Relazione Fisica Computazionale
DINAMICA DI PACCHETTI D'ONDA NEL CASO QUANTISTICO

Rebecca Ghidoni
Alessandro Piras

25 giugno 2020

Sommario

Lo scopo del progetto è studiare la dinamica di un pacchetto d'onda, inizialmente di forma gaussiana, che si muove in presenza di 2 barriere di potenziale. Il sistema verrà fatto evolvere utilizzando un algoritmo split-step, sfruttando la formula di Trotter-Suzuki.

Introduzione

Consideriamo una griglia spazio-temporale di $N \times (M+1)$ punti indipendenti,

- del dominio spaziale limitato al segmento [0,L) con condizioni periodiche al bordo
- del dominio temporale definito nell'intervallo [0,T]

ottenuta con una discretizzazione spaziale $\delta x = \frac{L}{N}$ e temporale $\delta t = \frac{T}{M}$

Consideriamo il sistema quantistico 1-dimensionale definito dall'Hamiltoniana:

$$\hat{\mathcal{H}} = \hat{\mathcal{K}} + \hat{\mathcal{V}} \qquad \hat{\mathcal{K}} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} \qquad \hat{\mathcal{V}} = V(x) \tag{1}$$

dove il potenziale V(x) è costituito da 2 barriere centrate in $x_1 = \frac{L}{3}$ e $x_2 = \frac{L}{2}$, rispettivamente di altezza E_1 e E_2 e di ampiezza $l_1 << L$ e $l_2 << L$, $\hat{\mathcal{K}}$ è l'operatore legato all'energia cinetica e $\hat{\mathcal{V}}$ è l'operatore potenziale dipendente solo dalle coordinate.

Per semplicità utilizzeremo la unità di misura atomiche:
$m_e = 1$ $\hbar = 1$

Al tempo t=0 la funzione d'onda, nel nostro caso, ha la forma di una gaussiana di ampiezza σ centrata in x_0 e con velocità di gruppo $\frac{\hbar k_0}{m}$

$$\psi(x,0) = \frac{1}{\sqrt[4]{2\pi\sigma^2}} \exp\left\{ -\frac{(x-x_0)^2}{4\sigma^4} \right\} e^{ik_0 x} \tag{2}$$

L'operatore di evoluzione temporale è dato da $\hat{\mathcal{U}}(x,t) = e^{-i\frac{\hat{\mathcal{H}}}{\hbar}t}$.

$$\psi(x,t) = \hat{\mathcal{U}}(x,t)\psi(x,0) \tag{3}$$

Algoritmo utilizzato

Utilizziamo la formula di Trotter-Suzuki, che approssima l'operatore di evoluzione temporale nel modo seguente:

$$\hat{\mathcal{U}}(x,\delta t) = \hat{\mathcal{G}}_1(\frac{\delta t}{2}) \cdot \hat{\mathcal{G}}_2(\delta t) \cdot \hat{\mathcal{G}}_1(\frac{\delta t}{2}) + \hat{\mathcal{O}}(x,\delta t^3) \tag{4}$$

$$\hat{\mathcal{G}}_1(\frac{\delta t}{2}) = \exp\left\{ -i\frac{\hbar}{2}V(x)\right\} \qquad \hat{\mathcal{G}}_2(\delta t) = \exp\left\{ i\frac{\hbar}{2}\frac{\partial^2}{\partial x^2}\right\} \tag{5}$$

dove δt è il passo temporale di cui si vuole fare evolvere il sistema ad ogni passo, e dove si sono poste unitarie la costante $\hbar$ e la massa dell'elettrone (usando il sistema di unità atomiche).

L'operatore $\hat{\mathcal{G}}_1(\frac{\delta t}{2})$ agisce così sulle coordinate spaziali, per metà del passo temporale δt; al contrario $\hat{\mathcal{G}}_2(\delta t)$ agisce sull'intero passo δt e contiene solo le derivate parziali, perciò risulterà diagonale nello spazio dei momenti.

L'algoritmo split-step spettrale applicato all'equazione di Schroedinger dipendente dal tempo prevede quindi cinque step, che andranno ripetuti ad ogni iterazione:

- Applicazione dell'operatore $\hat{\mathcal{G}}_1(\frac{\delta t}{2})$, diagonale nello spazio delle coordinate, direttamente alla funzione $\psi(x,t)$:

$$\psi'(x) = \hat{\mathcal{G}}_1(\frac{\delta t}{2})\psi(x,t) = e^{-i\frac{\hbar}{2}V(x)}\psi(x,t) \tag{6}$$

- Passaggio allo spazio dei momenti operando la trasformata della funzione d'onda:

$$\overline{\psi'}(k) = \int_{-\infty}^{\infty} e^{-ikx}\psi'(x)dx \tag{7}$$

  all'interno del programma questo sarà fatto attraverso una Fast Fourier Transform.

- Poiché l'operatore $\hat{\mathcal{G}}_2(\delta t)$ è diagonale nello spazio dei momenti, è adesso possibile applicarlo facilmente alla trasformata di ψ':

$$\overline{\psi''}(k) = \hat{\mathcal{G}}_2(\delta t)\overline{\psi'}(k) = e^{-i\frac{\delta t}{2}k^2}\overline{\psi'}(k) \tag{8}$$

- Dopodiché ritorno allo spazio delle coordinate calcolando l'antitrasformata di $\overline{\psi''}(k)$, sempre attraverso una FFT:

$$\psi''(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{ikx}\overline{\psi''}(k)dk \tag{9}$$

- Infine applicazione dell'operatore $\hat{\mathcal{G}}_1(\frac{\delta t}{2})$, ottenendo la funzione al tempo $t+\delta t$:

$$\psi(x,t+\delta t) = \hat{\mathcal{G}}_1(\frac{\delta t}{2})\psi''(x) = e^{-i\frac{\delta t}{2}V(x)}\psi''(x) \tag{10}$$

Poiché gli operatori $\hat{\mathcal{G}}_1(\frac{\delta t}{2})$ e $\hat{\mathcal{G}}_2(\delta t)$ sono unitari, e le FFT soddisfano il teorema di Parseval, la norma di ψ verrà conservata durante ogni passaggio. Infine, come è indicato nella formula (4), l'errore commesso nell'applicare la formula di Trotter-Suzuki sarà per ogni passaggio di ordine δt^3.

Codice Python

La parte iniziale del programma permette di leggere da un file di input di tipo testo i valori dei parametri iniziali da utilizzare all'interno del programma. Il file di input deve presentare le variabili in un ordine prestabilito:

- **L**=Lunghezza dello spazio in esame
- **T**=Intervallo temporale
- **N**=Numero di punti nello spazio
- **M**=Numero di punti nel tempo
- **k0**=Velocità di gruppo del pacchetto d'onda
- **E1**=Altezza della prima barriera di potenziale
- **E2**=Altezza della seconda barriera di potenziale
- **l1**=Spessore della prima barriera di potenziale
- **l2**=Spessore della seconda barriera di potenziale
- **sigma**=Varianza del pacchetto d'onda
- **freq**=Frequenza (mi dice ogni quanto salvare la traiettoria)

All'interno del programmo sono eseguiti dei test di controllo che verificano che i dati siano compatibili con i valori attesi. Ad esempio controlla che sia l'intervallo spaziale sia quello temporale vengano discretizzati in un numero sufficiente di punti; almeno 500. Oltre ad assegnare i valori indicati dal file di testo alle variabili all'inizio del programma vengono anche create le variabili **dx, dt, x0, x, t** che rappresentano rispettivamente la discretizzazione spaziale e temporale, la posizione iniziale del pacchetto d'onda, l'estremo spaziale iniziale e il tempo iniziale

Successivamente il programma crea una matrice di dimensioni $N \times (M+1)$ nella quale verrà salvata la traiettoria del pacchetto d'onda per ogni istante di tempo.

Si passa poi a creare la forma iniziale del pacchetto d'onda

Si utilizza poi un modulo per creare la forma del potenziale, nel nostro caso una doppia barriera. in questo punto viene eseguito un altro test di controllo per verificare che le barriere di potenziali siano abbastanza sottili da non sovrapporsi. In caso contrario viene chiamata l'uscita forzata dal programma.

In seguito il programma crea le espressioni per i propagatori G_1 e G_2

Infine si trova un ciclo for che fa evolvere temporalmente il sistema da $t = 0$ a $t = T$. Contemporaneamente controlla che la norma del pacchetto d'onda rimanga unitaria entro un ragionevole intervallo d'errore

All'interno del codice vengono anche richiamate dalla libreria numpy le funzioni fftfreq, fft e ifft che permettono di generare i punti dello spazio dei momenti, la trasformata di Fourier di

una funzione e l'antitrasformata di Fourier rispettivamente.

Viene poi eseguito un altro controllo che misura l'errore massimo percentile della norma; questo serve a verificare la diversa precisione del programma al variare dei parametri iniziali.

La traiettoria per tutti gli istanti di tempo viene invece salvata su un file di output di tipo testo dal nome 'traiettoria.txt' nel quale in ogni riga viene indicato il tempo corrispondente a ogni presa dati e il valore del pacchetto d'onda a ogni intervallo spaziale.

Successivamente viene graficato il pacchetto d'onda a intervalli di tempo regolari affinché l'utente possa verificare visivamente il funzionamento del codice.

Esperimenti numerici

Innanzitutto, come già ricordato nell'introduzione, abbiamo posto i valori della costante di Plank ridotta $\hbar$ e della massa dell'elettrone m_e uguali a 1; in questo modo il computer effettuerà i calcoli rimanendo sull'ordine delle unità, evitando quindi di dover lavorare con potenze troppo alte o troppo basse che rischiano di creare errori indesiderati.

Da ciò abbiamo definito l'unità di misura per le lunghezze pari a $l = \frac{\hbar^2(4\pi\epsilon_0)^2}{m_e e^2}$ e l'unità di tempo pari a $t = \frac{\hbar^3(4\pi\epsilon_0)^2}{m_e e^4}$.

Sempre per ragioni di efficienza nei calcoli poniamo il numero di intervalli spaziali N pari a una potenza di 2, in questo modo sarà più facile effettuare il cambio nello spazio dei momenti.

Per scegliere i valori dei parametri iniziali abbiamo effettuato diverse prove. Per quanto riguarda il valore di sigma abbiamo testato 3 possibili valori togliendo dal programma l'effetto del potenziale per vedere meglio la dispersione del pacchetto d'onda dovuta solo alla varianza. Abbiamo testato i valori $sigma$=2, 8 o 16. Nel primo caso ci siamo subito resi conto che il pacchetto d'onda si disperdeva troppo velocemente e all'istante finale presentava un'altezza massima pari a circa un quarto di quella iniziale. Nel caso di $sigma = 16$ invece il pacchetto d'onda risultava essere fin dall'inizio troppo basso e largo e ciò comporta il non poter trascurare il rumore, soprattutto nei tempi finali.

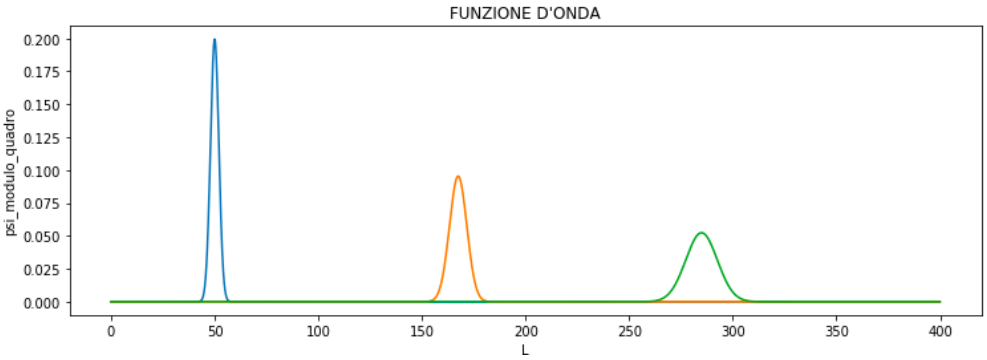

Figura 0.1: Evoluzione del pacchetto d'onda per sigma=2.

.

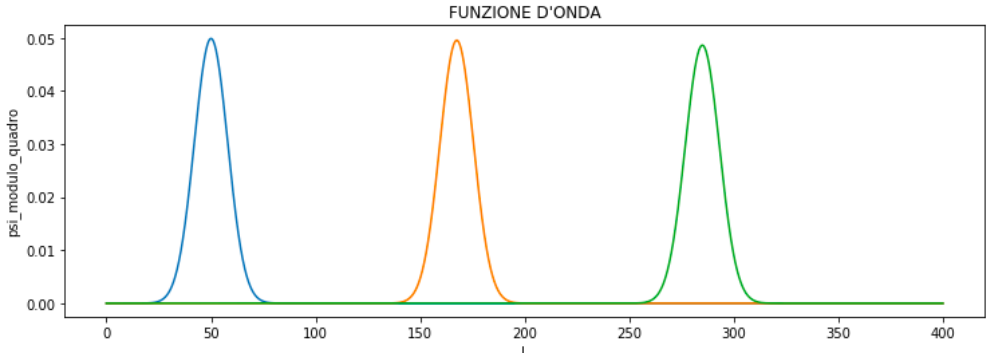

Figura 0.2: Evoluzione del pacchetto d'onda per sigma=8.

.

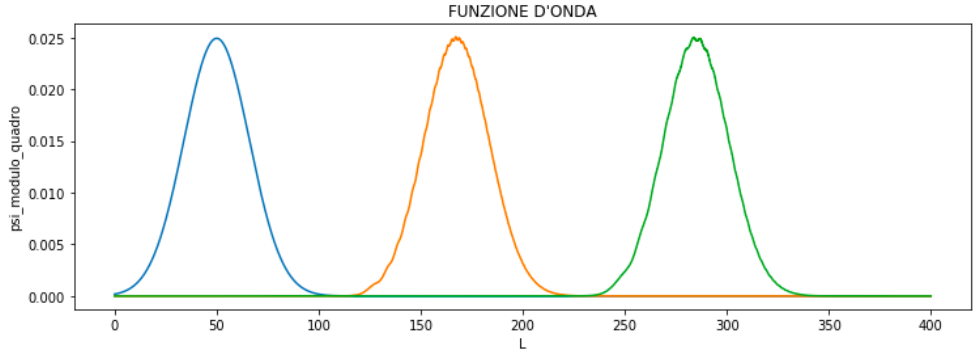

Figura 0.3: Evoluzione del pacchetto d'onda per sigma=16.

.

Un altro test effettuato riguarda l'errore massimo percentile della norma della funzione d'onda sempre al variare di sigma:

- Per sigma=2 l'errore massimo percentile è circa 0.2703 % e si verifica al tempo 28.5278
- Per sigma=8 l'errore massimo percentile è circa 0.1417 % e si verifica al tempo 28.1470
- Per sigma=16 l'errore massimo percentile è circa 0.0675 % e si verifica al tempo 17.9663

Per eseguire questo test abbiamo mantenuto tutte gli altri valori numerici pari a quelli del test di riferimento, visibili in appendice.

Infine abbiamo fatto variare il rapporto $\frac{dx}{dt}$ per poter confrontare l'accuratezza dell'integrazione in diversi casi:

- $\frac{dx}{dt} = 3.2552$ ottenuto con M=1000
- $\frac{dx}{dt} = 9.7657$ ottenuto con M=3000
- $\frac{dx}{dt} = 16.2761$ ottenuto con M=5000

In tutti i casi i valori per le altre variabili sono pari a quelli iniziali di prova.

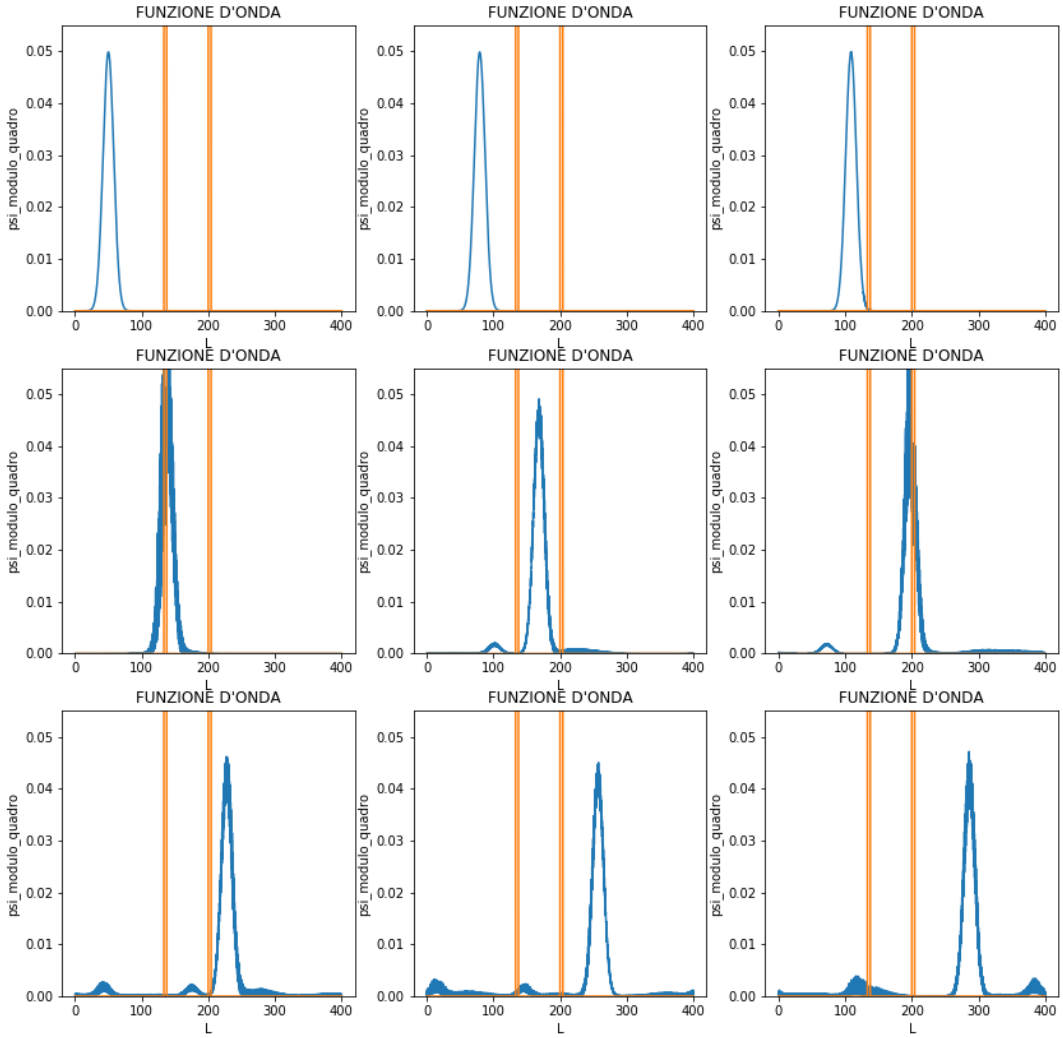

Figura 0.4: Evoluzione del pacchetto d'onda per $\frac{dx}{dt} = 3.2552$.

.

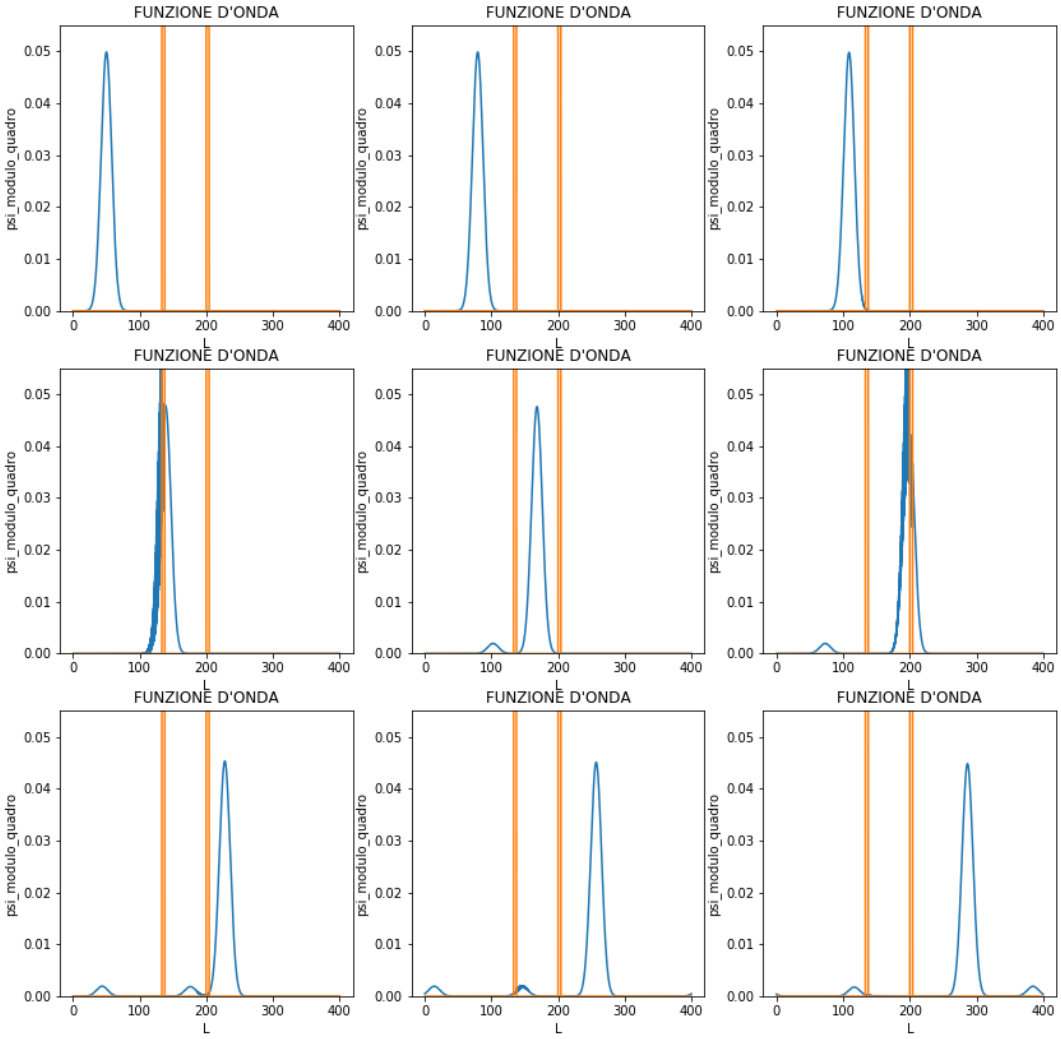

Figura 0.5: Evoluzione del pacchetto d'onda per $\frac{dx}{dt} = 9.7656$.

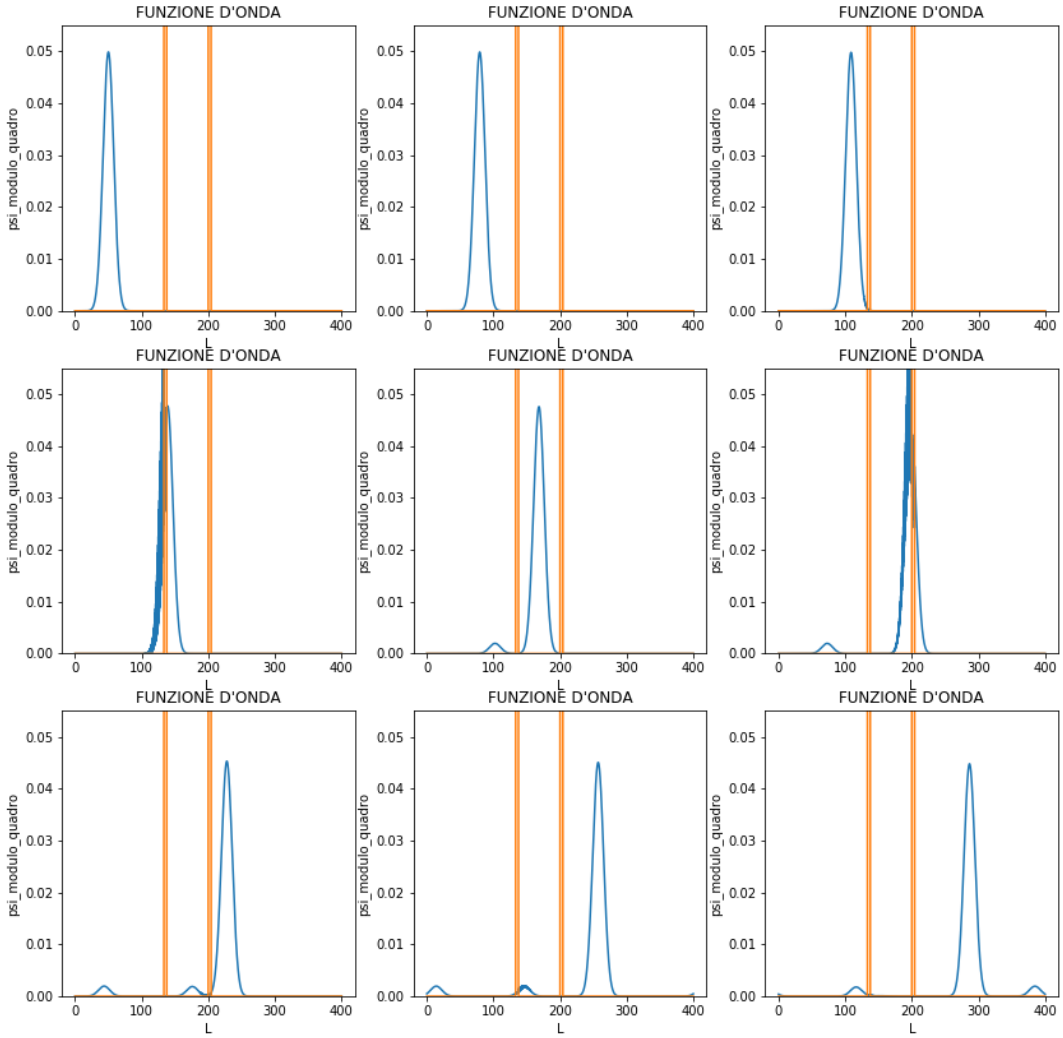

Figura 0.6: Evoluzione del pacchetto d'onda per $\frac{dx}{dt} = 16.2767$.

Analizzando i grafici si nota immediatamente come al crescere del rapporto tra $\frac{dx}{dt}$ cresca anche l'accuratezza dell'integrazione. Bisogna però tenere anche presente che aumentare il valore di M porta ad un aumento del tempo di lavoro della CPU, infatti:

- Per M= 1000 il tempo di esecuzione dell'algoritmo di Trotter-Susuki è pari a circa 9 secondi
- Per M= 3000 il tempo di esecuzione dell'algoritmo di Trotter-Susuki è pari a circa 27 secondi
- Per M= 5000 il tempo di esecuzione dell'algoritmo di Trotter-Susuki è pari a circa 44 secondi

In tutti i casi l'oscillazione del tempo di lavoro della CPU si aggirava attorno a pochi secondi.

Per l'ultimo test abbiamo deciso di osservare i coefficienti di trasmissione e di riflessione all'aumentare dell'energia del potenziale. Per semplicità abbiamo utilizzato, nel primo caso, una sola barriera di energia pari a 15 nelle apposite unità di misura, confrontabile con l'energia di gruppo del pacchetto d'onda; nel secondo caso invece abbiamo utilizzato una barriera di energia pari a 150 nelle apposite unità di misura.
Per entrambi i casi la barriera si trovava a metà del dominio spaziale

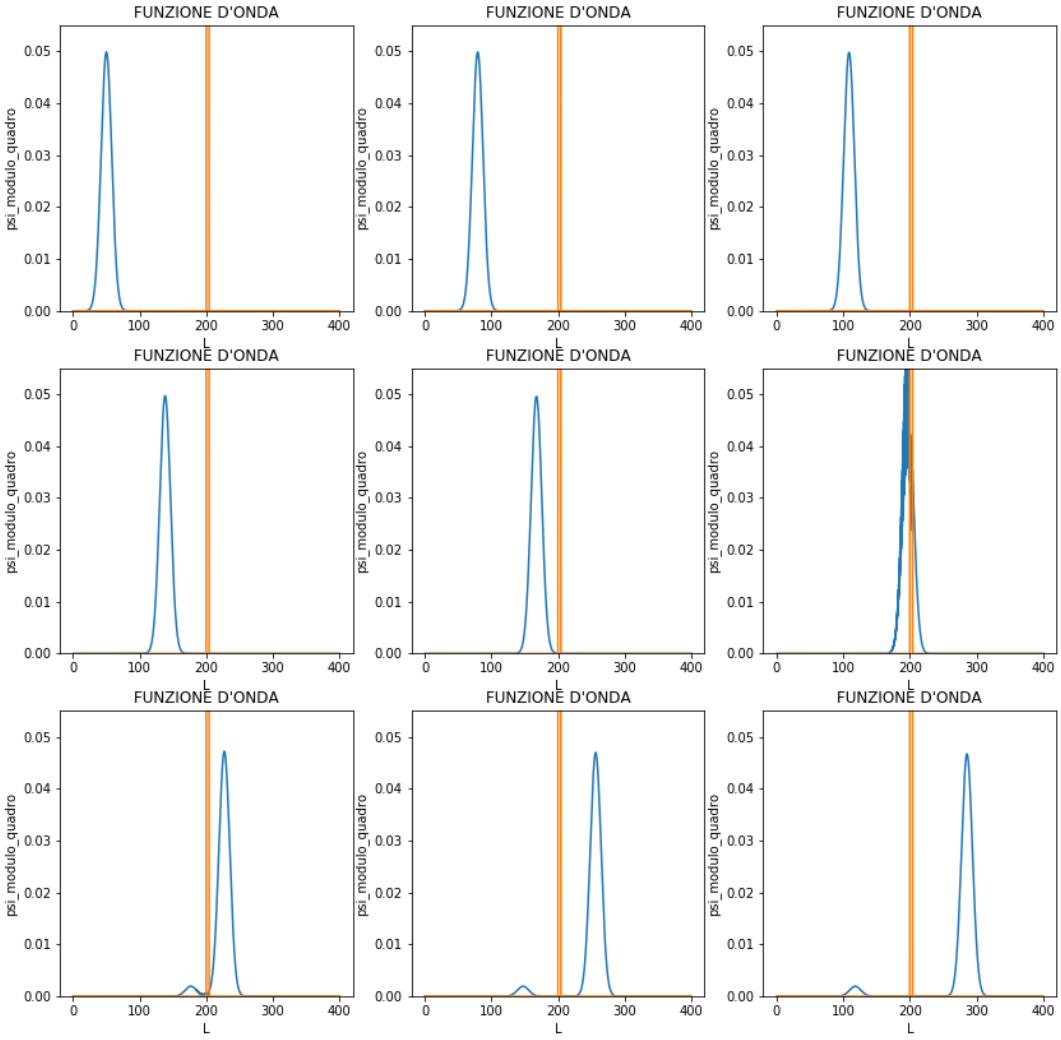

Figura 0.7: Evoluzione del pacchetto d'onda contro barriera di altezza 15

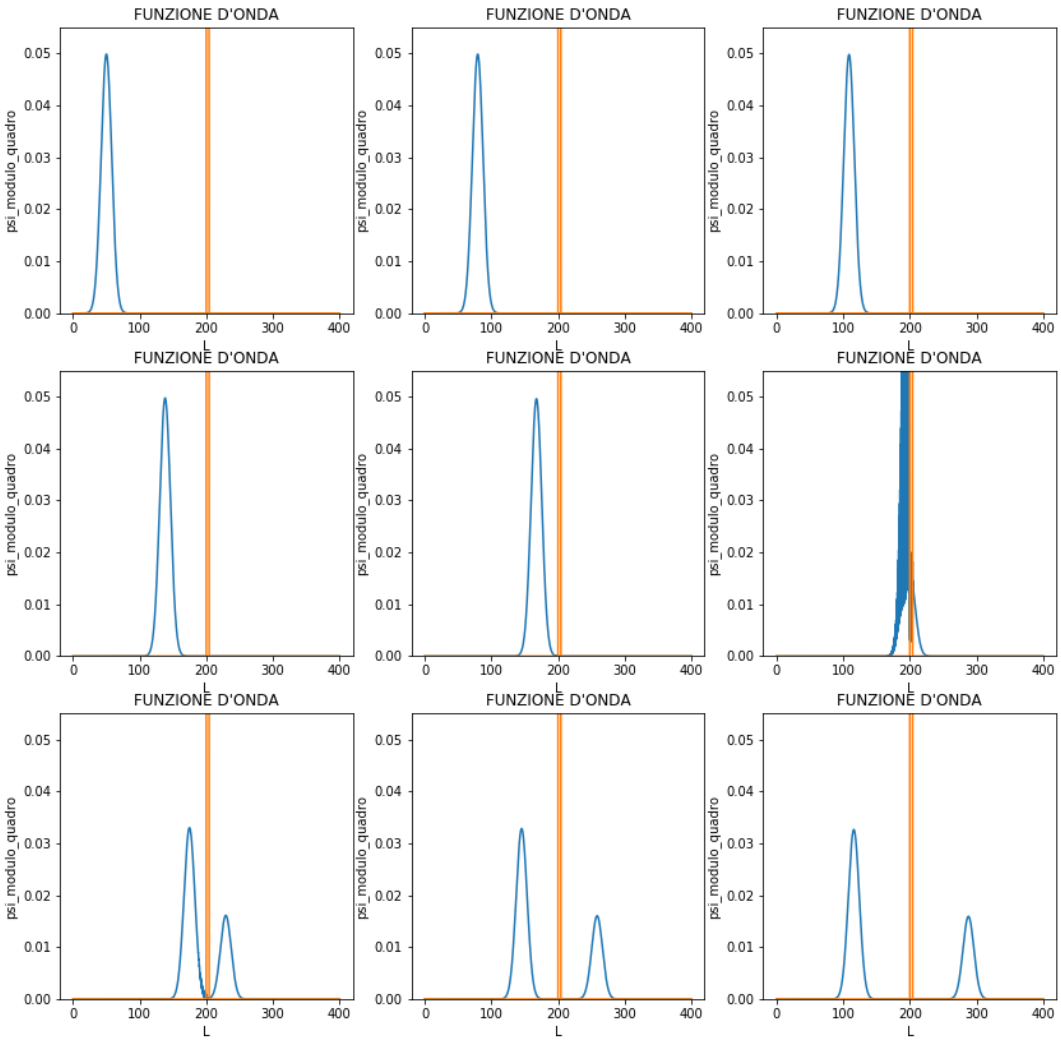

Figura 0.8: Evoluzione del pacchetto d'onda contro barriera di altezza 150

Avendo aumentato l'altezza delle barriere di potenziale di circa un ordine di grandezza possiamo notare come il coefficiente di riflessione, ovvero la percentuale di onda riflessa dalla barriera, aumenti e di conseguenza il coefficiente di trasmissione, ovvero la percentuale di onda che riesce a superare la barriera cali.

Risultati dell'esperienza

Studiando il rapporto $\frac{dx}{dt}$ si nota facilmente che all'aumentare di questo il rumore risulti via via sempre più trascurabile. Quando questo rapporto supera il valore 9 il rumore è molto ridotto, consigliamo comunque di mantenerlo sempre attorno alla decina.

Per quanto riguarda la conservazione della norma, si nota che l'algoritmo è molto buono lasciando l'errore sempre al di sotto dello 0.5%, dovuto forse agli errori di macchina che lo rendono via via più significativo a ogni iterata.

Dallo studio dal caso con barriera di potenziale pari a 150 nelle opportune unità di misura, valore ben maggiore dell'energia totale del pacchetto d'onda, abbiamo potuto constatare come la trasmissione dell'onda oltre la barriera sia effettivamente dovuta all'effetto tunnel.

Consigli operativi

Dopo aver effettuato diverse prove siamo arrivati alla conclusione che, benchè tutti i valori numerici siano teoricamente accettabili, è buona norma porre delle condizioni nella loro scelta

- Il valore di N deve essere una potenza di 2; ciò permette di ridurre al minimo l'errore quando si effettuano la trasformata e l'antitrasformata di Fourier
- La velocità di propagazione del pacchetto d'onda deve essere abbastanza piccola da evitare che, avendo imposto le condizioni di periodicità al bordo, l'onda riflessa dalla prima barriera e l'onda trasmessa da entrambe le barriere si scontrino
- I valori sia di M che di N dovrebbero essere mantenuti maggiori di 500 per ottenere una suddivisione spaziale e temporale abbastanza fitta
- Per come vogliamo che sia costruito il potenziale è necessario utilizzare valori di **l1** e **l2** molto minori di **L**

Appendice

Come valori di prova iniziale abbiamo utilizzato i seguenti valori:

```
400
30
4096
3000
-8
15
15
4
4
8
200
```

Listing 1: File di testo contenente i valori di prova iniziali utilizzati per eseguire il codice