Università degli Studi di Genova

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Fisica

Risoluzione dell'equazione di Schrödinger mediante la simulazione di un pacchetto d'onda gaussiano

Tesina per Fisica Computazionale 2

Daniele Rapetti

Sommario

Ho sviluppato e generalizzato il metodo di Crank-Nicolson per adattarlo all'equazione di Schrödinger . Ho quindi analizzato i risultati e provato l'evoluzione con un salto di potenziale e una barriera rettangolare.

Indice

Parte I
Teoria

1 Introduzione: matematica

1.1 Derivate numeriche

Prima di tutto faccio un piccolo elenco di metodi di calcolo per le derivate, usando il metodo delle differenze finite:

La derivata prima (in avanti) è:

$$\frac{\partial F}{\partial x}(a) \simeq \frac{F(a+h) - F(a)}{h} + O(h) \tag{1.1.1}$$

Ma la sua precisione è al primo ordine, per cui si può utilizzare la cosiddetta definizione "centrale":

$$\frac{\partial F}{\partial x}(a) \simeq \frac{F(a+h) - F(a-h)}{2 * h} + O(h^2) \tag{1.1.2}$$

Per la derivata seconda il discorso è simile ma in entrambi i casi la precisione è sempre al secondo ordine:

$$\frac{\partial^2 F}{\partial x^2}(a) \simeq \frac{\frac{\partial F}{\partial x}(a+h) - \frac{\partial F}{\partial x}(a)}{h} = \frac{F(a+2h) + F(a) - 2F(a+h)}{h^2} + O(h^2) \tag{1.1.3}$$

$$\frac{\partial^2 F}{\partial x^2}(a) \simeq \frac{F(a+h) + F(a-h) - 2F(a)}{h^2} + O(h^2) \tag{1.1.4}$$

Definito come si calcolano le derivate si può iniziare a pensare di risolvere delle equazioni differenziali in maniera numerica.

1.2 Equazione del calore

Il metodo che voglio illustrare inizialmente è stato concepito per la risoluzione di equazioni differenziali iperboliche, come quella del calore:

$$\frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial x^2} \tag{1.2.1}$$

La prima cosa da fare è calcolare i valori delle derivate:

$$\frac{\partial T}{\partial t}(x,t) = \frac{T(x, t+\Delta t) - T(x,t)}{\Delta t} \tag{1.2.2}$$

e

$$\frac{\partial^2 T}{\partial x^2}(x,t) = \frac{T(x+\Delta x, t) + T(x-\Delta x, t) - 2T(x,t)}{\Delta x^2} \tag{1.2.3}$$

1.2.1 Eulero esplicito, "in avanti"

Quindi prendo i risultati e li sostituisco nell'equazione le derivate (con queste definizioni si ha precisione Δx^2 nello spazio e Δt nel tempo):

$$\frac{T(x, t+\Delta t) - T(x,t)}{\Delta t} = k \frac{T(x+\Delta x, t) + T(x-\Delta x, t) - 2T(x,t)}{\Delta x^2} \tag{1.2.4}$$

e andando avanti:

$$T(x, t+\Delta t) = k \frac{\Delta t}{\Delta x^2} \left(T(x+\Delta x, t) + T(x-\Delta x, t) - 2T(x,t)\right) + T(x,t) \tag{1.2.5}$$

A questo punto calcolo ogni punto conoscendo il corrispondente e i due primi vicini del passo precedente.

1.2.2 Eulero implicito, "all'indietro"

Posso, volendo, calcolare la derivata spaziale nell'istante temporale successivo:

$$\frac{T(x,t+\Delta t)-T(x,t)}{\Delta t}=k\frac{T(x+\Delta x,t+\Delta t)+T(x-\Delta x,t+\Delta t)-2T(x,t+\Delta t)}{\Delta x^2} \tag{1.2.6}$$

proseguo:

$$T(x,t+\Delta t)-k\frac{\Delta t}{\Delta x^2}\left(T(x+\Delta x,t+\Delta t)+T(x-\Delta x,t+\Delta t)-2T(x,t+\Delta t)\right)=T(x,t) \tag{1.2.7}$$

Anche in questo caso non approfondisco, l'approccio è simile al metodo di Crank-Nicolson, ma preferisco parlarne direttamente di quest'ultimo per poi generalizzarlo e descriverne l'algoritmo utilizzato per la risoluzione dell'equazione di Schrödinger

1.3 Un esempio di come ricavare il metodo di Crank Nicolson: l'equazione del calore

Innanzitutto utilizziamo la definizione centrale della derivata prima in modo da avere una precisione di Δt^2 anche per quanto riguarda il tempo, ma calcolandola in $t+\Delta t/2$ con incremento $\Delta t/2$:

$$\frac{\partial T}{\partial t}(x,t+\Delta t/2)\simeq\frac{T(x,t+\Delta t/2+\Delta t/2)-T(x,t-\Delta t/2+\Delta t/2)}{2*\Delta t/2}=\frac{T(x,t+\Delta t)-T(x,t)}{\Delta t} \tag{1.3.1}$$

Per lo spazio utilizziamo le derivate calcolate negli esempi precedenti. A questo punto abbiamo $\frac{\partial T}{\partial t}(x,t+\Delta t/2)$, $\frac{\partial^2 T}{\partial x^2}(x,t)$ e $\frac{\partial^2 T}{\partial x^2}(x,t+\Delta t)$ per calcolare l'equazione facciamo la media delle due derivate spaziali ai tempi t e $t+\Delta t$.

$$\begin{aligned}\frac{T(x,t+\Delta t)-T(x,t)}{\Delta t}=\frac{K}{2}\Bigg(&\frac{T(x+\Delta x,t)+T(x-\Delta x,t)-2T(x,t)}{\Delta x^2}\\&+\frac{T(x+\Delta x,t+\Delta t)+T(x-\Delta x,t+\Delta t)-2T(x,t+\Delta t)}{\Delta x^2}\Bigg)\end{aligned} \tag{1.3.2}$$

In pochi passaggi si arriva a separare le parti a tempo differente, con $\eta=K\frac{\Delta t}{\Delta x^2}$:

$$\left(\frac{2}{\eta}+2\right)T(x,t+\Delta t)-T(x+\Delta x,t+\Delta t)-T(x-\Delta x,t+\Delta t)=\left(\frac{2}{\eta}-2\right)T(x,t)+T(x+\Delta x,t)+T(x-\Delta x,t) \tag{1.3.3}$$

Per ogni istante di tempo ho una matrice tridiagonale per il passo temporale che conosco e una per quello successivo. In sottosezione 2.2 descriverò come si risolve una di queste matrici.

2 Costruzione dell'algoritmo

2.1 Ottenere il sistema

Ho fatto un esempio con l'equazione del calore. Prima di procedere alla spiegazione su come si semplifica e si risolve il sistema di equazioni riconducibile ad una matrice tridiagonale spiegherò come condurre la PDE più generale ad un sistema del genere.

Scrivo la più generica equazione risolvibile con questo metodo:

$$\partial_t F = D_2 \partial_x^2 F + D_1 \partial_x F + V(x,t)F + U(x,t) \tag{2.1.1}$$

Il coefficiente della derivata temporale è ignorato perché è incluso negli altri coefficienti e non deve essere diverso da 0, ovviamente D_2, il coefficiente della derivata seconda spaziale non deve essere mai uguale a zero! Inoltre preferisco lasciare D_1 e D_2 come costanti nel tempo e nello spazio, ma cambiare le costanti costa poco a livello di complessità del calcolo, ma lascio però la dipendenza temporale e spaziale a V e U.

In seguito indicherò la il passo nello spazio a pedice con $_i$ e quello nel tempo in apice con ^{j}. Discretizzo l'equazione, calcolando la derivata temporale in $j+1/2$ con passo $\Delta t/2$ e usando la definizione centrale, per quanto riguarda le derivate spaziali faccio la media tra quelle calcolate in j e in $j+1$. I potenziali sono noti.

$$\begin{aligned}\frac{F_i^{j+1}-F_i^j}{\Delta t} = \frac{1}{2}\Bigg(& D_2\frac{F_{i+1}^j+F_{i-1}^j-2F_i^j}{\Delta x^2} + D_1\frac{F_{i+1}^j-F_{i-1}^{j+1}}{2\Delta x} + F_i^j V_i^{j+1} + U_i^{j+1/2} + \\ & D_2\frac{F_{i+1}^{j+1}+F_{i-1}^{j+1}-2F_i^{j+1}}{\Delta x^2} + D_1\frac{F_{i+1}^{j+1}-F_{i-1}^{j+1}}{2\Delta x} + F_i^j V_i^j + U_i^{j+1/2}\Bigg)\end{aligned} \tag{2.1.2}$$

ho usato la definizione di derivata centrale anche per ∂_x in modo da mantenere la precisione Δx^2.

Non mostro i passaggi che portano al risultato. Per comodità indico $\eta = \frac{D_2\Delta t}{\Delta x^2}$ e scrivo:

$$\begin{aligned}\left(-\eta + D_1\frac{\Delta t}{2\Delta x}\right)F_{i-1}^{j+1} + \left(2+2\eta-\Delta t V_i^{j+1/2}\right)F_i^{j+1} + \left(-\eta - D_1\frac{\Delta t}{2\Delta x}\right)F_{i+1}^{j+1} - \Delta t U_i^{j+1/2} = \\ \left(\eta - D_1\frac{\Delta t}{2\Delta x}\right)F_{i-1}^{j} + \left(2-2\eta+\Delta t V_i^{j+1/2}\right)F_i^{j} + \left(\eta + D_1\frac{\Delta t}{2\Delta x}\right)F_{i+1}^{j} + \Delta t U_i^{j+1/2}\end{aligned} \tag{2.1.3}$$

Per rendere più chiara spiegazione e risoluzione della matrice tridiagonale riassumo l'equazione precedente in:

$$a_i^j F_{i-1}^{j+1} + d_i^j F_i^{j+1} + c_i^j F_{i+1}^{j+1} + e_i^j = ak_i^j F_{i-1}^j + dk_i^j F_i^j + ck_i^j F_{i+1}^j + ek_i^j \tag{2.1.4}$$

con:

$$\begin{array}{llr} a_i^j = -1 + \frac{D_1}{D_2}\frac{\Delta x}{2} & ak_i^j = 1 - \frac{D_1}{D_2}\frac{\Delta x}{2} & \text{parametri dei } (F_{i-1}^*) \\ d_i^j = \frac{1}{\eta}\left(2-\Delta t V_i^{j+1/2}\right)+2 & dk_i^j = \frac{1}{\eta}\left(2+\Delta t V_i^{j+1/2}\right)-2 & \text{parametri dei } (F_i^*) \\ c_i^j = -1 - \frac{D_1}{D_2}\frac{\Delta x}{2} & ck_i^j = 1 + \frac{D_1}{D_2}\frac{\Delta x}{2} & \text{parametri dei } (F_{i+1}^*) \\ e_i^j = -\frac{\Delta x^2}{D_2}U_i^{j+1/2} & ek_i^j = \frac{\Delta x^2}{D_2}U_i^{j+1/2} & \text{funzioni esterne} \end{array} \tag{2.1.5}$$

In realtà i parametri e_i^j e ek_i^j dato che non moltiplicano la funzione possono essere accorpati. Ho preferito mantenerli separati perché mi sembrava che in questo modo fosse più chiaro capirne la provenienza. Per lo stesso motivo ho preferito mantenere separati i valori che moltiplicano la funzione nota e quelli che moltiplicano il passo successivo.

2.2 La matrice Tridiagonale: soluzione

D'ora in avanti ometto la dipendenza temporale delle componenti della matrice. Per ogni istante di tempo j ho un sistema di N equazioni nella forma:

$$a_i F_{i-1}^{j+1} + d_i F_i^{j+1} + c_i F_{i+1}^{j+1} + e_i = ak_i F_{i-1}^j + dk_i F_i^j + ck_i F_{i+1}^j + ek_i \tag{2.2.1}$$

Dove j rappresenta l'istante di tempo che conosco e $j+1$ quello che sto calcolando. La prima cosa da fare è portare nel membro a destra tutti i parametri noti, dando per scontato che l'unica incognita dell'equazione è la funzione:

$$a_i F_{i-1}^{j+1} + d_i F_i^{j+1} + c_i F_{i+1}^{j+1} = ak_i F_{i-1}^{j} + dk_i F_i^{j} + ck_i F_{i+1}^{j} + ek_i - e_i \tag{2.2.2}$$

Per proseguire con la risoluzione è meglio passare alla rappresentazione matriciale del sistema (rappresento gli N punti rispettando le convenzioni del C, quindi $i = 0 \to N-1$):

$$\begin{pmatrix} d_0 & c_0 & & & & \\ a_1 & d_1 & c_1 & & & \\ & & ... & & & \\ & & & a_{N-2} & d_{N-2} & c_{N-2} \\ & & & & a_{N-1} & d_{N-1} \end{pmatrix} \boldsymbol{F}^{j+1} = \begin{pmatrix} dk_0 & ck_0 & & & & \\ ak_1 & dk_1 & ck_1 & & & \\ & & ... & & & \\ & & & ak_{N-2} & dk_{N-2} & ck_{N-2} \\ & & & & ak_{N-1} & dk_{N-1} \end{pmatrix} \boldsymbol{F}^{j} + \begin{pmatrix} ek_0 - e_0 \\ ek_1 - e_1 \\ ... \\ ek_{N-2} - e_{N-2} \\ ek_{N-1} - e_{N-1} \end{pmatrix} \tag{2.2.3}$$

Per comodità compatto il lato conosciuto in un vettore $\boldsymbol{B}^j$ le cui componenti sono:

$$b_i^j = ak_i F_{i-1}^{j} + dk_i F_i^{j} + ck_i F_{i+1}^{j} + ek_i - e_i \tag{2.2.4}$$

$$\begin{pmatrix} d_0 & c_0 & & & & \\ a_1 & d_1 & c_1 & & & \\ & & ... & & & \\ & & & ... & & \\ & & & a_{N-2} & d_{N-2} & c_{N-2} \\ & & & & a_{N-1} & d_{N-1} \end{pmatrix} \boldsymbol{F}^{j+1} = \boldsymbol{B}^j \tag{2.2.5}$$

A questo punto procedo con il trasformare la matrice nella somma di una matrice identità e di una matrice con valori non nulli solo nelle celle sopra alla diagonale. Svolgo i primi passaggi:

$$\begin{pmatrix} d_0 & c_0 & 0 & ... & 0 \\ a_1 & d_1 & c_1 & ... & 0 \\ & . & . & . & \end{pmatrix} F^{j+1} = \begin{pmatrix} b_0 \\ b_1 \\ ... \end{pmatrix} \to \begin{pmatrix} 1 & \frac{c_0}{d_0} & 0 & ... & 0 \\ a_1 & d_1 & c_1 & ... & 0 \\ & . & . & . & \end{pmatrix} F^{j+1} = \begin{pmatrix} \frac{b_0}{d_0} \\ b_1 \\ ... \end{pmatrix} \tag{2.2.6}$$

Proseguendo, chiamando $h_0 = \frac{c_0}{d_0}$ e $p_0 = \frac{b_0}{d_0}$, moltiplico la prima riga per a_1 e la sottraggo alla seconda, in modo da eliminare a_1 dalla seconda riga:

$$\begin{pmatrix} 1 & h_0 & 0 & ... & 0 \\ a_1 - a_1 & d_1 - a_1 h_0 & c_1 & ... & 0 \\ & . & . & . & \end{pmatrix} F^{j+1} = \begin{pmatrix} p_0 \\ b_1 - a_1 p_0 \\ ... \end{pmatrix} \to \begin{pmatrix} 1 & h_0 & 0 & ... & 0 \\ 0 & 1 & \frac{c_1}{d_1 - a_1 h_0} & ... & 0 \\ & . & . & . & \end{pmatrix} F^{j+1} = \begin{pmatrix} p_0 \\ \frac{b_1 - a_1 p_0}{d_1 - a_1 h_0} \\ ... \end{pmatrix} \tag{2.2.7}$$

A questo punto chiamo $h_1 = \frac{c_1}{d_1 - a_1 h_0}$ e $p_1 = \frac{b_1 - a_1 p_0}{d_1 - a_1 h_0}$ e ripeto il ragionamento precedente sottraendo la seconda riga alla terza:

$$\begin{pmatrix} 1 & h_0 & 0 & ... & 0 \\ 0 & 1 & h_1 & ... & 0 \\ 0 & a_3 - a_3 & d_3 - a_3 h_1 & c_3 & ... \\ & . & . & . & \end{pmatrix} F^{j+1} = \begin{pmatrix} p_0 \\ p_1 \\ b_3 - a_3 p_1 \\ ... \end{pmatrix} \to \begin{pmatrix} 1 & h_0 & 0 & ... & 0 \\ 0 & 1 & h_1 & ... & 0 \\ 0 & 0 & 1 & \frac{c_3}{d_3 - a_3 h_1} & ... \\ & . & . & . & \end{pmatrix} F^{j+1} = \begin{pmatrix} p_0 \\ p_1 \\ \frac{b_3 - a_3 p_1}{d_3 - a_3 h_1} \\ ... \end{pmatrix} \tag{2.2.8}$$

A questo punto chiamo $h_3 = \frac{c_3}{d_3 - a_3 h_1}$ e $p_3 = \frac{b_3 - a_3 p_1}{d_3 - a_3 h_1}$ e proseguo, ottengo così le regole:

$$h_i = \frac{c_i}{d_i - a_i h_{i-1}} \tag{2.2.9}$$

e

$$p_i = \frac{b_i - a_i p_{i-1}}{d_i - a_i h_{i-1}} \tag{2.2.10}$$

A questo punto ho semplificato il sistema:

(2.2.11)

$$\begin{pmatrix} 1 & h_0 & & & & \\ & 1 & h_1 & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & 1 & h_{N-2} \\ & & & & & 1 \end{pmatrix} \boldsymbol{F}^{j+1} = \boldsymbol{P} \tag{2.2.12}$$

Per risolvere il sistema devo calcolare il vettore delle $\boldsymbol{P}$ per poi ottenere i valori di F^{j+1} a partire dall'ultimo $F_{N-1}^{j+1} = p_{N-1}$ con la formula:

$$F_i^{j+1} = p_i + h_i F_{i+1}^{j+1} \tag{2.2.13}$$

A questo punto ho bisogno di conoscere come trattare le condizioni al contorno.

3 Condizioni al contorno

In seguito espongo come è possibile adottare alcune condizioni al contorno:

- Dirichlet: Conosco i valori della funzione negli estremi del dominio
- Neumann: Conosco i valori della derivata della funzione negli estremi del dominio
- Robin: Conosco una combinazione lineare tra il valore della funzione e la sua derivata negli estremi del dominio
- Miste: Negli estremi ho tipi differenti di condizioni al contorno

3.1 Dirichlet

Conosco il valore della funzione negli estremi del dominio.

$$F(x,t) = f(x,t) \forall x \in \partial D \tag{3.1.1}$$

Assegno a F_0^{j+1} e F_{N-1}^{j+1} il valore noto, e‘ quindi inutile calcolare la prima e l'ultima riga della matrice $N \times N$ e posso trattare tutto come se la matrice fosse $N-2 \times N-2$, con indici da 1 a $N-2$. Per tenere conto delle condizioni, senza dover apportare modifiche all'algoritmo devo cambiare i valori:

$$\begin{array}{llll|llll} a'_0 = 0 & ak'_0 = 0 & & & a'_1 = 0 & & ak'_1 = ak_1 \\ d'_0 = 1 & dk'_0 = 0 & & & d'_1 = d_1 & & dk'_1 = dk_1 \\ c'_0 = 0 & ck'_0 = 0 & & & c'_1 = c_1 & & ck'_1 = ck_1 \\ e'_0 = -F_0^{j+1} & ek'_0 = 0 & & & e'_1 = e_1 + a_1 F_0^{j+1} & & ek'_1 = ek_1 \end{array} \tag{3.1.2}$$

Che equivale a scrivere:

$$\begin{array}{lll} b'_0 = F_0^{j+1} & h'_0 = 0 & p'_0 = F_0^{j+1} \\ b'_1 = b_1 - a_1 F_0^{j+1} & h'_1 = \frac{c_1}{d'_1} & p'_1 = \frac{b'_1}{d'_1} \end{array} \tag{3.1.3}$$

Di conseguenza $F_1^{j+1} = p'_1 + h'_1 F_2^{j+1}$ e $F_0^{j+1} = p'_0 + h'_0 F_1^{j+1} = F_0^{j+1}$.

Mentre se la condizione si presenta come l'ultimo punto del dominio:

$$\begin{array}{llll|llll} a'N-2 = a_{N-2} & & ak'_{N-2} = ak_{N-2} & & a'_{N-1} = 0 & & ak'_{N-1} = 0 \\ d'_{N-2} = d_{N-2} & & dk'_{N-2} = dk_{N-2} & & d'_{N-1} = 1 & & dk'_{N-1} = 0 \\ c'_{N-2} = 0 & & ck'_{N-2} = ck'_{N-2} & & c'_{N-1} = 0 & & ck'_{N-1} = 0 \\ e'_{N-2} = e_{N-2} + c_{N-2} F_{N-1}^{j+1} & & ek'_{N-2} = ek_{N-2} & & e'_{N-1} = -F_{N-1}^{j+1} & & ek'_{N-1} = 0 \end{array} \tag{3.1.4}$$

Che posso riscrivere:

$$\begin{array}{lll} b'_{N-2} = b_{N-2} - c_{N-2} F_{N-1}^{j+1} & h'_{N-2} = 0 & p'_{N-2} = p_{N-2} \\ b'_{N-1} = F_{N-1}^{j+1} & h'_{N-1} = 0 & p'_{N-1} = F_0^{j+1} \end{array} \tag{3.1.5}$$

e di conseguenza $F_{N-1}^{j+1} = p'_{N-1} = F_{N-1}^{j+1}$ e $F_{N-2}^{j+1} = p'_{N-2} + h'_{N-2} F_{N-1}^{j+1} = p'_{N-2}$.

3.2 Neumann

Conosco il valore della derivata negli estremi del dominio.

$$\frac{\partial F}{\partial x}(x,t) = f(x,t) \forall x \in \partial D \tag{3.2.1}$$

La spiegazione e l'esempio per questa risoluzione lo fornisco nel paragrafo dedicato a Robin.

3.3 Robin

Conosco una combinazione lineare tra la derivata e il valore della funzione negli estremi del dominio.

$$\frac{\partial F}{\partial x}(x,t) + r(x,t)F(x,t) = g(x,t) \forall x \in \partial D \tag{3.3.1}$$

Come prima, per mantenere la precisione del metodo (Δx^2) non posso usare la definizione centrale, ho quindi bisogno di inventarmi un "nodo fantasma" F_{-1}^{j+1} (o F_N^{j+1} se fosse la condizione nell'ultimo punto del dominio):

$$\frac{\partial F}{\partial x}(x(i=0), t(j=j+1)) = \frac{F_1^{j+1} - F_{-1}^{j+1}}{2\Delta x} \tag{3.3.2}$$

Sostituisco nella (3.3.1) per $i = 0$ e al tempo geerico $j = n$:

$$\frac{F_1^n - F_{-1}^n}{2\Delta x} + R_0^n F_0^n = g_0^n \rightarrow F_{-1}^n = F_1^n + 2\Delta x \left(R_0^n F_0^n - g_0^n\right) \tag{3.3.3}$$

Partendo dalla forma matriciale del problema generico:

$$a_0 F_{-1}^{j+1} + d_0 F_0^{j+1} + c_0 F_1^{j+1} + e_0 = ak_0 F_{-1}^j + dk_0 F_0^j + ck_0 F_1^j + ek_0 \tag{3.3.4}$$

e sostituendo F_{-1}^n:

$$a_0 \left(F_1^{j+1} + 2\Delta x \left(R_0^{j+1} F_0^{j+1} - g_0^{j+1}\right)\right) + d_0 F_0^{j+1} + c_0 F_1^{j+1} = ak_0 \left(F_1^j + 2\Delta x \left(R_0^j F_0^j - g_0^j\right)\right) + dk_0 F_0^j + ck_0 F_1^j \tag{3.3.5}$$

A questo punto raccolgo i termini dello stesso punto della funzione:

$$\left(d_0 + 2a_0 R_0^{j+1}\Delta x\right) F_0^{j+1} + (c_0 + a_0) F_1^{j+1} - 2a_0 g_0^{j+1}\Delta x = \left(dk_0 + 2ak_0 R_0^j \Delta x\right) F_0^j + (ck_0 + ak_0) F_1^j - 2ak_0 g_0^j \Delta x \tag{3.3.6}$$

E quindi i parametri interessati della matrice diventano:

$$\begin{array}{ll} a_0' = 0 & ak_0' = 0 \\ d_0' = d_0 + 2a_0 R_0^{j+1}\Delta x & dk_0' = dk_0 + 2ak_0 R_0^j \Delta x \\ c_0' = a_0 + c_0 & ck_0' = ak_0 + ck_0 \\ e_0' = e_0 - 2a_0 g_0^{j+1}\Delta x & ek_0' = ek_0 - 2ak_0 g_0^j \Delta x \end{array} \tag{3.3.7}$$

Che equivale a scrivere:

$$\begin{array}{l} b_0' = dk_0' F_0^j + ck_0' F_1^j + 2\Delta x \left(a_0 g_0^{j+1} - ak_0 g_0^j\right) + ek_0 - e_0 \\ h_0' = \frac{c_0'}{d_0'} \\ p_0' = \frac{b_0'}{d_0'} \end{array} \tag{3.3.8}$$

e di conseguenza $F_0^{j+1} = p_0' + h_0' F_1^{j+1}$.

Mentre se la condizione si presenta come ultimo punto del dominio:

$$\frac{F_N^n - F_{N-2}^n}{2\Delta x} + R_{N-1}^n F_{N-1}^n = g_{N-1}^n \rightarrow F_N^n = F_{N-2}^n - 2\Delta x \left(R_{N-1}^n F_{N-1}^n - g_{N-1}^n\right) \tag{3.3.9}$$

Salto i passaggi, molto simili a quelli della spiegazione precedente, per il calcolo di b_{N-1}' andranno usati i seguenti parametri:

$$\begin{array}{ll} a_{N-1}' = a_{N-1} + c_{N-1} & ak_{N-1}' = ak_{N-1} + ck_{N-1} \\ d_{N-1}' = d_{N-1} - 2c_{N-1} R_{N-1}^{j+1}\Delta x & dk_{N-1}' = dk_{N-1} - 2ck_{N-1} R_{N-1}^j \Delta x \\ c_{N-1}' = 0 & ck_{N-1}' = 0 \\ e_{N-1}' = e_{N-1} + c_{N-1} g_{N-1}^{j+1}\Delta x & ek_{N-1}' = ek_{N-1} + ck_{N-1} g_{N-1}^{j+1}\Delta x \end{array} \tag{3.3.10}$$

Che equivale a scrivere:

$$\begin{aligned} b_{N-1} &= ak'_{N-1}F^j_{N-2} + dk'_{N-1}F^j_{N-1} - 2\Delta x\left(c_{N-1}g^{j+1}_{N-1} - ck_{N-1}g^j_{N-1}\right) + ek_{N-1} - e_{N-1} \\ h'_{N-1} &= 0 \\ p'_{N-1} &= \frac{b'_{N-1}+a'_{N-1}p_{N-2}}{d'_{N-1}-a'_{N-1}h_{N-2}} \end{aligned} \tag{3.3.11}$$

e di conseguenza $F^{j+1}_{N-1} = p'_{N-1}$, anche perchè è il primo punto da cui si parte per calcolare il valore della funzione in $j+1$.

Se faccio in modo di eliminare il coefficiente che moltiplica il valore della funzione (gli R) ottengo le condizioni a contorno di Neuman.

3.4 Osservazione

Il modo in cui ho trattato i parametri per quanto riguarda le condizioni al contorno di Dirichlet nei punti 0 e $N-1$, non è matematicamente corretto infatti i parametri andrebbero messi tutti a 0 in quanto quei punti non fanno parte dell'algoritmo.

Ho impostato i valori per avere un algoritmo che possa svolgere il calcolo rispettando le condizioni al contorno senza sapere quali siano, mettendo nelle mani dell'utente, che si occuperà di impostare i corretti parametri della matrice, la gestione delle condizioni.

4 Applicazioni

4.1 Equazione del calore

Riprendiamo l'equazione del calore:

$$\frac{\partial T}{\partial t}(x,t) = k\frac{\partial^2}{\partial x^2}T(x,t) \tag{4.1.1}$$

Per rispettare la convenzione che ho scelto $D_2 = k$, $D_1 = U = V(x,t) = 0$.
A questo punto posso sostituire, con $\eta = k\frac{\Delta t}{\Delta x^2}$:

$$\begin{array}{ll} a_i^j = -1 & ak_i^j = 1 \\ d_i^j = \frac{2}{\eta} + 2 & dk_i^j = \frac{2}{\eta} - 2 \\ c_i^j = -1 & ck_i^j = 1 \\ e_i^j = 0 & ek_i^j = 0 \end{array} \tag{4.1.2}$$

sostituire i parametri nella matrice, scegliere le condizioni al contorno e procedere con i calcoli.

4.2 Equazione di Schrödinger

Lavoriamo con l'equazione di Schrödinger dipendente dal tempo:

$$i\hbar\frac{\partial \psi}{\partial t}(x,t) = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x,t)\right]\psi(x,t) \tag{4.2.1}$$

prima di tutto portiamola in una forma tale che non ci sia nulla a moltiplicare la derivata temporale:

$$\frac{\partial \psi}{\partial t}(x,t) = \left[i\frac{\hbar}{2m}\frac{\partial^2}{\partial x^2} + \frac{\Lambda(x,t)}{i\hbar}\right]\psi(x,t) \tag{4.2.2}$$

Ho usato Λ per indicare il potenziale per rispettare la convenzione. $D_2 = i\frac{\hbar}{2m}$, $D_1 = U = 0$ e $V(x,t) = \frac{\Lambda(x,t)}{i\hbar}$.
A questo punto, con $\eta = i\frac{\hbar}{2m}\frac{\Delta t}{\Delta x^2}$, posso sostituire, usando k come indice spaziale:

$$\begin{array}{ll} a_k^j = -1 & ak_k^j = 1 \\ d_k^j = \frac{1}{\eta}\left(2 - \Delta t V_k^{j+1}\right) + 2 & dk_k^j = \frac{1}{\eta}\left(2 + \Delta t V_k^j\right) - 2 \\ c_k^j = -1 & ck_k^j = 1 \\ e_k^j = 0 & ek_k^j = 0 \end{array} \tag{4.2.3}$$

subsectionStabilità Per mostrare che lo schema proposto è stabile utilizzo il principio di Von Neumann: perché un metodo si riveli stabile non deve propagare eventuali errori che nascono dal calcolo. Definiamo l'errore come:

$$E_k^j = F_k^j - u_k^j \tag{4.2.4}$$

Dove ho usato k invece che i per non creare confusione con l'unità immaginaria. F_k^j è il valore della soluzione calcolata con l'algoritmo mentre u_k^j è il valore reale della funzione nel punto (k, j).

Si definisce il fattore crescita:

$$G = \left| \frac{E_k^{j+1}}{E_k^j} \right| \tag{4.2.5}$$

che serve come criterio per comprendere la stabilità dello schema. Come vedremo più avanti è comodo studiare G in funzione delle frequenze spaziali. Per ogni frequenza:

- $G < 1$: l'algoritmo è stabile e gli errori vengono attenuati passo per passo.
- $G > 1$: l'algoritmo non è stabile e gli errori vengono amplificati passo per passo.
- $G = 1$: gli errori non vengono nè amplificati nè ridotti dall'evoluzione, è stabile.

se $G < 1$ per tutte le frequenze allora il l'algoritmo si considera incondizionatamente stabile.

Per fare l'analisi in frequenza faccio la trasformata di Fourier spaziale dell'errore della funzione:

$$E_k^j = \sum_\omega \hat{\epsilon}_\omega^j e^{i\omega x} \tag{4.2.6}$$

Della trasformata prendiamo un solo termine per una data ω. Mi aspetto che la propagazione dell'errore soddisfi la stessa equazione che sto simulando e quindi procedo sostituendo l'espressione nello schema , a partire da (2.2.2), e ignorando per semplicità i termini e e ek ottengo:

$$\left(a_k e^{i\omega(k-1)\Delta x} + d_k e^{i\omega k\Delta x} + c_k e^{i\omega(k+1)\Delta x}\right) \hat{\epsilon}_\omega^{j+1} = \left(ak_i e^{i\omega(k-1)\Delta x} + dk_i e^{i\omega k\Delta x} + ck_k e^{i\omega(k+1)\Delta x}\right) \hat{\epsilon}_\omega^j \tag{4.2.7}$$

abbiamo:

$$\frac{\hat{\epsilon}^{j+1}}{\hat{\epsilon}^j} = \frac{ak_i e^{-i\omega\Delta x} + dk_i + ck_k e^{i\omega\Delta x}}{a_k e^{-i\omega\Delta x} + d_k + c_k e^{i\omega\Delta x}} \tag{4.2.8}$$

dalla (2.1.5) deduco:

$$\begin{aligned} a_k^j &= -ak_k^j \\ c_k^j &= -ck_k^j \\ c_k^j &= -a_k^j - 2 \end{aligned}$$

e sostituisco nell'uguaglianza:

$$\frac{\hat{\epsilon}_\omega^{j+1}}{\hat{\epsilon}_\omega^j} = -\left(1 - \frac{d + dk}{d - 2\cos(\omega\Delta x) - 2i(1+a)\sin(\omega\Delta x)}\right) \tag{4.2.9}$$

da cui ricavo G:

$$G = \left|1 - \frac{d + dk}{d - 2\cos(\omega\Delta x) - 2i(1+a)\sin(\omega\Delta x)}\right| \tag{4.2.10}$$

Non posso più essere generale. Per fare il valore assoluto devo conoscere se a, d e dk sono reali o complessi. Per esempio, sostituendo i valori che trovo in (4.1.2) per l'equazione del calore, dopo le sostituzioni e varie semplificazioni trovo che:

$$G = \frac{\left|\Delta x^2 - \Delta t k(1 - \cos(\omega\Delta t))\right|}{\Delta x^2 + \Delta t k(1 - \cos(\omega\Delta t))} \tag{4.2.11}$$

In cui si vede che $G < 1$ per tutte le frequenze diverse dai multipli di $\omega = 2\pi/\Delta x$, in quei casi $G = 1$ e quindi tenderanno a rimanere delle oscillazioni nella soluzione.

Mentre usando i valori di (4.2.3) che uso per la risoluzione dell'equazione di Schrödinger , utilizzando le stesse notazioni ottengo:

$$G = \left| \frac{\left(\Delta x^2 \Lambda m - \hbar^2 \left(1 - \cos\left(\omega \Delta x\right)\right)\right) \Delta t + 2i \Delta x^2 \hbar m}{\left(\Delta x^2 \Lambda m - \hbar^2 \left(1 - \cos\left(\omega \Delta x\right)\right)\right) \Delta t - 2i \Delta x^2 \hbar m} \right| \tag{4.2.12}$$

Si nota facilmente che $G = 1$ per tutte le frequenze; vuol dire l'algoritmo propagherà gli errori, ma senza amplificarli.

Parte II
Simulazione

5 Il programma: descrizione e scelta dei parametri

Per semplicità e alleggerire i calcoli ho impostato $\hbar = 1$, Gli eseguibili compilabili dal Makefile sono i seguenti, nei commenti è spiegato come eseguirli:

```
#il progama principale
single: MainC.cpp TridiagC.o CrankSolverC.o impostazioniC.o experimentC.o
	@echo Compilo $@
	@$(CC11) $(CFLAGS) -o $@ $^  -DUSECOMPLEX
#funziona come single, ma ripete l'esperimento per una lista di potenziali
experiment: Experiments.cpp TridiagC.o CrankSolverC.o impostazioniC.o experimentC.o
	@echo Compilo $@
	@$(CC11) $(CFLAGS) -o $@ $^  -DUSECOMPLEX
#disegna soluzione, errore e pesi dell'integrale del file di dati in argomento
drawer:drawer.cpp preparedraw.o
	@echo Compilo $@
	@$(CC11) $(CFLAGS) -o $@ $^ $(LIBROOT) $(CFLAGSROOT)
#controlla i .dat letti in namelist.txt, varie opzioni con --help
analisi:analisi.cpp preparedraw.o
	@echo Compilo $@
	@$(CC11) $(CFLAGS) -o $@ $^ $(LIBROOT) $(CFLAGSROOT)
#visualizza vari potenziali in diversi files
CVD: CVDrawer.cpp impostazioniC.o TridiagC.o
	@echo Compilo $@
	@$(CC11) $(CFLAGS) -o $@ $^  -DUSECOMPLEX $(LIBROOT) $(CFLAGSROOT)
#una interfaccia grafica per osservare un esperimento alla volta
preview: preview.cpp visual.o libVisual.so
	@echo Compilo $@
	@$(CC11) $(CFLAGS) -o $@ -DSTANDALONE $^ $(LIBROOT) $(CFLAGSROOT)
```

- Il programma più semplice (*single*) è composto da un blocco *impostazioni* che carica i file con le informazioni sulla simulazione, crea la trimatrice e le condizioni iniziali ,e un blocco *CrankSolver* che si occupa di svolgere risolvere il sistema passo per passo (nel nostro caso non avendo potenziali che dipendono dal tempo la riduzione della trimatrice viene fatta solo una volta, nella costruzione del *CrankSolver*). Ho scelto di impostare il *CrankSolver* in modo che contenga solo i punti del passo attuale perché l'elevato numero di punti rischia di intasare la memoria del sistema piuttosto in fretta: i punti calcolati vengono salvati su un file in una cartella "./results", che deve essere creata prima di lanciare l'eseguibile, il numero di punti e la frequenza di essi viene deciso nei file di impostazione.

- *experiment* funziona come *single* ma ripete la simulazione per ogni potenziale che trova nella lista *namelist.txt* nella cartella in cui viene chiamato.

- *drawer*, *analisi*, *CVD* e *preview* sono i programmi che vengono utilizzati nell'analisi delle simulazioni:

  - *analisi* carica i file *.dat* contenuti in *namelist.txt* e ne fa varie analisi (plot degli errori, plot degli integrali nella prima/seconda metà del dominio, plot della posizione dei massimi).
  - *drawer* fa le stesse operazioni di analisi, ma relative a un singolo file *.dat*, inoltre mostra l'evoluzione dell'onda grazie a un *TGraph2D*.
  - *CVD* mostra i vari potenziali il cui elenco è contenuto nel file *infonamelist.txt*, carica le impostazioni del dominio da un file *settings.set* dato.
  - *preview* deve essere lanciato con *./runscript.sh* (che imposta **export** LD_LIBRARY_PATH=$LD_LIBRARY_PATH è una piccola interfaccia grafica che permette di fare qualche analisi sui risultati della simulazione.

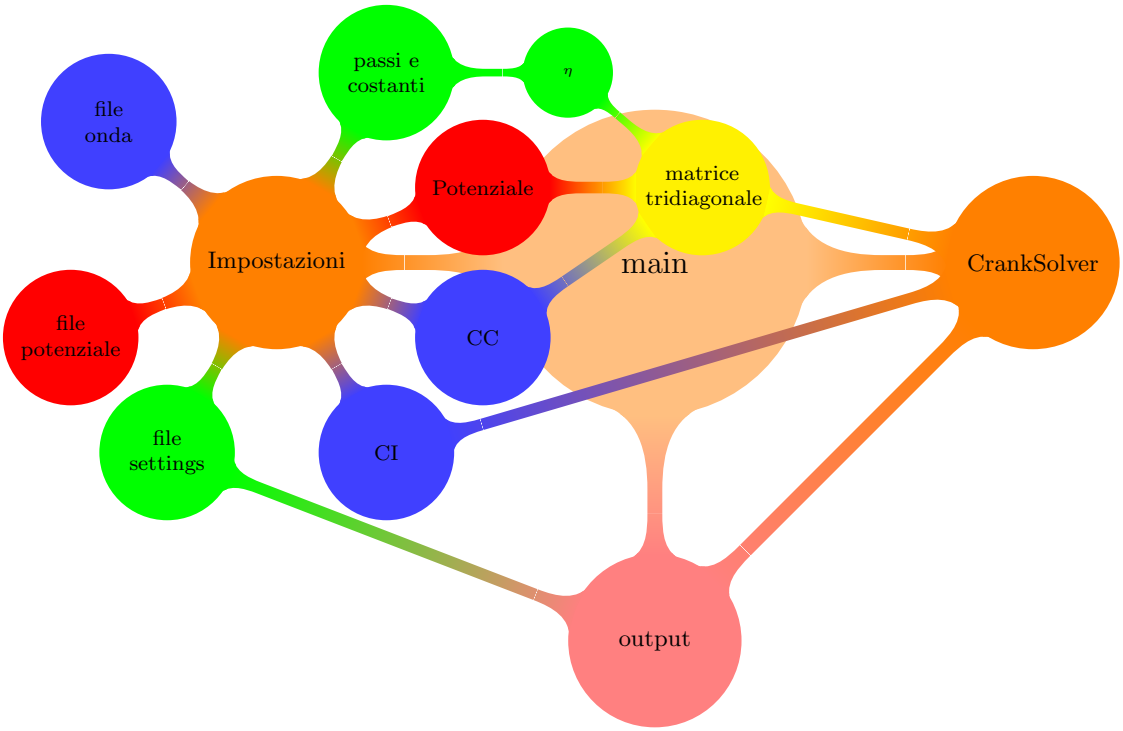


Figura 1: Collegamenti tra i blocchi del programma

5.1 Impostazioni

Il programma più semplice, compilabile con $> make single carica i dati da tre file di impostazioni, in cui viene alternata una riga di commento, anticipata da un "#" e una con i valori dell'impostazione, l'ordine è importante e il commento non deve contenere spazi.

Il file generale delle impostazioni:

Listing 1: Impostazioni principali

```
#massa
1
#lunghezza
60
#tmax
10
#Ndipende_da_intervallo=0
0
#Nl
10000
#spacestep.
0.0015
#Nt
1000
#timestep
0.001
#timeskip
100
#spaceskip
1
#precisione
5e-3
```

In questo file fornisco i parametri principali della simulazione, quelli che ho previsto che avrei cambiato più di rado durante la simulazione. La massa, la lunghezza sono ovvi, tmax è il tempo a cui si ferma la simulazione. L'opzione "*#Ndipende_da_intervallo=0*" se impostata uguale a 0 fa sì che il numero di passi (spaziali e temporali) venga calcolato conoscendo la lunghezza totale e il passo, se impostata diversa da 0 invece fa sì che il programma calcoli il passo a partire dalla lunghezza totale e il numero di punti voluti. Le opzioni "*#timeskip*" e "*#spaceskip*" servono nel momento di salvare i risultati: il primo imposta ogni quanti passi temporali esportare il vettore di punti, che vengono salvati ad intervalli decisi dal secondo parametro. Il parametro "*#precisione*" invece blocca la simulazione prima del tempo massimo se l'errore (vedi sottosezione 5.3) supera il valore impostato, se lo si imposta a 0 salta il controllo.

Il file delle condizioni iniziali:

Listing 2: Impostazioni CI

```
#Bump_Gauss
G
#impostazioni_onda_norm_stdev_midpoint
1 2 5
#impostazionicc0_012DNR_val_pesorobin
2 0 0
#impostazioniccN_012DNR_val_pesorobin
2 0 0
#Energia
10
```

Il primo parametro deve essere una lettera e serve a indicare al programma la forma del pacchetto d'onda nelle condizioni iniziali, se viene scritto "*b*" il programma disegnerà la funzione "bump" con parametri presi dalla riga successiva, nell'ordine "altezza del massimo", "larghezza" e "punto del massimo"; negli altri casi una gaussiana con i parametri "altezza", "deviazione standard" e "punto medio". I due set successivi alle impostazioni dell'onda riguardano le condizioni al contorno, la prima che può essere 0, 1 o 2 indica il tipo di condizione, rispettivamente Dirichlet, Neumann o Robin sulla stessa riga il primo numero indica il valore della funzione, della derivata o della combinazione lineare, il terzo parametro viene solo utilizzata da Robin ed è il peso della funzione. L'ultima impostazione è l'energia che si vuole impostare per l'onda.

Il file del potenziale:

Listing 3: Impostazioni potenziale

```
#tipo_Gauss_Muro_Salto
M
#impostazioni_Val_Pos_Vpar
10 30 4
```

Il primo parametro imposta la forma del potenziale, che può essere una gaussiana, un "bump", un rettagolo oppure un salto. Nella riga successiva il primo parametro imposta l'altezza del punto più alto del potenziale, se messo a 0 dice all'algoritmo di calcolare l'evoluzione in assenza di potenziale, il secondo parametro indica il centro della funzione se è pari, oppure la posizione del salto, l'ultimo parametro indica la larghezza o la deviazione standard del potenziale non viene utilizzato se la forma è il salto.

5.2 Condizioni iniziali

La mia idea è di simulare l'evoluzione di un pacchetto di onde piane quindi nella forma

$$\Psi(x,t) = \psi(x,t)e^{ik_0x-\omega t} \rightarrow \Psi(x,0) = \psi(x,0)e^{ik_0x} \tag{5.2.1}$$

O, più correttamente:

$$\Psi(x,t) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} A(k)e^{ikx-\omega t}\mathrm{d}k \tag{5.2.2}$$

Dato che utilizzerò una forma gaussiana per il pacchetto, nel programma non farò mai l'integrale in quanto la trasformata di Fourier di una gaussiana è una gaussiana. Ovvero avrò:

$$\Psi(x,0) = Ae^{-\frac{(x-x_0)^2}{2\sigma}} e^{ikx} \tag{5.2.3}$$

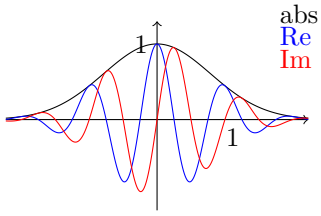


Figura 2: Le condizioni iniziali, con $x_0 = 0,\ A = 1,\ \sigma = 1$ e $k = 7$

Dove x_0 è il punto del massimo dell'onda, σ la sua deviazione standard e A un parametro che cambia l'altezza del massimo. La velocità di gruppo k dipende dall'energia secondo la relazione di dispersione $k(E) = \sqrt{\frac{2m}{\hbar^2}E}$, tenendo conto che nelle condizioni iniziali le particelle del treno sono da considerarsi libere (ovvero non sono influenzate dal potenziale nel punto in cui le ho depositate).

Poiché la forma dell'onda è moltiplicata per e^{ikx} e deve essere discretizzata devo come minimo scegliere un passo che non cada in problemi di aliasing, e quindi campionare con una frequenza superiore a k, ovvero il passo deve essere più piccolo di $1/k$.

La discretizzazione delle condizioni iniziali, gaussiane:

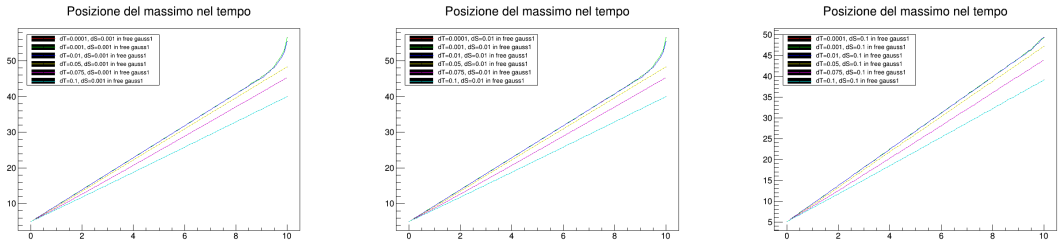
$$CI\,[j] = Ae^{-\frac{(j*\mathrm{d}x-x_0)^2}{2\sigma}+ikj*\mathrm{d}x} \tag{5.2.4}$$

5.3 Lancio senza potenziale: errore

Prima di tutto voglio effettuare qualche simulazione senza potenziale per poter osservare come si comporta l'onda.

Ho bisogno di definire un "errore" per capire quanto la simulazione possa essere andata a buon fine: la mia scelta è stata di utilizzare la norma della funzione, ricordando che le funzioni d'onda sono L^2 e quindi la loro norma è $|f|^2 = \int_X |f|^2\, d\mu$. Ho utilizzato la definizione dell'integrale discreto di Simpson, il che mi limita a poter fare solo simulazioni con un numero di passi pari (e in particolare multipli di 4 per aver un integrale preciso anche quando calcolo l'integrale solo nella prima o seconda metà del dominio) nello spazio.

Nella simulazione blocco il calcolo se la norma della funzione si discosta troppo dalla norma delle condizioni iniziali, il limite viene scelto nel file delle impostazioni.



(a) Le differenze per $dS = 0.001$ (b) Le differenze per $dS = 0.01$ (c) Le differenze per $dS = 0.1$

Figura 3: Le posizioni dei massimi nel tempo della simulazione a vari passi temporali e spaziali

Per capire come impostare i parametri delle simulazioni ho effettuato diversi lanci con varie combinazioni di passo temporale e passo spaziale. Riporto i vari grafici degli errori. Da questi lanci capisco che devo scartare l'idea di usare un passo temporale grande in quanto ho notato che le funzioni d'onda vengono "rallentate", come si può vedere in Figura 3. Dalle figure noto che le simulazioni devono

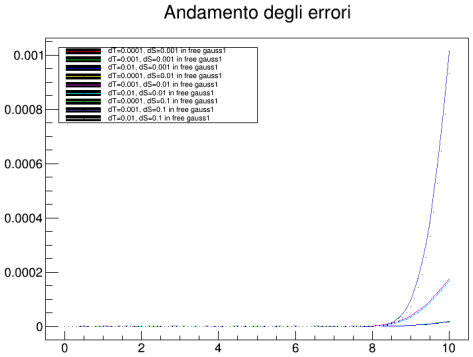
avere come massimo passo temporale 0.01. Il punto in cui gli andamenti perdono l'andamento lineare e' dovuto al fatto che il pacchetto sta impattando contro il bordo del dominio e quindi inizia a rimbalzare. Mi aspetto che la simulazione ritorni:

$$\Psi(x,t) = Ae^{-\frac{(x-x_0-\omega t)^2}{2\sigma}} e^{ik(x-\omega t)} \tag{5.3.1}$$

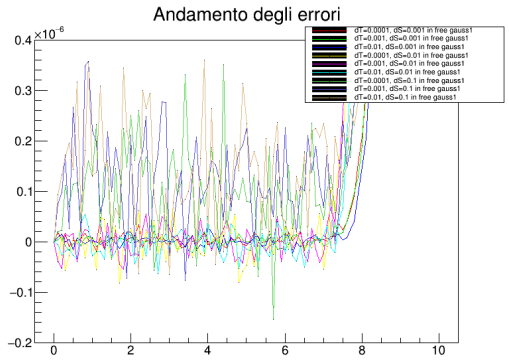
e quindi che i massimi delle onde da me simulate si spostino nel tempo seguendo la legge:

$$x_0(t) = x_0(0) + \omega t \tag{5.3.2}$$

Poi ho quindi guardato l'errore di alcune simulazioni, scartando a priori i passi più grandi di 0.01, Figura 4a e 4b.



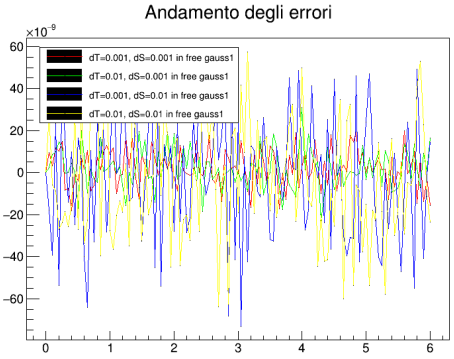
(a) Il grafico completo degli errori



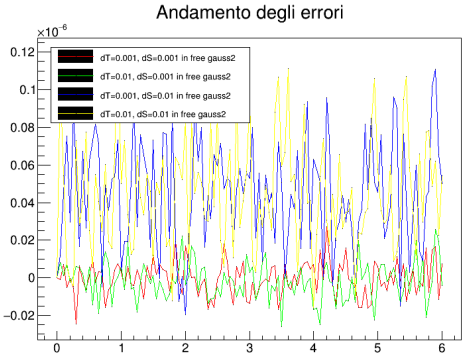
(b) Zoom per mostrare dove la riflessione non ha influenza

Figura 4: Il grafico degli errori, si vede anche quanto la riflessione influisca sul modulo della funzione.

A questo punto ho fatto una simulazione diminuendo il tempo massimo a 6 in modo da non avere a che fare con l'esplosione dell'errore dovuta al limite del dominio. e cosi' da poter osservare quale fosse il miglior passo spaziale. Ho fatto i lanci con due diverse condizioni iniziali (gaussiana con σ 1 e 2).



(a) CI: gaussiana($\sigma = 2$)



(b) CI: gaussiana($\sigma = 2$)

Figura 5: I passi che mi sembravano più opportuni a confronto

Osservando Figura 5 utilizzare come passi spaziali e temporali valori compresi tra 0.01 e 0.001 ha un buon rapporto "errore sulla simulazione/peso sul calcolatore", in particolare si nota che il passo spaziale influenza di più l'errore. Ho scelto di continuare con $dT = 0.01$ e $dS = 0.005$ in quanto hanno errore molto simile a $dS = 0.001$ (Figura 6) ma sono più rapidi nei calcoli (sulla mia macchina).

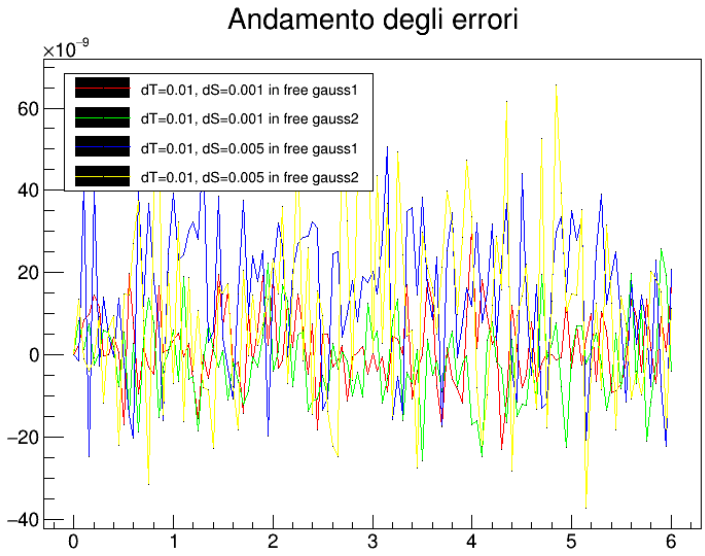


Figura 6: Gli errori dei passi che ritengo più convenienti

5.4 La forma delle condizioni iniziali

I test sull'errore li ho effettuati utilizzando come condizione iniziale un pacchetto d'onda gaussiano. Per effetto della dispersione è un pacchetto deltiforme che nel tempo si è disperso allargandosi in una gaussiana.

Ho provato ad utilizzare una forma differente per il pacchetto: la funzione "bump":

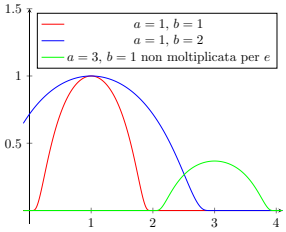


Figura 7: Alcuni esempi della funzione bump

$$bump = (x) = \begin{cases} e^{-\frac{b^2}{b^2-(x-a)^2}} & \text{per } |x-a| < b \\ 0 & \text{altrove} \end{cases} \tag{5.4.1}$$

Che è una funzione di test quindi è definita differente da 0 solo in un intervallo aperto, ed ha la caratteristica di avere tutte le derivate continue, questo mi permetterebbe di ignorare completamente le condizioni al contorno quando la funzione si trova lontano dai bordi della simulazione. Inoltre moltiplico il risultato per e per avere un pacchetto con il massimo alto 1, vedi Figura 7.

A questo punto simulo il pacchetto e confronto l'errore con uno gaussiano. Come si può vedere dalla Figura 8 non è una buona idea.

L'errore cresce molto verso la fine della simulazione, questo potrebbe essere dovuto al fatto che sta impattando contro i limiti del dominio, inoltre se si osserva l'evoluzione della funzione si vede che perde praticamente subito la forma iniziale e quindi anche i vantaggi di avere le code pari a 0.

A questo punto l'idea migliore per il pacchetto d'onda sembra essere la gaussiana.

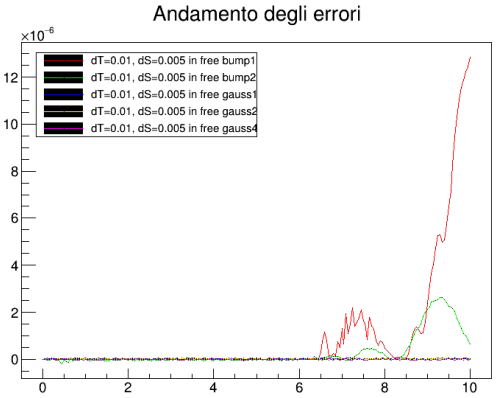
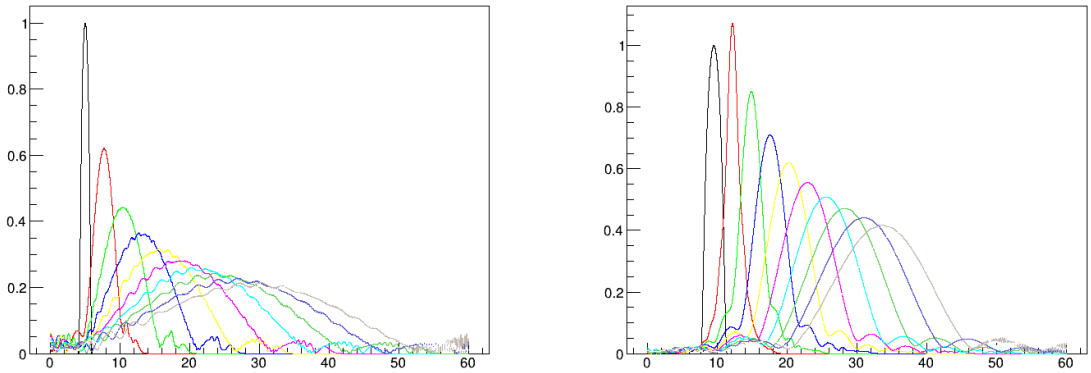


Figura 8: l'evoluzione dell'errore del "bump" confrontata con quella di un pacchetto gaussiano



(a) larghezza 1

(b) larghezza 2

Figura 9: L'evoluzione del "bump", le "increspature" sull'onda che si vedono ai lati sono le interferenze dovute alla riflessione

5.5 Dispersione dei pacchetti d'onda

La prima cosa che si può osservare nelle simulazioni (e in particolare in Figura 10) è la dispersione dell'onda, ovvero il pacchetto si allarga nel tempo. Questa è una caratteristica dell'equazione di Schrödinger , in quanto la sua forma è appunto quella di una qualsiasi equazione che descriva la dispersione.

Voglio calcolare la teoria di questo fenomeno per poterlo confrontare con i risultati che ho ottenuto nelle prime simulazioni. Per ottenere la soluzione dell'equazione con condizioni iniziali $\psi(x,0) = e^{-\frac{x^2}{2\sigma^2}}$ si passa dalla trasformata di Fourier spaziale $\hat{\psi}(x,0) = \sqrt{2\pi\sigma^2}e^{-\frac{k^2\sigma^2}{2}}$, che porta ad avere la funzione, con dipendenza temporale:

$$\hat{\psi}(k,t) = \sqrt{2\pi\sigma^2}e^{-\frac{k^2\sigma^2}{2}}e^{-i\frac{E}{\hbar}t} = \sqrt{2\pi\sigma^2}e^{-\frac{k^2\sigma^2}{2}}e^{-i\frac{\hbar^2k^2/2m}{\hbar}t} = \sqrt{2\pi\sigma^2}e^{-k^2\frac{\left(\sigma^2+i\hbar t/m\right)}{2}} \tag{5.5.1}$$

e poi ritrasformando:

$$\psi(x,t) = \frac{\sigma}{\sqrt{\sigma^2 + i\hbar t/m}}e^{-\frac{k^2}{2(\sigma+i\hbar t/m)}} \tag{5.5.2}$$

ora calcolo il valore assoluto della funzione:

$$|\psi(x,t)| = \frac{\sigma}{\sqrt[4]{\sigma^4 + (\hbar t/m)^2}}e^{-\frac{k^2}{2}\frac{\sigma^2}{\sigma^4+(\hbar t/m)^2}} = \sqrt{\frac{\sigma_0}{\sigma(t)}}e^{-\frac{k^2}{2\sigma^2(t)}} \tag{5.5.3}$$

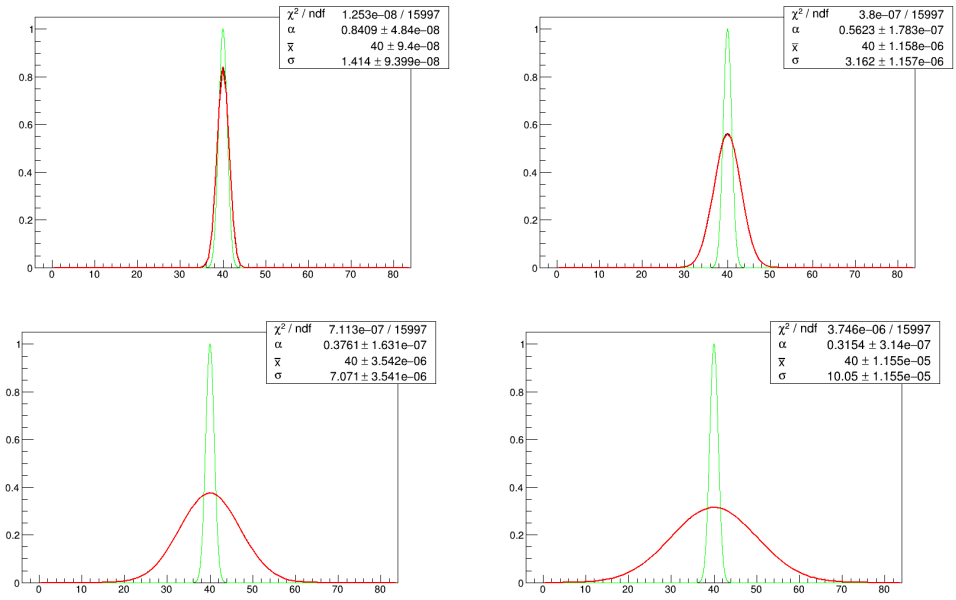
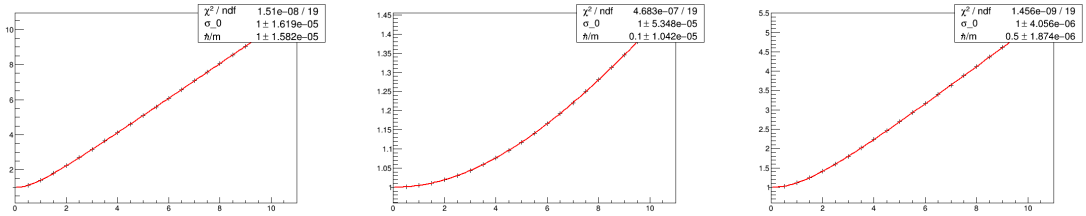


Figura 10: La dispersione delle gaussiana a $\sigma = 1$, in verde le condizioni iniziali

E, chiamando σ_0 il valore assegnato nelle condizioni iniziali, mi aspetto dalla simulazione che σ evolva come

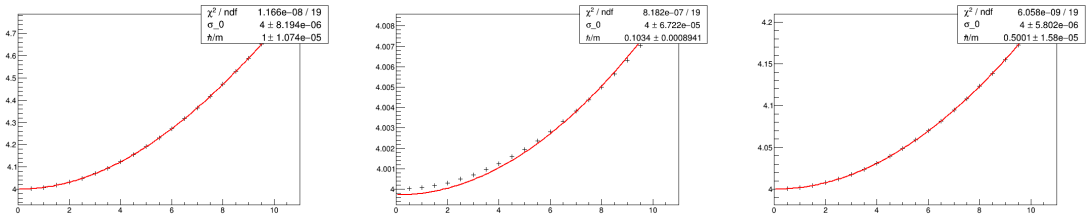
$$\sigma(t) = \sqrt{\frac{\sigma_0^4 + \left(\frac{\hbar}{m}t\right)^2}{\sigma_0^2}} \tag{5.5.4}$$

Ho fatto diversi fit, sul parametro σ, con diverse masse e diversi valori di σ iniziali, utilizzando come funzione da fittare (5.5.4). L'algoritmo sembra rispettare la dispersione, come si può vede re in Figura 11, in Figura 12 e in Figura 13 dove la funzione ha energia diversa da 0



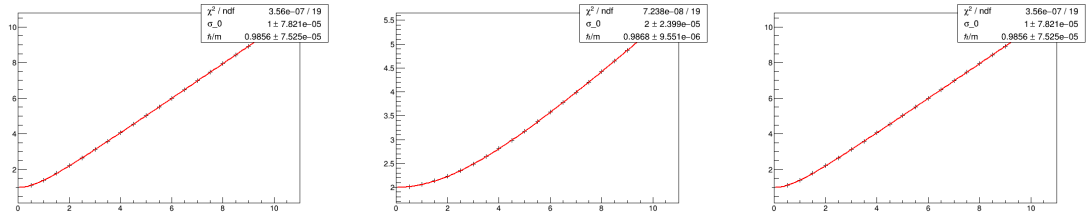
(a) $\sigma_0 = 1$, $m = 1$ (b) $\sigma_0 = 1$, $m = 10$ (c) $\sigma_0 = 1$, $m = 2$

Figura 11: Alcuni fit dell'andamento della dispersione



(a) $\sigma_0 = 4$, $m = 1$ (b) $\sigma_0 = 4$, $m = 10$ (c) $\sigma_0 = 4$, $m = 2$

Figura 12: Alcuni fit dell'andamento della dispersione



(a) $\sigma_0 = 1$, $m = 1$, con $E = 10$ (b) $\sigma_0 = 2$, $m = 1$, con $E = 10$ (c) $\sigma_0 = 4$, $m = 1$, con $E = 10$

Figura 13: Alcuni fit dell'andamento della dispersione

t	α	$\bar{x}$	σ	t	α	$\bar{x}$	σ
0	1	40	1	5.5	0.422955	40	5.59001
0.5	0.94574	40	1.11803	6	0.405467	40	6.08259
1	0.840898	40	1.4142	6.5	0.389951	40	6.57628
1.5	0.744796	40	1.80272	7	0.376066	40	7.07086
2	0.668766	40	2.23596	7.5	0.363549	40	7.56615
2.5	0.609451	40	2.69242	8	0.352191	40	8.06202
3	0.562349	40	3.16219	8.5	0.341825	40	8.55837
3.5	0.524146	40	3.63995	9	0.332317	40	9.05512
4	0.492486	40	4.12299	9.5	0.323555	40	9.55221
4.5	0.465765	40	4.60964	10	0.315447	40	10.0496
5	0.442856	40	5.09887				

Tabella 1: I dati di una simulazione, con $\alpha = 1$, $\sigma = 1$ e $m = 1$

5.6 Condizioni al contorno

Quello che vorrei simulare è un pacchetto d'onda che sia partito da $-\infty$ e arrivi nel "punto interessante", l'area che osservo nella simulazione, e poi prosegua fino all'infinito (o venga eventualmente riflesso).

Per quanto riguarda le condizioni al contorno ho avuto idee:

- Non tengo conto delle condizioni al contorno, faccio in modo di usare pacchetti d'onda ristretti e blocco la simulazione quando questa va ad avvicinarsi troppo ai bordi.
- Uso una forma d'onda tale per cui conosco la funzione e la sua derivata in ogni punto
- Emulo un ambiente chiuso (con ai lati muri di potenziale alti ∞)

La prima ipotesi è stata quella che ho usato in tutte le prove che ho effettuato prima di iniziare a raccogliere dati, considerando 0 il valore della funzione negli estremi.

Per la seconda ipotesi dovrei usare come condizione iniziale una funzione e calcolarne la derivata per ogni passo negli estremi a priori il discorso non si applica ma i paccheti d'onda hanno una dispersione tendono ad allargarsi, per cui se per esempio avessi un pacchetto d'onda generico ($\Psi(x,t) = e^{ikx}\psi(x,t)$)la relazione dovrebbe essere $\partial_x\Psi(x,t) = ik\Psi(x,t) + e^{ikx}\partial_x\psi(x,t)$ e potrei quindi utilizzare le CC di Robin in caso conoscessi come la funzione si disperde nel tempo potrei ricalcolare ad ogni passo le condizioni al contorno ed applicarle. Questo però mi porterebbe a svariati problemi quali l'incertezza che la simulazione e la teoria siano esattamente parallele sulla dispersione

La terza idea equivale a utilizzare Dirichlet con il valore della funzione pari a 0 nei bordi. Devo mostrare che l'errore non cresce per quest'ipotesi

5.7 Prime Conclusioni

Dopo i primi esperimenti sono arrivato a queste conclusioni:

- Farò le simulazioni utilizzando una gaussiana come forma del pacchetto d'onda, perché mantiene la propria forma.
- Per controllare la dispersione dell'onda, vista la forma di (5.5.4) posso scegliere di utilizzare un σ_0 grande oppure una massa grande
  - con una massa grande per avere la stessa velocità di gruppo devo aumentare l'energia, ma in compenso posso utilizzare dei σ_0 piuttosto piccoli e quindi questo mi rende più semplice il compito di fare confronti tra due parti di spazio limitate per analizzare la trasmissività delle barriere di potenziale
  - con σ_0 grande il pacchetto d'onda non ha bisogno di una grande energia per muoversi, ma il fatto che esso sia esteso può comportare problemi per esempio quando si riflette contro un potenziale o quando lo attraversa o nel momento di definire le condizioni iniziali, perché rischio di sovrapporre l'onda al potenziale.
- Come condizioni al contorno simulerò la scatola chiusa con pareti infinite, sebbene una prima idea fosse quella di utilizzare le condizioni "trasparenti", ma questo oltre ad appesantire il programma farebbe uscire il pacchetto d'onda dalla mia "zona d'osservazione" e quindi non sarei più in grado di fare l'integrale di tutto il pacchetto (comprese le sue riflessioni e i pezzi in cui si è diviso).
- Per ritardare il più possibile l'interazione con le condizioni al contorno devo aumentare l'intervallo spaziale se aumento il tempo della simulazione

Con queste conclusioni in mano mi appresto a simulare le situazioni per cui ho progettato questo algoritmo e inizialmente confrontarle con qualche caso reale.

6 Dati e analisi

Con le impostazioni che ho ricavato nella sezione precedente ho quindi proceduto con la simulazione del comportamento delle onde in presenza di vari potenziali.

6.1 Il salto di potenziale

Il primo potenziale che analizzo è il salto di potenziale, posiziono un gradino a metà del dominio e, con varie altezze e ne calcolo il coefficiente di trasmissività.

$$\begin{cases} V_0 & x > a \\ 0 & \text{altrove} \end{cases} \tag{6.1.1}$$

Prima di tutto farò una piccola parentesi teorica sul risultato che mi aspetto. Prendiamo un dominio infinito con in 0 il salto di potenziale, l'equazione, indipendente dal tempo, avrà la forma:

$$E\psi = \left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V_0 H(x) \right) \psi \tag{6.1.2}$$

Divido lo spazio in due porzioni ($x < 0$ e $x > 0$) in cui:

$$\begin{aligned} k_1 &= \sqrt{\tfrac{2mE}{\hbar^2}} & x < 0 \\ k_2 &= \sqrt{\tfrac{2m(E-V_0)}{\hbar^2}} & x > 0 \end{aligned} \tag{6.1.3}$$

Dato che il potenziale ha una discontinuità finita in 0, funzione e derivata devono essere continue. Se per esempio avessimo delle onde piane in ciascuna porzione dello spazio ne avrei una che si muove verso destra ($\psi_\rightarrow$) e una verso sinistra ($\psi_\leftarrow$) e quindi in 0 avrei, chiamando i coefficenti delle onde piane A per il lato sinistro e B quello destro:

$$\begin{cases} A_\rightarrow e^{ik_1 x} + A_\leftarrow e^{-ik_1 x} & x < 0 \\ B_\rightarrow e^{ik_2 x} + B_\leftarrow e^{-ik_2 x} & x > 0 \end{cases} \Rightarrow \begin{cases} A_\rightarrow + A_\leftarrow = B_\rightarrow + B_\leftarrow \\ k_1 \left(A_\rightarrow - A_\leftarrow \right) = k_2 \left(B_\rightarrow - B_\leftarrow \right) \end{cases} \tag{6.1.4}$$

nel caso più generico. Nel nostro caso consideriamo un'onda piana da $-\infty$ che si muove verso destra prima di incontrare la barriera:

$$\begin{cases} A_\rightarrow = 1 \quad , \quad A_\leftarrow = r \\ B_\rightarrow = t \quad , \quad B_\leftarrow = 0 \end{cases} \Rightarrow \begin{cases} 1 + r = t \\ k_1 \left(1 - r \right) = k_2 t \end{cases} \Rightarrow \begin{cases} r = \frac{k_1 - k_2}{k1 + k_2} \\ t = \frac{2k_1}{k1 + k_2} \end{cases} \tag{6.1.5}$$

Quelli che mi interessano sono però il coefficiente di riflessione e trasmissione, dato che stiamo parlando di funzioni con norma L2 sono:

$$\begin{cases} R = |r|^2 = \frac{(k_1 - k_2)^2}{(k1 + k_2)^2} \\ T = 1 - R = |t|^2 \frac{k_2}{k_1} = \frac{4k_1 k_2}{(k1 + k_2)^2} \end{cases} \tag{6.1.6}$$

Quindi proseguiamo ed eseguiamo delle simulazioni.

Nelle simulazioni i lanci sono stati fatti con pacchetti gaussiani inizialmente con $\sigma = 0.5$, energia 100 e massa 10. Il potenziale è stato impostato perché il salto fosse al centro del dominio e con altezza diversa per ogni lancio e ne ho analizzato l'andamento temporale del rapporto tra l'integrale su tutto il dominio e quello fatto nella prima metà, ricavando così il coefficiente di riflessione nel tempo. Il coefficiente di riflessione possiamo dire che è il valore del peso dell'integrale quando si è stabilizzato. Dall'andamento nel tempo si può vedere un piccolo abbassamento quando l'onda impatta il potenziale per le barriere più alte, prima di venire riflessa all'indietro penetra leggermente nell'area classicamente vietata.

Come si può notare dalla Tabella 2 sebbene stia simulando un pacchetto d'onda in movimento e non un un'onda piana il peso dei coefficienti ricalca abbastanza bene i coefficienti teorici, e sembra avvicinarli di più man mano che allargo il pacchetto. Notare che la simulazione si allontana dalla teoria per le onde piane man mano che si avvicini al regime in cui il potenziale è maggiore dell'energia dell'onda.

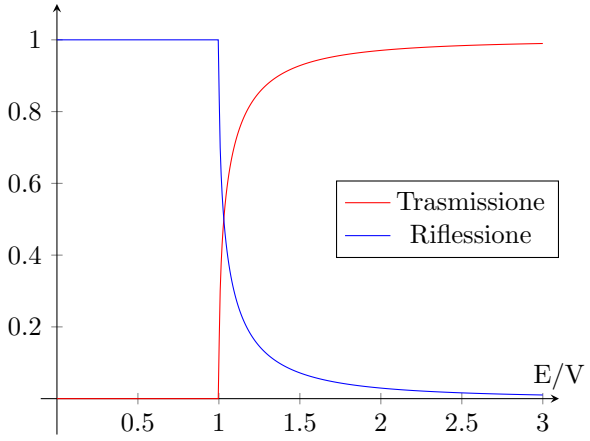


Figura 14: Coefficienti di trasmissione in rosso e riflessione in blu

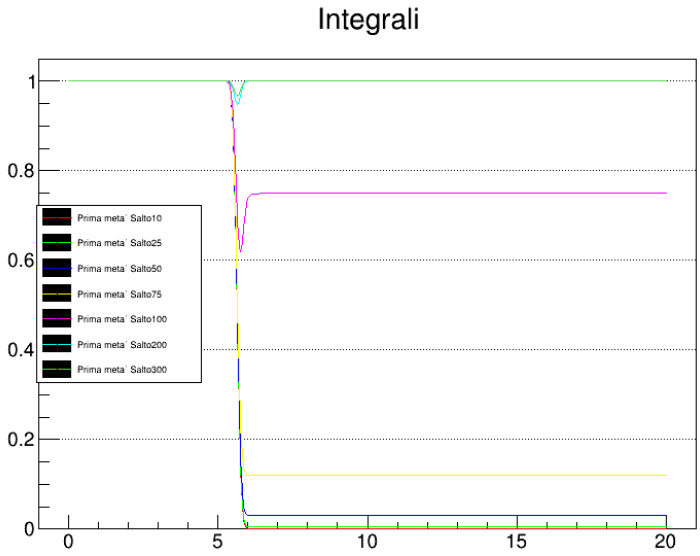


Figura 15: I pesi dell'integrale della prima metà del dominio per $\sigma = 0.5$

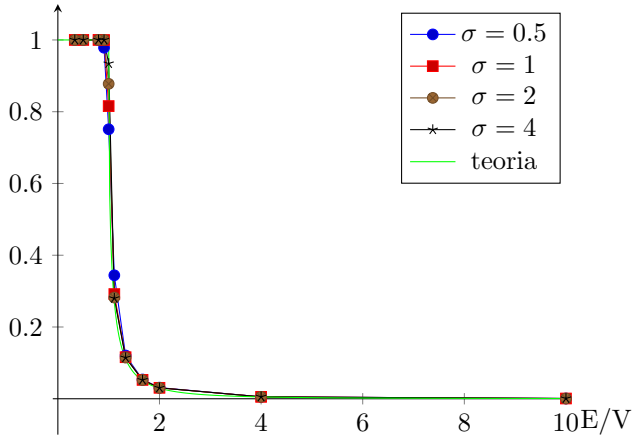


Figura 16: La visualizzazione dei dati e il confronto con la teoria

E/V	simulazione σ				teoria
	0.5	1	2	4	
10	0.000724389	0.000718394	0.000716913	0.000716543	0.000693482
4	0.00539149	0.00533663	0.00532312	0.00531976	0.00515478
2	0.0309848	0.0304801	0.0303576	0.0303272	0.0294373
1.66	0.0536892	0.0525454	0.0522718	0.0522041	0.0506914
1.33	0.120552	0.1158	0.114741	0.114483	0.111112
1.11	0.344071	0.291725	0.281975	0.279959	0.269876
1	0.751078	0.815668	0.8777	0.934287	1
0.9090	0.978315	0.999843	1	1	1
0.8	0.999979	1	1	1	1
0.5	1	1	1	1	1
0.33	1	1	1	1	1

Tabella 2: Alcuni dati delle simulazioni per il salto di potenziale

6.2 La barriera rettangolare

Similmente a prima usiamo le definizioni dei k in (6.1.3). Il potenziale avrà la forma:

$$\begin{cases} V_0 & 0 < x < a \\ 0 & \text{altrove} \end{cases} \tag{6.2.1}$$

Per l'esempio $x_0 = 0$.

Dato che il potenziale ha una discontinuità finita in 0 e in a, funzione e derivata devono essere continue. Come prima:

$$\begin{cases} A_{\rightarrow}e^{ik_1x} + A_{\leftarrow}e^{-ik_1x} & x < 0 \\ C_{\rightarrow}e^{ik_2x} + C_{\leftarrow}e^{-ik_2x} & 0 < x < a \\ B_{\rightarrow}e^{ik_1x} + B_{\leftarrow}e^{-ik_1x} & x > a \end{cases} \Rightarrow \begin{cases} A_{\rightarrow} + A_{\leftarrow} = C_{\rightarrow} + C_{\leftarrow} & \text{in } 0 \\ C_{\rightarrow}e^{ik_2a} + C_{\leftarrow}e^{-ik_2a} = B_{\rightarrow}e^{ik_1a} + B_{\leftarrow}e^{-ik_1a} & \text{in } a \\ k_1\left(A_{\rightarrow}e^{-ik_1a} - A_{\leftarrow}e^{ik_1a}\right) = k_2\left(C_{\rightarrow}e^{-ik_2a} - C_{\leftarrow}e^{ik_2a}\right) & \text{in } 0 \\ k_2\left(C_{\rightarrow}e^{ik_2a} - C_{\leftarrow}e^{-ik_2a}\right) = k_1\left(B_{\rightarrow}e^{ik_1a} - B_{\leftarrow}e^{-ik_1a}\right) & \text{in } a \end{cases} \tag{6.2.2}$$

nel caso più generico. Nel nostro caso invece consideriamo un'onda piana da $-\infty$ che si muove verso destra prima di incontrare la barriera:

$$\begin{cases} A_{\rightarrow} = 1 \quad , \quad A_{\leftarrow} = r \\ B_{\rightarrow} = t \quad , \quad B_{\leftarrow} = 0 \\ C_{\rightarrow} = m_1 \quad , \quad C_{\leftarrow} = m_2 \end{cases} \Rightarrow \begin{cases} 1 + r = m_1 + m_2 \\ m_1e^{ik_2a} + m_2e^{-ik_2a} = te^{ik_1a} \\ k_1\left(1 - r\right) = k_2\left(m_1 - m_2\right) \\ k_2\left(m_1e^{ik_2a} - m_2e^{-ik_2a}\right) = k_1te^{ik_1a} \end{cases} \Rightarrow \begin{cases} r = \frac{\left(e^{2iak_2}-1\right)\left(k_1^2-k_2^2\right)}{(k_1-k_2)^2e^{2iak_2}-(k_1+k_2)^2} \\ t = \frac{4e^{-ia(k_1-k_2)}k_1k_2}{(k_1+k_2)^2-(k_1-k_2)^2e^{2iak_2}} \\ m_1 = \frac{2k_1(k_1+k_2)}{(k_1+k_2)^2-(k_1-k_2)^2e^{2iak_2}} \\ m_2 = \frac{2e^{2iak_2}k_1(k_2-k_1)}{(k_1+k_2)^2-(k_1-k_2)^2e^{2iak_2}} \end{cases} \tag{6.2.3}$$

ma a noi interessano:

$$\begin{cases} R = |r|^2 = \left|\frac{\sin(ak_2)\left(k_1^2-k_2^2\right)}{(k_1+k_2)^2\sin(ak_2)+2ik_1k_2\cos(ak_2)}\right|^2 = \frac{\left(k_1^2-k_2^2\right)^2\sin^2(ak_2)}{4k_1^2k_2^2\cos^2(ak_2)+\left(k_1^2+k_2^2\right)^2\sin^2(ak_2)} \\ T = |t|^2 = \left|\frac{2e^{-iak_1}k_1k_2}{i(k_1+k_2)^2\sin(ak_2)-2k_1k_2\cos(ak_2)}\right|^2 = \frac{4k_1^2k_2^2}{4k_1^2k_2^2\cos^2(ak_2)+\left(k_1^2+k_2^2\right)^2\sin^2(ak_2)} \end{cases} \tag{6.2.4}$$

Che messi in termini di E e V ed m:

$$\begin{cases} R = \frac{V_0^2\sin^2\left(a\sqrt{2m(E-V_0)}\right)}{4(E^2-EV_0)+V_0^2\sin^2\left(a\sqrt{2m\sqrt{E-V_0}}\right)} \\ T = \frac{4E(E-V_0)}{4(E^2-EV_0)+V_0^2\sin^2\left(a\sqrt{2m\sqrt{E-V_0}}\right)} \end{cases} \tag{6.2.5}$$

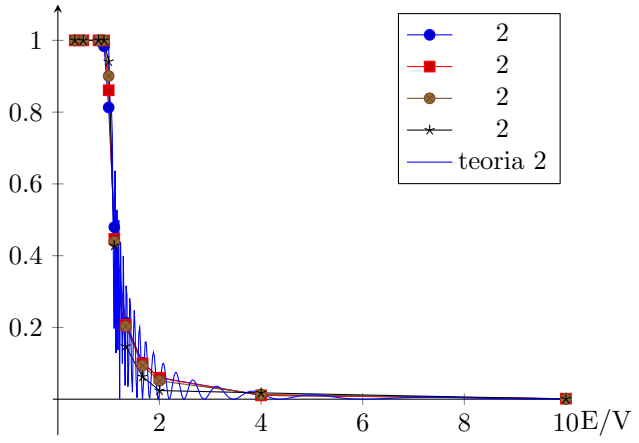


Figura 17: La visualizzazione dei dati e il confronto con la teoria

Dai dati sembra che se la barriera è più grande della larghezza del treno la simulazione non rispetta la teoria

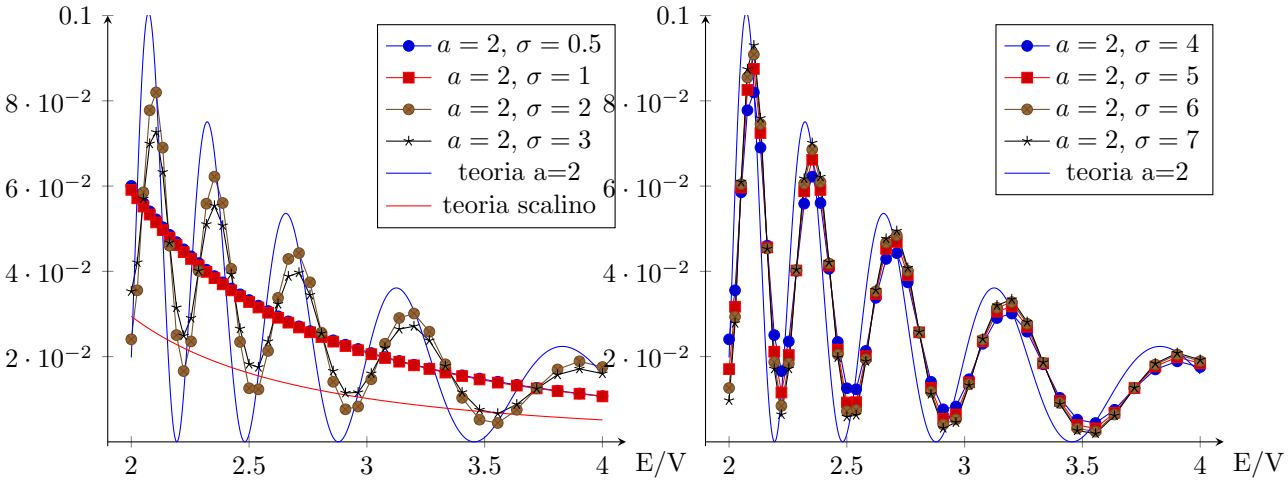


Figura 18: I dati tra 2 e 4 per una barriera larga 2

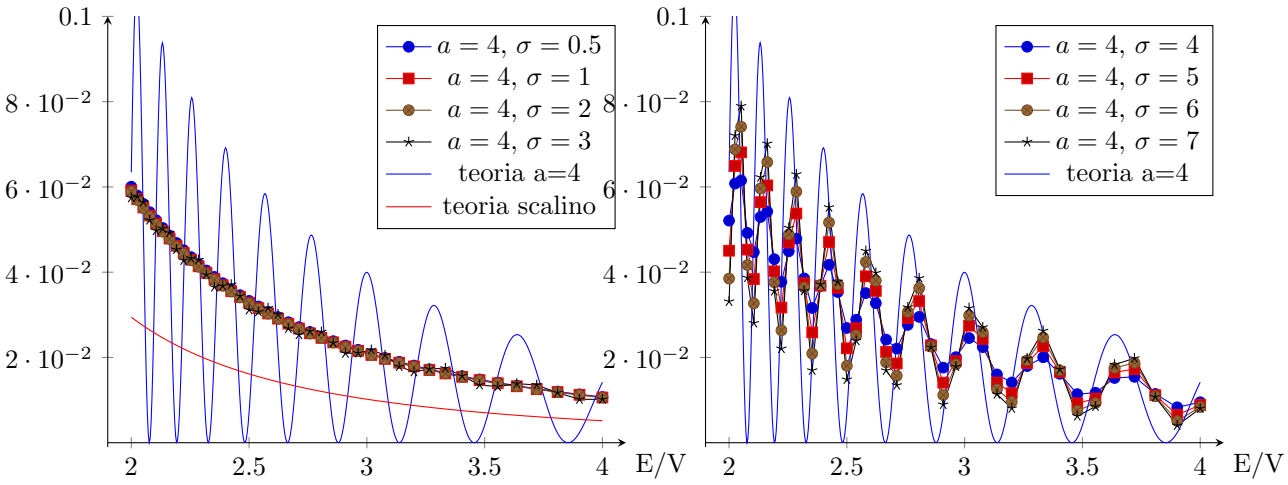


Figura 19: I dati tra 2 e 4 per una barriera larga 4

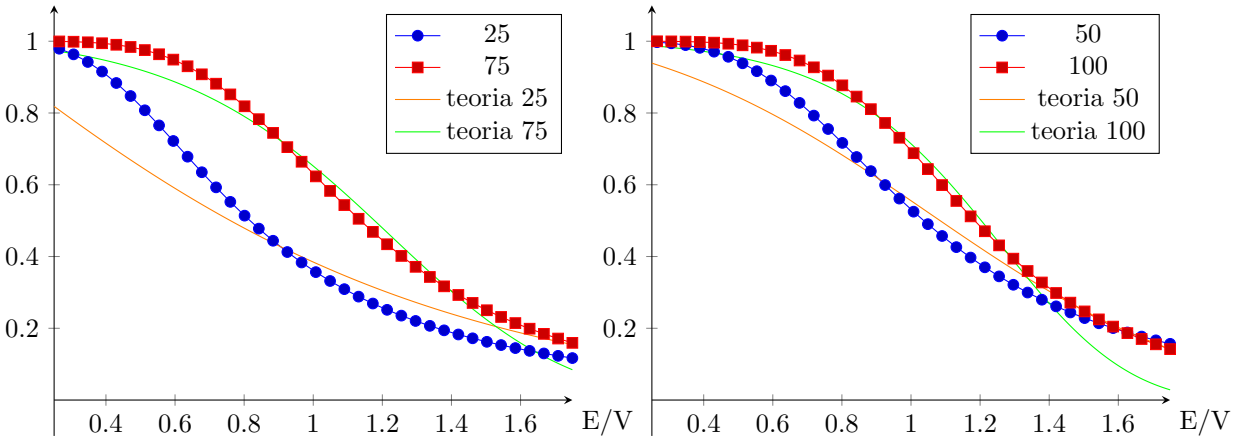


Figura 20: La visualizzazione dei dati e il confronto con la teoria, con massa 5

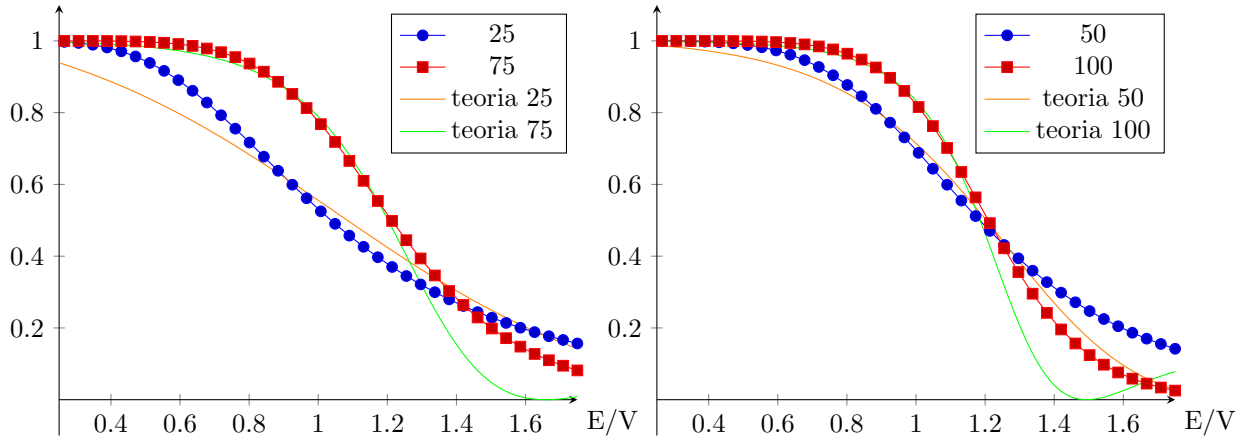


Figura 21: La visualizzazione dei dati e il confronto con la teoria, con massa 10

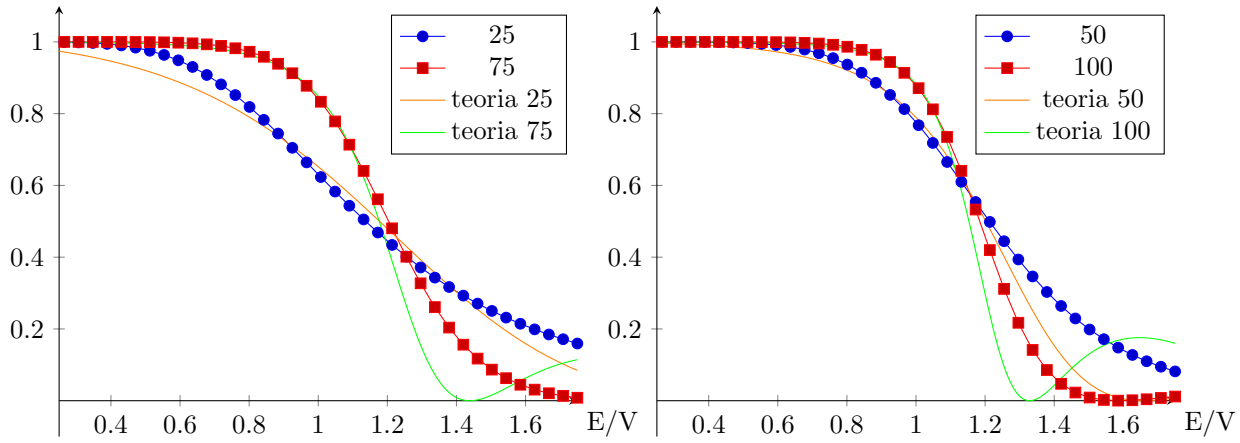


Figura 22: La visualizzazione dei dati e il confronto con la teoria, con massa 15