

# Quantum Walk e applicazioni

Relatore: **Prof. Dario Trevisan**

Di: **Riccardo Daluiso**

TESI DI LAUREA TRIENNALE

UNIVERSITÀ DI PISA



DIPARTIMENTO DI MATEMATICA  
CORSO DI LAUREA IN MATEMATICA



# Indice

<b>1</b>	<b>Introduzione</b>	<b>7</b>
1.1	Formalismo matematico e algebra lineare . . . . .	7
1.2	I postulati della Meccanica Quantistica . . . . .	9
1.3	Operatore di Densità . . . . .	11
1.4	Quantum Noise . . . . .	12
1.5	Introduzione al Quantum Computing . . . . .	13
<b>2</b>	<b>Quantum Walk</b>	<b>21</b>
2.1	Random Walk a tempi discreti . . . . .	21
2.2	Introduzione alla Quantum Walk . . . . .	24
2.3	Distribuzione della Quantum Walk . . . . .	28
2.4	Implementazione in Cirq . . . . .	33
2.5	Noisy Quantum Walk . . . . .	38
<b>3</b>	<b>Algoritmo di Grover</b>	<b>43</b>
3.1	Introduzione . . . . .	43
3.2	Procedura . . . . .	44
3.3	Interpretazione Grafica . . . . .	45
3.4	Implementazione in Cirq . . . . .	47
3.5	Algoritmo di Grover in termini di Quantum Walk . . . . .	49



# Abstract

La Meccanica Quantistica rappresenta una perfetta sintesi tra formalismo matematico e profondità del contesto fisico che essa descrive. Una formulazione rigorosa e completa della Meccanica Quantistica utilizza la teoria degli spazi di Hilbert e la teoria spettrale degli operatori non limitati su tali spazi. Analizzeremo qui solo gli aspetti matematici fondamentali necessari a comprendere il fenomeno delle *quantum walk*, che consideriamo il fulcro dello studio di questa tesi.

Dopo aver posto i fondamenti della Meccanica e della Computazione Quantistica, riprenderemo brevemente lo studio delle *random walk*, processo stocastico largamente studiato in ambito matematico, in modo da comprendere meglio le sue differenze con l'analogo quantistico. Le conoscenze algoritmiche sviluppate nella prima parte ci permetteranno di implementare le *quantum walk* su un computer classico, grazie al pacchetto Python Cirq in grado di simulare il funzionamento di *circuiti quantistici*. Per ultimo, descriveremo un importante algoritmo di ricerca spaziale, l'algoritmo di Grover, utile a comprendere, anche se in minima parte, le potenzialità dei Computer Quantistici, e lo presenteremo in termini di camminata aleatoria su un grafo completo.



# Capitolo 1

## Introduzione

In questo capitolo introdurremo la matematica necessaria alla comprensione della meccanica quantistica e porremo le basi per la computazione quantistica. Sottolineiamo che le osservazioni fisiche che hanno portato alla definizione di questo formalismo esulano dagli scopi del progetto.

### 1.1 Formalismo matematico e algebra lineare

Sia  $\mathcal{H}$  un  $\mathbb{C}$ -spazio vettoriale munito di un prodotto interno  $\langle \cdot | \cdot \rangle$ <sup>1</sup>. Assumiamo che  $\langle \cdot | \cdot \rangle$  sia lineare rispetto al secondo argomento e antilineare rispetto al primo, vale a dire

$$\langle \alpha \phi | \beta \psi \rangle = \bar{\alpha} \beta \langle \phi | \psi \rangle.$$

Se la distanza indotta da  $\langle \cdot | \cdot \rangle$  rende  $\mathcal{H}$  completo come spazio metrico, diremo che  $\mathcal{H}$  è uno *spazio di Hilbert*. Usando la notazione di Dirac, un vettore  $\psi \in \mathcal{H}$  sarà denotato con  $|\psi\rangle$  (*ket*). Invece,  $\langle \phi |$  (*bra*) denoterà un funzionale su  $\mathcal{H}$  definito da:

$$\langle \phi | : \psi \rightarrow \langle \phi | \psi \rangle.$$

Questi funzionali formano lo spazio duale  $\mathcal{H}^*$  di  $\mathcal{H}$ . Nel caso di  $\mathcal{H} = \mathbb{C}^n$ , possiamo pensare a  $\langle \psi | = |\psi\rangle^*$  come al trasposto coniugato del vettore colonna  $|\psi\rangle$ .

Con queste notazioni è possibile descrivere in maniera conveniente gli operatori su  $\mathcal{H}$  come prodotto esterno di vettori *bra* e *ket*. In particolare, la notazione  $A = |\psi\rangle \langle \phi |$  indicherà l'operatore di rango uno che agisce su un generico vettore  $|\eta\rangle$  come segue:

$$A |\eta\rangle = \langle \phi | \eta \rangle |\psi\rangle.$$

Data una base ortonormale  $\{|i\rangle\}_i$  possiamo decomporre un vettore  $|\phi\rangle$  come

$$|\phi\rangle = \sum_i \langle i | \phi \rangle |i\rangle,$$

ovvero

$$\sum_i |i\rangle \langle i| = I$$

dove  $I$  è l'operatore identità su  $\mathcal{H}$ .

Dato un operatore  $A$  di  $\mathcal{H}$ , denotiamo con  $A^*$  il suo *aggiunto*, definito dalla relazione

$$\langle \phi | A^* \psi \rangle = \langle A \phi | \psi \rangle.$$

---

<sup>1</sup>Per semplicità, ci occuperemo solo di spazi di dimensione finita, che sono anche quelli di maggiore rilievo per un'analisi computazionale.

Un operatore  $A$  si dice *Hermitiano* se  $A = A^*$ . Una classe importante di operatori Hermitiani è data dai *proiettori ortogonali*, che soddisfano la proprietà aggiuntiva di idempotenza, ovvero, se  $P$  è un proiettore,  $P^2 = P$ . L'esempio classico di proiettore è l'operatore  $|\psi\rangle\langle\psi|$  dove  $\|\psi\| = 1$ , infatti

$$(|\psi\rangle\langle\psi|)^2 = |\psi\rangle\langle\psi|\psi\rangle\langle\psi| = |\psi\rangle\langle\psi|.$$

Si dice che  $|\psi\rangle\langle\psi|$  è il proiettore *sul* vettore unitario  $|\psi\rangle$ , in quanto questo vettore genera il sottospazio dei punti fissi per il proiettore.

Un operatore  $U$  si dice *unitario* se  $U^*U = UU^* = I$ . Nel caso finito-dimensionale, è sufficiente richiedere che  $U^*U = I$ .

Come conseguenza del teorema spettrale (valido per ogni operatore *normale*  $N$ , cioè che soddisfa la proprietà  $NN^* = N^*N$ ), per ogni operatore Hermitiano  $A$  esiste una opportuna base ortonormale  $\{|i\rangle\}_i$  di autovettori per  $A$  con corrispondenti autovalori *reali*  $\alpha_i$  tale che

$$A = \sum_i \alpha_i |i\rangle\langle i|.$$

La decomposizione spettrale di  $A$  si può scrivere in una forma conveniente se si 'raggruppano' gli autovettori corrispondenti a uno stesso autovalore. Formalmente, si definisce

$$P_\alpha = \sum_{i:\alpha_i=\alpha} |i\rangle\langle i|$$

che è un proiettore, e si ottiene

$$A = \sum_{\alpha \in \text{Spec}(A)} \alpha P_\alpha.$$

Questa rappresentazione è unica a meno di riordinare, e la collezione di proiettori  $\{P_\alpha\}$  forma una *risoluzione ortogonale all'identità*, cioè

$$P_\alpha P_{\alpha'} = \delta_{\alpha\alpha'} P_\alpha \quad \sum_{\alpha \in \text{Spec}(A)} P_\alpha = I$$

dove  $\delta_{\alpha\alpha'} = 1$  se  $\alpha = \alpha'$ ,  $\delta_{\alpha\alpha'} = 0$  altrimenti. Come conseguenza, la collezione  $\{P_\alpha\}$  spezza  $\mathcal{H}$  in una somma diretta ortogonale data dalle immagini dei proiettori, cioè

$$\mathcal{H} = \bigoplus_{\alpha} \text{Im} P_\alpha.$$

Siano ora  $(\mathcal{H}_1, \langle \cdot | \cdot \rangle_1)$  e  $(\mathcal{H}_2, \langle \cdot | \cdot \rangle_2)$  due spazi di Hilbert muniti di prodotto scalare. Ricordiamo che un generico  $\psi \in \mathcal{H}_j$  definisce un funzionale  $|\phi\rangle \mapsto \langle\psi|\phi\rangle$ . Per ogni  $|\psi_j\rangle \in \mathcal{H}_j$ ,  $j = 1, 2$ , definiamo  $\langle\psi_1 \otimes \psi_2|$  come il funzionale definito su  $\mathcal{H}_1 \times \mathcal{H}_2$  tale che

$$\langle\psi_1 \otimes \psi_2|(|\phi_1, \phi_2\rangle) = \langle\psi_1|\phi_1\rangle_1 \cdot \langle\psi_2|\phi_2\rangle_2. \quad (1.1.1)$$

Definiamo anche un prodotto interno sullo spazio  $\mathcal{L}$  delle combinazioni lineari  $\sum_j c_j \psi_{1j} \otimes \psi_{2j}$  che soddisfano (1.1.1) come segue:

$$\langle\psi_1 \otimes \psi_2|\phi_1 \otimes \phi_2\rangle \doteq \langle\psi_1|\phi_1\rangle_1 \cdot \langle\psi_2|\phi_2\rangle_2$$

Si può dimostrare che uno spazio  $\mathcal{L}$  così definito esiste ed è unico, ed è uno spazio di Hilbert, che si indica con  $\mathcal{H}_1 \otimes \mathcal{H}_2$ . Si può inoltre dimostrare la seguente proprietà, che coinvolge il prodotto tensoriale di due spazi di Hilbert:

**Proprietà 1.1.1.** *Se  $\{|i\rangle\}_i$  è una base ortonormale per  $\mathcal{H}_1$  e  $\{|j\rangle\}_j$  è una base ortonormale per  $\mathcal{H}_2$ ,  $\{|i\rangle \otimes |j\rangle\}_{i,j}$  è una base ortonormale per  $\mathcal{H}_1 \otimes \mathcal{H}_2$ .*

## 1.2 I postulati della Meccanica Quantistica

In questa sezione si descrivono e si analizzano i postulati che stanno alla base della meccanica quantistica, e forniscono una connessione profonda tra i risultati fisici sperimentali e il formalismo matematico necessario a descriverli. Non esiste un sistema di postulati unico, e non deve dunque stupire se si osservano leggere differenze tra un testo e l'altro.

**Postulato 1. (Stati)** *Gli stati fisici di un sistema isolato sono in corrispondenza con classi di equivalenza di uno spazio di Hilbert separabile  $\mathcal{H}$ . La relazione di equivalenza è data da*

$$|\psi\rangle \sim |\eta\rangle \Leftrightarrow \exists \lambda \in \mathbb{C} \setminus \{0\} : |\psi\rangle = \lambda |\eta\rangle.$$

Per questa ragione nel corso della trattazione considereremo come rappresentanti canonici gli stati *unitari*, cioè tali che  $\langle \psi | \psi \rangle = 1$ . Chiameremo questi stati anche *funzioni d'onda*. Lo spazio  $\mathcal{H}$  è chiamato *spazio degli stati*.

**Postulato 2. (Dinamica, tempo discreto)** *L'evoluzione di uno sistema quantistico chiuso è descritta da operatori unitari; se lo stato del sistema al tempo  $t_0$  è descritto da  $|\psi_0\rangle$ , allora esiste un operatore unitario  $U$  dipendente solo da  $t_0$  e  $t_1$  tale che lo stato al tempo  $t_1$  è descritto da  $|\psi_1\rangle$ , legato a  $|\psi_0\rangle$  tramite la relazione*

$$|\psi_1\rangle = U |\psi_0\rangle.$$

Il presente postulato descrive la relazione che intercorre tra gli stati di un sistema in due diversi istanti di tempo. Esiste una versione più raffinata di questo postulato che descrive l'evoluzione di un sistema a tempo continuo. L'operatore  $H$  utilizzato nella sua definizione è chiamato *hamiltoniana* del sistema e non deve essere confuso con l'operatore di *Hadamard* che sarà introdotto più avanti. Per lo scopo della tesi, sarà sufficiente il Postulato 2, ma introduciamo anche quest'altra versione per completezza.

**Postulato 2'. (Dinamica, tempo continuo)** *L'evoluzione temporale di un sistema quantistico chiuso è descritta dall'equazione di Schrödinger:*

$$i\hbar \frac{d|\psi\rangle}{dt} = H |\psi\rangle.^2$$

Il terzo postulato descrive l'effetto di una misurazione sul sistema, che dunque non è più isolato. È anche detta *legge di Born*.

**Postulato 3. (Misure)** *Le misure sono descritte da una collezione  $\{M_a\}_{a \in \Lambda}$  di operatori di misura. Essi soddisfano la relazione di completezza,*

$$\sum_{a \in \Lambda} M_a^\dagger M_a = I$$

*e agiscono su  $\mathcal{H}$ , lo spazio di Hilbert del sistema che deve essere misurato. L'indice  $a$  si riferisce al risultato della misura. Se il sistema è nello stato  $|\psi\rangle$  prima della misura, la probabilità di ottenere  $a$  è data da*

$$p(a) = \langle \psi | M_a^\dagger M_a | \psi \rangle$$

*e lo stato del sistema dopo la misura (avendo osservato come risultato  $a$ ) è*

$$\frac{M_a |\psi\rangle}{\sqrt{\langle \psi | M_a^\dagger M_a | \psi \rangle}}.$$

<sup>2</sup>Spesso, in questo caso, lo spazio dell'equazione è  $L^2(-\pi, \pi)$  o  $L^2(\mathbb{C})$ , spazi infinito-dimensionali.

La relazione di completezza ci garantisce che la somma delle probabilità sia 1, infatti

$$\sum_a p(a) = \sum_a \langle \psi | M_a^\dagger M_a | \psi \rangle = \langle \psi | \sum_a M_a^\dagger M_a | \psi \rangle = \langle \psi | \psi \rangle = 1.$$

Una classe importante di misure è data dalle *misure proiettive*. Una misura proiettiva è descritta da un'osservabile, un operatore Hermitiano  $M$  sullo spazio  $\mathcal{H}$  che ammette una decomposizione spettrale

$$M = \sum_a a |a\rangle \langle a| = \sum_a a P_a.$$

I possibili risultati della misura corrispondono agli autovalori dell'osservabile. Dato uno stato  $|\psi\rangle$ , la probabilità di osservare  $a$  con la misura è data da

$$p(a) = \langle \psi | P_a | \psi \rangle$$

e lo stato dopo la misura sarà nello stato

$$\frac{P_a |\psi\rangle}{\sqrt{p(a)}}.$$

Dalla definizione si evince che le misure proiettive possono essere pensate come un caso particolare delle misure introdotte nel Postulato 3, se gli operatori  $M_a$  soddisfano la relazione di completezza e sono proiettori ortogonali.

Utilizzando le notazioni del Postulato 3, possiamo infine definire

$$E_a = M_a^\dagger M_a.$$

Si osserva subito che l'operatore  $E_a$  è semi-definito positivo, e la conoscenza dell'insieme  $\{E_a\}_a$  è condizione sufficiente per determinare le probabilità dei diversi risultati delle misure. Questi operatori sono chiamati *elementi POVM* associati alle misure (abbreviazione di 'Positive Operator-Valued Measure', *misura a valori operatoriali positivi*). L'insieme completo  $\{E_a\}_a$  è chiamato POVM.

## 1.3 Operatore di Densità

Supponiamo che un sistema quantistico si trovi in uno stato  $|\psi_i\rangle$  con rispettiva probabilità  $p_i$ , per  $i = 1, \dots, k$ . Chiamiamo l'insieme  $\{(\psi_i, p_i)\}_{i=1, \dots, k}$  un *ensemble di stati puri*. Definiamo la matrice di densità del sistema

$$\rho := \sum_{i=1}^k p_i |\psi_i\rangle \langle \psi_i|.$$

Se il sistema evolve tramite un operatore unitario  $U$ , cioè  $|\psi_i\rangle \rightarrow U|\psi_i\rangle$ , l'evoluzione della matrice di densità è descritta da

$$\rho \xrightarrow{U} \sum_{i=1}^k p_i U |\psi_i\rangle \langle \psi_i| U^\dagger = U \rho U^\dagger.$$

Anche le misure possono essere descritte in termini di operatore di densità. Sappiamo infatti dal postulato 3 che se lo stato iniziale è  $|\psi_i\rangle$ , la probabilità di osservare  $m$  è data da

$$p(m|i) = \langle \psi_i | M_m^\dagger M_m | \psi_i \rangle = \text{Tr}(M_m^\dagger M_m |\psi_i\rangle \langle \psi_i|),$$

dove per l'ultima uguaglianza abbiamo utilizzato il fatto che se  $\{|i\rangle\}_i$  è una base ortonormale,

$$\text{Tr}(A |\psi\rangle \langle \psi|) = \sum_i \langle i | A | \psi \rangle \langle \psi | i \rangle = \langle \psi | A | \psi \rangle.$$

Applicando la legge delle probabilità totali si ottiene allora che

$$\begin{aligned} p(m) &= \sum_i p_i p(m|i) = \sum_i p_i \text{Tr}(M_m^\dagger M_m |\psi_i\rangle \langle \psi_i|) \\ &= \text{Tr}(M_m^\dagger M_m \sum_i p_i |\psi_i\rangle \langle \psi_i|) \\ &= \text{Tr}(M_m^\dagger M_m \rho). \end{aligned}$$

Abbiamo ottenuto il postulato 3 in termini di operatore di densità.

**Teorema 1.3.1** (Caratterizzazione dell'operatore di densità). *Un operatore  $\rho$  è l'operatore di densità associato ad un qualche ensemble  $\{p_i, |\psi_i\rangle\}$  se e solo se soddisfa le seguenti proprietà:*

- (**Traccia**)  $\text{Tr}(\rho) = 1$ .
- (**Positività**)  $\rho > 0$ .

*Dimostrazione.* ( $\Rightarrow$ ) Sia  $\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i|$  un operatore di densità. Allora

$$\text{Tr}(\rho) = \sum_i p_i \text{Tr}(|\psi_i\rangle \langle \psi_i|) = \sum_i p_i = 1.$$

Inoltre, per ogni  $|\phi\rangle$  nello spazio degli stati si ha che

$$\langle \psi | \rho | \psi \rangle = \sum_i p_i \langle \phi | \psi_i \rangle \langle \psi_i | \phi \rangle = \sum_i p_i |\langle \phi | \psi_i \rangle|^2 \geq 0.$$

( $\Leftarrow$ ) Se  $\rho$  soddisfa la condizione di positività, ammette una decomposizione spettrale

$$\rho = \sum_i \lambda_i |i\rangle \langle i|$$

dove i vettori  $|i\rangle$  sono ortogonali, e gli scalari  $\lambda_i$  sono gli autovalori reali non negativi di  $\rho$ .<sup>3</sup> La condizione di traccia ci garantisce che  $\sum_i \lambda_i = 1$ . A questo punto, è immediato verificare che  $\rho$  è l'operatore di densità associato all'ensemble di stati  $\{\lambda_i, |i\rangle\}_i$ .  $\square$

<sup>3</sup>Un operatore positivo in uno spazio di Hilbert complesso è necessariamente autoaggiunto, e dunque vale il teorema spettrale.

## 1.4 Quantum Noise

Nel seguito vogliamo analizzare le operazioni che possono essere eseguite sulla matrice di densità di un sistema quantistico, dovute ad esempio ad interazioni con un ipotetico ambiente esterno. Per evidenziare l'azione di quest'ultimo utilizziamo il formalismo degli *operatori di Kraus*  $\{E_k\}_k$ , definiti secondo la seguente equazione:

$$\mathcal{E}(\rho) = \sum_{k=1}^m E_k \rho E_k^\dagger, \quad (1.4.1)$$

dove con  $\mathcal{E}(\cdot)$  abbiamo indicato l'azione dell'ambiente esterno sul sistema, con conseguente cambiamento dello stato del sistema e della sua matrice di densità:

$$\rho \mapsto \mathcal{E}(\rho),$$

$$|\psi\rangle \mapsto \mathcal{E}(|\psi\rangle) = \mathcal{E}(|\psi\rangle \langle\psi|).$$

Gli operatori di Kraus  $\{E_k\}_k$  in generale non sono unitari, ma devono soddisfare la proprietà di conservazione della traccia (*trace-preserving property*):

$$\sum_{k=1}^m E_k^\dagger E_k = I. \quad (1.4.2)$$

Nella sezione 1.5 descriveremo i tipi più comuni di rumore nei sistemi di qubit.

## 1.5 Introduzione al Quantum Computing

### Qubit

Il qubit (abbreviazione di *quantum bit*) è l'unità di informazione quantistica. Per i nostri scopi ci concentreremo sugli aspetti matematici di questa entità, presentandola da un punto di vista astratto, senza preoccuparci della sua esistenza come 'oggetto fisico'.

In accordo con il primo postulato della MQ, il qubit è rappresentato da un vettore unitario in uno spazio di Hilbert  $\mathcal{H}$ , e analogamente al bit classico, ammette due stati, che indicheremo per convenzione con  $|0\rangle$  (*ground state*) e  $|1\rangle$  (*excited state*). Tuttavia, grazie al principio di sovrapposizione (una immediata conseguenza del primo postulato), un qubit ammette anche una qualsiasi combinazione lineare unitaria di questi due stati,  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  con  $\alpha, \beta \in \mathbb{C}$  e  $|\alpha|^2 + |\beta|^2 = 1$ .

Detto in altri termini, un qubit è un vettore *unitario*  $|\psi\rangle \in \mathcal{H} := \text{Span}\{|0\rangle, |1\rangle\}$ . Una tale base per  $\mathcal{H}$  è detta *base computazionale*, ed è una base ortonormale per questo spazio vettoriale.

Il punto fondamentale della teoria dell'informazione quantistica è che non possiamo in alcun modo determinare lo stato esatto di un qubit, ovvero determinare  $\alpha, \beta$  come sopra. Consideriamo la misura di un qubit nella base computazionale. Abbiamo qui due possibili risultati, 0 e 1, definiti rispettivamente dagli operatori di misura  $M_0 = |0\rangle\langle 0|$  e  $M_1 = |1\rangle\langle 1|$ . Ciasuno dei due operatori è Hermitiano ( $M_{0,1}^\dagger = M_{0,1}$ ) e idempotente. La relazione di completezza è soddisfatta, infatti  $M_0^\dagger M_0 + M_1^\dagger M_1 = M_0^2 + M_1^2 = M_0 + M_1 = I$ .

Supponiamo adesso che lo stato da misurare sia  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ . Applicando il Postulato 3 si ottiene che la probabilità di ottenere il risultato 0 è

$$p(0) = \langle \psi | M_0^\dagger M_0 | \psi \rangle = \langle \psi | M_0 | \psi \rangle = |\alpha|^2.$$

Analogamente,  $p(1) = |\beta|^2$ . Lo stato  $|\psi'\rangle$  dopo la misura sarà quindi

$$|\psi'\rangle = \begin{cases} \frac{\alpha}{|\alpha|} |0\rangle & \text{con probabilità } |\alpha|^2 \\ \frac{\beta}{|\beta|} |1\rangle & \text{con probabilità } |\beta|^2 \end{cases}$$

che differiscono rispettivamente dagli stati  $|0\rangle$  e  $|1\rangle$  per un fattore di fase globale, e dunque ininfluente per il postulato 1.

### La sfera di Bloch

Gli stati di un qubit hanno una rappresentazione geometrica come punti di una sfera tridimensionale. La sfera di Bloch può essere vista come una generalizzazione della rappresentazione geometrica di un numero complesso di norma 1 come punto della circonferenza unitaria:  $z = x + iy, |z|^2 = x^2 + y^2 = 1$  che è l'equazione di  $S^1$ . Questa rappresentazione rende ben agevole l'osservazione dell'effetto del noise sui qubit ([3]). Cerchiamo di ricavare una descrizione di uno stato  $|\psi\rangle$  in funzione di angoli.

Come abbiamo visto nella sezione 1.5, un generico stato di un qubit può essere scritto nella forma  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  con  $|\alpha|^2 + |\beta|^2 = 1$ .

Possiamo esprimere  $\alpha$  e  $\beta$  in coordinate polari, ottenendo

$$|\psi\rangle = \rho_\alpha e^{i\phi_\alpha} |0\rangle + \rho_\beta e^{i\phi_\beta} |1\rangle.$$

Tuttavia moltiplicare lo stato per un fattore arbitrario  $e^{i\gamma}$  non ha conseguenze osservabili e dunque possiamo ridurre i gradi di libertà:

$$|\psi\rangle \equiv \rho_\alpha |0\rangle + \rho_\beta e^{i(\phi_\beta - \phi_\alpha)} |1\rangle = \rho_\alpha |0\rangle + \rho_\beta e^{i\phi} |1\rangle.$$

Ci siamo ridotti a una descrizione con soli 3 gradi di libertà. Utilizzando adesso il vincolo di normalizzazione, e tornando in coordinate cartesiane,  $\rho_\beta e^{i\phi_\beta} = x + iy$ , otteniamo:

$$|\rho_\alpha|^2 + |x + iy|^2 = \rho_\alpha^2 + x^2 + y^2 = 1.$$

Che è l'equazione di  $S^2$ . Prima di procedere, ricordiamo le equazioni di passaggio da coordinate cartesiane a coordinate sferiche:

$$\begin{cases} x = \rho \sin \theta \cos \phi \\ y = \rho \sin \theta \sin \phi \\ z = \rho \cos \theta \end{cases}$$

con  $\rho \in [0, +\infty)$ ,  $\theta \in [0, \pi]$ ,  $\phi \in [0, 2\pi)$ .

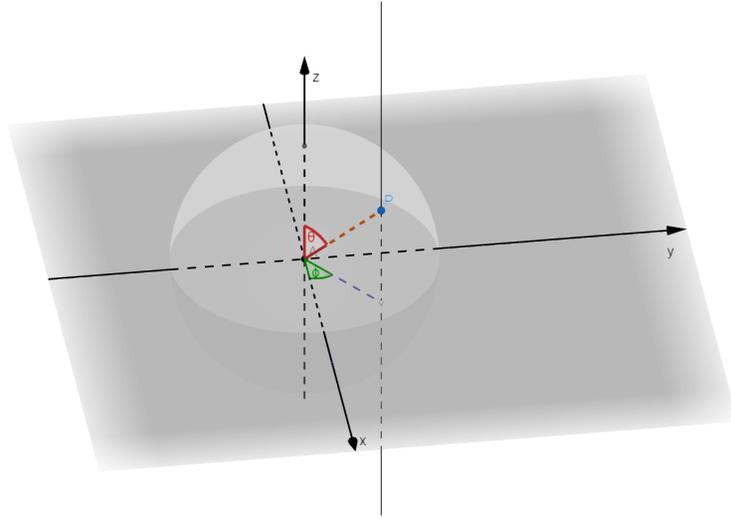


Figura 1.1: Angoli  $\theta$  e  $\phi$  del passaggio in coordinate sferiche.

Interpretando  $z$  come il coefficiente di  $|0\rangle$ ,  $\rho_\alpha$ , e ricordando che  $\rho = 1$ , possiamo scrivere:

$$\begin{aligned} |\psi\rangle &= \cos \theta |0\rangle + (\cos \phi + i \sin \phi) \sin \theta |1\rangle \\ &= \cos \theta |0\rangle + e^{i\phi} \sin \theta |1\rangle. \end{aligned} \tag{1.5.1}$$

In questo modo siamo riusciti ad eliminare un altro grado di libertà, abbiamo solo due parametri che descrivono i punti su una sfera unitaria.

Consideriamo adesso uno stato  $|\psi'\rangle$  corrispondente al punto antipodale a  $|\psi\rangle$  sulla sfera, e avente cioè coordinate sferiche ( $\rho = 1$ ,  $\theta' = \pi - \theta$ ,  $\phi' = \pi + \phi$ ). Scrivendo  $|\psi'\rangle$  come in (1.5.1) si ottiene subito che  $|\psi'\rangle = -|\psi\rangle$ , dunque i due stati differiscono solo per un fattore di fase di  $-1$ , e sono perciò equivalenti. Si può quindi considerare solo la semisfera superiore, parametrizzata da  $\theta \in [0, \frac{\pi}{2}]$ . Effettuando la sostituzione  $\theta \rightarrow 2\theta$  si possono mappare i punti sulla semisfera superiore sui punti di una sfera e si ottiene così la formula convenzionale per la rappresentazione di un qubit sulla sfera di Bloch:

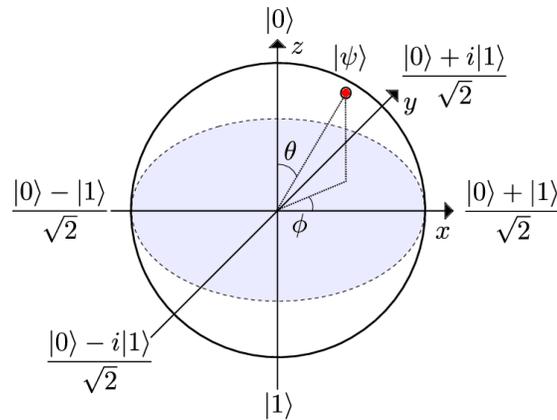
$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle, \quad \theta \in [0, \pi], \quad \phi \in [0, 2\pi).$$

Dimostriamo adesso che con questa definizione una qualsiasi coppia di punti antipodali è ortogonale : siano dunque

$$\begin{aligned}
 |\psi\rangle &= \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \\
 |\eta\rangle &= \cos \frac{\pi - \theta}{2} |0\rangle + e^{i(\phi + \pi)} \sin \frac{\pi - \theta}{2} |1\rangle \\
 &= \cos \frac{\pi - \theta}{2} |0\rangle - e^{i\phi} \sin \frac{\pi - \theta}{2} |1\rangle
 \end{aligned} \tag{1.5.2}$$

e calcoliamo  $\langle \eta | \psi \rangle$

$$\begin{aligned}
 \langle \eta | \psi \rangle &= \cos \frac{\pi - \theta}{2} \cos \frac{\theta}{2} - \sin \frac{\pi - \theta}{2} \sin \frac{\theta}{2} \\
 &= \cos \frac{\pi - \theta}{2} + \frac{\theta}{2} \\
 &= \cos \frac{\pi}{2} = 0.
 \end{aligned} \tag{1.5.3}$$



## Matrici di Pauli

Le matrici di Pauli sono matrici  $2 \times 2$  complesse autoaggiunte e unitarie. Sono così definite:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \equiv X$$

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \equiv Y$$

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \equiv Z$$

Esse soddisfano le seguenti proprietà:

- $\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = I$
- $[\sigma_i, \sigma_j] = \sigma_i \sigma_j - \sigma_j \sigma_i = 2i \varepsilon_{ijk} \sigma_k$

$\varepsilon$  è il simbolo di Levi-Civita definito da

$$\varepsilon_{ijk} = \begin{cases} +1 & \text{se } (i, j, k) \text{ permutazione pari} \\ -1 & \text{se } (i, j, k) \text{ permutazione dispari} \\ 0 & \text{altrimenti} \end{cases}$$

dove con parità della permutazione indichiamo il numero di trasposizioni necessario a portare  $(i, j, k)$  nella permutazione  $(1, 2, 3)$ .

- $\det(\sigma_i) = -1, \quad i = 1, 2, 3$
- $\text{tr}(\sigma_i) = 0, \quad i = 1, 2, 3$

In Quantum Computing, le matrici di Pauli sono tra le più importanti porte logiche a singolo qubit. Nella tesi faremo uso principalmente della matrice di permutazione  $X$  (*bit-flip*).

## Logica Quantistica

Una computer classico è spesso rappresentato come un registro finito di  $n$  bit sul quale è possibile agire tramite operazioni logiche elementari quali AND, OR, o NOT. Un qualsiasi intero  $i \in [0, 2^n - 1]$  ammette una rappresentazione binaria della forma

$$i = \sum_{j=0}^{n-1} 2^j i_j$$

con  $i_j \in \{0, 1\} \forall j = 0, \dots, n-1$ . La stringa  $i_{n-1} \dots i_1 i_0$  costituisce allora la rappresentazione binaria di  $i$ .

Analogamente, un computer quantistico può essere visto come lo stato composto degli  $n$  qubit  $\{|i_j\rangle\}_{j=0}^{n-1}$  che costituiscono un suo registro, e un generico stato  $|\psi\rangle$  è della forma

$$|\psi\rangle = \sum_{j=0}^{2^n-1} c_j |\psi_j\rangle = \sum_{j=0}^{2^n-1} c_j |i_{n-1}\rangle \otimes \dots \otimes |i_i\rangle \otimes |i_0\rangle$$

con la condizione di normalizzazione

$$\sum_{j=0}^{2^n-1} |c_j|^2 = 1.$$

In accordo con il postulato 1 della meccanica quantistica, lo stato di un computer quantistico a  $n$  qubit è una funzione d'onda in uno spazio di Hilbert di dimensione  $2^n$ .

## Quantum Gate

I *gate quantistici* sono alla base della descrizione dei circuiti quantistici, e rappresentano l'analogo delle porte logiche nei circuiti classici. È bene sottolineare che si lavora in coordinate nella base computazionale  $\{|0\rangle, |1\rangle\}$ , cioè

$$\alpha |0\rangle + \beta |1\rangle \leftrightarrow \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Inoltre, ogni operazione logica applicata ad uno stato deve preservarne la condizione di normalizzazione, cioè l'operatore associato deve essere unitario.

L'operatore di maggior rilievo per gli scopi della tesi è sicuramente l'operatore di *Hadamard*, definito come

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Si può verificare facilmente che  $H$  agisce sulla base computazionale come segue

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \equiv |+\rangle$$

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \equiv |-\rangle$$

Gli stati  $|+\rangle, |-\rangle$  formano una nuova base composta dalla sovrapposizione degli stati della base computazionale con eguale ampiezza, infatti dopo una misura avremo probabilità  $1/2$  di osservare 0 e probabilità  $1/2$  di osservare 1.

Gli operatori possono operare anche su due o più qubit. Gli stati composti possono essere rappresentati in coordinate come segue <sup>4</sup>

$$|0\rangle \equiv |00\rangle \equiv \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad |1\rangle \equiv |01\rangle \equiv \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}; \quad |2\rangle \equiv |10\rangle \equiv \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}; \quad |3\rangle \equiv |11\rangle \equiv \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}; \quad \dots$$

cioè  $|j\rangle$  è lo stato composto che rappresenta  $j \in \mathbb{N}$  in forma binaria, dipendente implicitamente dal numero di qubit disponibili. Osserviamo che la rappresentazione vettoriale degli stati composti è coerente con il *prodotto di Kronecker*, o prodotto tensoriale matriciale - interpretando i vettori come matrici  $n \times 1$  - definito come segue:

**Definizione 1.5.1** (Prodotto di Kronecker). *Siano  $A \in \mathbb{C}^{m \times n}$  e  $B \in \mathbb{C}^{p \times q}$ . Il prodotto di Kronecker tra  $A$  e  $B$  è la matrice  $A \otimes B \in \mathbb{C}^{mp \times nq}$  definita a blocchi come*

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}$$

Il principale operatore a due qubit è sicuramente il *CNOT*, o controlled-not gate. Il primo qubit agisce da controllo, mentre il secondo da bersaglio (*target qubit*). Lo stato del qubit bersaglio viene scambiato solo se il qubit di controllo è nello stato  $|1\rangle$ , altrimenti viene lasciato invariato. Questo permette di costruire un operatore unitario (e dunque invertibile), infatti la rappresentazione matriciale del *CNOT* è

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

E' interessante notare che grazie all'operatore *CNOT* è possibile osservare il fenomeno dell'*entanglement quantistico*. Dato uno spazio  $\mathcal{H}_A \otimes \mathcal{H}_B$ , esistono stati non esprimibili come prodotto tensoriale di un elemento di  $\mathcal{H}_A$  e un elemento  $\mathcal{H}_B$ . Se uno stato può essere scritto nella forma

$$|\psi\rangle = |\psi\rangle_A \otimes |\psi\rangle_B$$

si dice che è *separabile*. In caso contrario lo stato è *entangled*.

Applicando *CNOT* ad uno stato della forma  $(\alpha|0\rangle + \beta|1\rangle)|0\rangle$  con  $\alpha, \beta \neq 0$ , otteniamo

$$CNOT(\alpha|0\rangle + \beta|1\rangle)|0\rangle = \alpha|00\rangle + \beta|11\rangle,$$

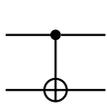
<sup>4</sup>Per descrivere stati composti  $|i\rangle \otimes |j\rangle$  utilizzeremo spesso le notazioni compatte  $|i\rangle|j\rangle$  o  $|ij\rangle$ .

che è uno stato entangled in quanto non può essere espresso nella forma

$$\gamma|i\rangle \otimes \delta|j\rangle$$

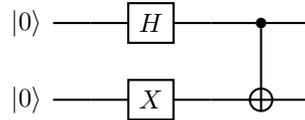
con  $i, j \in \{0, 1\}$ .

Riportiamo qui i principali gate utilizzati nell'articolo, con le rispettive abbreviazioni utilizzate nella modellazione di circuiti.

- Hadamard  :  $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
- Pauli-X  :  $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
- Pauli-Y  :  $\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
- Pauli-Z  :  $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
- Controlled-NOT (CNOT)  :  $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
- Misura (nella base canonica) 

Sottolineiamo che nei circuiti utilizzeremo sempre *misure proiettive*.

Riportiamo infine un semplice esempio di circuito quantistico evidenziando di passo in passo (formalmente, ad ogni *momento* del circuito) lo stato del sistema:



$$|0\rangle \otimes |0\rangle \longrightarrow H|0\rangle \otimes X|0\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes |1\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|11\rangle \longrightarrow \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle.$$

## Quantum Channel

I *quantum channel* definiscono le operazioni non unitarie sui qubit, e simulano perciò il rumore in un sistema non isolato. Come studiato nella sezione 1.4, utilizzeremo il formalismo delle matrici di densità per descriverli. Ne riportiamo alcuni:

- Bit-flip Channel:

$$\rho \longmapsto (1-p)\rho + pX\rho X;$$

- Asymmetric-Depolarizing Channel:

$$\rho \longmapsto (1-p_x-p_y-p_z)\rho + p_xX\rho X + p_yY\rho Y + p_zZ\rho Z;$$

- Amplitude-Damping Channel:

$$\rho \longmapsto E_0\rho E_0^\dagger + E_1\rho E_1^\dagger,$$

$$\text{dove } E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix} \text{ e } E_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix};$$

con  $p, p_x, p_y, p_z, \gamma \in [0, 1]$  e  $p_x + p_y + p_z = 1$ .



## Capitolo 2

# Quantum Walk

In questo capitolo introdurremo brevemente la *random walk discreta*, ovvero il moto una particella vincolata a muoversi lungo  $\mathbb{Z}$  in tempi discreti. Successivamente descriveremo l'analogo quantistico di questo processo, la *quantum walk discreta*, la cui evoluzione, in accordo al secondo postulato, è descritta da un operatore unitario  $U$ . Il modello che analizzeremo è chiamato *modello monetizzato*, per l'utilizzo di un operatore che rappresenta il lancio di una moneta e influenza il moto del sistema. Le quantum walk sono un argomento profondo ed interessante e sono largamente studiate in quanto grazie ad esse possono essere implementati diversi algoritmi quantistici (vedremo un esempio nel capitolo successivo) e possono simulare molti sistemi fisici complessi.

### 2.1 Random Walk a tempi discreti

Una particella puntiforme è vincolata a muoversi lungo  $\mathbb{Z}$  in entrambe le direzioni. All'istante iniziale  $t = 0$  essa si trova in 0, e ad ogni unità di tempo si muove di un passo a destra (+1) o di un passo a sinistra (-1) con eguale probabilità 1/2.

Formalmente, si definisce *passeggiata aleatoria* un processo stocastico  $\{S_t\}_{t \in \mathbb{N}}$  con

$$\begin{cases} S_0 = 0 \\ S_t = \sum_{k=1}^t X_k, \quad t \geq 1 \end{cases}$$

dove  $\{X_k\}$  sono variabili aleatorie i.i.d. Ci soffermeremo qui sulla passeggiata aleatoria *semplice e simmetrica*, dove  $X_k = \pm 1$  e  $P[X_k = +1] = P[X_k = -1] = 1/2$ .

Calcoliamo la probabilità di trovare la particella in  $n$  al tempo  $t$ , cioè  $P[S_t = n] = p(t, n)$ . Siano dunque

- $\Omega = \{\text{cammini possibili al tempo } t\}$
- $A = \{\text{cammini che portano in } n \text{ al tempo } t\}$

Si verifica subito che  $|\Omega| = 2^t$ , troviamo dunque la cardinalità di  $A$ . Supponiamo che la particella si sia spostata di  $\alpha$  passi verso destra e  $\beta$  passi verso sinistra. Vogliamo per prima cosa trovare  $\alpha$  e  $\beta$  in funzione di  $t$  e  $n$ . Molto più agevole è esplicitare prima  $t$  e  $n$  in funzione di  $\alpha$  e  $\beta$ , infatti

$$\begin{cases} t = \alpha - \beta \\ n = \alpha + \beta \end{cases}$$

Con una semplice sostituzione si trova che

$$\begin{cases} \alpha = \frac{t+n}{2} \\ \beta = \frac{t-n}{2} \end{cases}$$

Per trovare la cardinalità di  $A$ , osserviamo che i modi di giungere in  $n$  al tempo  $t$ , cioè di fare  $\alpha$  passi a destra in  $t$  unità di tempo, sono in bigezione con i modi di selezionare (senza ordine)  $\alpha$  elementi su un totale di  $t$  elementi, e dunque

$$\binom{t}{\alpha} = \binom{t}{\frac{n+t}{2}},$$

da cui segue che la probabilità finale è data da

$$p(t, n) = \frac{|A|}{|\Omega|} = \frac{1}{2^t} \binom{t}{\frac{n+t}{2}}.$$

Questa formula è valida solo se  $n \leq t$  e  $t+n$  è pari. Negli altri casi, la probabilità è 0. Dalla relazione trovata si può ricavare la probabilità di tornare all'origine al tempo  $t$ , infatti

$$p(t, 0) = \frac{1}{2^t} \binom{t}{\frac{t}{2}} = \sqrt{\frac{2}{\pi t}} \left(1 + O\left(\frac{1}{t}\right)\right),$$

dove si è usata l'approssimazione di Stirling,

$$t! = \sqrt{2\pi t} \left(\frac{t}{e}\right)^t \left(1 + O\left(\frac{1}{t}\right)\right).$$

Possiamo facilmente calcolare il valore atteso di  $S_t$ , infatti

$$\mathbb{E}[S_t] = \sum_{n \in \mathbb{Z}} n p(t, n) = 0$$

perché la distribuzione è simmetrica.

La deviazione standard è infine data da

$$\begin{aligned} \sigma(t) &= \sqrt{\text{Var}(S_t)} = \sqrt{\text{Var}\left(\sum_{k \leq t} X_k\right)} \\ &= \sqrt{\sum_{k \leq t} \text{Var}(X_k)} = \sqrt{\sum_{k \leq t} 1} = \sqrt{t}, \end{aligned} \tag{2.1.1}$$

dove si è usata l'indipendenza (e dunque la scorrelazione) delle  $\{X_k\}$ .

Osserviamo che  $P[|S_t| \leq t] = 1$ , tuttavia la disuguaglianza di Čebyšëv mostra che  $P[|S_t| \leq k\sqrt{t}] \geq 1 - 1/k^2 \forall k \geq 1$ . In effetti il Teorema del Limite Centrale darebbe stime anche più precise (gaussiane).

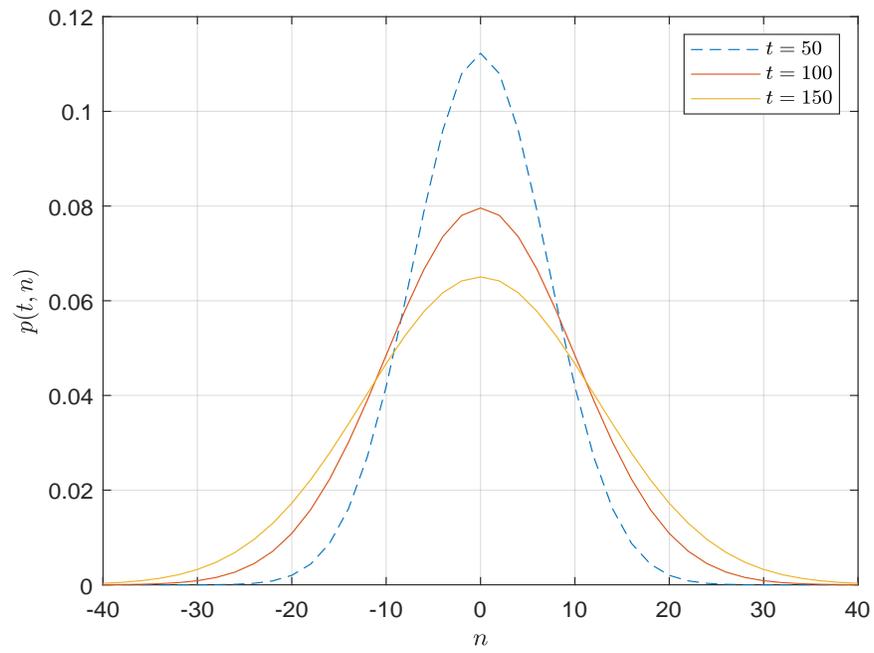


Figura 2.1: Distribuzione di probabilità di una random walk su  $\mathbb{Z}$  per i valori di  $t$  indicati.

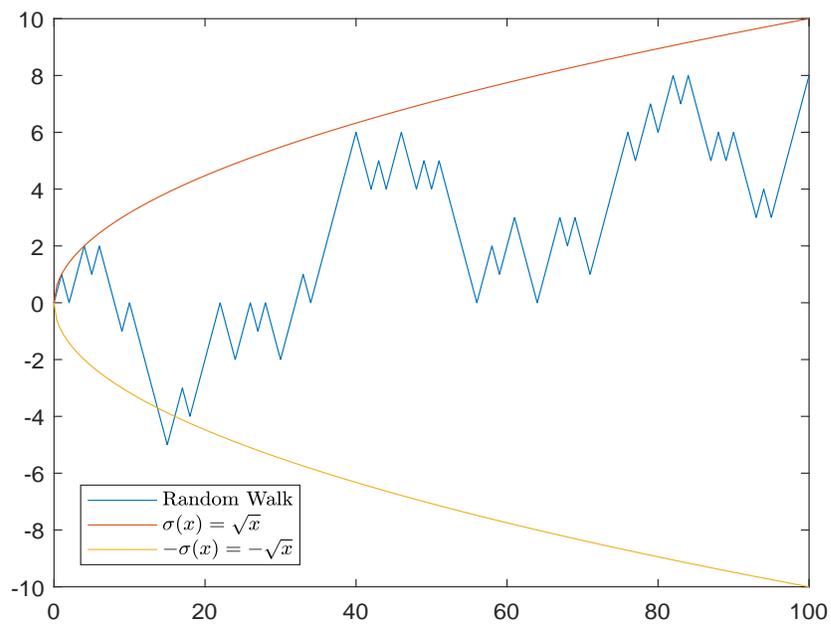


Figura 2.2: Evoluzione temporale di una random walk.

## 2.2 Introduzione alla Quantum Walk

La posizione della particella sulla retta al tempo  $t$  è descritta da un vettore  $|n\rangle$ ,  $n \in \mathbb{Z}$  in uno spazio di Hilbert  $\mathcal{H}_P$  di dimensione infinita una cui base computazionale è

$$\{|n\rangle, n \in \mathbb{Z}\}^1$$

In accordo al postulato 1, ogni sovrapposizione della forma  $\sum_n \alpha_n |n\rangle$  soggetta al vincolo  $\sum_n |\alpha_n|^2 = 1$  è uno stato valido per la posizione. L'evoluzione del sistema è condizionata dal lancio di una moneta, come nel caso classico. Se si ottiene 'testa' dopo il lancio della moneta, la particella si muoverà di una unità verso sinistra, viceversa se si ottiene 'croce' la particella si muoverà di una unità verso destra. Lo spazio di Hilbert del sistema è dunque

$$\mathcal{H} = \mathcal{H}_P \otimes \mathcal{H}_C$$

e una base computazionale di  $\mathcal{H}_C$  è data da  $\{|0\rangle, |1\rangle\}$ . In altri termini,  $|coin\rangle \in \mathcal{H}_C$  e un generico stato normalizzato della moneta può essere scritto come  $|coin\rangle = a|0\rangle + b|1\rangle$  con  $|a|^2 + |b|^2 = 1$ . L'evoluzione dello stato, cioè lo shift da  $|n\rangle$  in  $|n \pm 1\rangle$  è descritto da un operatore unitario, che indichiamo con  $A$ :

$$A|n\rangle|\epsilon\rangle = \begin{cases} |n+1\rangle|\epsilon\rangle & \text{se } \epsilon = 0 \\ |n-1\rangle|\epsilon\rangle & \text{se } \epsilon = 1. \end{cases}$$

L'operatore maggiormente utilizzato per la simulazione del lancio della moneta è l'*operatore di Hadamard*  $H$  (cf. 1.5), ricordiamo che  $H$  agisce sulla base computazionale come segue:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Consideriamo la particella posizionata inizialmente nell'origine  $|n=0\rangle$  e supponiamo ad esempio che il suo spin iniziale sia  $|0\rangle$  (*spin up*). Allora lo stato iniziale  $|\psi(t=0)\rangle$  è completamente descritto da

$$|\psi(0)\rangle = |n=0\rangle|0\rangle.$$

La quantum walk ha inizio appena viene applicato l'operatore  $I_P \otimes H$  allo stato iniziale ( $I_P$  è l'operatore identità dello spazio  $\mathcal{H}_P$ ), ovvero è realizzato il lancio della moneta; successivamente l'operatore di shift  $A$  modifica la posizione della particella come indicato dalla moneta. Possiamo esplicitare questo primo step:

$$|n=0\rangle|0\rangle \xrightarrow{I \otimes H} |n=0\rangle \left( \frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) = \frac{1}{\sqrt{2}} |n=0\rangle|0\rangle + \frac{1}{\sqrt{2}} |n=0\rangle|1\rangle$$

$$\xrightarrow{A} \frac{1}{\sqrt{2}} |n=1\rangle|0\rangle + \frac{1}{\sqrt{2}} |n=-1\rangle|1\rangle.$$

Se a questo punto eseguiamo una misura nella base computazionale, abbiamo probabilità  $1/2$  di osservare la particella in  $n=1$ , e probabilità  $1/2$  di osservarla in  $n=-1$ . Ritroviamo quindi il primo step della random walk classica. Con una immediata induzione si osserva che se ad ogni passo della quantum walk, dopo l'applicazione di  $U := A(I \otimes H)$  eseguiamo una misura proiettiva  $m$ , e dunque 'collassiamo' lo stato della particella, otteniamo proprio la random walk su  $\mathbb{Z}$ .

<sup>1</sup>Come già osservato in precedenza, qui eviteremo di trattare spazi di dimensione infinita restringendoci a  $T$  passi, dunque la base computazionale può essere definita come  $\{|n\rangle, n \in [-T, T]\}$ .

Per osservare qualcosa di nuovo e interessante, omettiamo dunque la misura del sistema ad ogni passo, inserendola solo una volta al termine dell'algoritmo. In questo modo, lo stato della particella al tempo  $t$  è descritto da

$$|\psi(t)\rangle = U^t |\psi(0)\rangle.$$

Riportiamo in tabella 2.1 le probabilità di osservare la particella in  $n \in \mathbb{Z}$  al tempo  $t$  per  $t = 0, \dots, 5$ , con stato iniziale  $|0\rangle|0\rangle$ .

$t \setminus n$	-5	-4	-3	-2	-1	0	1	2	3	4	5
0						1					
1					1/2		1/2				
2				1/4		1/2		1/4			
3			1/8		1/8		5/8		1/8		
4		1/16		1/8		1/8		5/8		1/16	
5	1/32		5/32		1/8		1/8		17/32		1/32

Tabella 2.1:  $p(t, n)$  per  $t = 0, \dots, 5$  con condizioni iniziali  $|0\rangle|0\rangle$ .

$t \setminus n$	-4	-3	-2	-1	0	1	2	3	4
0					1				
1				1/2		1/2			
2			1/4		1/2		1/4		
3		1/8		3/8		3/8		1/8	
4	1/16		12/32		1/8		12/32		1/16

Tabella 2.2:  $p(t, n)$  per  $t = 0, \dots, 4$  con condizioni iniziali  $|0\rangle \frac{|0\rangle - i|1\rangle}{\sqrt{2}}$ .

La distribuzione di probabilità di una quantum walk discreta non ha una forma analitica chiusa, ma può essere ottenuta mediante simulazioni numeriche (cf. sezione 2.4). Per esempio, i risultati ottenuti dopo 100 passi con stato iniziale  $|n=0\rangle|0\rangle$  sono riportati in figura 2.3. Si osserva la presenza di un picco massimo per  $n \sim 100/\sqrt{2}$ , e si può verificare che il picco conserva questa caratteristica anche per  $t$  molto maggiori, cioè si trova in  $n \sim t/\sqrt{2}$ . In questo picco, la probabilità è circa 10 volte maggiore rispetto all'origine, vale a dire che è molto probabile trovare la particella sempre più lontana dall'origine con il passare del tempo. Dalla figura si osserva dunque la sostanziale differenza con la camminata aleatoria classica, in cui la distribuzione Gaussiana ci garantiva una maggiore probabilità di trovare la particella nei dintorni dell'origine anche per  $t \gg 0$ .

Per capire perché la distribuzione della quantum walk tende verso destra, è sufficiente osservare che l'applicazione di un Hadamard allo stato  $|1\rangle$  introduce un segno  $-$ , mentre ciò non accade con lo stato  $|0\rangle$ . Ricordiamo che lo stato  $|0\rangle$  della moneta produce uno shift verso destra, mentre lo stato  $|1\rangle$  verso sinistra, e ciò produce l'asimmetria della distribuzione. In effetti, inizializzando il sistema nello stato  $|n=0\rangle|1\rangle$  si ottiene la stessa distribuzione riflessa rispetto all'asse  $y$ .

Per ottenere una distribuzione simmetrica, vorremo sovrapporre in qualche modo le due distribuzioni ottenute, evitando cancellazioni di termini. Questo può essere fatto con l'ausilio dell'unità immaginaria  $i$ , inizializzando la moneta nello stato  $\frac{|0\rangle - i|1\rangle}{\sqrt{2}}$ . Ciò che si

ottiene è riportato in figura 2.4, mentre in tabella 2.2 si possono osservare le probabilità esplicite nei primi 4 passi.

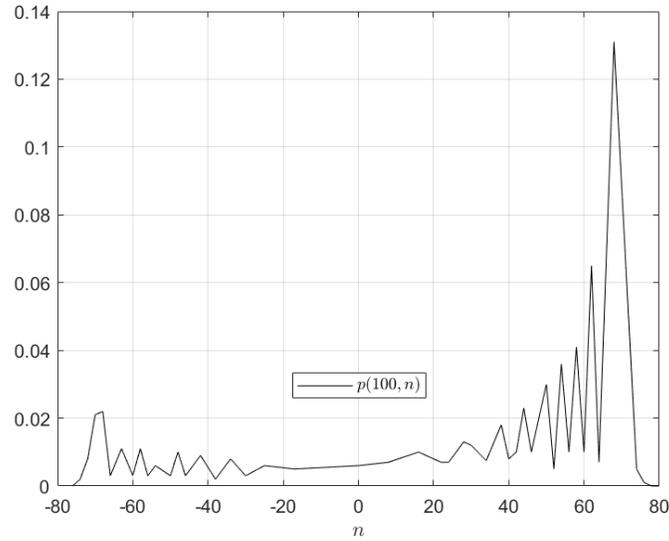


Figura 2.3: Distribuzione di probabilità dopo 100 passi di una QW con condizioni iniziali  $|n = 0\rangle |0\rangle$ .

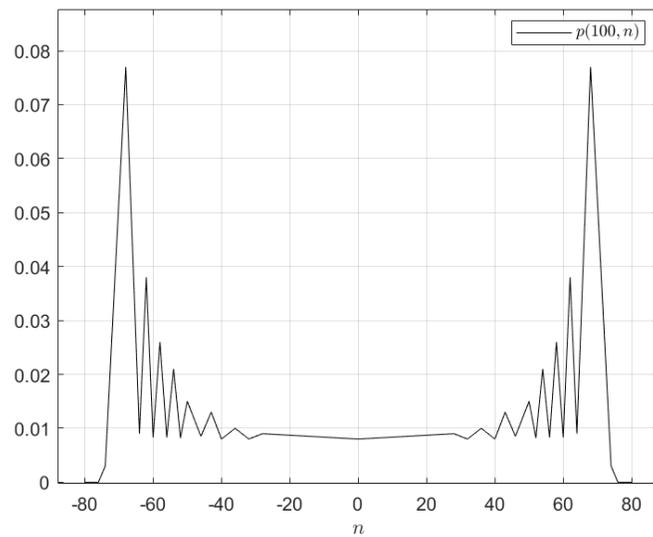


Figura 2.4: Distribuzione di probabilità dopo 100 passi di una QW con condizioni iniziali  $|n = 0\rangle \frac{|0\rangle - i|1\rangle}{\sqrt{2}}$ .

Se la distribuzione è simmetrica, come accade in figura 2.4, il valore atteso della posizione della particella, che si indica con  $\langle n \rangle$ , è 0. In questo caso è più agevole esplicitare la deviazione standard in funzione del tempo, infatti

$$\sigma(t) = \sqrt{\langle n^2 \rangle - \langle n \rangle^2} = \sqrt{\sum_{n=-\infty}^{+\infty} n^2 p(t, n)},$$

dove in analogia con il caso classico abbiamo indicato con  $p(t, n)$  la probabilità di trovarci in  $n$  al tempo  $t$ . Il calcolo esplicito del termine destro dell'equazione è piuttosto complesso, per questo ci limiteremo ad uno studio numerico di  $\sigma(t)$ . Utilizziamo dunque la varianza campionaria e mettiamo in un grafico i dati al variare di  $t = 0, \dots, 100$ . Quello che si ottiene è riportato in figura 2.5 ed è in accordo con quanto sperimentato in [5], cioè che

$$\sigma(t) \sim 0.54t.$$

In figura è riportata anche la deviazione standard nota della random walk classica,  $\sigma(t) = \sqrt{t}$ .

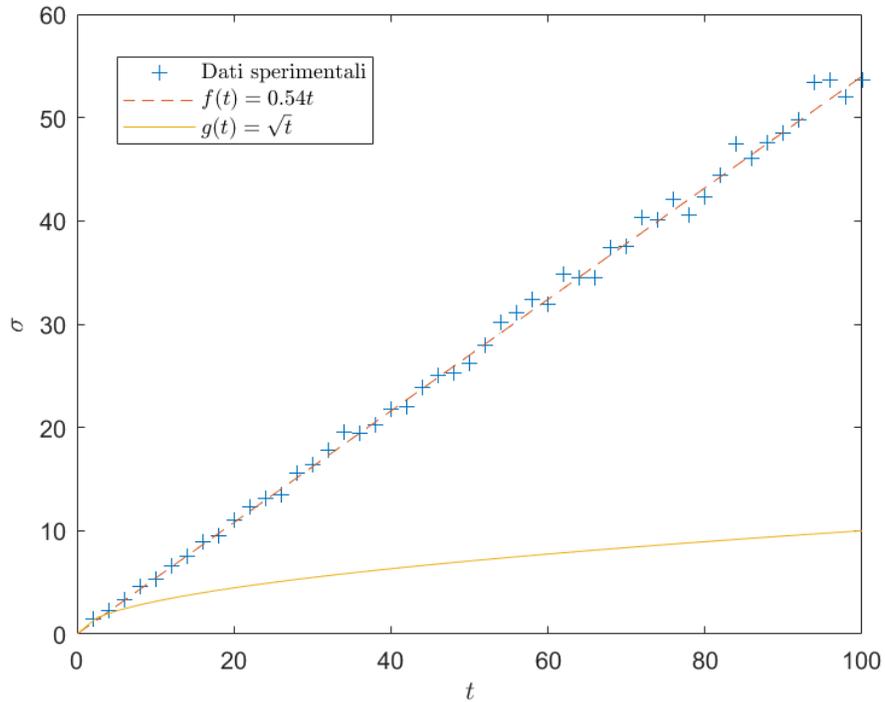


Figura 2.5: Deviazione standard di QW e RW a confronto.

## 2.3 Distribuzione della Quantum Walk

Abbiamo visto nella sezione 2.2 che una base computazionale associata alla quantum walk discreta è data da

$$\{|x, j\rangle : j \in \{0, 1\}, x \in \mathbb{Z}\}$$

nello spazio di Hilbert  $\mathcal{H} = \mathcal{H}_P \otimes \mathcal{H}_C$ . Lo stato della particella al tempo  $t$  è descritto dallo stato

$$|\psi(t)\rangle = \sum_{j=0}^1 \sum_{x \in \mathbb{Z}} \alpha_{x,j}(t) |x, j\rangle$$

dove le  $\alpha_{x,j}$  sono funzioni complesse e soddisfano la condizione di normalizzazione

$$\sum_{x \in \mathbb{Z}} |\alpha_{x,0}(t)|^2 + |\alpha_{x,1}(t)|^2 = 1.$$

Osserviamo comunque che per ogni  $t$ , solo un numero finito dei termini  $\alpha_{x,j}(t)$  è diverso da 0, dunque le somme sono finite.

Possiamo descrivere esplicitamente anche l'operatore di shift  $S$ , infatti

$$S = \sum_{j=0}^1 \sum_{x \in \mathbb{Z}} |x + (-1)^j, j\rangle \langle x, j|^2 \quad (2.3.1)$$

Studiamo adesso l'equazione di ricorrenza che lega i coefficienti  $\alpha_{x,j}$  dalla relazione  $|\psi(t+1)\rangle = S(I \otimes H) |\psi(t)\rangle$ :

$$\begin{aligned} |\psi(t+1)\rangle &= S(I \otimes H) \sum_{j=0}^1 \sum_{x \in \mathbb{Z}} \alpha_{x,j}(t) |x, j\rangle \\ &= \sum_{x \in \mathbb{Z}} S[\alpha_{x,0}(t) |x\rangle H |0\rangle + \alpha_{x,1}(t) |x\rangle H |1\rangle] \\ &= \sum_{x \in \mathbb{Z}} S[\alpha_{x,0}(t) |x\rangle \frac{|0\rangle + |1\rangle}{\sqrt{2}} + \alpha_{x,1}(t) |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}] \\ &= \sum_{x \in \mathbb{Z}} \frac{\alpha_{x,0}(t) + \alpha_{x,1}(t)}{\sqrt{2}} |x+1\rangle |0\rangle + \frac{\alpha_{x,0}(t) - \alpha_{x,1}(t)}{\sqrt{2}} |x-1\rangle |0\rangle \end{aligned} \quad (2.3.2)$$

D'altra parte vale che

$$|\psi(t+1)\rangle = \sum_{j=0}^1 \sum_{x \in \mathbb{Z}} \alpha_{x,j}(t+1) |x, j\rangle$$

Eguagliamo le somme termine a termine con un opportuno shift dell'indice muto  $x$  e otteniamo infine che

$$\begin{aligned} \alpha_{x,0}(t+1) &= \frac{\alpha_{x-1,0}(t) + \alpha_{x-1,1}(t)}{\sqrt{2}} \\ \alpha_{x,1}(t+1) &= \frac{\alpha_{x+1,0}(t) - \alpha_{x+1,1}(t)}{\sqrt{2}} \end{aligned} \quad (2.3.3)$$

con condizioni iniziali nel caso di inizializzazione in  $|0\rangle |0\rangle$ :

$$\alpha_{0,0}(0) = 1, \quad \alpha_{0,1}(0) = 0$$

<sup>2</sup>Si vede applicando  $S$  a un generico vettore della base computazionale  $|x', j'\rangle$ , infatti  $S|x', j'\rangle = |x' + (-1)^{j'}, j'\rangle$  dato che la base è ortonormale.

dove le relazioni valgono per  $|x| \leq t$  e  $x+t$  pari. Negli altri casi  $\alpha_{x,0}(t) = \alpha_{x,1}(t) = 0$ . Da queste equazioni per ricorrenza si può anche verificare la simmetria della distribuzione con la quantum walk inizializzata in  $|0\rangle|1\rangle$ , infatti detti  $\beta_{x,j}(t)$  i coefficienti definiti da (2.3.3) con condizioni iniziali

$$\beta_{0,0}(0) = 0, \quad \beta_{0,1}(0) = 1$$

si può verificare che vale

$$|\alpha_{x,0}(t)| = |\beta_{-x,1}(t)|, \quad |\alpha_{x,1}(t)| = |\beta_{-x,0}(t)|, \quad \forall x, t.$$

Per risolvere le (2.3.3), è necessario passare nella *base di Fourier*.

**Definizione 2.3.1** (Trasformata di Fourier Discreta). *Sia  $f : \mathbb{Z} \rightarrow \mathbb{C}$ . Si definisce trasformata di Fourier discreta (DTFT) di  $f$  la funzione*

$$\tilde{f}(\omega) = \sum_{x=-\infty}^{\infty} e^{-i\omega x} f(x),$$

purché  $\sum_x |f(x)| < \infty$ .

La trasformata inversa permette di ottenere la  $f$  originale a partire da  $\tilde{f}$ :

$$f(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i\omega x} \tilde{f}(\omega) d\omega$$

I coefficienti  $\alpha_{x,j}(t)$  introdotti nel paragrafo precedente sono funzioni discrete della variabile  $x$  e la loro DTFT è data da

$$\tilde{\alpha}_j(\omega, t) = \sum_{x=-\infty}^{\infty} e^{-i\omega x} \alpha_{x,j}(t) = \sum_{x=-\infty}^{\infty} e^{-i\omega x} \alpha_j(x, t) \quad (2.3.4)$$

Possiamo utilizzare la DTFT per trasformare la base computazionale  $\{|x\rangle\}$  definendo

$$|\tilde{\omega}\rangle = \sum_{x=-\infty}^{\infty} e^{i\omega x} |x\rangle \quad (2.3.5)$$

con  $\omega \in [-\pi, \pi]$  variabile continua. In questo modo abbiamo trasformato la base computazionale del sistema in una nuova base ortonormale in  $\mathcal{H} = L^2(-\pi, \pi) \otimes \mathbb{C}^2$

$$\{ |\tilde{\omega}\rangle |j\rangle : j \in \{0, 1\}, \omega \in [-\pi, \pi] \}$$

Questa nuova base è chiamata *base estesa di Fourier*. Vediamo come si può scrivere  $|\psi(t)\rangle$  nella nuova base:

$$\begin{aligned} |\psi(t)\rangle &= \sum_{j=0}^1 \sum_{x \in \mathbb{Z}} \alpha_{x,j}(t) |x\rangle |j\rangle \\ &= \sum_{j=0}^1 \sum_{x \in \mathbb{Z}} \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i\omega x} \tilde{\alpha}_j(\omega, t) d\omega |x\rangle |j\rangle \\ &= \frac{1}{2\pi} \sum_{j=0}^1 \int_{-\pi}^{\pi} \tilde{\alpha}_j(\omega, t) \left( \sum_{x \in \mathbb{Z}} e^{i\omega x} |x\rangle \right) |j\rangle d\omega \\ &= \frac{1}{2\pi} \sum_{j=0}^1 \int_{-\pi}^{\pi} \tilde{\alpha}_j(\omega, t) |\tilde{\omega}\rangle |j\rangle d\omega. \end{aligned} \quad (2.3.6)$$

Calcoliamo adesso l'azione dell'operatore di shift  $S$  sulla nuova base. Usando le relazioni (2.3.1) e (2.3.5), cioè le definizioni di  $S$  e di  $|\tilde{\omega}\rangle$ , otteniamo che

$$\begin{aligned}
S|\tilde{\omega}, j\rangle &= \sum_{x \in \mathbb{Z}} e^{i\omega x} S|x, j\rangle \\
&= \sum_{x \in \mathbb{Z}} e^{i\omega x} |x + (-1)^j, j\rangle \\
&= [x + (-1)^j \rightarrow x'] \\
&= \sum_{x' \in \mathbb{Z}} e^{i\omega x' - (-1)^j} |x', j\rangle \\
&= e^{-ik(-1)^j} |\tilde{\omega}, j\rangle
\end{aligned} \tag{2.3.7}$$

cioè  $|\tilde{\omega}, j\rangle$  è autovettore per  $S$  relativo all'autovalore  $\lambda = e^{-ik(-1)^j}$ . Vogliamo calcolare adesso l'azione dell'operatore  $U = S(I \otimes H)$  sulla nuova base di Fourier. Per prima cosa osserviamo che l'azione di  $H$  su uno stato  $|j'\rangle$ ,  $j' = 0, 1$  si può scrivere in notazione compatta come

$$H|j'\rangle = \sum_{j=0}^1 H_{j+1, j'+1} |j\rangle$$

dove  $H_{j+1, j'+1}$  denota l'entrata  $(j+1, j'+1)$  della matrice  $H$ . Infatti ricordiamo che

$$\begin{aligned}
H|0\rangle &= \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \\
H|1\rangle &= \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle
\end{aligned}$$

Dunque la formula è soddisfatta per ogni stato della base computazionale. L'azione di  $U$  sulla base è allora

$$\begin{aligned}
U|\tilde{\omega}', j'\rangle &= S(I \otimes H)|\tilde{\omega}', j'\rangle = S \sum_{j=0}^1 H_{j+1, j'+1} |\tilde{\omega}', j\rangle \\
&= \sum_{j=0}^1 e^{-ik(-1)^j} H_{j+1, j'+1} |\tilde{\omega}', j\rangle
\end{aligned} \tag{2.3.8}$$

e le entrate di  $U$  nella base di Fourier sono date da

$$\langle \tilde{\omega}, j | U | \tilde{\omega}', j' \rangle = e^{-i\omega(-1)^j} H_{j+1, j'+1} \delta_{\omega, \omega'} \tag{2.3.9}$$

Definiamo l'operatore  $\tilde{H}_\omega$  come segue:

$$\tilde{H}_\omega = \begin{bmatrix} e^{-i\omega} & 0 \\ 0 & e^{i\omega} \end{bmatrix} \cdot H = \frac{1}{\sqrt{2}} \begin{bmatrix} e^{-i\omega} & e^{-i\omega} \\ e^{i\omega} & -e^{i\omega} \end{bmatrix}$$

cioè in modo tale che

$$\tilde{H}_{j+1, j'+1}^{(\omega)} = e^{-i\omega(-1)^j} H_{j+1, j'+1}$$

Osserviamo adesso che se  $|\lambda_\omega\rangle$  è autovettore di  $\tilde{H}_\omega$  relativo all'autovalore  $\lambda_\omega$ , allora  $|\tilde{\omega}\rangle |\lambda_\omega\rangle$  è autovettore di  $U$  relativo all'autovalore  $\lambda_\omega$ , infatti

$$\lambda_\omega |\tilde{\omega}\rangle |\lambda_\omega\rangle = |\tilde{\omega}\rangle \lambda_\omega |\lambda_\omega\rangle = |\tilde{\omega}\rangle \left( \tilde{H}_\omega |\lambda_\omega\rangle \right) = U |\tilde{\omega}\rangle |\tilde{\lambda}\rangle \tag{2.3.10}$$

dove nell'ultima equazione si è usato che, per costruzione,  $\tilde{H}_\omega$  soddisfa

$$U |\tilde{\omega}\rangle |j\rangle = |\tilde{\omega}\rangle \left( \tilde{H}_\omega |j\rangle \right)$$

per ogni stato della base computazionale. Abbiamo mostrato che ogni autovalore di  $\tilde{H}_\omega$  è autovalore anche per  $U$ , e dunque abbiamo ridotto la diagonalizzazione di un operatore di dimensione infinita alla diagonalizzazione di un operatore di dimensione 2. Calcoliamo adesso il polinomio caratteristico di  $\tilde{H}_\omega$ :

$$p_{\tilde{H}_\omega}(t) = t^2 + (i\sqrt{2}\sin\omega)t - 1 \quad (2.3.11)$$

Per risolvere l'equazione definiamo un angolo  $\theta_\omega$  tale che

$$\theta_\omega \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \quad \sin\theta_\omega = \frac{1}{\sqrt{2}}\sin\omega \quad (2.3.12)$$

da cui segue

$$\begin{aligned} \sin^2\theta_\omega &= \frac{1}{2}\sin^2\omega \leftrightarrow \\ \cos^2\theta_\omega &= 1 - \sin^2\theta_\omega = 1 - \frac{1}{2}\sin^2\omega \end{aligned} \quad (2.3.13)$$

Dunque le soluzioni di (2.3.11) sono

$$\begin{aligned} t_{1,2}^{(\omega)} &= \frac{1}{2} \left( -i\sqrt{2}\sin\omega \pm \sqrt{4\left(1 - \frac{1}{2}\sin^2\omega\right)} \right) \\ &= -\frac{i}{\sqrt{2}}\sin\omega \pm \sqrt{\left(1 - \frac{1}{2}\sin^2\omega\right)} \end{aligned} \quad (2.3.14)$$

cioè

$$\begin{aligned} \lambda_\omega &= e^{-i\theta_\omega} \\ \mu_\omega &= e^{i(\pi+\theta_\omega)} \end{aligned} \quad (2.3.15)$$

Gli autovettori normalizzati associati ai due autovalori distinti si ottengono con un semplice conto che omettiamo per brevità, e sono

$$\begin{aligned} |\lambda_\omega\rangle &= \frac{1}{\sqrt{c_-}} [e^{-i\omega}, \sqrt{2}e^{-i\theta_\omega} - e^{-i\omega}]^T \\ |\mu_\omega\rangle &= \frac{1}{\sqrt{c_+}} [e^{-i\omega}, -\sqrt{2}e^{i\theta_\omega} - e^{-i\omega}]^T \end{aligned} \quad (2.3.16)$$

dove  $c_\pm$  sono le costanti di normalizzazione. Alla luce di quanto visto fino ad adesso siamo pronti per scrivere la decomposizione spettrale di  $U$ ,

$$\begin{aligned} U &= \frac{1}{2\pi} \int_{-\pi}^{\pi} (\lambda_\omega |\tilde{\omega}, \lambda_\omega\rangle \langle \tilde{\omega}, \lambda_\omega| + \mu_\omega |\tilde{\omega}, \mu_\omega\rangle \langle \tilde{\omega}, \mu_\omega|) d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \left( e^{-i\theta_\omega} |\tilde{\omega}, \lambda_\omega\rangle \langle \tilde{\omega}, \lambda_\omega| + e^{i(\pi+\theta_\omega)} |\tilde{\omega}, \mu_\omega\rangle \langle \tilde{\omega}, \mu_\omega| \right) d\omega \end{aligned} \quad (2.3.17)$$

e la  $k$ -esima potenza di  $U$  è data da

$$U^k = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left( e^{-i\theta_\omega k} |\tilde{\omega}, \lambda_\omega\rangle \langle \tilde{\omega}, \lambda_\omega| + e^{i(\pi+\theta_\omega)k} |\tilde{\omega}, \mu_\omega\rangle \langle \tilde{\omega}, \mu_\omega| \right) d\omega \quad (2.3.18)$$

Siamo pronti per risolvere analiticamente l'equazione che descrive lo stato di una quantum walk discreta al generico tempo  $t$ .

## Soluzioni Analitiche

Sia data una quantum walk con condizioni iniziali

$$|\psi(0)\rangle = |x=0\rangle |0\rangle$$

dalla decomposizione spettrale di  $U$  ottenuta in (2.3.18) segue che

$$\begin{aligned} |\psi(t)\rangle &= U^t |\psi(0)\rangle \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \left( e^{-i\theta_\omega t} |\tilde{\omega}, \lambda_\omega\rangle \langle \tilde{\omega}, \lambda_\omega | 0, 0\rangle + e^{i(\pi+\theta_\omega)t} |\tilde{\omega}, \mu_\omega\rangle \langle \tilde{\omega}, \mu_\omega | 0, 0\rangle \right) d\omega. \end{aligned} \quad (2.3.19)$$

Dalle definizioni di  $|\tilde{\omega}\rangle$ ,  $|\lambda_\omega\rangle$  e  $|\mu_\omega\rangle$  (cf. (2.3.5) e (2.3.16)) si ha che

$$\langle \tilde{\omega} | 0\rangle = 1, \quad \langle \lambda_\omega | 0\rangle = \frac{1}{\sqrt{c_-}} e^{i\omega}, \quad \langle \mu_\omega | 0\rangle = \frac{1}{\sqrt{c_+}} e^{i\omega}$$

da cui

$$\begin{aligned} \langle \tilde{\omega}, \lambda_\omega | 0, 0\rangle &= \frac{1}{\sqrt{c_-}} e^{i\omega} \\ \langle \tilde{\omega}, \mu_\omega | 0, 0\rangle &= \frac{1}{\sqrt{c_+}} e^{i\omega} \end{aligned} \quad (2.3.20)$$

Sostituiamo nella (2.3.19),

$$|\psi(t)\rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} |\tilde{\omega}\rangle \left( \frac{e^{-i\theta_\omega t + i\omega}}{\sqrt{c_-}} |\lambda_\omega\rangle + \frac{e^{i(\pi+\theta_\omega)t + i\omega}}{\sqrt{c_+}} |\mu_\omega\rangle \right) d\omega \quad (2.3.21)$$

Per concludere, scriviamo  $|\lambda_\omega\rangle$  e  $|\mu_\omega\rangle$  nella base computazionale, e uguagliamo termine a termine con la relazione (2.3.6) per trovare i coefficienti  $\tilde{\alpha}_j(\omega, t)$ .

$$\begin{aligned} |\psi(t)\rangle &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |\tilde{\omega}\rangle \left( \frac{e^{-i\theta_\omega t + i\omega}}{c_-} \left[ \sqrt{2} e^{-i\theta_\omega} - e^{-i\omega} \right] \right. \\ &\quad \left. + \frac{e^{i(\pi+\theta_\omega)t + i\omega}}{c_+} \left[ -\sqrt{2} e^{i\theta_\omega} - e^{-i\omega} \right] \right) d\omega \end{aligned} \quad (2.3.22)$$

Dopo aver sostituito opportunamente le costanti di normalizzazione  $c_\pm$  otteniamo che

$$\begin{aligned} \tilde{\alpha}_0(\omega, t) &= \frac{e^{-i\theta_\omega t}}{2} \left( 1 + \frac{\cos \omega}{\sqrt{1 + \cos^2 \omega}} \right) \\ &\quad + \frac{(-1)^t e^{i\theta_\omega t}}{2} \left( 1 - \frac{\cos \omega}{\sqrt{1 + \cos^2 \omega}} \right) \\ \tilde{\alpha}_1(\omega, t) &= \frac{e^{i\omega}}{2\sqrt{1 + \cos^2 \omega}} (e^{-i\theta_\omega t} - (-1)^t e^{i\theta_\omega t}) \end{aligned} \quad (2.3.23)$$

Utilizzando la trasformata inversa introdotta nel presente paragrafo riusciamo ad ottenere una formula compatta per i coefficienti  $\alpha_j(x, t)$ :

$$\begin{aligned} \alpha_0(x, t) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \left( 1 + \frac{\cos \omega}{\sqrt{1 + \cos^2 \omega}} \right) e^{-i(\theta_\omega t - \omega x)} d\omega \\ \alpha_1(x, t) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{e^{i\omega} e^{-i(\theta_\omega t - \omega x)}}{\sqrt{1 + \cos^2 \omega}} d\omega \end{aligned} \quad (2.3.24)$$

Ovviamente, ricordiamo che la formula vale solo per  $x + t$  pari (in tempi pari possiamo trovarci solo su interi pari, in tempi dispari solo su interi dispari). Se  $x + t$  è dispari,  $\alpha_0(x, t) = \alpha_1(x, t) = 0$ .

## 2.4 Implementazione in Cirq

Riportiamo qui il codice della Quantum Walk implementata con il pacchetto Cirq di Python ([6]). Il codice ne riprende uno presente nella documentazione ufficiale per la Quantum Walk su un grafo ed è stato modificato e riadattato al caso in esame. In particolare, consideriamo  $k + 1$  qubit,  $k$  per lo spazio  $\mathcal{H}_P$ , che rappresentano la posizione della particella, e uno per lo spazio  $\mathcal{H}_C$ , la moneta. Gli stati di estremali ( $|0\rangle$  e  $|1\rangle$ ) di un qubit di  $\mathcal{H}_P$  simulano una cifra della rappresentazione binaria della posizione della particella (cf. 1.5), e sono enumerati da 1 a  $k$ , in ordine decrescente di significatività. Il passo in avanti, ovvero l'addizione di 1, è implementato coerentemente con l'addizione di un'unità in rappresentazione binaria, con l'ausilio di una catena di *controlled-X*. Similmente la sottrazione. Per ultimo, è necessario osservare che per non incappare in rappresentazioni binarie di interi negativi, che sarebbe necessario gestire tramite metodi quali il complemento a due, si parte in posizione  $a$  (indicata nel codice con *trasl*, e possibile ottenere mediante gate  $X$  applicati inizialmente sui qubit opportuni) e si fa attenzione ad impostare un numero di passi inferiore o uguale ad  $a$ . Infine, tramite la funzione implementata appositamente `bin_to_int_minus_trasl`, lo stato verrà convertito in un intero e gli sarà sottratto  $a$ , in modo da ottenere la posizione della particella come se si fosse lavorato con una Quantum Walk inizializzata in  $z = 0$ .

```
[1]: import cirq
```

```
[2]: number_qubits = 9
qubits = cirq.GridQubit.rect(1, number_qubits)

# coefficiente additivo di traslazione
trasl = 100

#FUNZIONI AUSILIARIE

# Conversione della misura ottenuta in intero e traslazione
def bin_to_int_minus_trasl(bits):
    a = 0
    for i in range(number_qubits):
        a = a + ( ( 2** ( number_qubits-i-1 ) ) * bits[i] )
    return a - trasl

# Creazione di un gate G t.c.  $G|0\rangle = 1/\sqrt{2} (|0\rangle - i|1\rangle)$ 
class Zero_Average(cirq.Gate):
    def __init__(self):
        super(Zero_Average, self)

    def _num_qubits_(self):
        return 1

    def _unitary_(self):
        return np.array([
            [1.0, 1.0],
            [0.-1j, 0.+1j]
        ]) / np.sqrt(2)

    def _circuit_diagram_info_(self, args):
        return "G"
```

```

zero_average = Zero_Average()

#-----

def initial_state():

    #inizializzazione a trasl = 100 (0 dopo traslazione) ->
    →|0001100100>
    yield cirq.X.on(cirq.GridQubit(0, 3))
    yield cirq.X.on(cirq.GridQubit(0, 4))
    yield cirq.X.on(cirq.GridQubit(0, 7))

    #inizializzazione moneta allo stato 1/sqrt(2) (|0> -i|1>)
    #yield zero_average.on(cirq.GridQubit(0, number_qubits))

    #inizializzazione moneta allo stato -|1>
    #yield cirq.X.on(cirq.GridQubit(0, number_qubits))
    #yield cirq.Z.on(cirq.GridQubit(0, number_qubits))

    #default:: inizializzazione moneta allo stato |0>

def walk_step():

    #Lancia la moneta
    yield cirq.H.on(cirq.GridQubit(0, number_qubits))

    #Operazione di addizione
    yield cirq.X.on(cirq.GridQubit(0, number_qubits))

    for i in range(number_qubits, 0, -1):

        controls = [cirq.GridQubit(0, v) for v in range(number_qubits,
→i-1, -1)]
        yield cirq.X.on(cirq.GridQubit(0, i-1)).
→controlled_by(*controls)
        if (i > 1):
            yield cirq.X.on(cirq.GridQubit(0, i-1))

        yield cirq.X.on(cirq.GridQubit(0, number_qubits))

    #Operazione di sottrazione
    for i in range(1, number_qubits+1):

        controls = [cirq.GridQubit(0, v) for v in range(number_qubits,
→i-1, -1)]
        yield cirq.X.on(cirq.GridQubit(0, i-1)).
→controlled_by(*controls)
        if (i < number_qubits):
            yield cirq.X.on(cirq.GridQubit(0, i))

iterator = 100
rep_number = 100

```

```

def generate_walk(number_qubits, iterator, rep_number):

    circuit = cirq.Circuit()
    circuit.append(initial_state())
    for j in range(0, iterator):
        circuit.append(walk_step())
    circuit.append(cirq.measure(*qubits, key='x'))
    #simulazione
    simulator = cirq.Simulator()
    result = simulator.run(circuit, repetitions = rep_number)
    final = result.histogram(key='x', fold_func =_
↪bin_to_int_minus_trasl)
    print(final)
    return final

    #-----

final = generate_walk(number_qubits, iterator, rep_number)

```

```

Counter({68:10, 66:8, 62:7, 70:5, -70:5, 28:5, 58: 5, 50: 4, -68:4, 44:
↪4, 46:4, 36:4, -58:2, -28:2, 4:2, 10:2, -62:2, 32:2, 54:2, -48:1, -54:
↪1, 2:1, -18:1, 0:1, -20:1, -64:1, -72:1, 64:1, 72:1, -40:1, -22:1, 22:
↪1, 24:1, 60:1, -52:1, 40:1, 20:1, -34:1, 34:1, -44:1})

```

---

Per visualizzare i risultati, è necessario importare i pacchetti matematici opportuni:

```

[1]: import math
import numpy as np
import matplotlib.pyplot as plt

```

Aggiungiamo infine il codice per ottenere il grafico e l'istogramma associati a `rep_number` iterazioni della quantum walk:

```

[2]: a = np.bincount(np.array(list(final.elements()))+iterator)
plt.figure()
plt.plot(np.arange(-iterator, len(a)-iterator), a)
plt.xticks(np.arange(-80, 81, 20))
plt.xlim(-70, 71)
#-----
plt.figure()
plt.hist(list(final.elements()), width = 3)

```

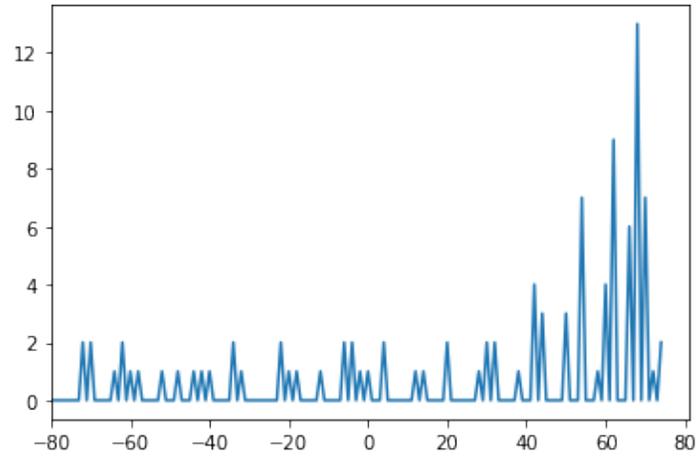
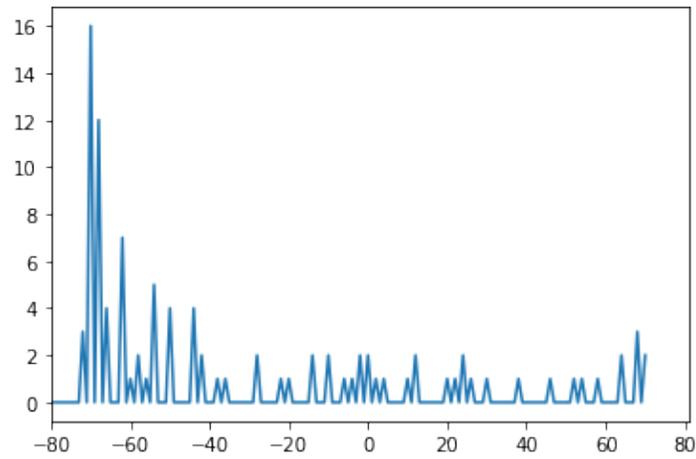
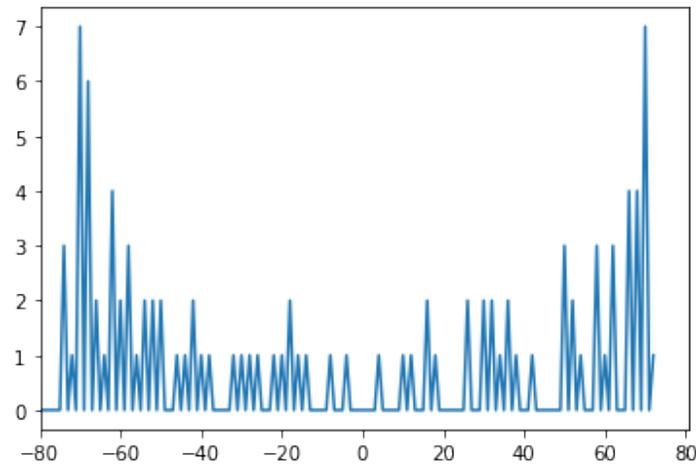
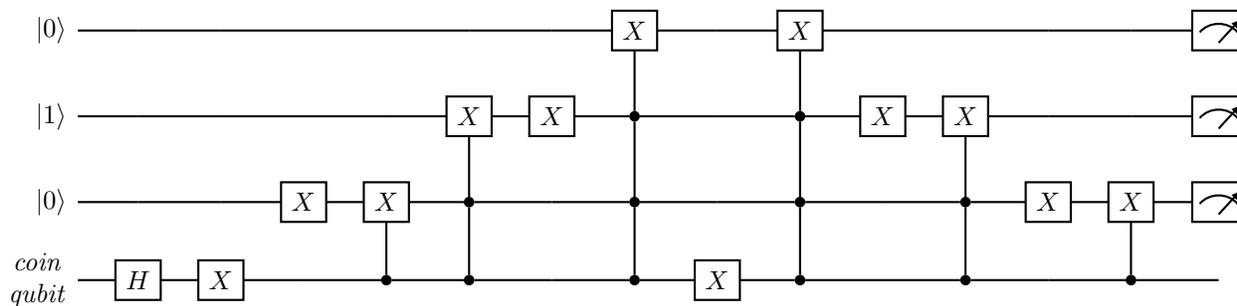
(a)  $|\psi(0)\rangle = |n=0\rangle |0\rangle$ (b)  $|\psi(0)\rangle = |n=0\rangle (-|1\rangle)$ (c)  $|\psi(0)\rangle = |n=0\rangle \frac{|0\rangle - i|1\rangle}{\sqrt{2}}$ 

Figura 2.6: 100 simulazioni di una quantum walk dopo 100 passi, ciascuna con condizioni iniziali indicate.



Riportiamo qui sopra una rappresentazione grafica del circuito utilizzato per implementare una quantum walk. Per semplicità sono stati utilizzati 4 qubit, compresa la moneta, ed è stato effettuato un solo passo dell'algoritmo. Si osservi che lo stato iniziale in  $\mathcal{H}_P$  è  $|010\rangle$ , che corrisponde a  $n = 2$  in conversione decimale, e pertanto nell'implementazione dovrà essere effettuato uno shift di due unità con la modifica del parametro `trasl`.

## 2.5 Noisy Quantum Walk

La quantum walk, così come altri sistemi quantistici, può essere affetta da rumore. Per semplicità ed efficienza computazionale supporremo sempre che il rumore coinvolga solo il *coin qubit*.

### Bit-Flip Channel

In accordo con quanto studiato nella sezione 1.5, vediamo inizialmente la perturbazione che un *bit-flip channel* comporta nel nostro sistema. Supponiamo che ad ogni passo della Quantum Walk, al *coin qubit* venga applicato il gate di Pauli  $X$  con probabilità  $p$ , e sia lasciato invariato con probabilità  $1 - p$ . Riportiamo qui il codice necessario all'implementazione del *noise*, e i risultati sperimentali al variare di quattro parametri diversi di  $p$ :

```
[1]: import cirq
bit_flip = cirq.bit_flip(p=0.1)

[...]

yield bit_flip.on(cirq.GridQubit(0, number_qubits))
```

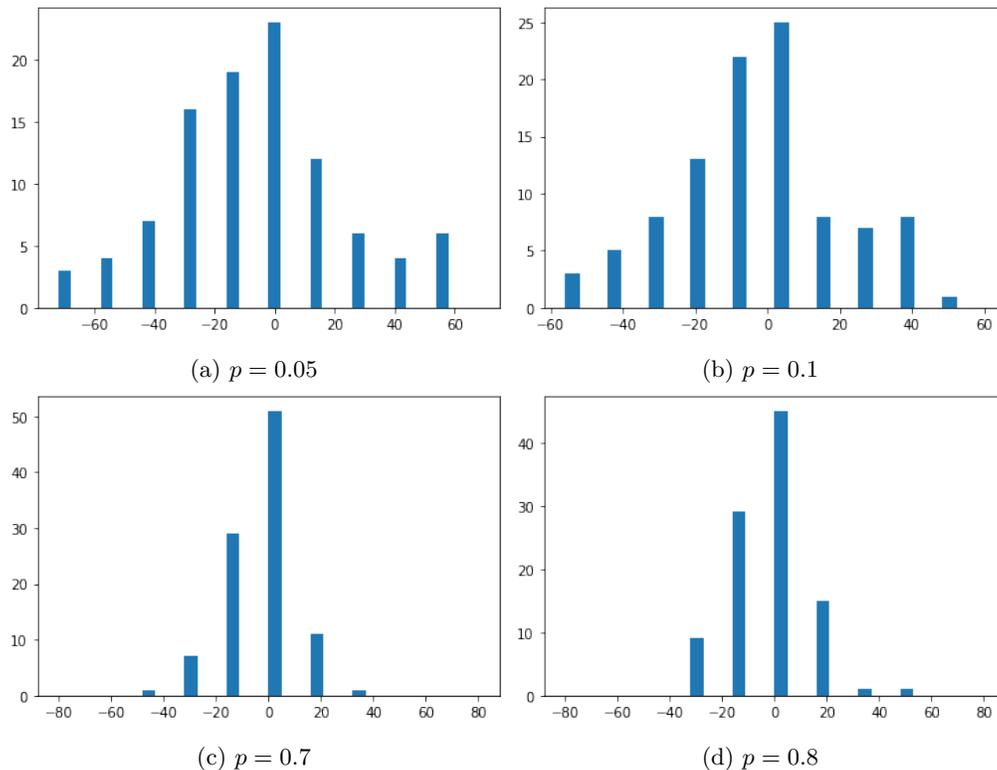


Figura 2.7: Distribuzione delle QW con *bit-flip* per parametri di  $p$  indicati.

## Asymmetric-Depolarizing Channel

Vediamo adesso l'effetto dell'*Asymmetric-Depolarizing Channel*, che applica le matrici di Pauli  $X$ ,  $Y$  o  $Z$  rispettivamente con probabilità  $p_x$ ,  $p_y$  e  $p_z$ . Si osservi che per  $p_y = p_z = 0$  si ritrova il caso precedente.

```
[1]: import cirq
depo = cirq.asymmetric_depolarize(
    p_x=0.10,
    p_y=0.05,
    p_z=0.15,
)

[...]

yield depo.on(cirq.GridQubit(0, number_qubits))
```

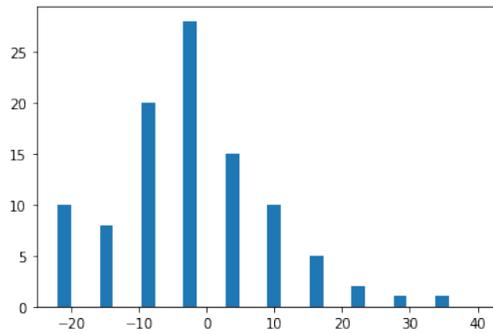
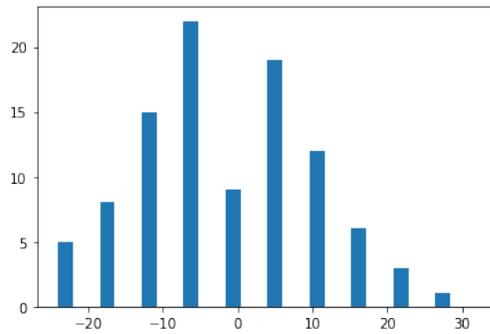
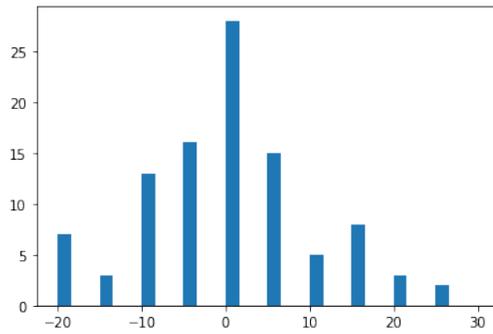
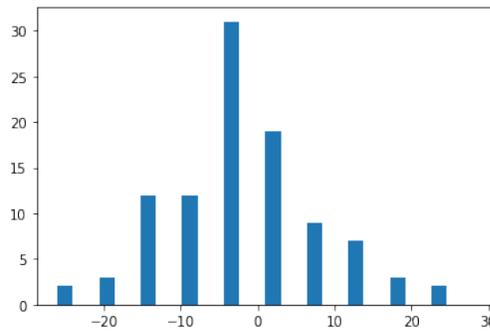
(a)  $p_x = 0.10$ ,  $p_y = 0.05$ ,  $p_z = 0.15$ (b)  $p_x = 0.20$ ,  $p_y = 0.10$ ,  $p_z = 0.20$ (c)  $p_x = 0.30$ ,  $p_y = 0.05$ ,  $p_z = 0.30$ (d)  $p_x = 0.05$ ,  $p_y = 0.35$ ,  $p_z = 0.50$ 

Figura 2.8: Distribuzione delle QW con *Asymmetric-Depolarization* per parametri di  $p_x$ ,  $p_y$ ,  $p_z$  indicati.

## Misura Intermedia

Il principale tipo di rumore che vogliamo studiare, è quello derivato dalla misura intermedia del qubit, ovvero effettuata prima della terminazione del circuito. Fissato un tempo  $t \in [0, t_{max})$ , supponiamo che al passo  $t$  la moneta sia misurata con probabilità 1. È bene precisare che Cirq nella versione attuale (0.13.1) non consente di effettuare misure intermedie nel modo standard, ed è quindi indispensabile utilizzare gli *operatori di Kraus* (cf. 1.4). La sintassi per la costruzione di un operatore che simula la misura intermedia di un qubit è la seguente:

```
[1]: import cirq
import numpy as np

[2]: ops = [
    np.array([[0, 0], [0, 1]]),
    np.array([[1, 0], [0, 0]]),
]
mid_meas = cirq.KrausChannel(ops)
```

Per prima cosa possiamo verificare la correttezza dell'operatore. Infatti, se ad ogni passo dell'algoritmo effettuiamo la misura della moneta dopo l'applicazione di  $H$ , è naturale aspettarci il comportamento della Random Walk classica, come già osservato nella sezione 2.2.

```
[3]: def walk_step():
    #Lancia la moneta
    yield cirq.H.on(cirq.GridQubit(0, number_qubits))

    #misura la moneta
    yield mid_meas.on(cirq.GridQubit(0, number_qubits))

    ...
```

I risultati osservati sono riportati in figura 2.9, e sono coerenti con quelli attesi.

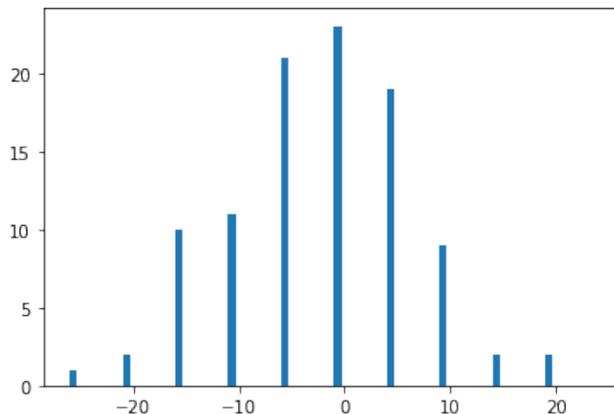


Figura 2.9: Istogramma di 100 simulazioni di QW con misura del *coin qubit* ad ogni passo dopo l'applicazione di  $H$ .

Studiamo adesso il caso della misura della moneta al primo passo.

Sia  $|\psi\rangle = |n=0\rangle|0\rangle$  lo stato iniziale della quantum walk. Supponiamo di effettuare una misura del *coin qubit* subito dopo l'applicazione di  $H$ . Il primo passo dell'algoritmo è dunque il seguente:

$$|n=0\rangle|0\rangle \xrightarrow{I \otimes H} |0\rangle \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \xrightarrow{\text{Misura}} \begin{cases} |0\rangle|0\rangle \xrightarrow{A} |1\rangle|0\rangle & \text{con probabilità } p = 1/2 \\ |0\rangle|1\rangle \xrightarrow{A} |-1\rangle|1\rangle & \text{con probabilità } p = 1/2 \end{cases}$$

Dopo il primo passo, possiamo trovarci con uguale probabilità in una delle due situazioni seguenti:

1. Una QW con la particella inizialmente in posizione  $z = 1$  e spin iniziale *up*
2. Una QW con la particella inizialmente in posizione  $z = -1$  e spin iniziale *down*

Fissato a priori un tempo  $t_{max}$  finito di passi, in entrambi i casi le nuove passeggiate avranno un tempo  $t_{max} - 1$ . Come già osservato, nel primo caso la distribuzione tende verso destra (figura 2.6a), mentre nel secondo caso la distribuzione è la stessa riflessa lungo le ordinate e dunque tende verso sinistra (figura 2.6b). Effettuare  $N \gg 0$  prove di questo processo corrisponde dunque ad eseguire  $k \sim N/2$  volte la camminata 1 e  $N - k \sim N/2$  volte la camminata 2. È naturale dunque aspettarci una distribuzione che interpoli queste due, e dunque ricalchi a meno di una traslazione di 1 unità la quantum walk a distribuzione simmetrica ottenuta inizializzando lo stato in  $\frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$ . I risultati osservati confermano quelli teorici e sono riportati in figura 2.10.

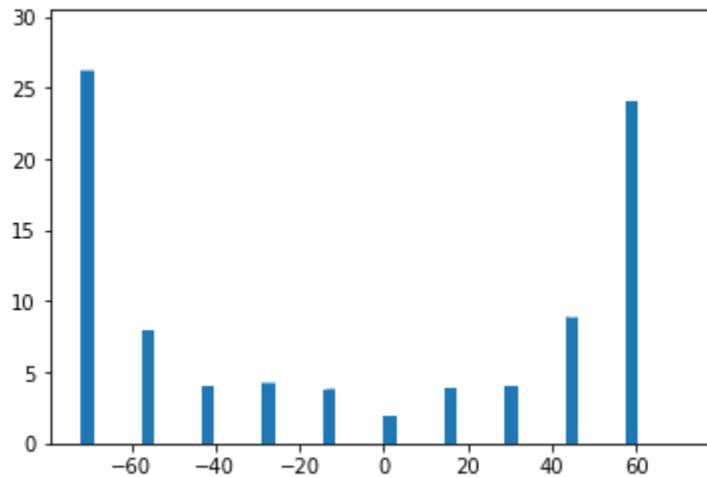


Figura 2.10: Istogramma di 100 simulazioni di QW con misura del *coin qubit* al primo passo dopo l'applicazione di  $H$ .

Per ultimo, studiamo come varia la distribuzione in funzione dell'istante di tempo in cui si effettua la misura. Come si osserva dalla figura 2.11 (a pagina successiva), l'effetto della misura al tempo  $t \leq t_{max}$  comporta una perturbazione nella distribuzione dipendente da  $t$ ; la distribuzione della QW (fig. 2.3) è interpolata ad un nuovo picco che si aggiunge inizialmente nel semiasse negativo delle ascisse, e che si sposta verso destra crescendo di intensità con il crescere di  $t$  fino a  $t \sim t_{max}/2$ . Il nuovo picco cattura proporzionalmente alla sua altezza il picco destro della distribuzione senza rumore, e

raggiunge la sua massima intensità in corrispondenza di  $t \sim t_{max}/2$ , dove cattura quasi completamente il picco destro e la misura riesce a far concludere circa il 30% delle QW precisamente in  $n = 0$ . Successivamente continua a spostarsi verso destra e la sua intensità decresce, fino ad inglobarsi completamente nel picco destro per  $t$  vicino a  $t_{max}$ .

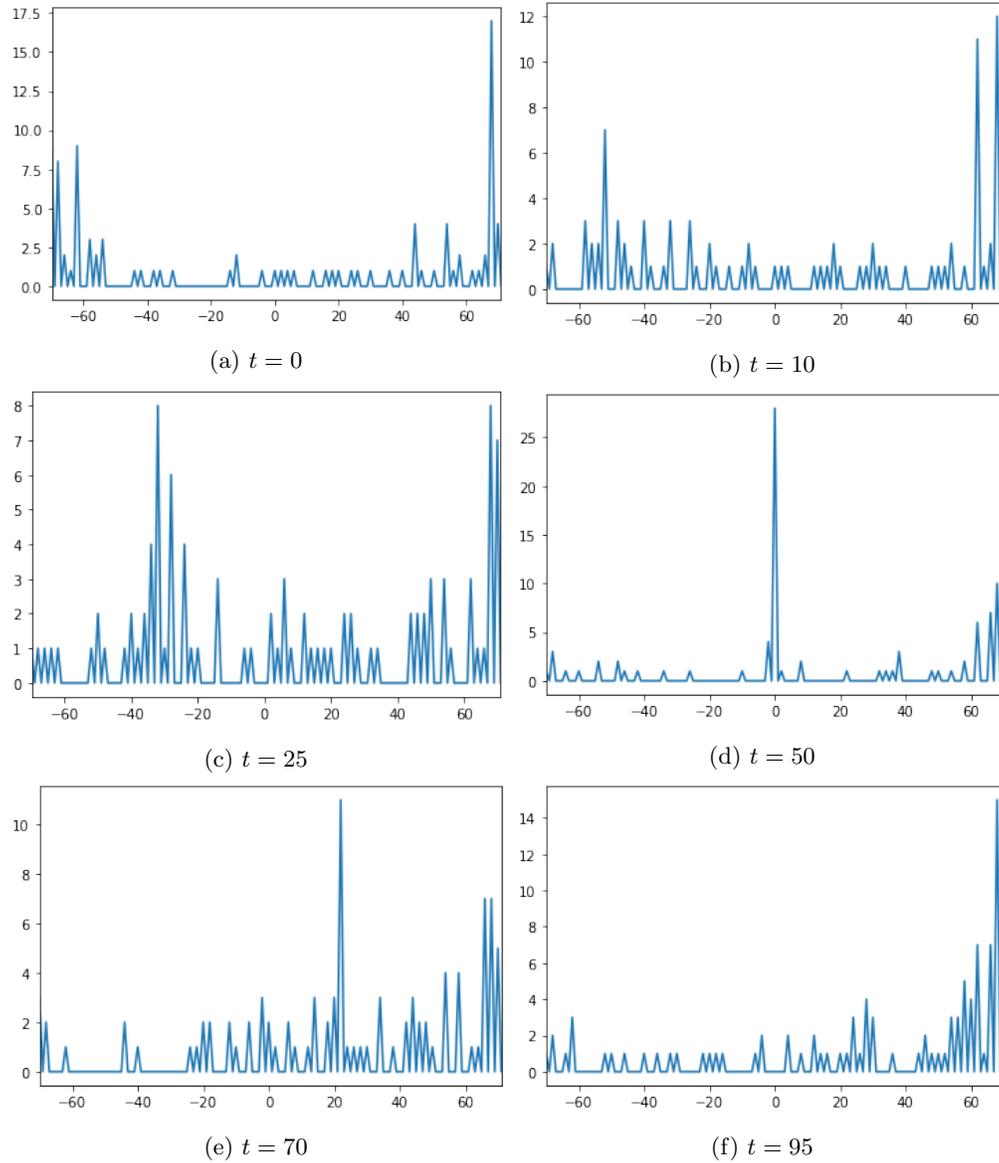


Figura 2.11: Ciascun grafico rappresenta 100 simulazioni di una QW con misura intermedia del *coin qubit* al tempo  $t$  indicato.

## Capitolo 3

# Algoritmo di Grover

### 3.1 Introduzione

Sia  $n \geq 1$ ,  $x_0 \in \{0, 1\}^n$  e si consideri una funzione  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  definita nel modo seguente:

$$f(x) = \begin{cases} 1 & \text{se } x = x_0 \\ 0 & \text{altrimenti} \end{cases}$$

Lo scopo dell'algoritmo di Grover è trovare l'unico  $\mathbf{x}_0$  in cui la funzione è non nulla. È noto che un computer classico richiede  $O(N)$  operazioni per la risoluzione del problema, dove  $N = 2^n = \#\{0, 1\}^n$ , in quanto nel caso pessimo si deve analizzare tutto il dominio di  $f$ . L'algoritmo di Grover risolve il problema in  $O(\sqrt{N})$  con probabilità  $p > 1/2$ , ovvero presenta una velocità quadratica rispetto ad un algoritmo classico, ed è stato inoltre provato che è ottimo, nel senso che nessuna macchina di Turing può risolvere il problema in meno di  $O(\sqrt{N})$  operazioni.

Si supponga per i nostri scopi di disporre di un oracolo, una struttura il cui funzionamento interno può non essere noto ma che è in grado di *riconoscere* una soluzione al problema di ricerca (una tale struttura è detta *black box*, scatola nera). Questo riconoscimento è segnalato da un *ancilla qubit*  $q$ . Più formalmente, detto  $O$  l'oracolo, nel nostro caso un operatore unitario, esso è definito dalla seguente operazione:

$$|\mathbf{x}\rangle |q\rangle \xrightarrow{O} |\mathbf{x}\rangle |f(\mathbf{x}) \oplus q\rangle$$

dove  $|\mathbf{x}\rangle = |x_1 \dots x_n\rangle$  e  $\oplus$  è la somma modulo 2. L'ancilla  $|q\rangle$  è invertita se l'oracolo individua  $|\mathbf{x}\rangle$  come soluzione, altrimenti è lasciata invariata. Nel caso banale, potremmo verificare la validità di una soluzione semplicemente preparando lo stato  $|\mathbf{x}\rangle |0\rangle$  e osservando dopo l'applicazione di  $O$  quando avviene il cambiamento dello stato  $|0\rangle$  in  $|1\rangle$ . Si noti che nessun risparmio operativo sarebbe ricavato da questo approccio.

La prima cosa fondamentale da osservare è cosa accade all'ancilla dopo l'applicazione di  $O$  se la inizializziamo nello stato

$$|q\rangle = |-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}} = H|1\rangle$$

In tal caso, infatti, se  $\mathbf{x} = \mathbf{x}_0$  lo stato risultante differisce da  $|-\rangle$  di una fase globale  $-1$ , altrimenti rimane invariato.

$$|\mathbf{x}\rangle |-\rangle \xrightarrow{O} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle |-\rangle$$

Questo ci permette di semplificare lo studio omettendo l'ancilla  $q$ ; con questa convenzione l'evoluzione dello stato dopo l'applicazione dell'oracolo può essere compattata come segue:

$$|\mathbf{x}\rangle \xrightarrow{O} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle$$

e in questo modo  $O$ , può anche essere esplicitato:

$$O = -|\mathbf{x}_0\rangle\langle\mathbf{x}_0| + \sum_{\mathbf{x} \neq \mathbf{x}_0} |\mathbf{x}\rangle\langle\mathbf{x}| = I - 2|\mathbf{x}_0\rangle\langle\mathbf{x}_0|$$

La puntualizzazione necessaria da fare adesso riguarda il ruolo e l'implementazione di  $O$ . In effetti, può sembrare controintuitivo l'utilizzo di un operatore che già conosce la soluzione  $\mathbf{x}_0$  per costruire un algoritmo che deve trovare questa stessa soluzione. Il punto chiave è tuttavia la significativa differenza che intercorre tra *conoscere* e *riconoscere* una soluzione. Per fissare le idee, possiamo pensare alla fattorizzazione in primi di un numero naturale. Supponiamo che ci venga fornito in input un intero  $a$  e ci venga detto che  $a = p_1 p_2$  con  $p_1$  e  $p_2$  primi (eventualmente distinti). Siamo interessati a trovare questi due primi. In questo esempio, l'azione dell'oracolo  $O$  è prendere in input uno stato  $|x\rangle$ , operare la divisione di  $a$  per  $x$ , e invertire l'ancilla se il resto della divisione è 0 e  $x$  è primo, o lasciarla invariata in caso contrario. Una volta trovato il primo più piccolo  $p_1$ , l'altro si trova immediatamente,  $p_2 = a/p_1$ . Al di là dell'implementazione effettiva di  $O$ , che va oltre i nostri scopi, è immediato osservare come esso non conosca a priori  $p_1$ , ma sia tuttavia in grado di svolgere le funzioni richieste dal programma. E' per questo che potremmo supporre senza perdere di generalità di conoscere già  $\mathbf{x}_0$  prima che l'algoritmo abbia inizio.

## 3.2 Procedura

Il sistema viene inizializzato allo stato  $|s\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle$ <sup>1</sup>. Questo può essere fatto applicando  $H^{\otimes n}$  allo stato iniziale  $|0\rangle^{\otimes n}$ ; questo stato rappresenta la sovrapposizione di tutti gli stati possibili ciascuno con la stessa ampiezza.

L'algoritmo di Grover consiste essenzialmente nell'applicazione ripetuta dell'operatore

$$H^{\otimes n} P H^{\otimes n} O$$

dove

$$P|\mathbf{x}\rangle = \begin{cases} |\mathbf{x}\rangle & \text{se } |x\rangle = |0\rangle \\ -|\mathbf{x}\rangle & \text{altrimenti} \end{cases}$$

Cioè il terzo gate esegue uno shift di fase a tutti gli stati eccetto lo  $|0\rangle$ , e dunque

$$P = 2|0\rangle\langle 0| - I$$

da cui segue che la scrittura dell'applicazione degli ultimi tre gate può essere scritta in forma compatta:

$$H^{\otimes n} P H^{\otimes n} = H^{\otimes n} (2|0\rangle\langle 0| - I) H^{\otimes n} = 2|s\rangle\langle s| - I$$

dove si è usato che  $|s\rangle = H^{\otimes n} |0\rangle$  e il fatto che  $H^{\otimes n}$  è unitario.<sup>2</sup>

Definendo  $\mathcal{R}_s := 2|s\rangle\langle s| - I$ , l'iterazione di Grover è completamente descritta dall'operatore  $G := \mathcal{R}_s O$ . Verifichiamo infine qual è l'azione di  $\mathcal{R}_s$  su un generico stato  $\sum_j a_j |j\rangle$ :

$$(2|s\rangle\langle s| - I) \sum_j a_j |j\rangle = \sum_j (2\langle a| - a_j) |j\rangle$$

dove  $\langle a| = \frac{\sum_j a_j}{N}$  indica la media delle ampiezze negli stati di base. Dall'equazione si osserva come l'ampiezza di ciascuno stato  $|j\rangle$  sia riflessa rispetto alla media delle ampiezze.

<sup>1</sup>Indichiamo con  $|j\rangle$  la conversione di  $j$  in stringa binaria, eventualmente preceduta da degli zeri iniziali in modo che la lunghezza della stringa sia  $n$ ; ad esempio, se disponiamo di 4 qubit,  $|3\rangle$  rappresenta lo stato  $|0011\rangle$ .

<sup>2</sup>Svolgiamo il calcolo per completezza:  $H^{\otimes n} (2|0\rangle\langle 0| - I) H^{\otimes n} = (2H^{\otimes n} |0\rangle\langle 0| - H^{\otimes n}) H^{\otimes n} = 2|s\rangle\langle s| - H^{\otimes n} H^{\otimes n} = 2|s\rangle\langle s| - I$ .

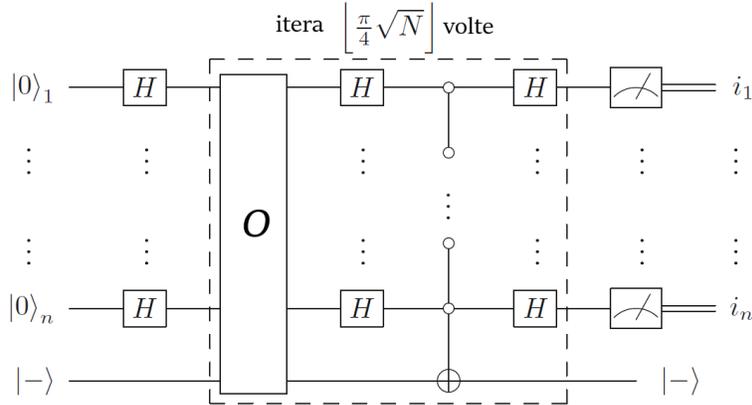


Figura 3.1: Circuito di Grover

### 3.3 Interpretazione Grafica

Verifichiamo per prima cosa che  $O$  è una riflessione; con un ragionamento analogo seguirà che lo stesso vale per  $\mathcal{R}_s$ . Definiamo dunque i sottospazi vettoriali di  $\mathcal{H}^N$ :

$$\mathcal{V} := \text{Span}\{|\mathbf{x}_0\rangle\}$$

$$\mathcal{W} := \text{Span}\{|\mathbf{x}\rangle : \mathbf{x} \neq \mathbf{x}_0\}$$

Si verifica immediatamente che  $\dim\mathcal{V} = 1$ ,  $\dim\mathcal{W} = N - 1$  e  $\mathcal{V} \perp \mathcal{W}$ . In altre parole abbiamo decomposto  $\mathcal{H}^N$  in una somma diretta ortogonale; per verificare che  $O$  è la riflessione rispetto a  $\mathcal{W}$  procediamo nel seguente modo: sia  $|\psi\rangle \in \mathcal{H}^N$ . Allora  $\exists |\psi_v\rangle \in \mathcal{V}$ ,  $\exists |\psi_w\rangle \in \mathcal{W}$  tali che  $|\psi\rangle = |\psi_v\rangle + |\psi_w\rangle$ . Poiché  $|\psi_v\rangle \perp |\psi_w\rangle$ :

$$O|\psi\rangle = O(|\psi_v\rangle + |\psi_w\rangle) = O|\psi_v\rangle + O|\psi_w\rangle = -|\psi_v\rangle + |\psi_w\rangle$$

Allo stesso modo, dalla definizione di  $\mathcal{R}_s$  segue che  $\mathcal{R}_s|s\rangle = |s\rangle$  e  $\mathcal{R}_s|s^\perp\rangle = -|s^\perp\rangle$ . Abbiamo dimostrato dunque che  $\mathcal{R}_s$  è la riflessione rispetto a  $|s\rangle$ .

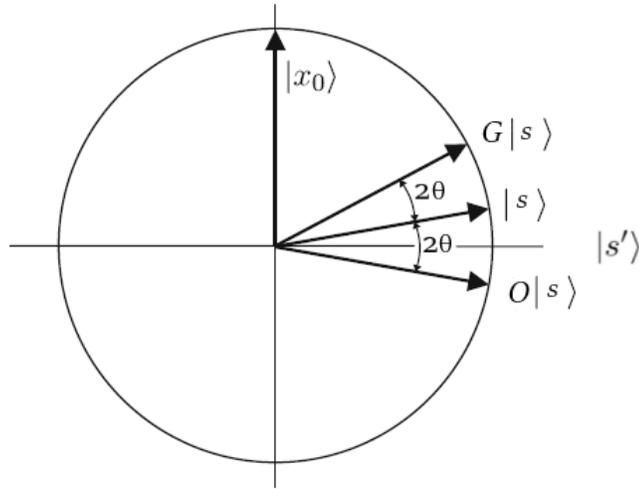


Figura 3.2: Rappresentazione grafica della prima applicazione della procedura di Grover. Il piano di rotazione è individuato dagli stati ortogonali  $|s'\rangle$  e  $|x_0\rangle$ , ed è la proiezione della situazione più generale che ha luogo in uno spazio di dimensione  $N$ .

L'azione di  $G$  sullo stato iniziale  $|s\rangle$  restituisce un vettore che si trova nel sottospazio coperto da  $|s\rangle$  e  $|x_0\rangle$ . Verosimilmente nelle coordinate  $\{|\mathbf{x}\rangle : |\mathbf{x}\rangle \neq |x_0\rangle\}$  e  $|x_0\rangle$ , il vettore iniziale  $|s\rangle$  è molto lontano da  $|x_0\rangle$ , in quanto  $\langle s|x_0\rangle = \frac{1}{\sqrt{N}} \rightarrow 0$  per  $N \rightarrow \infty$ .

Consideriamo lo stato addizionale  $|s'\rangle$  nello span di  $|x_0\rangle$  e  $|s\rangle$  e ortogonale a  $|x_0\rangle$ , ottenuto dunque rimuovendo da  $|s\rangle$  lo stato  $|x_0\rangle$  e riscalandolo:  $|s'\rangle = \frac{1}{\sqrt{N-1}} \sum_{\mathbf{x} \neq x_0} |\mathbf{x}\rangle$ .

Cerchiamo di esplicitare l'angolo  $\theta$  tra  $|s\rangle$  e  $|s'\rangle$ . Questo può essere fatto in due modi: esplicitando  $|s\rangle$  in funzione di  $|\mathbf{x}\rangle$  e  $|s'\rangle$ , che sono ortogonali, o calcolando il prodotto scalare tra  $|s\rangle$  e  $|s'\rangle$ . In ogni caso, si trova che

$$\theta = \arcsin \frac{1}{\sqrt{N}} \quad (3.3.1)$$

e dunque per  $N$  molto grande  $\theta = \frac{1}{\sqrt{N}} + O(\frac{1}{N\sqrt{N}})$ . Partendo dalla condizione iniziale  $|s\rangle$ , una prima applicazione di  $G$  ruota lo stato di un angolo  $2\theta \sim 2/\sqrt{N}$  radianti rispetto a  $|s'\rangle$  verso  $|x_0\rangle$ . Ripetendo il procedimento  $\bar{t} = \lfloor \frac{\pi}{4} \sqrt{N} \rfloor$  volte, si ottiene una rotazione di all'incirca  $\frac{\pi}{2}$ , e dunque un vettore molto vicino a  $x_0$ . In realtà con un'analisi al secondo ordine di  $\theta$  si può mostrare che si rimane sempre nel primo quadrante, e dunque l'angolo è poco meno di  $\frac{\pi}{2}$ . La probabilità di ottenere lo stato  $|x_0\rangle$  dopo la misura al tempo  $\bar{t}$  è per definizione

$$p_{x_0} = |\langle \mathbf{x} | U^{\bar{t}} | s \rangle|^2 \geq \cos^2 \theta$$

dove abbiamo utilizzato la stima di  $\theta$  e la definizione di prodotto scalare per la disuguaglianza. Utilizzando l'equazione (3.3.1) si ottiene infine

$$p_{x_0} \geq 1 - \frac{1}{N} \xrightarrow{N \rightarrow \infty} 1$$

### 3.4 Implementazione in Cirq

Riportiamo qui il codice dell'algoritmo di Grover con  $n = 3$  qubit. Per semplicità è stato scelto  $|x_0\rangle = |111\rangle$ , in questo modo l'oracolo non è altro che un Toffoli Gate implementato su 3 target qubit. La generalizzazione è immediata.

```
[1]: import cirq
import math
import numpy as np
import matplotlib.pyplot as plt

[2]: n_qubits = 3
N = 2**n_qubits
qubits = cirq.GridQubit.rect(1, n_qubits)
ancilla = cirq.NamedQubit('ancilla')
print(qubits)

# Implementazione dell'oracolo con  $x_0 = 111$ 

def Oracolo():
    controls = [cirq.GridQubit(0,v) for v in range(0,n_qubits)]
    yield cirq.X.on(ancilla).controlled_by(*controls)

#Inizializzazione allo stato  $|s\rangle$  con ancilla in  $|-\rangle$ 

def Inizializzazione():
    yield cirq.H.on_each(*qubits)
    yield cirq.X.on(ancilla)
    yield cirq.H.on(ancilla)

# Iterazione di Grover
def Grover():
    # Primo passo, applicare H-H-H-H
    yield cirq.H.on_each(*qubits)
    yield cirq.H.on(ancilla)
    #Secondo passo, applicare X-X-X
    yield cirq.X.on_each(*qubits)
    #Terzo passo, applicare cZ
    yield cirq.Z.on(cirq.GridQubit(0,2)).controlled_by(cirq.
→GridQubit(0,0), cirq.GridQubit(0,1))
    #Quarto passo, applicare X-X-X
    yield cirq.X.on_each(*qubits)
    #Quinto passo, applicare H-H-H
    yield cirq.H.on_each(*qubits)

def Grover_Algorithm(qubits, ancilla, n_qubits):
    circuit = cirq.Circuit()
    circuit.append(Inizializzazione())
    circuit.append(Oracolo())
    step = math.floor( ( math.pi*math.sqrt(N) ) / 4)
    for t in range(0,step):
        circuit.append(Grover())
        circuit.append(Oracolo())
```

```

circuit.append(cirq.measure(*qubits, key = 'x'))
print(circuit)

simulator = cirq.Simulator()
result = simulator.run(circuit, repetitions=100)
counter = result.data
res = counter['x'].to_numpy()
Occ = (np.bincount(res))
plt.bar([0, 1, 2, 3, 4, 5, 6, 7], Occ, align='center')
plt.ylabel('Occorrenze (su 100)')
plt.yticks([0, 10, 20, 30, 40, 50, 60])
plt.xticks([0, 1, 2, 3, 4, 5, 6, 7], ['000', '001', '010', '011', '100', '101', '110', '111'])
plt.show()
return

final = Grover_Algorithm(qubits, ancilla, n_qubits)

```

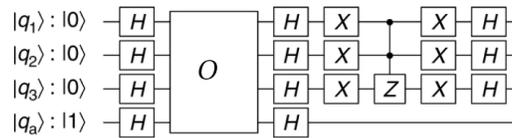
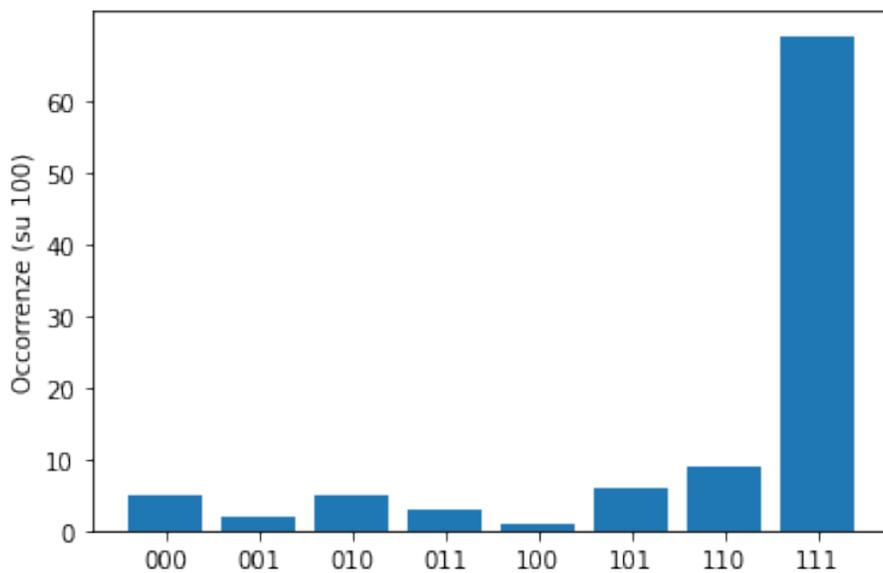


Figura 3.3: Esempificazione del circuito di Grover con  $n = 3$  qubit implementato in Cirq. Si noti che per semplicità è stata riportata solo una delle due iterazioni della procedura di Grover richieste con 3 qubit, ed è perciò omessa la misura finale.



### 3.5 Algoritmo di Grover in termini di Quantum Walk

L'algoritmo di Grover può essere visto come un algoritmo di ricerca su un grafo completo con cicli. Possiamo sostituire ogni arco non orientato  $\{v_1, v_2\}$  con una coppia di archi orientati  $(v_1, v_2)$  e  $(v_2, v_1)$  in modo da ottenere un grafo orientato. Lo spazio del sistema è

$$\mathcal{H}^{N^2} \mathcal{H}^N \otimes \mathcal{H}^N = \text{Span}\{|v_1, v_2\rangle : 0 \leq v_1, v_2 \leq N-1\},$$

dove  $v_1$  indica un vertice del grafo  $G$  e  $v_2$  il valore della moneta, che corrisponde all'etichetta di un altro qualsiasi vertice di  $G$ . Definiamo l'operatore *flip-flop*  $S$  come

$$S |v_1, v_2\rangle = |v_2, v_1\rangle$$

e l'operatore della moneta è

$$C = I_N \otimes \mathcal{R}_s$$

dove  $I_N$  è l'identità su  $\mathcal{H}^N$  e  $\mathcal{R}_s$  è l'operatore di Grover (vedi 3.2) definito da

$$\mathcal{R}_s = 2 |s\rangle \langle s| - I_N,$$

con  $|s\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle$  stato diagonale dello spazio della moneta. L'operatore di evoluzione della quantum walk su grafo completo con cicli senza vertici marcati è

$$U' = S(I_N \otimes \mathcal{R}_s).$$

Supponiamo adesso di marcare il vertice  $|0\rangle$ . Come già osservato nella sezione 3.2, l'oracolo è

$$\tilde{O} = (I_N - 2 |0\rangle \langle 0|) \otimes I_N,$$

ed ha la funzione di invertire il segno al solo stato marcato, e lasciare invariati gli altri. Possiamo scrivere  $\tilde{O}$  equivalentemente come operatore che agisce sull'intero spazio  $\mathcal{H}^{N^2}$  come

$$\tilde{O} = I_{N^2} - 2 \sum_{v \in V(G)} |0, v\rangle \langle 0, v|$$

ovvero l'oracolo marca ogni arco incidente in 0.

L'operatore di evoluzione modificato è

$$\begin{aligned} U &= U' \tilde{O} = S(I_N \otimes \mathcal{R}_s) \tilde{O} \\ &= S(O \otimes \mathcal{R}_s) \end{aligned} \tag{3.5.1}$$

dove  $O = I_N - 2 |0\rangle \langle 0| = \tilde{O}|_{\mathcal{H}^N}$ .

**Teorema 3.5.1.** *Sia  $|\psi(0)\rangle = |s_1, s_2\rangle$  lo stato iniziale dell'algoritmo di ricerca, dove  $s_1$  e  $s_2$  sono gli stati diagonali. Allora l'evoluzione del sistema dopo un numero pari  $2t$  di passi è data da*

$$U^{2t} |\psi(0)\rangle = (\mathcal{R}_s O)^t |s_1\rangle \otimes O (\mathcal{R}_s O)^{t-1} |s_2\rangle. \tag{3.5.2}$$

*Dimostrazione.* Notiamo intanto che vale

$$U |\psi(0)\rangle = S(O \otimes \mathcal{R}_s) |s_1, s_2\rangle = S(O |s_1\rangle \otimes |s_2\rangle) = |s_2\rangle \otimes O |s_1\rangle.$$

Dimostriamo il teorema per induzione su  $t$ .

( $t = 1$ )

$$\begin{aligned} U^2 |\psi(0)\rangle &= U(|s_2\rangle \otimes O |s_1\rangle) \\ &= S(O \otimes \mathcal{R}_s)(|s_2\rangle \otimes O |s_1\rangle) \\ &= S(O |s_2\rangle \otimes \mathcal{R}_s O |s_1\rangle) \\ &= \mathcal{R}_s O |s_1\rangle \otimes O |s_2\rangle. \end{aligned}$$

$(t \Rightarrow t + 1)$

$$\begin{aligned}
 U^{2t} |\psi(0)\rangle &= U^2 \cdot U^{2t-2} |\psi(0)\rangle = U^2 ((\mathcal{R}_s O)^{t-1} |s_1\rangle \otimes O(\mathcal{R}_s O)^{t-2} |s_2\rangle) \\
 &= U ((\mathcal{R}_s O)^{t-1} |s_2\rangle \otimes O(\mathcal{R}_s O)^{t-1} |s_1\rangle) \\
 &= S (O(\mathcal{R}_s O)^{t-1} |s_2\rangle \otimes (\mathcal{R}_s O)(\mathcal{R}_s O)^{t-1} |s_1\rangle) \\
 &= (\mathcal{R}_s O)^t |s_1\rangle \otimes O(\mathcal{R}_s O)^{t-1} |s_2\rangle.
 \end{aligned}$$

□

A questo punto non ci resta che osservare che il primo termine del prodotto tensoriale di (3.5.2) è proprio l'evoluzione dell'algoritmo di Grover descritta in 3.2. Eseguendo  $2t = 2\lfloor \frac{\pi}{4} \sqrt{N} \rfloor$  passi dell'algoritmo e misurando infine la posizione della particella si ottiene dunque lo stesso risultato dell'algoritmo di Grover. Notiamo che l'algoritmo di Grover in termini di quantum walk su grafo richiede il doppio delle iterazioni dell'algoritmo standard (ma comunque un  $O(\sqrt{n})$ ).

È bene precisare che l'algoritmo di Grover non è un'eccezione, si può infatti dimostrare ([7]) che le Quantum Walk forniscono un modello universale di computazione quantistica.

# Ringraziamenti

Desidero innanzi tutto ringraziare il Professor Dario Trevisan per la grande disponibilità, pazienza e professionalità dimostrate in questi mesi di lavoro, e per avermi introdotto ad argomenti così affascinanti.

Un ringraziamento a mia madre, mio padre, mio fratello, che hanno contribuito a rendermi chi sono, mi hanno sostenuto in questo lungo percorso e grazie ai quali ho trovato la voglia e la motivazione di dedicarmi a questo progetto.

Un ringraziamento ai miei nonni, perché il loro amore non mi ha mai fatto sentire solo.

Infine, un ringraziamento a zia e agli amici di una vita, Lorenzo, Paolo, Marco, Alessio, Fares, Enzo, Francesco, Patrick, che mi sono sempre stati vicino, anche nei momenti più difficili.



# Bibliografia

- [1] I. L. Chuang A. Nielsen. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- [2] A. Holevo. *Probabilistic and Statistical Aspects of Quantum Theory*. Edizioni Della Normale, 2011.
- [3] Lorenzo Bagnasacco. *Quantum operations on a q-bit, a geometric visualization on Bloch sphere*. 2021.
- [4] G. Gaspari J. Rudnick. *Elements of the Random Walk*. Cambridge University Press, 2004.
- [5] Renato Portugal. *Quantum Walks and Search Algorithms, second edition*. Springer, 2018.
- [6] Cirq Developers. *Cirq*. Ver. v0.12.0. See full list of authors on Github: <https://github.com/quantumlib/Cirq/graphs/contributors>. Ago. 2021.
- [7] Andrew M. Childs. “Universal Computation by Quantum Walk”. In: *Phys. Rev. Lett.* 102 (mag. 2009).