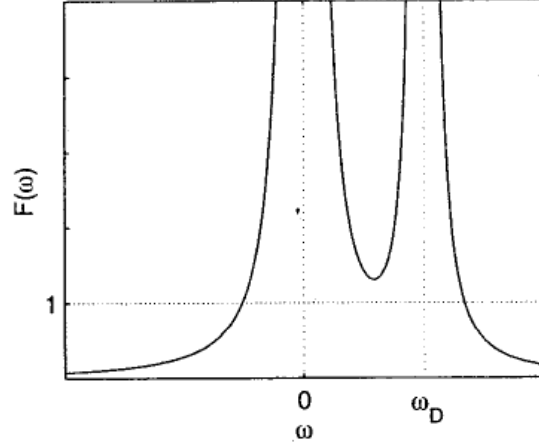


Figura 3.2:

Grafico qualitativo della funzione $F(\omega)$ in caso ω_D sia comparabile a ω_{p1} per $F_{min} > 1$.

Al fine di semplificare il problema supponiamo che la densità del fluido di elettroni in movimento abbia una densità molto minore di quella del fluido stazionario ($n_{02} \ll n_{01}$ o, equivalentemente $\omega_{p2} \ll \omega_{p1}$). Si avrà quindi $\alpha \ll 1$. In questo limite:

$$\omega_{min} \approx \omega_D(1 - \alpha^{2/3}), \quad F_{min} \approx \frac{\omega_{p1}^2}{\omega_D^2}(1 + 3\alpha^{2/3}). \quad (3.9)$$

Se $\omega_{p1}^2/\omega_D^2 < (1 + 3\alpha^{2/3})$, abbiamo che $F_{min} < 1$ e quindi l'equazione $F(\omega) = 1$ ha quattro soluzioni reali approssimate dalle soluzioni delle (3.5) e (3.6).

Se invece $\omega_{p1}^2/\omega_D^2 > (1 + 3\alpha^{2/3})$, $F_{min} > 1$ e quindi l'equazione $F(\omega) = 1$ ha solo due soluzioni reali; una più piccola di $-\omega_{p1}$ e l'altra più grande di $\omega_D + \omega_{p2}$ (fig 3.2). Le soluzioni rimanenti sono complesse e coniugate: $\omega = \omega_{re} \pm i\omega_{im}$. Questo implica un fenomeno del tutto nuovo: l'ampiezza della EEPW ha una soluzione (quella col segno +) che, durante la perturbazione, cresce nel tempo come:

$$\exp[i(kx - \omega_{re}t) + \omega_{im}t] = \exp[i(kx - \omega_{re}t)]\exp[\omega_{im}t]. \quad (3.10)$$

Questo tipo di instabilità è chiamata *instabilità di due stream*. Il rate di crescita dell'instabilità può essere approssimativamente ottenuto come di seguito:

Svliuppiamo la funzione $F(\omega)$ intorno al punto di minimo:

$$F(\omega) \approx F(\omega_{min}) + F'(\omega_{min})(\omega - \omega_{min}) + \frac{1}{2}F''(\omega_{min})(\omega - \omega_{min})^2 + \dots, \quad (3.11)$$

considerando l'equazione $F(\omega) = 1$ e fermandoci al secondo ordine in ω , poichè $F'(\omega_{min}) = 0$ e $F''(\omega_{min}) > 0$, assumendo anche che $F(\omega_{min}) > 1$ abbiamo che:

$$\omega - \omega_{min} = \pm i \sqrt{\left[2 \frac{F(\omega_{min}) - 1}{F''(\omega_{min})} \right]}, \quad (3.12)$$

con la parte reale e immaginaria di ω data da:

$$\omega_{re} = \omega_{min}, \quad \omega_{im} = \pm \sqrt{\left[2 \frac{F(\omega_{min}) - 1}{F''(\omega_{min})} \right]}. \quad (3.13)$$

L'espressione per ω_{im} vale esattamente:

$$\omega_{im} = \frac{\alpha^{1/3}}{\sqrt{3}(1 + \alpha^{2/3})^{5/2}} \frac{\omega_D^2}{\omega_{p1}} \sqrt{\left[\frac{\omega_{p1}^2}{\omega_D^2} (1 + \alpha^{2/3})^3 - 1 \right]}. \quad (3.14)$$

Nel limite $\alpha \ll 1$, ω_{re} e ω_{im} possono essere semplificati come:

$$\begin{aligned} \omega_{re} &\approx \omega_D(1 - \alpha^{2/3}), \\ \omega_{im} &\approx \pm \frac{\alpha^{1/3}}{\sqrt{3}} \frac{\omega_D^2}{\omega_{p1}} \sqrt{\left(\frac{\omega_{p1}^2}{\omega_D^2} - 1 \right)} = \frac{\alpha^{1/3}}{\omega_D} \sqrt{\left(1 - \frac{\omega_D^2}{\omega_{p1}^2} \right)}. \end{aligned} \quad (3.15)$$

Evidenziamo che la parte reale di ω non coincide con ω_D ma ne è molto vicino. Infatti dalla relazione $\omega_{re} = \omega_D(1 - \alpha^{2/3})$, con $\omega_D = kv_0$ (2.29) implica che la velocità di fase della EEPW, sarà data da:

$$v_f \equiv \frac{\omega_{re}}{k} \approx v_0(1 - \alpha^{2/3}) \quad (3.16)$$

Quindi, l'instabilità si verifica quando l'EEPW e lo stream del fluido di elettroni sono vicini alla risonanza $v_f \approx v_0$. In questa situazione l'accoppiamento tra lo streaming di elettroni e l'onda EEPW diventa molto forte e viene trasferita energia dagli elettroni, i quali si muovono con una velocità leggermente più grande rispetto alla velocità di fase, all'onda, che di conseguenza cresce.

3.3 Analisi lineare di due stream con velocità uguali ed opposte

Supponiamo ora che le due specie diverse del fluido abbiano velocità diverse v_{01} e v_{02} . Usando le nozioni del precedente paragrafo, scriviamo la relazione di dispersione (3.3) esplicitando le rispettive *Doppler frequency*:

$$\omega_{D1} = kv_{01}, \quad \omega_{D2} = kv_{02}. \quad (3.17)$$

Abbiamo una relazione di dispersione (3.3) del tipo:

$$1 = \frac{\omega_{p1}^2}{(\omega - kv_{01})^2} + \frac{\omega_{p2}^2}{(\omega - kv_{02})^2}, \quad (3.18)$$

anch'essa di quarto grado rispetto ad ω . Se supponiamo che:

$$\omega_{p1} = \omega_{p2} \equiv \omega_p, \quad v_{01} = -v_{02} \equiv v_0. \quad (3.19)$$

Le quattro soluzioni indipendenti della (3.18) saranno, in maniera esplicita:

$$\omega = \pm \sqrt{\left[k^2 v_0^2 + \omega_p^2 \pm \omega_p \sqrt{(4k^2 v_0^2 + \omega_p^2)} \right]}, \quad (3.20)$$

in cui, per:

$$0 < \frac{kv_0}{\omega_p} < \sqrt{2}, \quad (3.21)$$

abbiamo due soluzioni reali e due immaginarie. Se

$$\sqrt{2} < \frac{kv_0}{\omega_p}, \quad (3.22)$$

tutte e quattro le soluzioni sono reali. Se invece

$$\frac{kv_0}{\omega_p} = \frac{\sqrt{3}}{2}, \quad (3.23)$$

la parte immaginaria di ω_p , $\omega_{im} = \omega_p/2$, rappresenta il massimo rate di crescita dell'instabilità. Nella figura (3.3) e (3.4) sono rappresentati rispettivamente il diagramma $\omega - k$ per le soluzioni (3.19) e il rate di crescita di ω_{im} .

Dalla fig. 3.4 e dalla condizione (3.21) si ottiene che si ha instabilità quando:

$$\frac{\omega_p L}{v_0} > \frac{2\pi}{\sqrt{2}}, \quad (3.24)$$

3.3 Analisi lineare di due stream con velocità uguali ed opposte 29

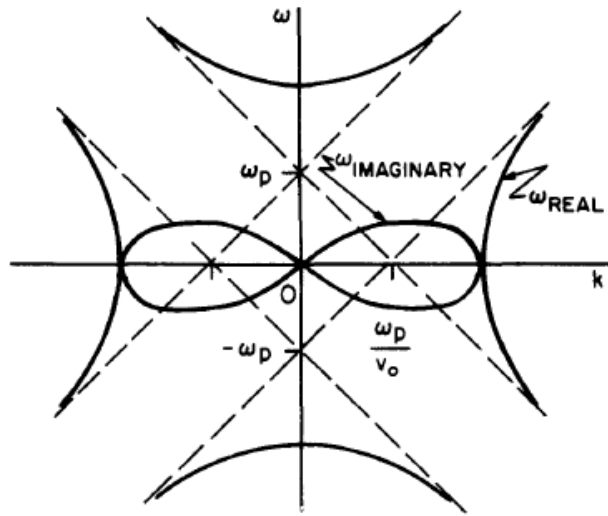


Figura 3.3:

Diagramma $\omega - k$ per due stream di velocità opposta, k reale e ω complesso. Notiamo che per ogni valore di k corrispondono quattro valori di ω corrispondenti alle quattro onde linearmente indipendenti.

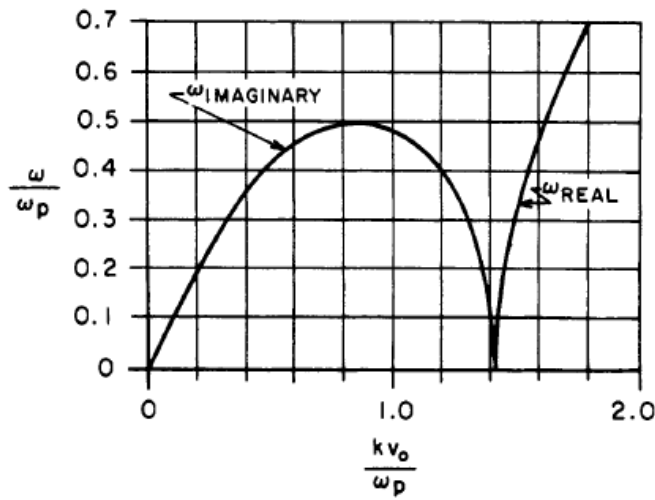


Figura 3.4:

Rate di crescita per ω_{im}

in cui definiamo $L = 2\pi/k_0$ come *lunghezza minima di instabilità del sistema* (con k_0 il più basso numero d'onda del sistema).

L'aumento di ω_{im} , innescato per piccole ampiezze dell'oscillazione, continua

3.3 Analisi lineare di due stream con velocità uguali ed opposte 30

finchè lo stream dei due fasci di elettroni non è completamente distrutto, la distribuzione tenderà quindi a ritornare Maxwelliana. Si dice in questo caso che gli stream collidenti sono stati *termalizzati*. Gli effetti collettivi del sistema innescano però dei campi elettrici la cui oscillazione ha una lunghezza d'onda maggiore della lunghezza di separazione delle particelle. Sono questi campi a distribuire le particelle con la tipica configurazione a vortici nello spazio delle fasi.

Chiamiamo $f(v_x)$ la distribuzione delle velocità lungo x. Quando l'instabilità cresce, osserveremo il cambiamento nella distribuzione di velocità come mostrato in (fig. 3.5). Per $t > 0$ osserviamo quindi una diminuzione della

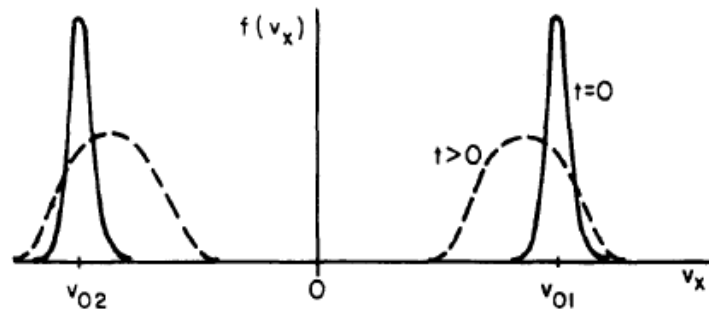


Figura 3.5:
rappresentazione della distribuzione di velocità dei due stream per $t = 0$ e $t > 0$.

velocità media \bar{v} e l'aumento della larghezza della $f(v_x)$ per i due stream a $t > 0$ (collegato all'incremento della temperatura associata ad ogni stream). Ci aspettiamo che, al diminuire di \bar{v} e all'aumentare di $v_{termico}$, che le condizioni per una crescita lineare dell'instabilità cessino di essere soddisfatte (la soglia per due streams elettrone-elettrone si dimostra essere: $\bar{v} \approx 1.3 \times v_{term}$) e che la crescita esponenziale si arresti. Ciò nonostante, in questi frangenti, le particelle di uno stream attraverseranno quelle dell'altro stream e si troveranno invischiate in vortici nello spazio delle fasi, saranno quindi *intrappolate*. Questa condizione di intrappolamento diviene autoconsistente in tutto l'arco dell'instabilità. Come è possibile osservare in (fig.3.6), gli elettroni, soddisfacendo la relazione di continuità nel fluido ed essendo attratti dalla zona positiva al centro del vortice, aumentano di velocità. Quest'aumento di velocità si traduce nello spazio delle fasi in un allargamento del singolo vortice.

La crescita dell'instabilità, in queste condizioni, non potrà dunque fermarsi, sebbene si potrebbe essere tentati di cercare un cambiamento nelle caratteristiche di questa crescita dal momento in cui $v_{termico}$ supera $\bar{v}/1.3$. Nel pro-

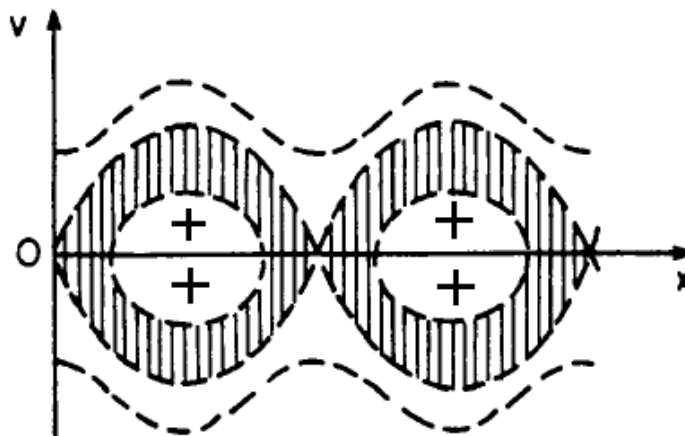


Figura 3.6: L'intrappolamento delle particelle di carica negativa nei vortici dello spazio delle fasi è una condizione autoconsistente: durante l'instabilità, le particelle nella zona segnata saranno attratte dalla zona di carica positiva (il centro del vortice), e aumentando la loro velocità consequenzialmente a questo, contribuiranno ad allargare il vortice stesso.

gramma che esporremo di seguito, ci aspettiamo una crescita dell'instabilità per $v_0 = 2 \times v_{termico}$ o $3 \times v_{termico}$ e stabilità per $v_0 = v_{termico}$

3.4 L'instabilità di due stream: una simulazione col metodo Particle-in-Cell

Consideriamo un plasma uniforme, non magnetizzato e unidimensionale composto da N elettroni ed N ioni. Questi ioni avranno una massa molto superiore a quella degli elettroni, di conseguenza, considerati i tempi relativamente brevi per la simulazione, saranno considerati come un background stazionario. Ci focalizzeremo esclusivamente sul moto degli elettroni. Siano x_i le coordinate di ciascuno di questi elettroni. Le equazioni del moto per ciascuno di questi elettroni possono essere scritte come:

$$\frac{dx_i}{dt} = v_i, \tag{3.25}$$

$$\frac{dv_i}{dt} = -\frac{eE(x_i)}{m_e}, \tag{3.26}$$

dove abbiamo indicato con e la carica dell'elettrone, con m_e la sua massa e con $E(x)$ la componente x del campo elettrico nella posizione x . Il campo

elettrico potrà a sua volta essere scritto come:

$$E(x) = -\frac{d\phi(x)}{dx}, \quad (3.27)$$

e quindi, usando l'equazione di Poisson-Maxwell, abbiamo:

$$\frac{d^2\phi(x)}{dx^2} = -\frac{e}{\varepsilon_0}(n_0 - n(x)), \quad (3.28)$$

con ε_0 costante dielettrica del vuoto, $n(x)$ densità di numero degli elettroni ($n(x)dx$ è il numero di elettroni nell'intervallo da x a $x + dx$) ed n_0 la densità di numero del background di ioni. Per ottenere la neutralità del plasma, abbiamo ovviamente che il valore medio della densità di numero degli elettroni $\langle n(x) \rangle$ deve essere uguale a n_0 , poichè il numero degli elettroni deve essere uguale a quello degli ioni. La distribuzione (Maxwelliana) iniziale degli elettroni consta di due stream di velocità v_b e $-v_b$ e la funzione di distribuzione $f(x, v)$ si può scrivere come:

$$f(x, v) = \frac{n_0}{2} \left[\frac{1}{\sqrt{2\pi}v_{th}} e^{-(v-v_b)^2/2v_{th}^2} + \frac{1}{\sqrt{2\pi}v_{th}} e^{-(v+v_b)^2/2v_{th}^2} \right], \quad (3.29)$$

con v_{th} è la velocità termica definita come:

$$v_{th} = \sqrt{\frac{k_b T}{m_e}}, \quad (3.30)$$

Con k_b costante di Boltzmann e T temperatura del "gas" elettronico.

$f(x, v)dx dv$ è il numero di elettroni tra x e $x + dx$ con velocità tra v e $v + dv$ e ovviamente $n(x) = \int_{-\infty}^{\infty} f(x, v) dv$.

Nella simulazione, i tempi saranno normalizzati alla frequenza di plasma (2.18), le posizioni x_i delle particelle alla lunghezza di Debye (2.5) e le velocità delle particelle rispetto alla velocità termica v_{th} (Notiamo che $\lambda_D = v_{th}/\omega_p$). Le (3.25, 3.26, 3.27, 3.28) diverranno:

$$\frac{dx_i}{dt} = v_i, \quad (3.31)$$

$$\frac{dv_i}{dt} = -E(x_i), \quad (3.32)$$

$$E(x) = -\frac{d\phi(x)}{dx}, \quad (3.33)$$

$$\frac{d^2\phi(x)}{dx^2} = \frac{n(x)}{n_0} - 1. \quad (3.34)$$

Notando che nel nostro schema di normalizzazione $v_{th} = 1$, la distribuzione di velocità iniziale diventerà:

$$f(x, v) = \frac{n_0}{2} \left[\frac{1}{\sqrt{2\pi}} e^{-(v-v_b)^2/2} + \frac{1}{\sqrt{2\pi}} e^{-(v+v_b)^2/2} \right]. \quad (3.35)$$

Risolveremo questo sistema di equazioni nel dominio $0 \leq x \leq L$ e adotteremo condizioni al contorno periodiche tali per cui:

$$\begin{aligned} n(0) &= n(L), \\ \phi(0) &= \phi(L), \\ E(0) &= E(L). \end{aligned} \quad (3.36)$$

Inoltre, imporremo che ogni elettrone che superi il confine destro del dominio riapparirà entro il confine di sinistra con la stessa velocità e viceversa.

3.5 Il codice

Viene presentato qui di seguito il codice Particle-in-Cell in linguaggio di programmazione C utilizzato per simulare l'instabilità di due stream.

Alcuni tra i più importanti parametri del programma sono stati definiti come costanti e inizializzati come tali fuori dal *main*:

```
const int J = 1000; // Numero di celle della griglia spaziale//
const int N = 80000; //numero di particelle computate//
const double L = 100; //lunghezza totale della griglia (in
lunghezza di Debye)//
const double vb = 3; velocità iniziale dei due fasci di
particelle (in velocità termica degli elettroni)
const double dt = 0.1; //step temporale (in frequenza di plasma
inversa)//
const double tmax = 21; //condizioni iniziali rk4//
const double alpha_l = 1; //condizioni iniziali rk4//
const double beta_l = 0; //condizioni iniziali rk4//
const double gamma_l = 0; //condizioni iniziali rk4//
const double alpha_h = 1; //condizioni iniziali rk4//
const double beta_h = 0; //condizioni iniziali rk4//
const double gamma_h = 0; //condizioni iniziali rk4//
const double dx = 0.1; // L/J (dimensioni della
singola cella in lunghezza di Debye)//
```

All'interno del *main* sono invece definiti i vettori principali del programma:

```

double r[N]; //vettore per le posizioni delle particelle//
double v[N]; //vettore per le velocità delle particelle//
double n[J]; // vettore densità di numero di elettroni
per ogni cella della griglia//
double rho[J]; // vettore densità di carica totale//
double phi[J]; // vettore per il potenziale elettrico per
unità di griglia//
double E[J]; //vettore campo elettrico per unità di griglia//
double E1[N]; //vettore campo elettrico per particella//
double dydt[2]; //vettori usati nella funzione rk4//
double sol[2]; //vettore usato nella funzione rk4//

```

Presentiamo ora una breve rassegna delle funzioni principali del programma:

La funzione *boundary* seleziona le particelle che hanno oltrepassato il dominio L (sia da destra che da sinistra) e le riposiziona all'interno di quest'ultimo, rispettivamente nel lato opposto rispetto a quello da cui sono sfuggite:

```

void boundary( double r[]){
int i;
for(i = 0; i < N; i++) {
if(r[i]< 0.) r[i] += L;
if(r[i] > L) r[i] -=L;
}
}

```

La funzione di distribuzione Maxwelliana per le velocità iniziali degli elettroni è stata implementata con un metodo chiamato *rejection method*:

```

double distribution (double vb)
{
// inizializza il generatore di numeri casuali
static int flag = 0;
if (flag == 0)
{
int seed = time (NULL);
srand (seed);
flag = 1;
}

\\ genera una velocità randomica

```

```

double fmax = 0.5 *(1. + exp (-2. * vb * vb));
double vmin = - 5. * vb; // vmin = - 5 vb
double vmax = + 5. * vb; //vmax = + 5 vb
double v = vmin + (vmax - vmin) * (double)rand ()/
(double)RAND_MAX; //v è un numero random tra vmin e vmax

\\accetta/rigetta il valore
double f = 0.5 *(exp -(v - vb) * (v - vb) / 2.) +
exp -(v + vb) * (v + vb) / 2.);
double x = fmax * (double)rand () / (double)RAND_MAX;
if (x > f) return distribution (vb);
else return v;
}

```

La funzione *density* carica il vettore densità $n[J]$ dal vettore posizione $r[N]$. Al fine di ottenere la densità di numero degli elettroni dalle posizioni r_i adottiamo il metodo Particle-in-Cell già introdotto nel capitolo 1. Avendo già definito un set di J celle equamente distanziate e localizzate nelle coordinate:

$$X_j = j\delta X, \quad (3.37)$$

per $j = 0, 1, \dots, J-1$ con $\delta X = L/J$. Supponiamo sia $n_j \equiv n(X_j)$ e supponiamo che l' i -esimo elettrone si trovi tra il j^{th} e il $(j+1)^{\text{th}}$ punto della griglia, quindi: $X_j < r_i < X_{j+1}$. Abbiamo quindi che:

$$n_j \rightarrow n_j + \left(\frac{X_{j+1} - r_i}{X_{j+1} - X_j} \right) / \delta x, \quad (3.38)$$

$$n_{j+1} \rightarrow n_{j+1} + \left(\frac{r_i - X_j}{X_{j+1} - X_j} \right) / \delta x. \quad (3.39)$$

Quindi, n_j aumenta di uno se l'elettrone è nel j^{th} punto della griglia e n_{j+1} aumenta di uno se l'elettrone è nel $(j+1)^{\text{th}}$ punto della griglia. Se invece l'elettrone è a metà rispetto a due punti successivi j e $j+1$ della griglia, sia n_j che n_{j+1} incrementeranno di $1/2$. Un simile algoritmo riempirà quindi il vettore $n[J]$ a partire dalle coordinate dei punti, come già detto. Qui di seguito la funzione:

```

void density(double r[], double n[])
{
int azz,i;
for(azz=0; azz< J; azz++) n[azz]=0; //azzeramento densità

```

```

for (i = 0; i < N; i++){
int j = (int)(r[i]/dx);
double y = r[i]/dx - (double)j;
n[j] += (1.- y)/dx; //da
if(j+1 == J) n[0] += y/dx;
else n[j+1] += y/dx;
}
}

```

La funzione *density1* terrà conto della densità degli ioni n_0 secondo la (3.52), determinando la densità di carica reale da utilizzare nell'equazione di Poisson:

```

void density1(double ne[], double rho[])
{
double n0 = (double) (N) / L;
int j;
for (j = 0; j < J; j++) rho[j] = ne[j]/n0 - 1.;
}

```

Risolviamo ora l'equazione di Poisson e carichiamo dal vettore $n[j]$ il vettore potenziale $\phi[J]$. Il metodo scelto per la soluzione è chiamato metodo della matrice tridiagonale (vedi Appendice A)

```

void tridiagonal (double a1[], double b1[], double c1[],
double w1[], double u[])
{
\\dichiarazione dei vettori\\
double x1[J-2], y1[J-2];
int i;

x1[J-3]= - a1[J-2]/ b1[J-2];
y1[J-3]= w1[J-2] / b1[J-2];
for (i = J-4; i > 0; i--)
{
x1[i] = - a1[i+1] / (b1[i+1] + c1[i+1] * x1[i+1]);
y1[i] = (w1[i+1] - c1[i+1] * y1[i+1])/
(b1[i+1] + c1[i+1] * x1[i+1]);
}
x1[0] = 0.;
y1[0] = (w1[1] - c1[1] * y1[1] )/

```



```

(b1[1] + c1[1] * x1[1]);

u[1] = y1[0];
for (i = 1; i < J-2; i++)
u[i+1] = x1[i] * u[i] + y1[i];
}

void poisson1D ( double u[], double rho[])
{
// dichiarazione vettori//
double a1[J], b1[J], c1[J], w1[J];
int i;

// inizializzazione matrice tridiagonale//
for (i = 2; i <= J-2; i++) a1[i] = 1.;
for (i = 1; i <= J-2; i++) b1[i] = -2.;
b1[1] -= beta_l / (alpha_l * dx - beta_l);
b1[J-2] += beta_h / (alpha_h * dx + beta_h);
for ( i = 1; i <= J-3; i++) c1[i] = 1.;

// inizializzazione vettori di destra//
for (i = 1; i <= J-2; i++)
w1[i] = rho[i] * dx * dx;
w1[1] -= gamma_l * dx / (alpha_l * dx - beta_l);
w1[J-2] -= gamma_h * dx / (alpha_h * dx + beta_h);

// inversione matrice tridiagonale//
tridiagonal (a1, b1, c1, w1, u);

// Calcolo dei valori da i=0 a i=N-1//
u[0] = (gamma_l * dx - beta_l * u[1]) / (alpha_l * dx - beta_l);
u[J-1] = (gamma_h * dx + beta_h * u[J-2]) / (alpha_h * dx + beta_h);
}

```

Con la funzione *electric* calcoliamo il vettore campo magnetico $E[J]$ per ogni punto della griglia a partire dal potenziale precedentemente calcolato. Facciamo utilizzo della seguente equazione discreta:

$$E_j = \frac{\phi_{j-1} - \phi_{j+1}}{2\delta x}. \quad (3.40)$$

Ovviamente i casi in cui $j = 0$ e $j = J - 1$ saranno risolti utilizzando le condizioni al contorno (3.55). Il codice per questa funzione sarà:

```
void electric (double u[], double E[])
{
int j;
for (j = 1; j < J-1; j++)
E[j] = (u[j-1] - u[j+1]) / (dx*2.);
E[0] = (u[J-1] - u[1]) / (dx*2.);
E[J-1] = (u[J-2] - u[0]) / (dx*2.);
}
```

Usiamo invece la funzione *electric1* per ottenere il vettore dei campi elettrici puntati sulle singole particelle $E[N]$ a partire da $E[J]$ puntato sulle celle della griglia. Per far ciò, supponiamo che l' i^{th} elettrone si trovi tra il j^{th} e il $(j+1)^{\text{th}}$ punto della griglia. Possiamo usare un'interpolazione lineare per valutare il campo elettrico sentito dal i^{th} elettrone come:

$$E(r_i) = \left(\frac{x_{j+1} - r_i}{x_{j+1} - x_j} \right) E_j + \left(\frac{r_i - x_j}{x_{j+1} - x_j} \right) E_{j+1}, \quad (3.41)$$

che può essere considerato l'algoritmo inverso di quello utilizzato per interpolare il potenziale dalle posizioni delle particelle. Il codice sarà:

```
void electric1(double r0[], double E[], double E1[])
{
int i;
for (i = 0; i < N; i++){

int j = (int)(r0[i]/dx);
double y = r0[i]/dx - (double)j;

if(j+1 == J) E1[i] = E[j]*(1. - y) + E[0]*y;
else E1[i] = E[j]*(1.-y) + E[j+1]*y;
}
}
```

Chiude la rassegna delle funzioni utilizzate nel programma l'algoritmo Runge-Kutta di ordine 4 *rk4* utilizzato per risolvere numericamente il set di $2N$ equazioni differenziali del primo ordine per il moto delle particelle:

```
void rk4_fixed (double y[], double h, double dydt[])
```

```
{  
  
int j;  
  
double k1[2], k2[2], k3[2], k4[2], f[2];  
  
//1 step  
  
for (j = 0; j < 2; j++)  
{  
k1[j] = h * dydt[j];  
f[j] = y[j] + k1[j] / 2.;  
}  
  
// primo step intermedio  
  
for (j = 0; j < 2; j++)  
{  
k2[j] = h * dydt[j];  
f[j] = y[j] + k2[j] / 2.;  
}  
  
// secondo step intermedio  
  
for (j = 0; j < 2; j++)  
{  
k3[j] = h * dydt[j];  
f[j] = y[j] + k3[j];  
}  
  
// terzo step intermedio  
  
for (j = 0; j < 2; j++)  
{  
k4[j] = h * dydt[j];  
}  
  
for (j = 0; j < 2; j++)  
{  
y[j] += k1[j] / 6. + k2[j] / 3. + k3[j] / 3. + k4[j] / 6.;  
}
```

```
return;  
}
```

Mostriamo infine il corpo del programma, il *main*, sintetizzandone i passaggi:

```
double t = 0;  
for(t= 0; t <= tmax; t= t+ dt){  
  
    boundary(r);  
    density(r,n);  
    density1(n, rho);  
    poisson1D(phi, rho);  
    electric(phi, E);  
    electric1(r, E, E1);  
  
    for(i = 0; i < N; i++){  
  
        dydt[0] = v[i];  
        dydt[1] = -E1[i];  
        sol[0] = r[i];  
        sol[1] = v[i];  
  
        rk4_fixed(sol, dt, dydt);  
  
        r[i] = sol[0];  
        v[i] = sol[1];
```

Come si può osservare dal codice, in ogni singolo ciclo temporale vengono richiamate le funzioni esposte in precedenza, fino ad ottenere il vettore $E[N]$ col campo elettrico sentito da ogni particella. Nel ciclo *for* successivo viene calcolata la nuova posizione e la nuova velocità per ogni singola particella. Per far ciò, ci serviamo del campo elettrico $E[N]$ sentito dalla particella e delle sue coordinate posizione e velocità possedute nel ciclo precedente a quello considerato (queste ultime prese come condizioni iniziali delle equazioni differenziali del moto).

3.6 Output del programma

La figura 3.6 mostra l'evoluzione della distribuzione degli elettroni nello spazio delle fasi. Si sono processate $8 \cdot 10^4$ particelle e si è riportato il grafico dello

spazio delle fasi per intervalli di tempo equidistanziati. Come si può notare, la distribuzione iniziale di elettroni ha la forma di due bande corrispondenti ai due stream di elettroni distribuiti in maniera Maxwelliana, con velocità opposta $v_b = 3v_{th}$. Con l'andare del tempo, le bande sviluppano spontaneamente strutture che crescono in grandezza fino a convertire la distribuzioni in un'insieme di vortici connessi. Nello stato finale, gli elettroni, sottoposti ad un potenziale elettrico periodico generato da disomogeneità nella densità di carica, oscillano avanti e indietro. In dettaglio, per $t = 0.0$ e fino a $t = 5.0$ i due stream hanno velocità di deriva distribuite in maniera maxwelliana intorno ai valori $v_b = \pm 3$ e soffrono di una lieve perturbazione che cresce esponenzialmente nel tempo. La densità di carica, il potenziale e il campo elettrico oscillano in maniera sinusoidale fino a $t = 12.00$, momento in cui la perturbazione diventa rilevante e la densità di carica sulla griglia comincia a presentare diversi picchi. In questo frangente, tali picchi nella densità di carica generano forti campi elettrici che frenano gli stream, e alcune particelle (già a $t = 15.00$) si fermano e invertono il proprio percorso, formando quindi vortici nello spazio delle fasi. Si noti che a causa di questi campi intensi, la velocità media delle particelle supera anche di due unità il valore iniziale di $3v_{th}$. L'instabilità ha quindi effettivamente distrutto i due stream iniziali. È per questa ragione che l'instabilità di due stream è uno dei problemi fondamentali nella realizzazione tecnica di acceleratori di particelle, nei quali spesso si considerano due fasci di particelle cariche di opposta velocità che interagiscono l'uno con l'altro.

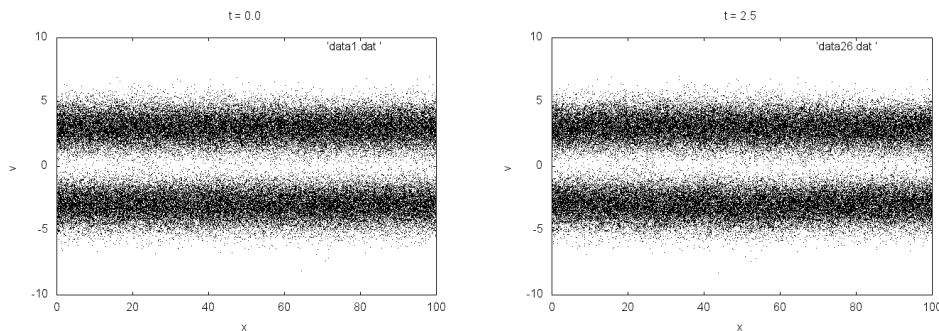


Figura 3.7: Rappresentazione del sistema nello spazio delle fasi per $t = 0.0$ e $t = 2.5$. Le posizioni sono normalizzate in lunghezza di Debye λ_D , i tempi nell'inverso della frequenza di plasma ω_p e le velocità nella velocità termica v_{th}

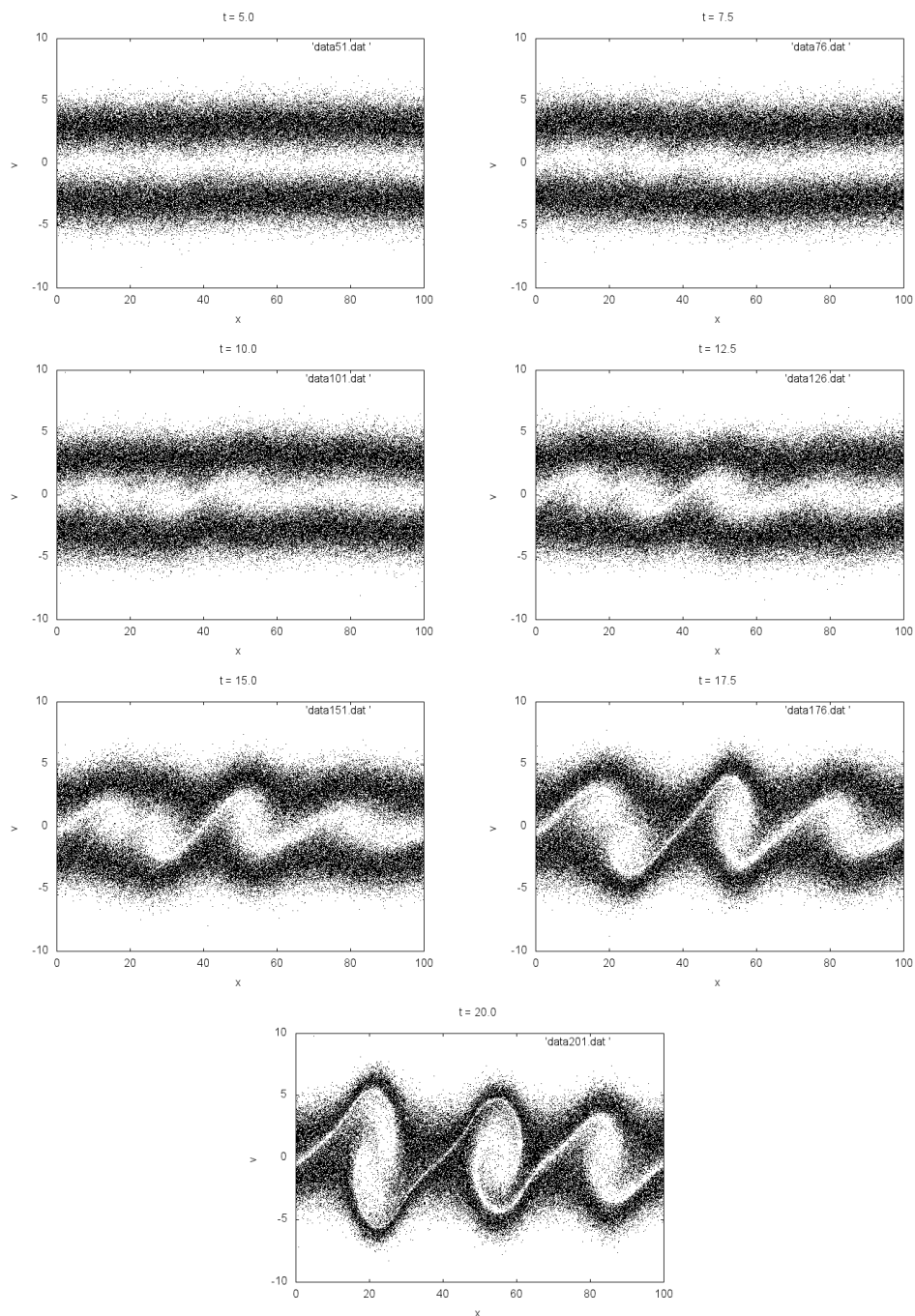


Figura 3.8: Rappresentazione del sistema nello spazio delle fasi per $t = 5.0, 7.5, 10.0, 12.5, 15.0, 17.5, 20.0$. Le posizioni sono normalizzate in lunghezza di Debye λ_D , i tempi nell'inverso della frequenza di plasma ω_p e le velocità nella velocità termica v_{th}

Capitolo 4

Conclusioni

Successivamente all'introduzione delle basi concettuali di un semplice algoritmo unidimensionale ed elettrostatico (ES1) per lo studio di un plasma freddo e non magnetizzato, sono state illustrate alcune possibili implementazioni delle singole funzioni usate. Tra questi, è stato introdotto il metodo Particle-In-Cell per l'interpolazione della densità di carica dalle particelle distribuite sulla griglia spaziale. Dopo aver introdotto lo stato di plasma e i suoi parametri fondamentali (lunghezza di Debye, parametro di plasma, frequenza di plasma), ci si è focalizzati sulle oscillazioni di plasma, definendone alcune proprietà necessarie per la trattazione successiva. La trattazione teorica si è conclusa con lo studio delle instabilità di plasma, con particolare riferimento all'instabilità di due stream. Con riferimento a quest'ultima, si è determinata l'equazione di dispersione per oscillazioni di plasma conseguenti all'introduzione di due stream di velocità opposta in un background di ioni stazionari. Si è presentato un metodo di soluzione approssimata di quest'ultima e le sue soluzioni esplicite, studiandone le caratteristiche salienti. L'elaborato introduce infine un esempio di programma in linguaggio C per la simulazione della suddetta instabilità, aggiungendovi una breve spiegazione per ogni funzione usata e l'output prodotto.

Nella fisica dei plasmi, simulazioni PIC sono tuttora ampiamente usate nello studio di interazioni laser-plasma, in numerosi modelli di studio della ionosfera (assorbimenti, propagazioni e riflessioni di microonde) o nello studio di microinstabilità nei plasmi confinati da macchine di tipo Tokamak.

Appendice A

Soluzione numerica dell'equazione di Poisson con il metodo della matrice tridiagonale.

Supponiamo di dover risolvere l'equazione di Poisson unidimensionale:

$$\frac{d^2u(x)}{dx^2} = v(x), \quad (\text{A.1})$$

per $x_l \leq x \leq x_h$ con condizioni al contorno dette di Dirichlet:

$$u(x_l) = u_l, \quad u(x_h) = u_h. \quad (\text{A.2})$$

Come primo passo, dividiamo il dominio $x_l \leq x \leq x_h$ in segmenti uguali i cui vertici sono localizzati nei seguenti punti di un'ipotetica griglia:

$$x_i = x_l + \frac{i(x_h - x_l)}{N + 1}, \quad (\text{A.3})$$

per $i = 1, \dots, N$, con gli estremi x_l e x_h , corrispondenti a $i = 0$ e $i = N + 1$ rispettivamente. Successivamente, discretizziamo il termine differenziale d^2u/dx^2 sui punti della griglia in questa maniera:

$$\frac{d^2u(x)}{dx^2} = \frac{u_{i-1} - 2u_i + u_{i+1}}{(\delta x)^2} + O(\delta x)^2, \quad (\text{A.4})$$

in cui $\delta x = (x_h - x_l)/(N + 1)$ e $u_i \equiv u(x_i)$. Questo tipo di discretizzazione è detto schema *discond-order, central difference*.

La nostra versione discretizzata dell'equazione di Poisson avrà la forma:

$$u_{i-1} - 2u_i + u_{i+1} = v_i(\delta x)^2, \quad (\text{A.5})$$

per $i = 1, \dots, N$ con $v_i \equiv v(x_i)$. Per di più $u_0 = u_l$ e $u_{N+1} = u_h$.

È d'aiuto l'utilizzo della forma matriciale di queste equazioni discrete. Sia $\mathbf{u} = (u_1, u_2, \dots, u_N)$ il vettore dei valori di u e

$$\mathbf{w} = [v_1(\delta x)^2 - u_l, v_2(\delta x)^2, v_3(\delta x)^2, \dots, v_{N-1}(\delta x)^2, v_N(\delta x)^2 - u_h], \quad (\text{A.6})$$

il vettore dei termini sorgente. Le equazioni discrete (A.5) possono quindi essere scritte come:

$$\mathbf{M}\mathbf{u} = \mathbf{w}, \quad (\text{A.7})$$

con la matrice \mathbf{M} che prende la forma:

$$M = \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 1 & 0 & 0 & 1 & -2 \end{bmatrix} \quad (\text{A.8})$$

Per $N = 6$. La matrice \mathbf{M} è chiamata matrice tridiagonale. La soluzione formale all'equazione (A.7) sarà:

$$\mathbf{u} = \mathbf{M}^{-1}\mathbf{w}, \quad (\text{A.9})$$

con \mathbf{M}^{-1} la matrice inversa di \mathbf{M} .

Esponiamo ora un algoritmo per l'inversione della matrice tridiagonale chiamato *gauss-Jordan elimination*. Siano \mathbf{a} , \mathbf{b} e \mathbf{c} i vettori rispettivamente della diagonale sinistra, centro e destra della matrice \mathbf{M} . Notiamo che a_1 e c_N sono indefiniti, li poniamo quindi a zero. L'equazione (A.7) potrà scriversi come:

$$a_i u_{i-1} + b_i u_i + c_i u_{i+1} = w_i, \quad (\text{A.10})$$

per $i = 1, \dots, N$. Cerchiamo una soluzione nella forma:

$$u_{i+1} = x_i u_i + y_i. \quad (\text{A.11})$$

Sostituendo in (A.10) abbiamo:

$$a_i u_{i-1} + b_i u_i + c_i (x_i u_i + y_i) = w_i, \quad (\text{A.12})$$

che può essere riscritta come:

$$u_i = \frac{a_i u_{i-1}}{b_i + c_i x_i} + \frac{w_i - c_i y_i}{b_i + c_i x_i}, \quad (\text{A.13})$$

Se scriviamo l'equazione (A.11) come:

$$u_i = x_{i-1}u_{i-1} + y_{i-1}, \quad (\text{A.14})$$

e comparando con la (A.13)

$$x_{i-1} = -\frac{a_i}{b_i + c_i x_i}, \quad (\text{A.15})$$

$$y_{i-1} = \frac{w_i - c_i y_i}{b_i + c_i x_i}. \quad (\text{A.16})$$

Risolveremo l'equazione per la matrice tridiagonale in due fasi. Nella prima fase, prendiamo i valori della diagonale principale da $i = N$ a 1 usando le (A.15) e (A.16). Quindi:

$$x_{N-1} = -\frac{a_N}{b_N}, \quad y_{N-1} = \frac{w_N}{b_N}, \quad (\text{A.17})$$

poichè $c_N = 0$. Inoltre:

$$\begin{aligned} x_i &= \frac{a_{i+1}}{b_{i+1} + c_{i+1} x_{i+1}}, \\ y_i &= \frac{w_{i+1} - c_{i+1} y_{i+1}}{b_{i+1} + c_{i+1} x_{i+1}}, \end{aligned} \quad (\text{A.18})$$

per $i = N - 2, N - 1, \dots, 1$. E infine:

$$x_0 = 0, \quad y_0 = \frac{w_i - c_i y_i}{b_i + c_i x_i}, \quad (\text{A.19})$$

poiché $a_1 = 0$. Abbiamo quindi definito in questa fase tutti i x_i e y_i . Nella seconda fase prendiamo tutti i valori della matrice diagonale da $i = 0$ a $N - 1$ usando l'equazione (A.11):

$$u_1 = y_0, \quad (\text{A.20})$$

poiché $x_0 = 0$ e

$$u_{i+1} = x_i u_i + y_i, \quad (\text{A.21})$$

per $i = 1, 2, \dots, N - 1$.

Abbiamo quindi invertito la matrice tridiagonale. Chiaramente, usiamo quest'algoritmo di inversione per invertire (A.7) con i termini sorgente specificati da (A.6), e la matrice diagonale \mathbf{M} con $a_i = 1$ per $i = 2, 3, \dots, N$, $b_i = -2$ per $i = 1, 2, \dots, N$ e $c_i = 1$ per $i = 1, 2, \dots, N$

Appendice B

Codice completo per la simulazione dell'instabilità di due stream

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>

const int J = 1000;
const int N = 80000;
const double L = 100;
const double vb = 3;
const double dt = 0.1;
const double tmax = 21;
const double alpha_l = 1;
const double beta_l = 0;
const double gamma_l = 0;
const double alpha_h = 1;
const double beta_h = 0;
const double gamma_h = 0;
const double dx = 0.1;

void boundary( double r[]){
int i;
for(i = 0; i < N; i++) {
```

```
if(r[i]< 0.) r[i] += L;
if(r[i] > L) r[i] -=L;
}
}

void electric1(double r0[], double E[], double E1[])
{
int i;
for (i = 0; i < N; i++){

int j = (int)(r0[i]/dx);
double y = r0[i]/dx - (double)j;

if(j+1 == J) E1[i] = E[j]*(1. - y) + E[0]*y;
else E1[i] = E[j]*(1.-y) + E[j+1]*y;
}
}

void density(double r[], double n[])
{
int azz,i;
for(azz=0; azz< J; azz++) n[azz]=0;
for (i = 0; i < N; i++){
int j = (int)(r[i]/dx);
double y = r[i]/dx - (double)j;
n[j] += (1.- y)/dx; //da
if(j+1 == J) n[0] += y/dx;
else n[j+1] += y/dx;

}
}

void density1(double ne[], double rho[])
{
double n0 = (double) (N) / L;
int j;
for (j = 0; j < J; j++) rho[j] = ne[j]/n0 - 1.;
}

void tridiagonal (double a1[], double b1[], double c1[],
double w1[], double u[])
{
double x1[J-2], y1[J-2];
```

```
int i;

x1[J-3]= - a1[J-2]/ b1[J-2];
y1[J-3]= w1[J-2] / b1[J-2];
for (i = J-4; i > 0; i--)
{
x1[i] = - a1[i+1] / (b1[i+1] + c1[i+1] * x1[i+1]);
y1[i] = (w1[i+1] - c1[i+1] * y1[i+1])/ (b1[i+1] + c1[i+1] * x1[i+1]);
}
x1[0] = 0.;
y1[0] = (w1[1] - c1[1] * y1[1] )/ (b1[1] + c1[1] * x1[1]);

u[1] = y1[0];
for (i = 1; i < J-2; i++)
u[i+1] = x1[i] * u[i] + y1[i];
}

void poisson1D ( double u[], double rho[])
{
double a1[J], b1[J], c1[J], w1[J];
int i;

for (i = 2; i <= J-2; i++) a1[i] = 1.;
for (i = 1; i <= J-2; i++) b1[i] = -2.;
b1[1] -= beta_l / (alpha_l * dx - beta_l);
b1[J-2] += beta_h / (alpha_h * dx + beta_h);
for ( i = 1; i <= J-3; i++) c1[i] = 1.;

for (i = 1; i <= J-2; i++)
w1[i] = rho[i] * dx * dx;
w1[1] -= gamma_l * dx / (alpha_l * dx - beta_l);
w1[J-2] -= gamma_h * dx / (alpha_h * dx + beta_h);

tridiagonal (a1, b1, c1, w1, u);

u[0] = (gamma_l * dx - beta_l * u[1])
/ (alpha_l * dx - beta_l);
u[J-1] = (gamma_h * dx + beta_h * u[J-2])
/ (alpha_h * dx + beta_h);
}
```

```
void electric (double u[], double E[])
{
  int j;
  for (j = 1; j < J-1; j++)
    E[j] = (u[j-1] - u[j+1]) / (dx*2.);
  E[0] = (u[J-1] - u[1]) / (dx*2.);
  E[J-1] = (u[J-2] - u[0]) / (dx*2.);
}

void rk4_fixed (double y[], double h, double dydt[])
{
  int j;
  double k1[2], k2[2], k3[2], k4[2], f[2];

  for (j = 0; j < 2; j++)
  {
    k1[j] = h * dydt[j];
    f[j] = y[j] + k1[j] / 2.;
  }

  for (j = 0; j < 2; j++)
  {
    k2[j] = h * dydt[j];
    f[j] = y[j] + k2[j] / 2.;
  }

  for (j = 0; j < 2; j++)
  {
    k3[j] = h * dydt[j];
    f[j] = y[j] + k3[j];
  }

  for (j = 0; j < 2; j++)
  {
    k4[j] = h * dydt[j];
  }

  for (j = 0; j < 2; j++)
  {
    y[j] += k1[j] / 6. + k2[j] / 3. + k3[j] / 3. + k4[j] / 6.;
  }
}
```

```
}
return;
}

double distribution (double vb)
{

static int flag = 0;
if (flag == 0)
{
int seed = time (NULL);
srand (seed);
flag = 1;
}

double fmax = 0.5 *(1. + exp (-2. * vb * vb));
double vmin = - 5. * vb; // vmin = - 5 vb
double vmax = + 5. * vb; //vmax = + 5 vb
double v = vmin + (vmax - vmin) *
(double)rand ()/ (double)RAND_MAX;

double f = 0.5 *(exp -(v - vb) * (v - vb) / 2.)
+ exp -(v + vb) * (v + vb) / 2.);
double x = fmax *
(double)rand () / (double)RAND_MAX;
if (x > f) return distribution (vb);
else return v;
}

int main(int argc, char *argv[]) {

double dydt[2];
double sol[2];
double E1[N],r[N], v[N];
double n[J], rho[J], phi[J], E[J];
int i;

int seed = time (NULL);
srand (seed);
for(i = 0;i < N; i++){
```

```
r[i] = L*(double) (rand()/(double)(RAND_MAX));
v[i] = distribution(vb);
}

char filename[64];
FILE *fdoc;
fdoc=fopen("output.dat", "w"); //apri il file di scrittura

double t = 0;
for(t= 0; t <= tmax; t= t+ dt){

boundary(r);
density(r,n);
density1(n, rho);
poisson1D(phi, rho);
electric(phi, E);
electric1(r, E, E1);

for(i = 0; i < N; i++){

dydt[0] = v[i];
dydt[1] = -E1[i];
sol[0] = r[i];
sol[1] = v[i];

rk4_fixed(sol, dt, dydt);

r[i] = sol[0];
v[i] = sol[1];
if(r[i]< 0.) r[i] += L;
if(r[i] > L) r[i] -=L;

printf("\n%f\t\tparticella %d:
%f          %f", t,i,r[i],v[i]);
}

\\SCRITTURA SU FILE\\
fdoc=fopen("output.dat", "a");
fprintf(fdoc, "set xrange[0:100]\n");
fprintf(fdoc, "set yrange[-10:10]\n");
fprintf(fdoc, "plot 'data%.0f.dat 'with dots
black\n set title \"t = %.1f\" \n set xlabel \"x\""
```



```
\n set ylabel \"v\"\nset term png\nset  
output 'data_%f.png'\n", t*10, t,t);  
  
fclose(fdoc);  
  
sprintf(filename, "data%.0f.dat", 10*t);  
fdoc=fopen(filename, "w");  
for(i = 0; i < N; i++) fprintf(fdoc,  
"%f      %f\n",r[i],v[i]);  
fclose(fdoc);  
  
}  
return 0;  
}
```

Bibliografia

- [1] C.K. Birdsall, A.B Langdon *Plasma Physics via computer simulation* (CBC Press, 01 Ott 2004)
- [2] E.M. Lifshitz, L.P.Pitaevskii, *Landau and Lifshitz, Course of theoretical Physics, volume 10, Physical Kinetics*
- [3] Dwight R. Nicholson *introduction to Plasma theory* (Krieger, 1992)
- [4] C.K. Birdsall, *Particle-in-Cell Charged-Particle Simulation plus Monte Carlo collision with neutral atoms, PIC-MCC*
- [5] D.Anderson, R.Fedele, M. Lisak, *a tutorial presentation of the two stream instability and Landau damping*
- [6] Sven Gunnar Bilén, *Pulse propagation along conductor in low-density, cold plasmas and applied to electrodynamic tethers in the ionosphere*
- [7] Chen, F.F, *Introduction to plasma physics and controlled fusion, 1984*
- [8] Lieberman, M.A, Lichtenberg A.J., *Principles of plasma discharges and material processing, 2005*
- [9] Dawson, J.M. *Particle simulation of plasmas". Reviews of Modern Physics, 1983*

Ringraziamenti

“Dispongo di una meravigliosa lista di ringraziamenti che non può essere contenuta nel margine troppo stretto della pagina.”

[Semicit.]