

Università degli Studi di Milano-Bicocca

Dipartimento di Fisica

Corso di Laurea Triennale



Solunzione numerica dell'equazione di Schrödinger

**Relatore:**

**Carlo Oleari**

**Candidato:**

**Luca Falzoni**

**ANNO ACCADEMICO 2021/2022**

# Indice

<b>Introduzione</b>	<b>1</b>
0.1 L'equazione di Schrödinger . . . . .	2
<b>1 Metodo</b>	<b>4</b>
1.1 Espansione di Lie-Trotter . . . . .	4
1.2 Soluzione numerica . . . . .	6
1.2.1 Trasformata discreta di Fourier . . . . .	7
1.3 Condizioni e vincoli sul sistema . . . . .	9
<b>2 Validazione del codice</b>	<b>11</b>
2.1 Particella libera . . . . .	11
2.1.1 Pacchetto d'onde gaussiano . . . . .	11
2.1.2 Evoluzione analitica . . . . .	12
2.2 Oscillatore armonico . . . . .	13
2.3 Stati coerenti . . . . .	14
2.4 Potenziale di Pöschl-Teller . . . . .	15
<b>3 Applicazioni</b>	<b>17</b>
3.1 Buca finita di potenziale . . . . .	17
3.2 Barriera coulombiana semplificata . . . . .	18
3.3 Pacchetto gaussiano di onde piane in oscillatore armonico . . . . .	19
3.4 Pacchetto gaussiano di onde piane in potenziale di Pöschl-Teller . . . . .	20
3.5 Barriera infinita . . . . .	21
3.6 Potenziale dipendente dal tempo . . . . .	23
<b>4 Descrizione software</b>	<b>25</b>
4.1 Algoritmo di simulazione . . . . .	26
<b>Conclusioni</b>	<b>30</b>

# Introduzione

Quando si tratta l'equazione di Schrödinger,

$$i\hbar \frac{\partial}{\partial t} \psi(\vec{x}, t) = \hat{H}(\vec{x}, t) \psi(\vec{x}, t) , \quad (1)$$

in meccanica quantistica il numero di problemi che si possono risolvere in maniera esatta è piuttosto limitato. Per fare previsioni e ottenere risultati concreti, è quindi di fondamentale importanza utilizzare metodi di soluzione numerici.

Si propone un metodo per la soluzione numerica dell'eq. (1) che consiste nell'applicazione ripetuta dell'espressione infinitesima dell'operatore di evoluzione temporale, approssimato tramite l'espansione di Lie-Trotter.

È stato sviluppato un software che consiste di tre componenti principali:

- un ambiente grafico per poter gestire i parametri della simulazione,
- l'algoritmo simulativo,
- una serie di strumenti grafici per mostrare i risultati tramite animazioni.

Si è utilizzato come linguaggio di programmazione Python, ma parte dell'algoritmo è stato implementato in modo che fosse compatibile con il compilatore Numba, questo per ridurre i tempi di simulazione.

Il metodo è stato innanzitutto validato tramite il confronto delle simulazioni con alcune soluzioni esatte, tra cui la dinamica degli stati coerenti dell'oscillatore armonico e, nel potenziale di Pöschl-Teller, l'evoluzione di un pacchetto gaussiano di autostati. In seguito, si è utilizzato il software per risolvere alcuni dei problemi monodimensionali generalmente proposti nei corsi di meccanica quantistica e altri di particolare interesse.

## 0.1 L'equazione di Schrödinger

La più generica forma dell'equazione di Schrödinger risulta essere quella in eq. (1), dove  $\hat{H}$  rappresenta l'operatore Hamiltoniano <sup>1</sup>

$$\hat{H}(\vec{x}, t) = \frac{1}{2m}(\hat{p} - q\hat{A}(\vec{x}, t))^2 + \hat{V}(\vec{x}, t) - g\frac{q}{2m}\hat{S} \cdot \hat{B}(\vec{x}, t) . \quad (2)$$

Di seguito, nei casi presi in esame, si utilizza una forma semplificata dell'eq. (1) e dell'operatore in eq. (2), poiché verranno utilizzati nella loro forma monodimensionale e senza considerare interazioni di natura elettromagnetica. Questa semplificazione non modifica il formalismo matematico utilizzato, ma permette di rendere più esplicative e chiare le espressioni che seguiranno; è dunque possibile riadattare tali espressioni e generalizzarle utilizzando lo stesso procedimento. L'equazione risolta nelle simulazioni risulta essere

$$i\hbar\frac{\partial}{\partial t}\psi(x, t) = \hat{H}(\vec{x}, t)\psi(\vec{x}, t) \quad \text{con} \quad \hat{H}(\vec{x}, t) = \frac{\hat{p}^2}{2m} + \hat{V}(\vec{x}, t) = \hat{T} + \hat{V} , \quad (3)$$

dove si ricorda che, nella rappresentazione degli  $x$ ,

$$\hat{p} = -i\hbar\frac{\partial}{\partial x} . \quad (4)$$

La soluzione formale dell'equazione di Schrödinger fa uso dell'operatore unitario di evoluzione temporale

$$\psi(x, t) = \hat{U}(t, t_0) \psi(x, t_0) \quad \text{con} \quad \hat{U}(t, t_0) = T \exp \left( \frac{-i}{\hbar} \int_{t_0}^t dt' \hat{H}(x, t') \right) . \quad (5)$$

Tale soluzione, non essendo esplicitamente analitica, non può essere risolta in maniera esatta se non in casi molto particolari. Se però si considera l'espressione infinitesimale dell'operatore  $\hat{U}$ , risulta

$$\hat{U}(t + \delta t, t) = \exp \left( \frac{-i}{\hbar} \hat{H}(x, t) \delta t \right) , \quad (6)$$

che può essere applicato senza difficoltà alla funzione d'onda  $\psi(x, t)$  il numero di volte necessarie a ottenere lo stesso risultato dell'eq. (5). L'unica difficoltà risiede

---

<sup>1</sup>Si ricorda che le grandezze riportate sotto il simbolo  $\hat{\phantom{x}}$  rappresentano degli operatori lineari in uno spazio di Hilbert

nell'espansione dell'operatore, poiché in forma esplicita

$$\hat{U}(t + \delta t, t) = \exp \left( \frac{-i}{\hbar} \delta t (\hat{V} + \hat{T}) \right) , \quad (7)$$

dove  $\hat{V}$  e  $\hat{T}$  sono due operatori che non commutano. Questo rende l'espansione dell'operatore  $\hat{U}$  non banale.

# 1. Metodo

Nel seguente capitolo vengono proposti gli aspetti teorici utili nella comprensione del modello, degli strumenti utilizzati e delle condizioni imposte dalla simulazione sulla natura del sistema.

## 1.1 Espansione di Lie-Trotter

Per espandere l'operatore, si è scelto di utilizzare un risultato noto con il nome di "Formula di Lie-Trotter":

$$\exp(A+B) = \lim_{n \rightarrow \infty} \left[ \exp\left(\frac{A}{n}\right) \exp\left(\frac{B}{n}\right) \right]^n, \quad (1.1)$$

dove  $A$  e  $B$  rappresentano due generici operatori lineari non commutanti. Dall'eq. (1.1) si può dimostrare che vale la seguente espansione

$$e^{\tau(A+B)} = \prod_{j=1}^n \exp(\tau A c_j) \exp(\tau B d_j) + O(\tau^n), \quad (1.2)$$

con  $c_i$  e  $d_i$  coefficienti reali.

All'interno del software sono stati implementati tre diversi livelli di profondità, troncando la produttoria al primo ordine

$$e^{\tau(A+B)} = \exp(\tau A) \exp(\tau B) + O(\tau), \quad (1.3)$$

al secondo ordine

$$e^{\tau(A+B)} = \exp\left(\frac{1}{2}\tau A\right) \exp(\tau B) \exp\left(\frac{1}{2}\tau A\right) + O(\tau^2), \quad (1.4)$$

e al terzo ordine

$$e^{\tau(A+B)} = \exp(\tau A) \exp\left(-\frac{1}{24}\tau B\right) \exp\left(-\frac{2}{3}\tau A\right) \exp\left(\frac{3}{4}\tau B\right) \exp\left(\frac{2}{3}\tau A\right) \exp\left(\frac{7}{24}\tau B\right) + O(\tau^3). \quad (1.5)$$

Per fissare i coefficienti  $c_i$  e  $d_i$  è necessario uguagliare l'espansione in serie di potenze con l'espansione di Lie-Trotter, troncata allo stesso ordine, in modo che termini uguali abbiano lo stesso peso. Al fine di confermare l'eq. (1.2) e esplicitare il processo seguito, si riporta il calcolo nel caso  $n = 1$ .

$$\begin{aligned} e^{\tau(A+B)} &= I + \tau(A+B) + \frac{\tau^2}{2}(A+B)^2 + O(\tau^3) \\ &= I + \tau(A+B) + \frac{\tau^2}{2}(A^2 + AB + BA + B^2) + O(\tau^3) , \end{aligned} \quad (1.6)$$

$$\begin{aligned} e^{\tau A c_1} e^{\tau B d_1} &= \left[ I + \tau A c_1 + \frac{\tau^2}{2} A^2 c_1^2 + O(\tau^3) \right] \left[ I + \tau B d_1 + \frac{\tau^2}{2} B^2 d_1^2 + O(\tau^3) \right] \\ &= I + \tau(A c_1 + B d_1) + \frac{\tau^2}{2}(A^2 c_1^2 + 2 A B c_1 d_1 + B^2 d_1^2) + O(\tau^3) . \end{aligned} \quad (1.7)$$

Come si può vedere dal confronto tra l'eq. (1.6) e l'eq. (1.7), è obbligatoria la scelta  $c_1 = 1$  e  $d_1 = 1$  per ottenere l'uguaglianza al primo ordine. Dalla differenza tra le due si ottiene che

$$\exp(\tau A) \exp(\tau B) = \exp\left(\tau(A+B) + \frac{\tau^2}{2}[A, B] + O(\tau^3)\right) , \quad (1.8)$$

in linea con l'eq. (1.2). Ad ordini superiori la complessità aumenta molto rapidamente, per trovare i coefficienti è, infatti, necessario risolvere un sistema non lineare di  $\sum_{i=1}^n 2^i$  equazioni in  $2n$  incognite. Un tale sistema risulterà quasi sempre indeterminato, permettendo ampia scelta tra soluzioni possibili. Al fine di ridurre il numero di operazioni, la scelta da operare dovrebbe in primo luogo azzerare il numero più alto di coefficienti in modo da rendere banali i termini esponenziali e inoltre prediligere quelle scelte per cui si ottengono coefficienti uguali. Ad esempio, al secondo ordine si ottiene come soluzione

$$c_1 = 1 + \frac{1}{2 + d_2 - 1} , \quad d_1 = 1 - d , \quad c_2 = -\frac{1}{2(d_2 - 1)} , \quad (1.9)$$

per cui per  $d_2 = 0$  si semplifica l'ultimo termine.

$$c_1 = \frac{1}{2} , \quad d_1 = 1 , \quad c_2 = \frac{1}{2} . \quad (1.10)$$

Sfortunatamente non è possibile azzerare nessun coefficiente al terzo al ordine. Non esiste un metodo che sia univocamente riconosciuto come migliore rispetto ad altri

per trovare i coefficienti ad ordini superiori. Il problema però è stato affrontato da diversi autori in letteratura, perciò è possibile fare uso dei loro risultati. Tra i metodi più utilizzati e di maggior successo, per  $n$  pari, si vuole ricordare il metodo frattale [citazione](#).

Il motivo della scelta di usare un'espansione del tipo in eq. (1.2) al posto di un'espansione del tipo

$$\exp(\tau(A+B)) = I + \tau(A+B) + \frac{\tau^2}{2}(A+B)^2 + \frac{\tau^3}{3!}(A+B)^3 + O(\tau^4), \quad (1.11)$$

risiede in una fondamentale proprietà dell'eq. (1.2), tale espansione infatti conserva la natura unitaria dell'operatore di evoluzione, ciò non accade utilizzando l'eq. (1.11)[1]. L'operatore di evoluzione in eq. (6) ha la forma di un esponenziale complesso, quindi approssimato tramite l'eq. (1.2)

$$\exp\left(\frac{-i\delta t}{\hbar}(\hat{V} + \hat{T})\right) = \prod_{j=1}^n \exp\left(\frac{-i\delta t}{\hbar}\hat{V}c_j\right) \exp\left(\frac{-i\delta t}{\hbar}\hat{T}d_j\right) + O(\tau^n), \quad (1.12)$$

diviene un prodotto di esponenziali complessi sempre unitario.

## 1.2 Soluzione numerica

Per risolvere numericamente è necessario definire uno spazio  $x$  finito e discreto di punti  $x_i$  dove campionare la funzione d'onda. In un tale spazio è possibile applicare un numero arbitrario di volte un operatore approssimato del tipo in eq. (1.12) ad una generica funzione  $\psi_0(x_i, t_0)$  e risolvere l'eq. (3).

Per rendere l'algoritmo più efficiente si utilizza un'ulteriore accortezza. Poiché gli algoritmi di differenziazione numerica sono poco efficienti [vorrei inserire un commento quantitativo ma non ho trovato referenze in merito](#) i termini in cui ad esponente compare l'operatore  $\hat{T}$  si applicano nella rappresentazione dei momenti. In tale rappresentazione  $\hat{p}^2$  si comporta come una semplice moltiplicazione

$$-\hbar^2 \frac{\partial^2}{\partial x^2} \longrightarrow p^2. \quad (1.13)$$

Per cambiare rappresentazione, come noto, si utilizza la trasformata di Fourier, i cui algoritmi numerici hanno un comportamento asintotico  $O(n \log n)$ . Alla proce-



dura da eseguire, per ogni passo  $\delta t$  e per ogni  $n$ , nello sviluppo si aggiungono una trasformata e una antitrasformata a seconda del termine che si sta applicando.

$$\exp\left(-\frac{i}{\hbar} c_j V(x)\right) \mathcal{F}^{-1} \left[ \exp\left(-\frac{i}{\hbar} d_j \frac{p^2}{2m}\right) \mathcal{F} [\psi(x, t)] \right] . \quad (1.14)$$

Poiché si utilizza uno spazio discreto, è necessario definire la trasformata discreta Fourier.

### 1.2.1 Trasformata discreta di Fourier

Si considera lo spazio degli  $x$  finito e discreto di  $N$  punti  $x_k$  tale per cui

$$0 \leq x \leq L \quad \text{con} \quad x_k = \frac{L}{N} k, \quad k = 0, \dots, N-1. \quad (1.15)$$

La trasformata discreta di Fourier è definita in analogia con la sua controparte continua

$$\begin{aligned} \tilde{\psi}(p) &\approx \frac{1}{\sqrt{2\pi\hbar}} \frac{L}{N} \sum_{k=0}^{N-1} \exp\left(-i\frac{p}{\hbar} x_k\right) \psi(x_k) \\ &= \frac{1}{\sqrt{2\pi\hbar}} \frac{L}{N} \sum_{k=0}^{N-1} \exp\left(-i\frac{p}{\hbar} \frac{L}{N} k\right) \psi\left(\frac{L}{N} k\right). \end{aligned} \quad (1.16)$$

Si noti che, poiché lo spazio degli  $x$  è finito, si induce una periodicità in  $\tilde{\psi}(p)$  dovuta al termine

$$\exp\left(-i\frac{p}{\hbar} \frac{L}{N} k\right). \quad (1.17)$$

Infatti

$$\tilde{\psi}(p) = \tilde{\psi}(p + P) \quad \text{per} \quad \exp\left(-i\frac{P}{\hbar} \frac{L}{N} k\right) = 1, \quad (1.18)$$

questo si ottiene quando  $P = 2\pi\hbar N/L$ . Per questo motivo si definisce la trasformata in modo che sia univoca limitando lo spazio dei  $p$

$$0 \leq p \leq 2\pi\hbar \frac{N}{L} \quad \text{con} \quad p_j = \frac{2\pi\hbar}{L} j, \quad j = 0, \dots, N-1. \quad (1.19)$$

In definitiva si definisce la trasformata

$$\tilde{\psi}_j = \frac{1}{\sqrt{2\pi\hbar}} \frac{L}{N} \sum_{k=0}^{N-1} \exp\left(-i\frac{2\pi}{N} j k\right) \psi_k \quad \text{con} \quad \psi_k \equiv \psi\left(\frac{L}{N} k\right), \quad (1.20)$$

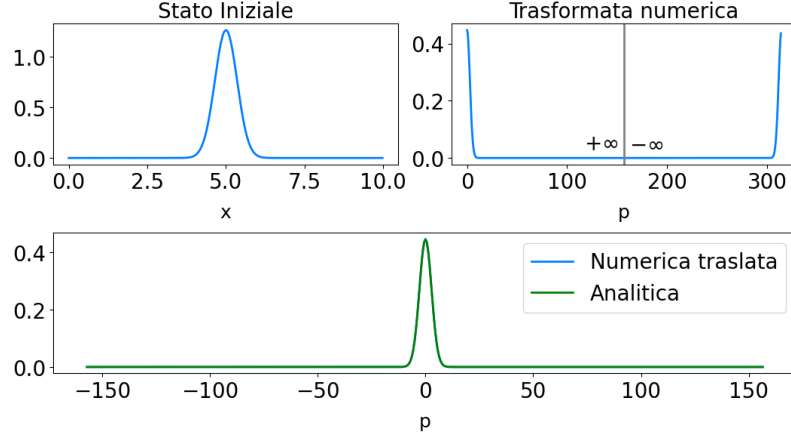


Figura 1.1: Modulo della trasformata di Fourier di una funzione gaussiana centrata in  $x_0 = 5$ . Notare come per ottenere la vera rappresentazione fisica nello spazio dei momenti sia necessario eseguire una traslazione rigida della soluzione numerica per  $p > \pi \hbar N/L$

mentre l'antitrasformata

$$\psi_k = \frac{\sqrt{2\pi\hbar}}{L} \sum_{j=0}^{N-1} \exp\left(-i\frac{2\pi}{N}jk\right) \tilde{\psi}_j. \quad (1.21)$$

La discretizzazione dei valori introduce un'altra problematica. La trasformata discreta mappa tutti i valori di  $\tilde{\psi}(p)$  in un intervallo che non è simmetrico rispetto all'origine, come fa invece la trasformata continua. Se si considera un periodo  $0 \leq p \leq 2\pi\hbar N/L$  in questo spazio sono mappati tutti i valori di  $p$  da  $-\infty$  a  $+\infty$ , ma sono disposti in ordine inverso. Nella prima metà, tra  $0 \leq p \leq \pi\hbar N/L$ , sono mappati i valori tra 0 e  $+\infty$ , mentre nella seconda metà tra  $\pi\hbar N/L < p \leq 2\pi\hbar N/L$  sono riportati i valori corrispondenti all'intervallo tra  $-\infty$  e 0. Per costruire, quindi, la vera rappresentazione fisica della funzione d'onda nello spazio dei  $p$  è necessario eseguire una traslazione rigida della  $\tilde{\psi}(p)$  dall'intervallo  $\pi\hbar N/L < p \leq 2\pi\hbar N/L$  all'intervallo  $-\pi\hbar N/L < p \leq 0$ , come si può vedere dall'esempio riportato in figura (1.1).

### 1.3 Condizioni e vincoli sul sistema

I principali limiti della simulazione sono dovuti al processo di discretizzazione. La discretizzazione dello spazio impone ovvi limiti sulla scelta dei parametri spaziali, poiché essendo lo spazio della simulazione limitato, si perde informazione su qualsiasi oggetto definito al di fuori dell'intervallo  $0 \leq x \leq L$ . È quindi banale, ma necessario, fare in modo che la funzione d'onda sia interamente definita all'interno dell'intervallo, a meno di porzioni infinitesime.

La discretizzazione spaziale introduce, inoltre, due criticità dovute alla natura della trasformata discreta di Fourier. In prima istanza bisogna porre particolare attenzione alla scelta dello stato iniziale, poiché non solo è limitato lo spazio diretto, ma lo è anche lo spazio dei momenti. La discretizzazione pone dei limiti non solo sui parametri spaziali della funzione d'onda, ma anche su quei parametri capaci di modificare la forma del profilo nello spazio nei momenti. Nuovamente è necessario che, a meno di infinitesimi, tutta l'informazione nello spazio dei momenti sia contenuta nell'intervallo  $-\pi\hbar N/L < p \leq \pi\hbar N/L$ . Inoltre la periodicità della trasformata di Fourier genera un effetto per cui un per profilo uscente da un estremo dello spazio viene propagata l'evoluzione come entrante dall'altro estremo dello spazio. Tale effetto disturba l'evoluzione della funzione d'onda originale, rendendola scorretta. Scegliendo opportunamente i parametri, è possibile ovviare a questo problema facendo in modo che le conseguenze di questo effetto si manifestino a tempi maggiori di quelli necessari ad ottenere i risultati desiderati. Se, però, si volesse simulare l'andamento di un sistema per tempi lunghi è possibile ovviare a questo problema modificando la forma del potenziale agli estremi dell'intervallo. Se, infatti, si modifica  $V$  in maniera tale per cui

$$V'(x) = \begin{cases} V(x) & \lambda < x < L - \lambda \\ \exp(-i s) & \text{altrove} \end{cases} \quad s, \lambda \in \mathbb{R}, \quad (1.22)$$

risulterà che la funzione d'onda verrà smorzata esponenzialmente ai bordi. La scelta di  $s$  e  $\lambda$  deve essere opportunamente valutata in modo da ottenere il risultato desiderato. Per valori di  $s$  ed  $\lambda$  troppo piccoli lo smorzamento non sarà sufficiente a

rendere l'effetto trascurabile. D'altra parte all'aumentare di  $s$  aumenta la riflessione dovuta allo smorzamento, altro effetto da voler minimizzare.

La discretizzazione temporale introduce un valore massimo possibile per il potenziale, dovuto ai termini  $\exp(-iVc_j \delta t)$ , che sono funzioni periodiche. Dovrei entrare più nel dettaglio? Bisogna, quindi, considerare la seguente condizione

$$\max_{[0,L]} |V(x)| < \frac{2\pi}{\delta t c_j}. \quad (1.23)$$

In alcuni casi è possibile ovviare a questa problematica tramite opportune estensioni della funzione d'onda, vedi la sezione 3.5

## 2. Validazione del codice

Per collaudare il metodo risolutivo sono stati implementati vari potenziali di cui si conosce la soluzione analitica per poter confrontare i risultati numerici con l'evoluzione esatta. In tutte le simulazioni è stato utilizzato come operatore di evoluzione quello in eq. (1.4), considerato come il miglior compromesso tra precisione e costo computazionale.

### 2.1 Particella libera

#### 2.1.1 Pacchetto d'onde gaussiano

Si ricorda la definizione di pacchetto d'onda per le onde piane, poiché è stato scelto come principale stato iniziale all'interno delle simulazioni.

$$\psi(q, 0) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dk \, g(k) \exp(ikq) , \quad (2.1)$$

dove  $g(k)$  rappresenta una distribuzione di probabilità che fissa i contributi per ogni valore di  $k$ . Per un pacchetto gaussiano<sup>1</sup> e sostituendo le definizioni

$$\hbar k \equiv p - p_0 , \quad q \equiv x - x_0 , \quad (2.2)$$

l'eq. (2.1) si riduce a

$$\psi(x, 0) = \left( \frac{2}{\pi\sigma^2} \right)^{1/4} \exp \left( -i \frac{p_0}{\hbar} (x - x_0) \right) \exp \left( -\frac{(x - x_0)^2}{\sigma^2} \right) , \quad (2.3)$$

dove  $p_0$  rappresenta lo sfasamento del pacchetto nello spazio dei momenti ed è la grandezza che controlla l'energia cinetica associata al pacchetto  $T = p_0^2/2m$ . In figura 2.1 viene riportato un esempio.

---

<sup>1</sup>Dove  $g(k)$  prende la forma di distribuzione gaussiana

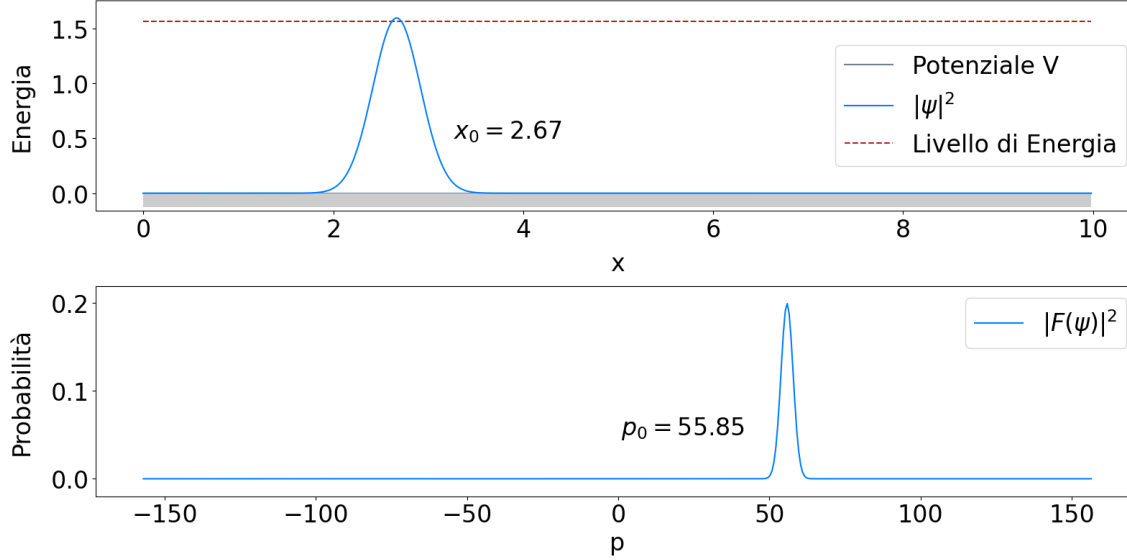


Figura 2.1: Rappresentazione dello stato iniziale per un pacchetto gaussiano di onde piane, sia nello spazio diretto sia nello spazio dei momenti.

### 2.1.2 Evoluzione analitica

Con potenziale di particella libera si intende una configurazione a potenziale costante  $V(x) = 0$ . In tale configurazione si può considerare la soluzione analitica per l'evoluzione di un pacchetto d'onde del tipo in eq. (2.3). Data l'eq. (3), con  $V(x) = 0$ , questa è risolta per ogni funzione del tipo

$$\psi(x, t) = A \exp(i(kx - \omega t)) \quad \text{con} \quad \omega = \frac{\hbar k^2}{2m}. \quad (2.4)$$

Dal principio di sovrapposizione si ottiene che

$$\psi(q, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dk g(k) \exp(i(kq - \omega t)). \quad (2.5)$$

Considerando un pacchetto gaussiano e sostituendo le definizioni in eq. (2.2), si ottiene una funzione analitica per l'evoluzione del pacchetto nel tempo [2]

$$\psi(x, t) = \left(\frac{2\sigma^2}{\pi}\right)^{1/4} \left(\sigma^4 + \frac{4\hbar^2 t^2}{m^2}\right)^{-1/4} \exp\left(i\frac{p_0}{\hbar}x\right) \exp\left(-\frac{[x - p_0 t/m]^2}{\sigma^2 - 2i\hbar t/m}\right) e^{i\phi}, \quad (2.6)$$

$$\text{con} \quad \phi = -\frac{1}{2} \arctan\left(\frac{2\hbar t}{m\sigma_0^2}\right) - \frac{p_0^2}{2m\hbar}t. \quad (2.7)$$

In figura (2.2) si riporta il confronto tra la soluzione numerica e quella analitica per alcuni tempi successivi.

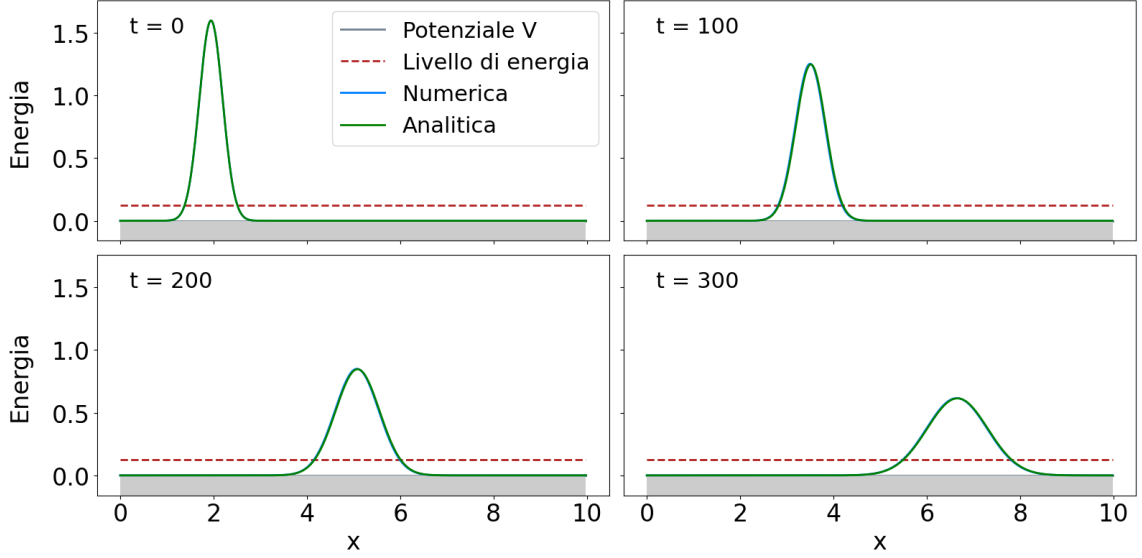


Figura 2.2: Confronto tra soluzione numerica e soluzione analitica per una particella libera. Si può notare un'ottima sovrapposizione tra le due.

## 2.2 Oscillatore armonico

L'oscillatore armonico quantistico è definito da

$$V(x) = \frac{1}{2}m\omega^2(x - x_0)^2. \quad (2.8)$$

Si ricordano i principali risultati ottenuti dalla soluzione dell'equazione di Schrödinger stazionaria

$$E_n = \left(\frac{1}{2} + n\right)\hbar\omega, \quad \psi_n = \left(\frac{\beta^2}{\pi}\right)^{1/4} \frac{1}{\sqrt{2^n n!}} \exp\left(-\frac{\beta^2 x^2}{2}\right) H_n(\beta x), \quad (2.9)$$

$$\text{con } \beta = \sqrt{\frac{m\omega}{\hbar}}. \quad (2.10)$$

È noto che se si considera l'eq. (5) per gli stati stazionari questa si riduce a

$$\psi_n(x, t) = \exp\left(-\frac{i}{\hbar}E_n t\right) \psi_n(x, 0). \quad (2.11)$$

Se inoltre si considera come stato iniziale una combinazione lineare di autostati è possibile trovare l'evoluto temporale combinando le evoluzioni di ogni singolo autostato

$$\psi(x, t) = \sum_k c_k \exp\left(-\frac{i}{\hbar}E_k t\right) \psi_k(x, 0) \quad \text{con} \quad \psi(x, 0) = \sum_k c_k \psi_k(x, 0), \quad (2.12)$$

che è una funzione analitica esatta.

In figura riportano il confronto tra la soluzione numerica e l'evoluzione analitica per la sovrapposizione di autostati dell'oscillatore armonico

$$\psi(x, 0) = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) . \quad (2.13)$$

## 2.3 Stati coerenti

Con stati coerenti [2] si intendono gli autostati dell'operatore di distruzione dell'oscillatore armonico. Sono divenuti celebri, poiché vennero utilizzati da Schrödinger come dimostrazione del principio di corrispondenza. Tali stati, infatti, sono caratterizzati da una dinamica molto simile al comportamento oscillatorio di un oscillatore armonico classico.

Non esiste una definizione analitica per gli stati coerenti, poiché sono definiti tramite una sovrapposizione infinita, del tipo in eq. (2.12), di autostati di oscillatore armonico nel seguente modo

$$|\alpha\rangle = \exp\left(-\frac{|\alpha|^2}{2}\right) \sum_{n=0}^{+\infty} \left(\frac{\alpha^n}{n!}\right) |n\rangle . \quad (2.14)$$

Numericamente non è possibile estendere la somma a  $+\infty$ , per questo è necessario troncarla ad un valore finito, all'interno del codice è stato fissato a 170.

L'evoluzione analitica per il modulo quadro di uno stato coerente corrisponde ad una traslazione rigida di una gaussiana con

$$\langle x \rangle_t = A \cos(\theta - \omega t), \quad \langle p \rangle_t = m\omega A \sin(\theta - \omega t) \quad \text{dove} \quad A = \sqrt{\frac{2\hbar}{m\omega}} |\alpha|, \quad (2.15)$$

che corrispondono alle soluzioni classiche dell'oscillatore armonico.

In figura (2.3) è riportato il confronto tra la soluzione analitica e numerica per l'evoluzione di uno stato coerente, mentre in sezione (3.3) si studia l'evoluzione di pacchetto gaussiano di onde piane anche in confronto ad uno stato coerente.



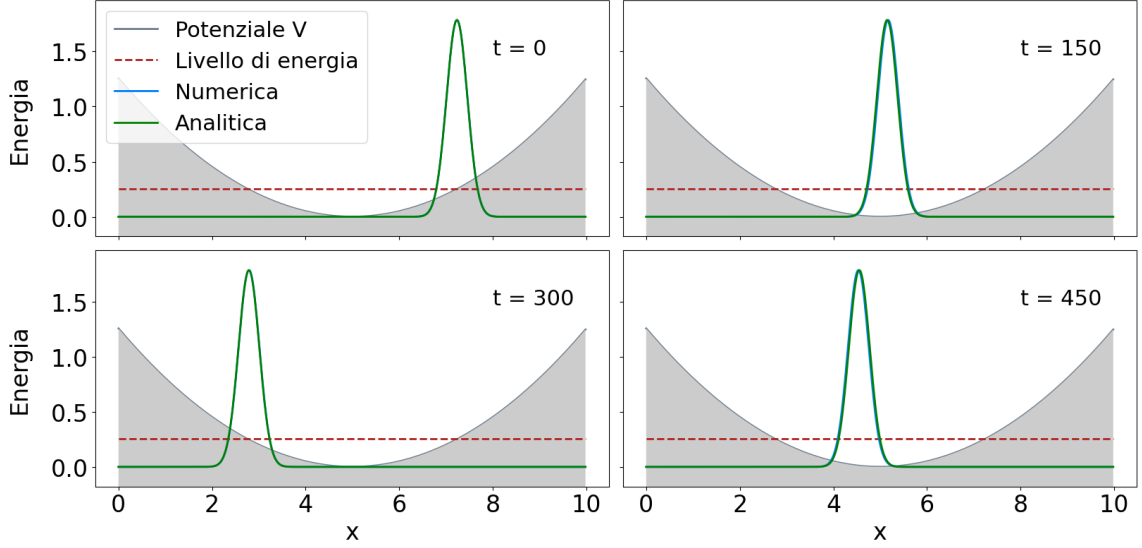


Figura 2.3: Confronto tra soluzione numerica e soluzione analitica per uno stato coerente. Si noti come le dimensioni del pacchetto di forma gaussiana non vengano deformate durante l'evoluzione.

## 2.4 Potenziale di Pöschl-Teller

Il potenziale di Pöschl-Teller (PT), applicato nel contesto della meccanica quantistica, prende la forma

$$V(x) = -\frac{\hbar^2 \alpha^2}{2m} \frac{\nu(\nu+1)}{\cosh^2 \alpha x} . \quad (2.16)$$

Questo potenziale è di particolare importanza poiché si può dimostrare che  $\forall \nu$  il coefficiente di trasmissione,  $T$ , per una funzione d'onda incidente sul potenziale risulta essere pari a 1. Per questo gli viene attribuito l'aggettivo *reflectionless*. Come per il potenziale armonico è possibile risolvere l'equazione agli autovalori per PT con un metodo algebrico, definendo gli operatori di innalzamento e abbassamento, quindi risolvere analiticamente il problema per una qualsiasi combinazione di autostati in analogia con quanto fatto in precedenza. [3]

Fissando  $\nu = 1$  si ottengono gli autostati e autovalori

$$\psi_k(x) = \left[ \frac{ik - \alpha \tanh(\alpha x)}{ik + a} \right] \exp(ikx) , \quad E_k = \frac{\hbar^2 k^2}{2m} \quad \text{con } k \in \mathbb{R} , \quad (2.17)$$

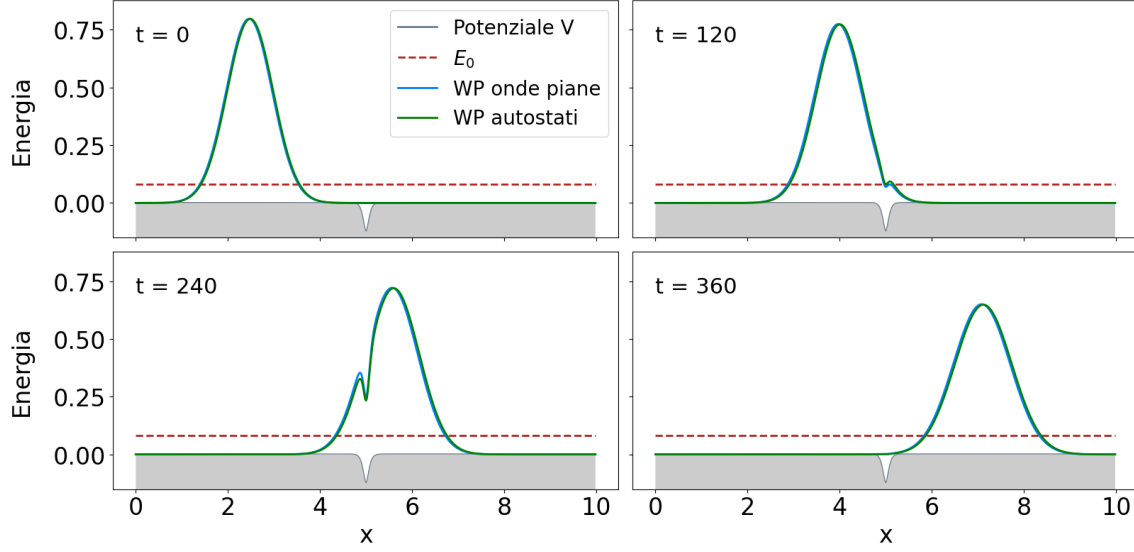


Figura 2.4: Confronto tra soluzione numerica e soluzione analitica per un pacchetto gaussiano di autostati di PL. Ancora una volta si ottiene un ottimo accordo tra la simulazione numerica e l'evoluzione esatta.

che generano uno spettro continuo e infinito. Si può, dunque, costruire un pacchetto d'onde di autostati

$$\psi(x, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dk A(k) \psi_k(x) \exp\left(-\frac{i}{\hbar} E_k t\right), \quad (2.18)$$

si ricorda che  $p - p_0 \equiv \hbar k$ . Fissando  $A(k) = (ik + \alpha)\phi_0(k)$ , dove  $\phi_0(k)$  è la trasformata di Fourier dell'eq. (2.3) e risolvendo l'integrale si ottiene la soluzione per l'evoluzione di un pacchetto d'onde gaussiano di autostati di PT

$$\psi(x, t) = N \left[ i \frac{p_0}{\hbar} - \frac{x - x_0 - v_g t}{\sigma_0 s_t} - \alpha \tanh(\alpha x) \right] \psi_G(x, t), \quad (2.19)$$

$$\text{con } s_t = \frac{\sigma_0}{2} \left( 1 + i \frac{2\hbar t}{m\sigma_0^2} \right) \quad (2.20)$$

e dove  $\psi_G(x, t)$  si riferisce all'evoluzione libera di un pacchetto gaussiano di onde piane,  $v_g = p_0/m$  è la velocità di gruppo.[4]

In figura (2.4) si riporta il confronto tra la soluzione numerica e la soluzione analitica, riscontrando una quasi perfetta sovrapposizione.

### 3. Applicazioni

L'utente tramite l'interfaccia ha la possibilità di scegliere tra alcuni potenziali pre-impostati:

- particella libera,
- oscillatore armonico,
- buca e barriera finita di potenziale,
- gradino di potenziale,
- barriera coulombiana semplificata ,
- potenziale *reflectionless* di Pöschl-Teller,
- barriera infinita.

Da combinare con i seguenti stati iniziali:

- pacchetto gaussiano di onde piane,
- autostati dell'oscillatore armonico e loro sovrapposizioni,
- stati coerenti,
- pacchetto gaussiano di autostati del potenziale di PL.

In questa sezione non solo si riportano alcuni dei risultati ottenuti partendo dagli stati sopracitati, ma anche altre soluzioni di particolare interesse.

#### 3.1 Buca finita di potenziale

La buca finita è definita tramite

$$V(x) = \begin{cases} -V_0 & \text{per } a \leq x \leq b \\ 0 & \text{altrove .} \end{cases} \quad (3.1)$$

La soluzione di questo problema permette di mettere in evidenza un fenomeno caratteristico della meccanica quantistica: la riflessione di una particella può avvenire anche in corrispondenza di una caduta di potenziale. Un tale comportamento è una totale novità rispetto alla meccanica classica. Secondo le equazioni di Newton

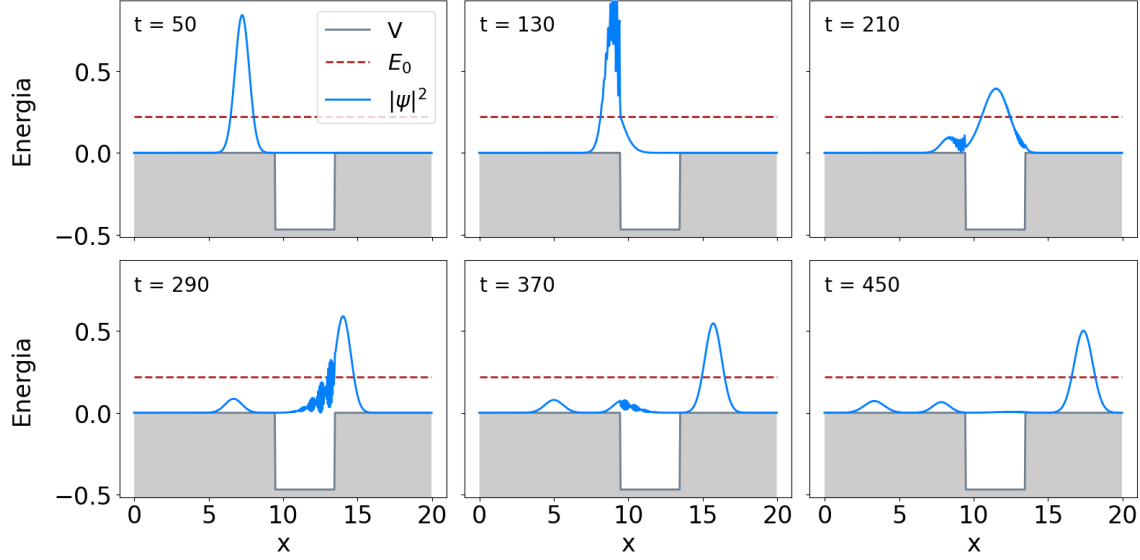


Figura 3.1: Evoluzione di un pacchetto gaussiano di onde piano che interagisce con una buca finita di potenziale, si noti il fatto che parte dell'onda viene riflessa, cosa non prevista in meccanica classica.

$m\ddot{\vec{x}} = -\nabla U(\vec{x}, t)$ , per cui risulta che in corrispondenza di una differenza di potenziale negativa la particella subisca con certezza una determinata accelerazione nella direzione del moto. In meccanica quantistica questo non è certo che accada, come mostrato in figura (3.1), infatti esiste una probabilità diversa zero che la particella si trovi nella zona precedente la buca con momento opposto.

## 3.2 Barriera coulombiana semplificata

È noto come, in prima approssimazione, il potenziale caratteristico per le particelle contenute nel nucleo abbia una forma del tipo

$$V(r) = \begin{cases} -V_N & \text{per } r \leq \tilde{R} \\ (Z e_0^2) (4\pi \epsilon_0) r^{-1} & \text{per } r > \tilde{R} . \end{cases} \quad (3.2)$$

Nel codice è stato implementato un potenziale con lo stesso andamento

$$V(x) = \begin{cases} 0 & \text{per } x \leq \tilde{x} \\ (x - \tilde{x} - (1/V_0))^{-1} & \text{per } x > \tilde{x} , \end{cases} \quad (3.3)$$

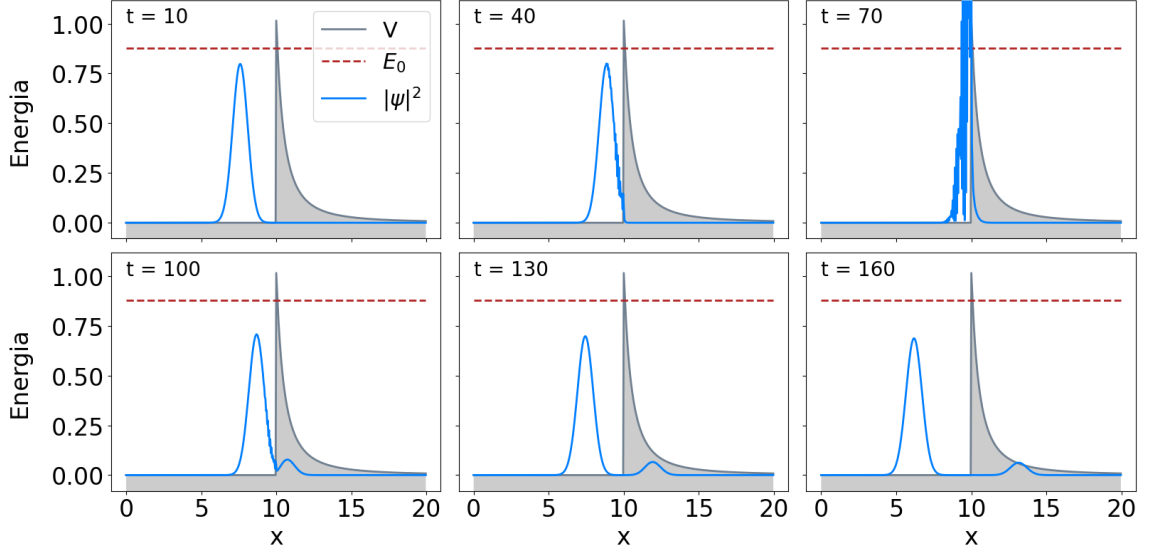


Figura 3.2: Visualizzazione grafica dell'effetto tunnel per un potenziale Coulombiano.

per mettere in evidenza un altro fenomeno caratteristico della meccanica quantistica: l'effetto tunnel. Fenomeno attraverso il quale si giustifica l'espulsione di particelle dal nucleo durante i decadimenti- $\alpha$ . In figura (3.2) si può vedere che nonostante la particella non abbia l'energia sufficiente ad oltrepassare la barriera, **secondo la meccanica classica**, una parte della funzione d'onda viene trasmessa.

### 3.3 Pacchetto gaussiano di onde piane in oscillatore armonico

Si ricorda che il comportamento classico di una particella soggetta ad un potenziale armonico è del tutto simile al comportamento di uno stato coerente, come già valutato in sezione (2.3). In questa sezione si vogliono evidenziare le differenze tra uno stato coerente e l'evoluzione di un pacchetto gaussiano di onde piane a cui è associata la stessa energia. Come si può vedere in figura (3.3), la dinamica del centro di massa del pacchetto di onde piane è facilmente interpretabile tramite la meccanica classica. Infatti il valor medio  $\langle x \rangle$  del pacchetto si sovrappone al valore medio dello stato coerente. D'altra parte mentre gli stati coerenti mantengono le dimensioni del profilo gaussiano, il pacchetto di onde piane subisce continue deformazioni lungo il moto.

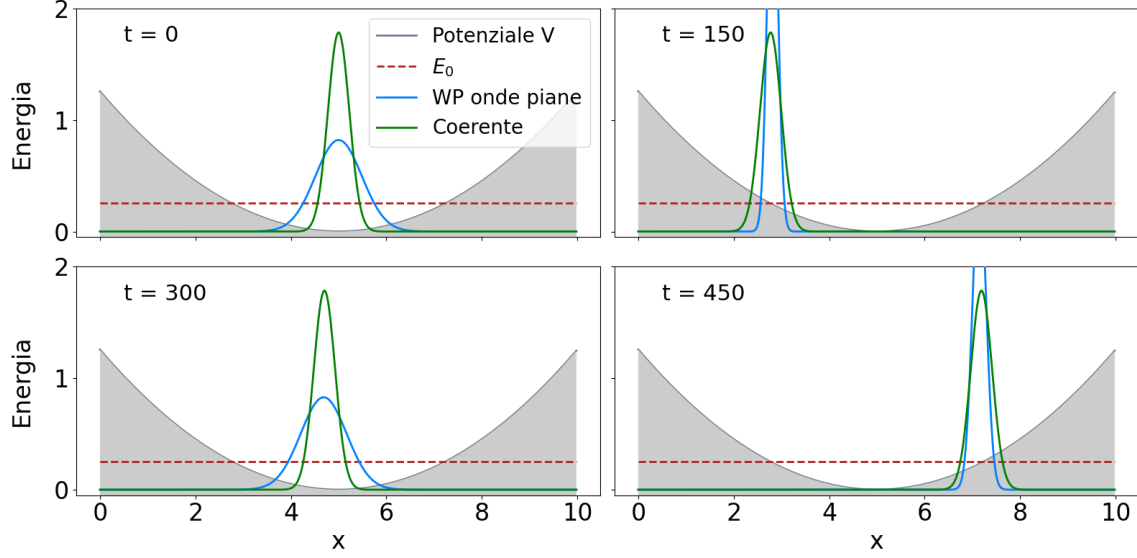


Figura 3.3: Confronto tra l'evoluzione di uno stato coerente e un pacchetto gaussiano di onde piane.

Emerge un comportamento puramente quantistico per cui la varianza del pacchetto subisce delle contrazioni e delle dilatazioni periodiche di periodo pari alla metà di quello dell'oscillatore armonico classico. [5]

### 3.4 Pacchetto gaussiano di onde piane in potenziale di Pöschl-Teller

Se si considera un pacchetto gaussiano di onde piane che attraversa un potenziale di Pöschl-Teller, risulterà che questo verrà interamente trasmesso, come sottolineato in sezione (2.4). Si può studiare la deformazione subita dal pacchetto durante il moto e confrontarla con un pacchetto di autostati di PL e l'evoluzione libera. Come si può riscontrare in figura (3.4), il pacchetto viene deformato in maniera simile al pacchetto di autosati, subisce un'accelerazione e si ottiene un pacchetto contratto rispetto all'evoluzione libera. Le linee verticali in figura (3.4) rappresentano il valore medio  $\langle x \rangle$  della corrispondente funzione d'onda: risulta che il pacchetto d'onde di autostati possiede una velocità di gruppo più alta rispetto al pacchetto gaussiano e risente

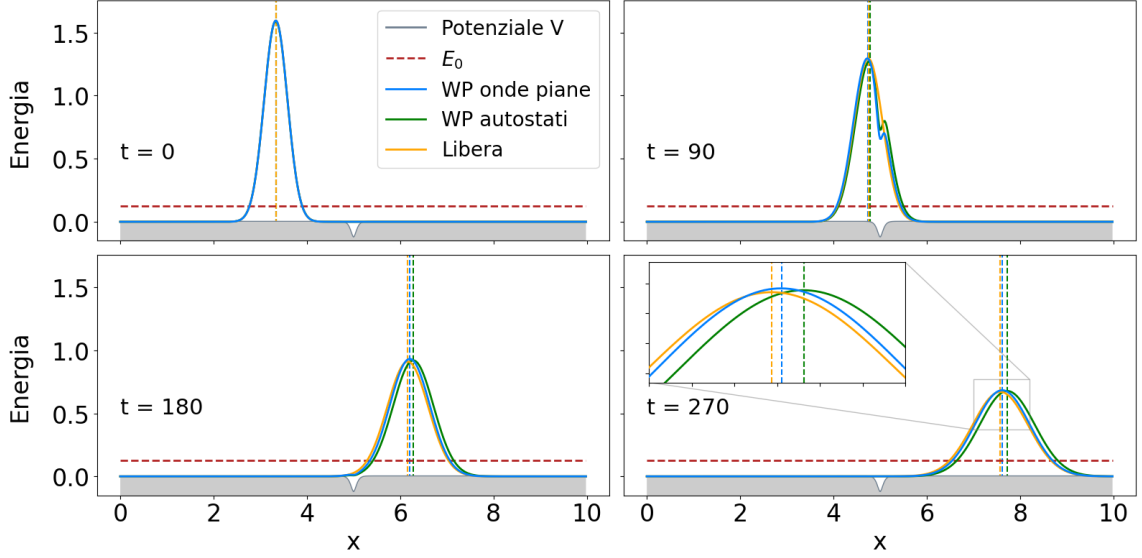


Figura 3.4: Confronto tra l'evoluzione di uno stato coerente e un pacchetto gaussiano di onde piane.

maggiormente degli effetti del potenziale, è infatti accelerato più velocemente.[4]

Si è inoltre confrontata l'evoluzione libera di un pacchetto gaussiano di onde piane con la sua evoluzione nel potenziale PL. Risulta che il pacchetto d'onde subisca una deformazione che lo trasla più velocemente rispetto alla propagazione libera lasciandolo più contratto.[6] *Se si immagina di implementare un reticolo ben dimensionato si potrebbe costruire una macchina che trasporta le particelle da un lato all'altro in maniera molto più efficiente, vedi figura . . . .*

### 3.5 Barriera infinita

Come riportato in sezione 1.3, non è possibile definire potenziali che non rispettino l'eq. (1.23). Il caso particolare della barriera infinita

$$V(x) = \begin{cases} 0 & \text{per } x < x_v \\ \infty & \text{per } x \geq x_v \end{cases} \quad (3.4)$$

è però risolvibile lasciando evolvere un pacchetto opportunamente costruito in un potenziale costante  $V = 0$ . La presenza di una barriera infinita si traduce matema-

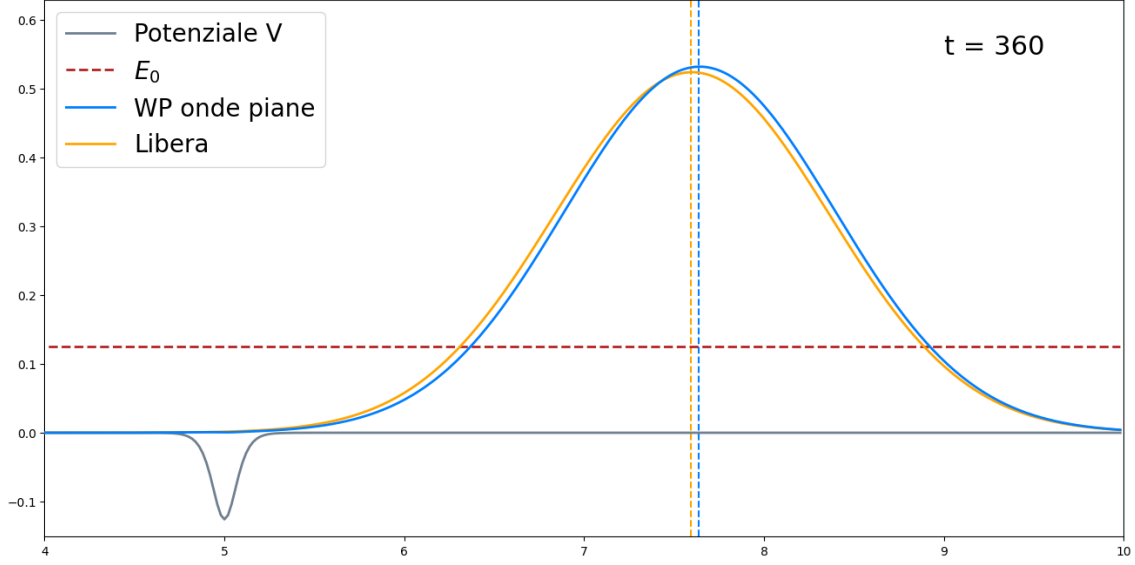


Figura 3.5: Confronto tra l'evoluzione di uno stato coerente e un pacchetto gaussiano di onde piane.

ticamente nell'affermare che

$$\psi(x_v, t) = 0 \quad \forall t, \quad (3.5)$$

nota anche come condizione al bordo di Dirichlet. Se si considera la funzione d'onda antisimmetrica rispetto a  $x_v$  di  $\psi(x, t)$ , costruita con

$$\psi_{\text{ANT}}(x - x_v, t) = -\psi(x_v - x, t), \quad (3.6)$$

questa, in un potenziale libero, evolverà in direzione opposta rispetto a  $\psi$ . Nel momento in cui si considera la sovrapposizione dei due,  $\Psi(x, t) = \psi(x, t) + \psi_{\text{ANT}}(x, t)$ , si ottiene per costruzione che l'eq. (3.5) è identicamente verificata  $\forall t$ . La soluzione del problema proposto risulta essere  $\Psi(x, t)$  valutata nell'intervallo  $(-\infty, 0]$ . In figura (3.6) si riportano i risultati di una simulazione numerica per un pacchetto gaussiano in un potenziale di eq. (3.4), risolto con il metodo appena descritto.



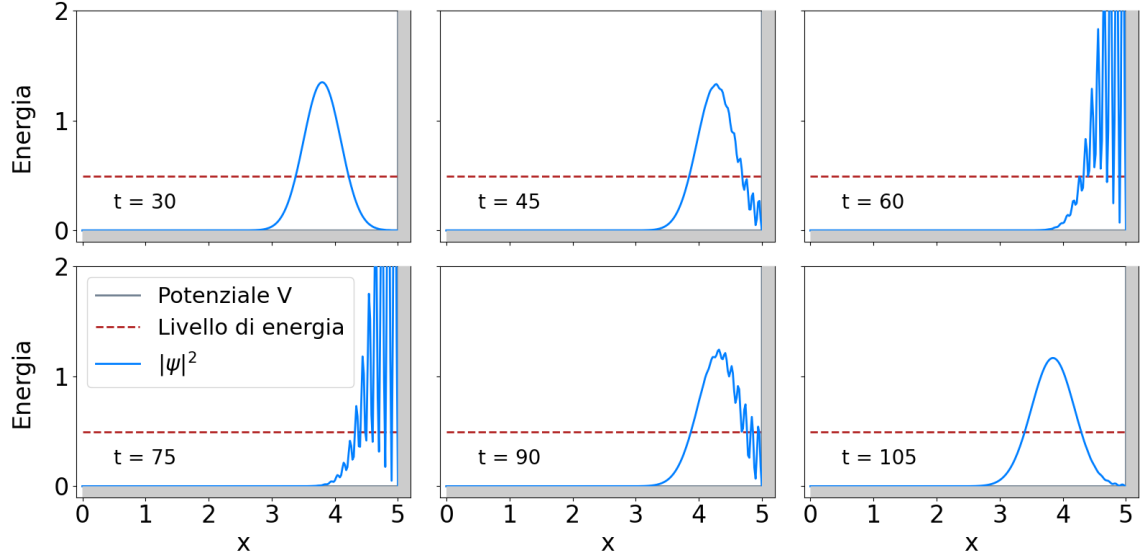


Figura 3.6: Evoluzione di un pacchetto gaussiano di onde piano che interagisce con una barriera di potenziale infinita. *Se può essere interessante posso sovrapporre questa soluzione con la soluzione di un potenziale step con  $V_0 \approx V_{MAX}$*

### 3.6 Potenziale dipendente dal tempo

Come anticipato è possibile risolvere anche problemi con potenziale dipendente dal tempo. Come esempio si è scelto di implementare un potenziale

$$V(x) = \alpha (x - x_0)^2 + \frac{|\sin(\omega_v t)|}{1 + \beta (x - x_0)^2}. \quad (3.7)$$

Come si può vedere in figura (3.7) è evidente come la funzione d'onda reagisca al potenziale mentre questo cambia forma.

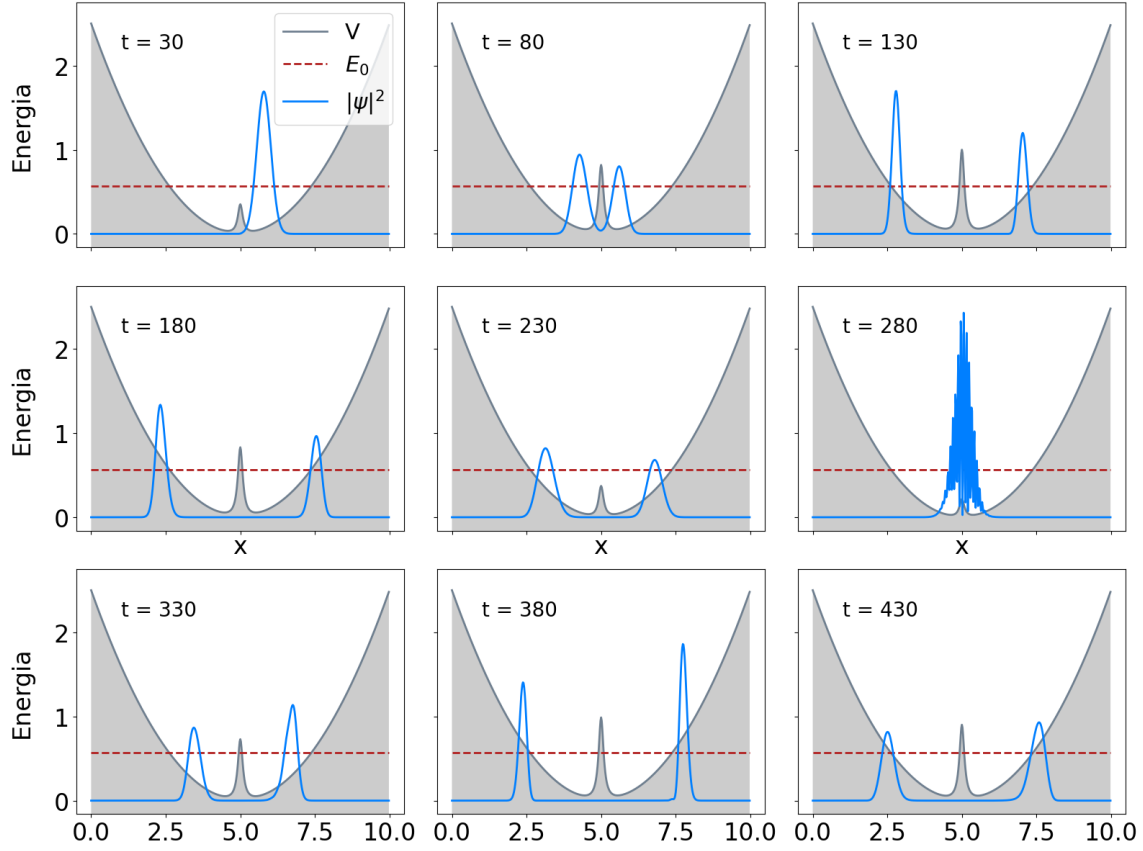


Figura 3.7: Evoluzione di un pacchetto gaussiano di onde piano che interagisce con un potenziale dipendente dal tempo. Notare come per  $t \approx 80$  la funzione d'onda venga separata tra le due regioni per poi sovrapporsi nuovamente per  $t \approx 280$

## 4. Descrizione software

Il software è diviso in tre componenti principali:

1. un ambiente grafico per la gestione dei parametri iniziali,
2. l'algoritmo di simulazione,
3. alcuni strumenti grafici per la visualizzazione dei risultati.

Ognuna di queste componenti è indipendente dalle altre, poiché le informazioni sono passate attraverso file che vengono ogni volta aggiornati, in questo modo è sempre possibile risalire ai parametri utilizzati, modificarli senza usare l'ambiente grafico e gestire tutto da riga di comando. In figura (4.1) è riassunto lo schema di funzionamento. Si crea tramite l'interfaccia un file *Par.py* che riassume tutti i parametri iniziali, si lancia la simulazione dall'interfaccia, l'algoritmo simulativo salva i risultati in una cartella precedentemente selezionata. I risultati sono poi animati tramite un altro strumento grafico. Si noti che da riga di comando è possibile lanciare in maniera indipendente sia l'algoritmo simulativo sia gli strumenti di animazione, basta che siano presenti i file intermedi necessari.

Il codice è stato scritto utilizzando il linguaggio Python appoggiandosi alle librerie esterne:

- Numpy, per le funzioni matematiche e la gestione dei vettori,
- Matplotlib, per generare i grafici e le animazioni,
- Tkinter, per la costruzione dell'interfaccia grafica,
- Numba, per compilare le funzioni di evoluzione temporale.

Per visualizzare i risultati, oltre che al semplice strumento di animazione, sono stati implementati:

- uno strumento per confrontare le soluzioni numeriche con le soluzioni analitiche,
- uno strumento per visualizzare l'evoluzione in contemporanea nello spazio diretto e nello spazio dei momenti,
- uno strumento per il calcolo dei coefficienti di riflessione e di trasmissione.

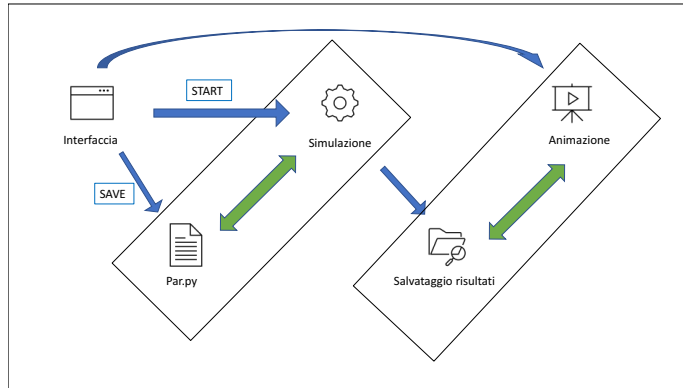


Figura 4.1: Schema di funzionamento del software. È una bozza se mi conferma che può andare bene la perfeziono

- uno strumento per il calcolo dei valori medi
- uno strumento per la valutazione dell'errore rispetto alla funzione analitica

## 4.1 Algoritmo di simulazione

Il cuore dell'algoritmo di simulazione è un ciclo che richiama ad ogni iterazione una funzione di evoluzione e salva i risultati nella cartella indicata dall'utente. Sono state implementate tre diverse funzioni di evoluzione che differiscono per generalità e prestazioni. Inoltre l'utente può scegliere tra due tipologie di algoritmi di simulazione: con o senza anteprima, dove con anteprima si intende una visualizzazioni in diretta dei risultati della simulazione. Questa scelta è dovuta alla perdita di velocità causata dalla creazione immediata dei grafici. Di seguito si riporta il codice utilizzato per la più generica funzione di evoluzione, con la quale è possibile simulare un potenziale dipendente dal tempo e selezionare l'ordine a cui viene troncata l'espansione.

```

1 @jit
2 def evolution_t_dependent(y, q, k, dt, V, m, mat_f, mat_i,
3     N, L, h, P_MAX, f_zero, coefficients, Level):
4     for i in range(Level):
5         if coefficients[i][0] != 0:
6             # V terms application
7             y = exp_V(y, V, dt, h, coefficients[i][0])

```

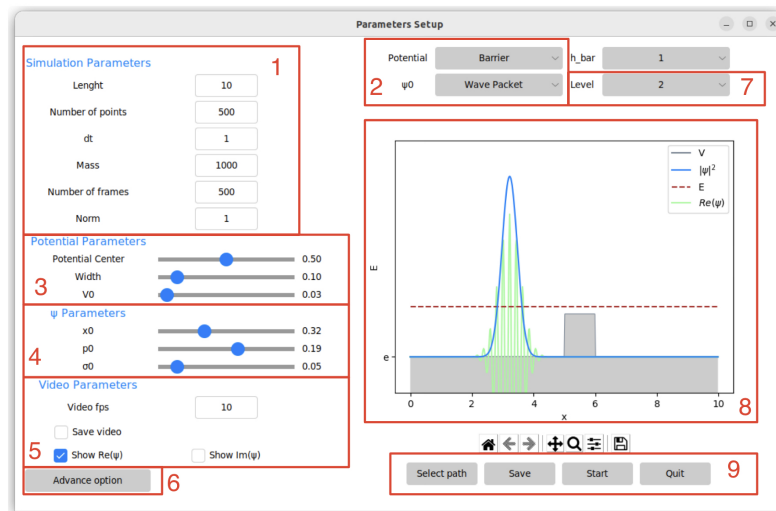


Figura 4.2: Illustrazione dell'interfaccia.

1. Si possono modificare i parametri caratteristici dello spazio e della particella, nonché impostare il numero e la grandezza dei passi temporali da eseguire.
2. Permette di scegliere tra gli stati e i potenziali citati al capitolo (3).
3. È possibile modificare i parametri caratteristici del potenziale selezionato.
4. È possibile modificare i parametri caratteristici dello stato iniziale selezionato. Quando si modificano parametri che influenzano la funzione d'onda nello spazio dei momenti viene messa in mostra un'anteprima dello stato iniziale nello spazio dei momenti.
5. Si possono modificare alcuni parametri per la creazione dell'animazione finale. Tra questi è possibile mostrare la parte reale e immaginaria della soluzione.
6. Tra le opzioni avanzate si può selezionare l'algoritmo di simulazione, vedi sezione (4.1), e inserire dei controlli per la conservazione della norma e dell'energia.
7. Il parametro *Level* indica l'ordine a cui viene troncata l'espansione in eq. (1.2).
8. Viene visualizzata in diretta un'anteprima dello stato iniziale.
9. Cliccando sui seguenti tasti si controllano le funzioni dell'interfaccia. Per eseguire in maniera corretta la simulazione la procedura prevista è: selezionare il percorso di salvataggio, salvare i risultati, far partire la simulazione e attendere l'animazione, uscire.

```

8         if coefficients[i][1] != 0:
9             # Fourier transform
10            f_y = DFT.Dft(y ,q, k, mat_f, N, L, h, f_zero)
11
12            f_y = DFT.shift(f_y, k, P_MAX)
13            k = DFT.shift_p(k, P_MAX)
14
15            # T terms application
16            f_y = exp_T(f_y, k, m, dt, h, coefficients[i][1])
17
18            f_y = DFT.back_shift(f_y, k)
19            k = DFT.back_shift_p(k, P_MAX)
20
21            # Anti-Fourier transform
22            y= DFT.iDft(f_y, q, k, mat_i, L, h, f_zero)
23
24    return y

```

Il simbolo `@jit` è detto *decorator* e serve per compilare la funzione attraverso Numba. Tale processo rende il calcolo molto più rapido, ma per poter sfruttare al meglio le potenzialità di questo compilatore bisogna rinunciare ad alcune comodità tipiche dei linguaggi interpretati come Python. Numba richiede, infatti, che tutte le variabili utilizzate nella funzione siano passate come parametri e che le variabili siano di tipo statico. Per questo vengono inseriti tra i parametri alcuni termini che non hanno a che fare con l'evoluzione del sistema, ma che sono utilizzati come variabili di appoggio, come ad esempio `f_zero`. DFT è la libreria personale utilizzata per raggruppare i metodi che computano la trasformata di Fourier. Si ricorda che per poter applicare i termini a cui esponente compare  $\hat{T}$  oltre che applicare la trasformata di Fourier è anche necessario traslare la soluzione, questo viene fatto tramite i metodi `shift`, vedi sezione (1.2.1). Per rendere il codice più efficiente, le matrici, `mat_f` e `mat_i`, utilizzate per calcolare le trasformate sono precalcolate e vengono passate alle funzioni come parametri. Il parametro `coefficients` rappresenta un vettore di dimensioni  $Level \times 2$  che raggruppa i coefficienti  $c_j$  e  $d_j$  dell'espansione. Le funzioni `exp_V` e `exp_T` restituiscono semplicemente i termini esponenziali che compaiono

nell'espansione. La funzione può essere resa più efficiente in quei casi in cui è possibile precalcolare alcuni termini. Per i potenziali non dipendenti dal tempo, infatti, i risultati restituiti dalle funzioni  $\text{exp}_V$  e  $\text{exp}_T$  sono sempre gli stessi. Quindi risulta più efficiente precalcolarli e passarli come parametri, in modo da dover eseguire solo una moltiplicazione. Con questa modifica si ottiene la seconda versione della funzione di evoluzione. Rinunciando ad un ulteriore grado di libertà si può rendere la funzione ancora più efficiente. Infatti le prestazioni del compilatore Numba sono infatti piuttosto sensibili alla presenza di condizioni *if* e di condizioni sui cicli. Se si elimina il ciclo *for* per sostituirlo con una serie di operazioni da eseguire tutte in fila e si eliminano manualmente i termini nulli si ottiene un considerevole aumento nelle prestazioni. Per poter implementare una tale funzione è quindi necessario conoscere in precedenza sia il livello sia i coefficienti associati a tale livello. Nel codice è stata implementata una funzione di quest'ultimo tipo per il livello che corrisponde a troncatura l'espansione a  $O(\delta t^3)$ .

# Conclusioni

Grazie al lavoro esposto in sezione (2) è possibile affermare che la soluzione numerica generata tramite la simulazione è del tutto compatibile con la soluzione esatta. È quindi possibile utilizzare questo strumento per risolvere un qualsiasi problema monodimensionale e ottenere risultati concreti come in sezione (3).

Il formalismo esposto in sezione (1) è dunque verificato e applicato. Tale formalismo è generalizzabile per qualsiasi operatore  $\hat{H}$  in un numero arbitrario di dimensioni, quindi si può estendere ad ogni sistema quantistico. I problemi che si possono riscontrare nella generalizzazione, oltre al fatto che la comprensione delle formule matematica diviene più difficoltosa, riguardano soprattutto il costo computazionale e le strategie di visualizzazione dei risultati. Ogni volta che si aumentano le dimensioni o i gradi di libertà del sistema (ad esempio considerando lo spin), è necessario ampliare la grandezza dei vettori che contengono la funzione d'onda. Nello spazio delle coordinate l'aumento di punti da considerare scala con  $N^d$ , dove  $d$  rappresenta le dimensioni. Inoltre per rappresentare funzioni d'onda multidimensionali è necessario animare uno spazio tridimensionale o quadridimensionale, il che anche se possibile rende i risultati più difficili da interpretare.



# Ringraziamenti

# Bibliografia

- [1] N. Hatano e M. Suzuki. «Finding Exponential Product Formulas of Higher Orders». In: *Quantum Annealing and Other Optimization Methods*. Springer Berlin Heidelberg, 2005, pp. 37–68. DOI: 10.1007/11526216\_2.
- [2] Claude Cohen-Tannoudji, Bernard Diu e Franck Laloë. *Quantum mechanics; 1st ed.* Trans. of : Mécanique quantique. Paris : Hermann, 1973. New York, NY: Wiley, 1977. URL: <https://cds.cern.ch/record/101367>.
- [3] Ryan Jaffe. «An Algebraic Approach to Reflectionless Potentials in One Dimension». In.
- [4] S. V. Mousavi. *Scattering by a reflectionless modified Pöschl-Teller potential: Bohmian trajectories and arrival times*. 2015. DOI: 10.48550/ARXIV.1507.03341. URL: <https://arxiv.org/abs/1507.03341>.
- [5] Hideo Tsuru. «Wave Packet Motion in Harmonic Potential». In: *Journal of the Physical Society of Japan* 60.11 (1991), pp. 3657–3663. DOI: 10.1143/JPSJ.60.3657. URL: <https://doi.org/10.1143/JPSJ.60.3657>.
- [6] John Lekner. «Reflectionless eigenstates of the sech<sup>2</sup> potential». In: *American Journal of Physics* 75.12 (2007), pp. 1151–1157. DOI: 10.1119/1.2787015. URL: <https://doi.org/10.1119/1.2787015>.