


Università degli Studi di Milano Bicocca

Dipartimento di Fisica

Corso di Laurea Triennale



Evoluzione temporale di un modello di Ising tramite Trotterization

Relatore: 
Prof. Andrea Giachero

Candidato:
Lorenzo del Prete

Correlatore:
Dr. Stefano Barison

ANNO ACCADEMICO 2023/2024

Indice

Abstract	1
1 La teoria della computazione quantistica	3
1.1 I quantum bits	3
1.1.1 La sfera di Bloch	4
1.1.2 I sistemi a più qubit e lo stato di Bell	4
1.2 I gate quantistici	6
1.2.1 I gate a singolo qubit e le matrici di Pauli	6
1.2.2 I gate di rotazione	7
1.2.3 Il CNOT Gate	8
2 Evoluzione temporale di una catena di Heisenberg	10
2.1 Modello quantistico di una catena di $\text{spin-}\frac{1}{2}$	10
2.2 XXX-Heisenberg Chain	11
2.3 Decomposizione di Suzuki-Trotter	13
2.3.1 Trotterization e Heisenberg Chain	13
2.4 Implementazione Trotterization	15
2.5 Trotterization in un sistema ideale	17
2.6 Andamento dell'errore	21
2.7 Il rumore nell'hardware	23
2.8 Ottimizzazione del circuito	27
2.9 Variazione dei parametri	31
2.10 Tecniche di mitigazione dell'errore	34
3 Simulazione su Quantum Processor Unit	37
Conclusioni	42

Elenco delle figure

1	Valori simulati della catena con 3 spin su QPU con MAE dei differenti metodi	2
1.1	Rappresentazione dello stato di un qubit sulla sfera di Bloch.	5
1.2	Circuito per la creazione di uno stato di Bell con un Hadamard Gate e un CNOT.	9
2.1	Probabilità nel tempo di rimanere nello stato $ 10\rangle$	12
2.2	Probabilità nel tempo di rimanere nello stato $ 110\rangle$	12
2.3	Circuito per l'implementazione di una rotazione Z su due qubit.	15
2.4	Circuito per l'implementazione di una rotazione X su due qubit.	16
2.5	Circuito per l'implementazione di una rotazione Y su due qubit.	16
2.6	Probabilità nel tempo di rimanere nello stato $ 10\rangle$ tramite Trotterization. .	18
2.7	Probabilità nel tempo di rimanere nello stato $ 110\rangle$ tramite Trotterization con 3 differenti Trotter steps.	19
2.8	Probabilità nel tempo di rimanere nello stato $ 110\rangle$ tramite Trotterization con 50 Trotter steps.	20
2.9	Errore relativo con $t = 1.5$ e Trotter steps variabili.	21
2.10	Errore relativo con $n = 12$ e tempo variabile in scala logaritmica.	22
2.11	Circuito quantistico per l'implementazione di uno step della Trotterization. .	22
2.12	Le curve a tre spin con noise bitphase-flip.	24
2.13	Rappresentazione sulla sfera di Bloch di come agisce il bitphase flip sugli stati della sfera. Gli assi x e z sono contratti come funzione della probabilità. .	25
2.14	Rappresentazione sulla sfera di Bloch della contrazione uniforme dovuta al depolarizing noise.	25
2.15	Curve per 3 spin con il noise model costruito.	26
2.16	Curve per 3 spin con una fake backend di Qiskit.	27

2.17	Processo di mappatura dei qubit virtuali sull'hardware. Un modo potrebbe essere quello di mappare i numeri dei qubit in modo corrispondente con quelli reali, oppure fare in modo di disporre i qubit in modo tale che i gate a due qubit agiscano su coppie adiacenti.	28
2.18	Confronto degli errori relativi di ciascuna curva rispetto ai tre livelli di ottimizzazione del transpile e ai dati non ottimizzati.	29
2.19	Curva a 3 spin con $n = 12$ ottimizzata.	29
2.20	Circuito di un singolo Trotter step con $n = 12$ dopo la transpilation su FakeMontrealV2.	30
2.21	Lunghezza del circuito nel caso in cui viene fatto aumentare nel tempo il numero di Trotter steps fino a $n = 12$	31
2.22	Confronto a $n = 12$ delle curve su tre spin con il numero di Trotter steps raggiunto linearmente e fissato a $t = 0$	32
2.23	Errore relativo con $n = 12$ fissato e raggiunto in modo lineare.	32
2.24	Confronto a $n = 60$ delle curve su tre spin con il numero di Trotter steps raggiunto linearmente e fissato a $t = 0$	33
2.25	Sostituzione di un operatore di misura con la TREX.	35
2.26	Applicazione della ZNE con fase di digital folding ed estrapolazione del dato noiseless.	36
3.1	Simulazione su QPU senza l'ottimizzazione del circuito e del noise. . . .	38
3.2	Confronto delle curve simulate su QPU ibm.kyiv da 127 qubit.	39

Abstract

Sono trascorsi più di quarant'anni dalla pubblicazione del celebre articolo "Simulating Physics with Computers" di R. Feynman, in cui fu evidenziato come la simulazione dei sistemi quantistici potesse essere più efficiente sfruttando i principi della meccanica quantistica rispetto all'approccio classico utilizzato dai computer classici. In questa tesi si tratterà, tramite l'utilizzo di un processore quantistico, la simulazione temporale di una catena di Heisenberg, un modello utile per la trattazione delle interazioni tra spin su un reticolo in fenomeni di magnetizzazione, dal punto di vista della dinamica. In particolare, definito uno stato iniziale, si analizzerà l'evoluzione nel tempo della probabilità che la catena nello stato di partenza. Si studierà il caso unidimensionale e in cui l'intensità dell'interazione è uniforme in ogni direzione. Questo modello prende il nome di *XXX-Heisenberg Chain*, trattando un modello a due spin e a tre spin. Al fine di studiare in maniera efficiente la dinamica su computer quantistico verrà utilizzata l'approssimazione di Trotter-Suzuki. Questa tecnica, detta anche Trotterization, permette di decomporre l'operatore di evoluzione temporale come prodotto di interazioni locali. L'introduzione della Trotterization per N_{spin} tramite gate quantistici richiede $6 \cdot (N_{spin} - 1)$ controlled NOT gate (CNOT), dunque nel caso a tre spin su cui verrà effettuata la simulazione 12 CNOT per step. Nel caso di 12 Trotter steps si arriva a 144 CNOT. Si analizzeranno anche gli errori dell'approssimazione di Trotter rispetto al numero di Trotter steps e al tempo. Per prima cosa si verificherà che nel caso in due spin l'errore di approssimazione è nullo, trattando solamente operatori che commutano. Dopodiché si verificherà che nel caso a tre spin l'errore algoritmico introdotto è $O(h^2)$, con $h = \frac{time}{Trotter_steps}$. I computer quantistici che abbiamo oggi a disposizione non permettono di correggere gli errori che avvengono durante la computazione, per questo motivo si indagherà quanto questi influiscano andando a costituire un *Noise Model* che possa approssimarne il comportamento, studiando i vari fenomeni di errore che potrebbe causare l'ambiente esterno sui gate del circuito, tra cui fenomeni di *bitphase flip*, *depolarizing channel* e *amplitude-phase damping*. Dopodiché si analizzeranno delle tecniche per ridurre l'errore analizzando quanto la lunghezza del circuito lo condiziona, provando un approccio *variable length* (VL) basato sull'incrementare linearmente la lunghezza

del circuito nel tempo. Si valuteranno i metodi della libreria Python *Qiskit* nel tradurre e ottimizzare il circuito per l'hardware tramite il processo di *transpilation* che permette di adattare il circuito alla topologia dell'hardware andando a modificare la disposizione delle porte. La libreria utilizza tre diversi livelli di ottimizzazione. Nella tesi si mostra che il secondo livello di ottimizzazione riduce maggiormente l'errore nel circuito utilizzato, riducendo il numero di CNOT per 12 Trotter step a 72 CNOT. Si sono analizzate inoltre delle tecniche di mitigazione dell'errore tra cui il *Twirled Readout Error eXtinction* (TREX), la *Zero Noise Extrapolation* (ZNE) e il *gate twirling*. In conclusione si effettuerà la simulazione tramite le *Quantum Processor Units* (QPU) di IBM Quantum confrontando le quattro tecniche studiate, dunque tramite *transpilation*, senza *transpilation* ma con *VL*, con *transpilation* e *VL* e con le tecniche di mitigazione dell'errore. Si ottiene in (1) che

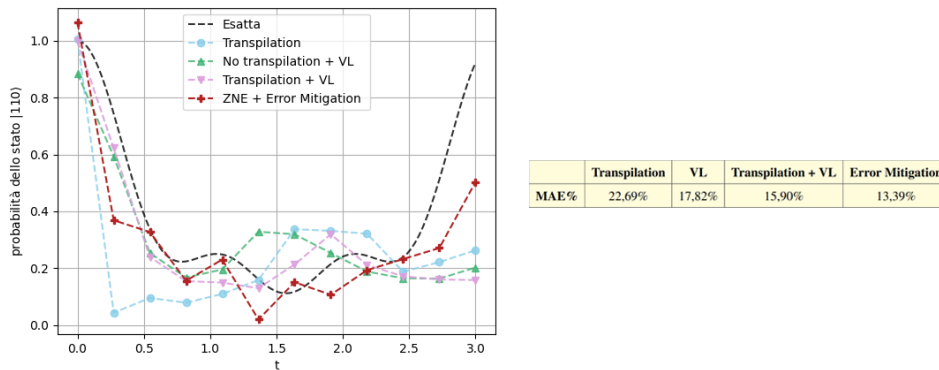


Figura 1: Valori simulati della catena con 3 spin su QPU con MAE dei differenti metodi

il metodo *VL* permette di ridurre notevolmente l'errore giungendo ad un errore assoluto medio (MAE) di 0.159. Si conclude che il rumore reale dell'hardware è tale da introdurre un errore significativo sulla simulazione che può essere ridotto tramite l'utilizzo di alcuni approcci, come la *VL* e la *transpilation*. Infine le tecniche di mitigazione dell'errore permettono di ridurre ulteriormente il MAE fino a 0.139. Questa simulazione mostra che l'utilizzo di un computer quantistico è efficace, al netto di un hardware sensibile al rumore ambientale su cui, tuttavia, la ricerca ottiene nuovi risultati ogni anno.

1. La teoria della computazione quantistica

1.1 I quantum bits

. La teoria della computazione quantistica e dell'informazione quantistica presenta un concetto analogo al bit di un computer classico: il *quantum bit* o *qubit*.

Un qubit può essere trattato e descritto come un oggetto matematico che viene poi realizzato tramite un sistema fisico. La possibilità di una trattazione matematica apre le porte alla possibilità di costruire una teoria della computazione e dell'informazione quantistica. Allo stesso modo di un bit, un qubit possiede uno *stato*. La differenza sostanziale tra il bit e il qubit consiste nella possibilità del secondo di trovarsi in ciò che viene definita una *sovrapposizione di stati*. Pertanto se un bit è nello stato $|0\rangle$ o $|1\rangle$ ¹, lo stato generico di un qubit $|\psi\rangle$ è definibile come

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1.1)$$

ovvero una combinazione lineare di stati dove α, β sono coefficienti complessi.

La eq. (1.1) permette una trattazione vettoriale dove $\{|0\rangle, |1\rangle\}$ sono i vettori linearmente indipendenti di una base ortonormale, detta *base computazionale*, appartenente ad uno spazio vettoriale complesso in due dimensioni.

La misurazione di $|\psi\rangle$ fa collassare la sovrapposizione in modo tale che la probabilità di misurare $|0\rangle$ o $|1\rangle$ sia

$$\begin{aligned} P(|0\rangle) &= |\alpha|^2 \\ P(|1\rangle) &= |\beta|^2 \end{aligned} \quad (1.2)$$

E' dunque necessario che lo stato $|\psi\rangle$ sia correttamente normalizzato in modo che la somma delle probabilità di ottenere ciascuna misura sia 1. La condizione è posta sui coefficienti della sovrapposizione:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (1.3)$$

¹Si utilizza il formalismo dei bra-ket di Dirac.

Il teorema di non-clonazione impedisce di clonare lo stato $|\psi\rangle$ in modo da determinare sia α che β poiché proibisce di copiare le informazioni dello stato senza che quest'ultimo collassi. Pertanto nel quantum computing è necessario preparare più volte lo stesso stato al fine di estrarre informazioni su α e β .

Formalmente:

Definito un processo di clonazione di uno stato correttamente normalizzato $|\psi\rangle$

$$|\psi\rangle \otimes |\phi\rangle \xrightarrow{\text{clonazione}} |\psi\rangle \otimes |\psi\rangle \quad (1.4)$$

il teorema di non-clonazione afferma che non esiste operatore U che rende possibile il processo.

1.1.1 La sfera di Bloch

La eq. (1.3) permette una visualizzazione geometrica dello stato di un qubit e delle trasformazioni applicabili su quest'ultimo. Si ricava che $|\psi\rangle$ dipende da due angoli θ e ϕ . Pertanto $|\psi\rangle$ è rappresentabile su un oggetto sferico detto *sfera di Bloch* come in fig. 1.1 come segue

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle \quad (1.5)$$

1.1.2 I sistemi a più qubit e lo stato di Bell

La trattazione di più qubit è un'estensione di quanto già detto in merito al singolo qubit. La base computazionale nel caso di due qubit è una base di quattro vettori linearmente indipendenti di cui lo stato può essere una combinazione lineare correttamente normalizzata.

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle$$

$$\sum_{i,j \in \{0,1\}} |\alpha_{ij}|^2 = 1 \quad (1.6)$$

Tra gli stati a due qubit è particolarmente importante lo *stato di Bell* definito come

$$\Psi = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad (1.7)$$

Dato un sistema con due qubit nello stato Ψ la misura di un primo qubit restituisce uno 0 o un 1 con il 50% di possibilità, lasciando il sistema nello stato, rispettivamente, $|00\rangle$ o

$|11\rangle$. A questo punto una misura del secondo qubit non può che restituire come risultato lo stesso misurato sul primo. Si tratta di un fenomeno in cui i risultati di due misure sono correlati.²

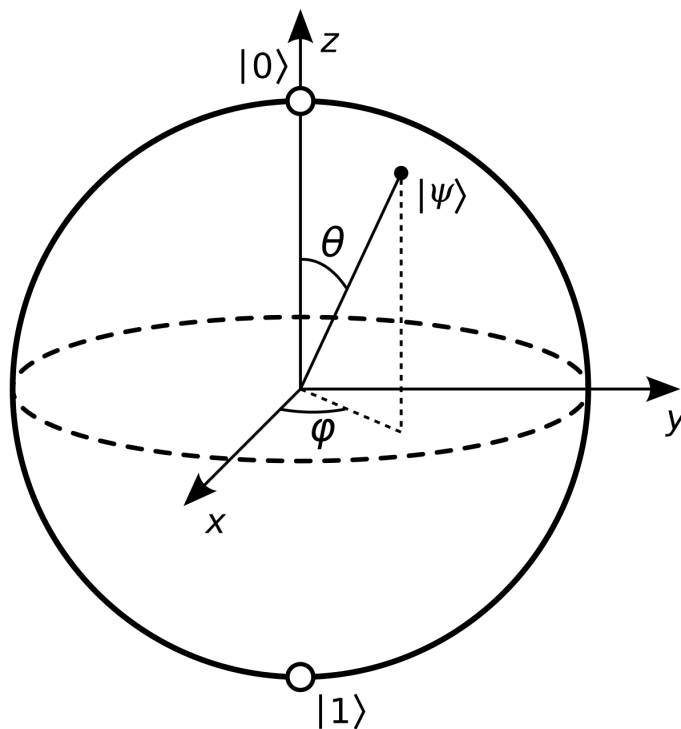


Figura 1.1: Rappresentazione dello stato di un qubit sulla sfera di Bloch.

²Michael A. Nielsen e Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

1.2 I gate quantistici

Le trasformazioni su un registro di qubit vengono effettuate tramite l'uso di *quantum gates*, equivalente quantistico dei gates utilizzati nella computazione classica. Si tratta di operazioni logiche che modificano lo stato dei qubit e che vengono, alla luce della teoria costruita, rappresentati come operatori unitari $A^\dagger A = I$ che operano nello spazio di Hilbert e che preservano la norma dello stato. L'unitarietà delle operazioni logiche su computer quantistici, a differenza della computazione classica, è tale da rendere tutte le operazioni reversibili.

1.2.1 I gate a singolo qubit e le matrici di Pauli

Le matrici di Pauli permettono la costruzione dei quantum gate logici. Sono definite come segue:

$$\sigma_x = X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (1.8)$$

$$\sigma_y = Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad (1.9)$$

$$\sigma_z = Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (1.10)$$

NOT Gate

Il NOT gate agisce sulla base computazionale come segue

$$\begin{aligned} |0\rangle &\rightarrow |1\rangle \\ |1\rangle &\rightarrow |0\rangle \end{aligned} \quad (1.11)$$

Il NOT Gate in forma matriciale corrisponde ad applicare la matrice di Pauli X sullo stato $|\psi\rangle$, che verrà dunque trasformato come

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \quad (1.12)$$

Z Gate

Lo Z gate, che corrisponde alla matrice di Pauli Z, agisce in questo modo

$$\begin{aligned} |0\rangle &\rightarrow |0\rangle \\ |1\rangle &\rightarrow -|1\rangle \end{aligned} \quad (1.13)$$

dunque

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ -\beta \end{bmatrix} \quad (1.14)$$

Hadamard Gate

L'Hadamard Gate è definito come segue

$$\begin{aligned} |0\rangle &\rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ |1\rangle &\rightarrow \frac{|0\rangle - |1\rangle}{\sqrt{2}} \end{aligned} \quad (1.15)$$

Ovvero

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (1.16)$$

La base $\{\frac{|0\rangle+|1\rangle}{\sqrt{2}}, \frac{|0\rangle-|1\rangle}{\sqrt{2}}\} = \{|+\rangle, |-\rangle\}$ è detta *base di Hadamard*.

1.2.2 I gate di rotazione

In virtù della rappresentazione geometrica sulla sfera di Bloch è possibile interpretare le operazioni sui qubit come rotazioni attorno ad un asse cartesiano codificate con tre angoli.

I gate di rotazione introducibili immediatamente sono costruiti tramite le matrici di Pauli e rappresentano una rotazione rispetto ad un piano fissato dal momento che

$$e^{iAX} = \cos(x)I + i\sin(x)A \quad (1.17)$$

$$R_x(\theta) = e^{-i\theta \frac{x}{2}} = \begin{bmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix} \quad (1.18)$$

$$R_y(\theta) = e^{-i\theta\frac{Y}{2}} = \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix} \quad (1.19)$$

$$R_z(\theta) = e^{-i\theta\frac{Z}{2}} = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \quad (1.20)$$

In generale ogni gate su un qubit può essere decomposto in rotazioni rispetto ai tre assi cartesiani di α, β, γ con una fase globale escludibile in quanto non varia la fisica del sistema. Pertanto sia U un gate unitario

$$U = e^{i\theta} R_z(\alpha) R_y(\beta) R_z(\gamma) \quad (1.21)$$

1.2.3 Il CNOT Gate

Nell'ambito dei gate che operano su più qubit è doveroso introdurre il *Control-NOT* o *CNOT* Gate. I due qubit su cui opera un CNOT hanno due ruoli differenti. Uno dei due qubit è un *control* su cui viene verificata una condizione rispetto alla quale viene effettuato un NOT sul secondo qubit definito *target*.³

Dal momento che si sta trattando un sistema a due qubit la rappresentazione matriciale del CNOT è una matrice 4x4:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (1.22)$$

Se il qubit di controllo del gate è nello stato $|0\rangle$ l'operazione sul target non viene effettuata e dunque rimane nel suo stato iniziale. Se il controllo è su uno stato $|1\rangle$ allora sul qubit target viene effettuato l'equivalente di un NOT Gate.

A partire da uno stato $|00\rangle$ è possibile ottenere il circuito fig. 1.2 che produce uno stato di

³Robert Lored. *Learn Quantum Computing with Python and IBM Quantum Experience*. Packt Publishing, 2020.

Bell tramite un gate di Hadamard e un CNOT gate come segue:

$$|00\rangle = |0\rangle \otimes |0\rangle$$

$$(H|0\rangle) \otimes |0\rangle = |+\rangle \otimes |0\rangle = \frac{|10\rangle + |00\rangle}{\sqrt{2}} \quad (1.23)$$

$$CNOT\left[\frac{|10\rangle + |00\rangle}{\sqrt{2}}\right] = \frac{|11\rangle + |00\rangle}{\sqrt{2}}$$

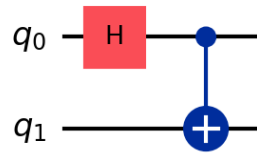


Figura 1.2: Circuito per la creazione di uno stato di Bell con un Hadamard Gate e un CNOT.

2. Evoluzione temporale di una catena di Heisenberg

2.1 Modello quantistico di una catena di spin- $\frac{1}{2}$

Il sistema in esame è composto da una catena in una dimensione di spin- $\frac{1}{2}$. Si tratta di un modello quantistico utilizzato per descrivere diversi fenomeni fisici tra cui i comportamenti ferromagnetici nella magnetizzazione dei materiali e tiene conto delle interazioni spin-spin nei vari siti della catena discreta.

Dati N spin disposti sugli N siti della catena, ogni spin vive in uno spazio di Hilbert di dimensione 2 e si trova in una sovrapposizione di stato up $|\uparrow\rangle$ e stato down $|\downarrow\rangle$.

Lo spazio di Hilbert del sistema \mathbb{H} è il prodotto tensoriale degli spazi di singolo spin¹

$$\mathbb{H} = \bigotimes_N \mathbb{C}^2 \quad (2.1)$$

Definito l'operatore dello spin i -esimo lungo la direzione cartesiana j S_i^j e considerando la sola interazione degli spin *nearest neighbours*, l'hamiltoniana del sistema è la seguente

$$\mathcal{H}_{XYZ} = \sum_{i,j}^{N,\{x,y,z\}} J_j S_i^j \cdot S_{i+1}^j \quad (2.2)$$

nel caso in cui $J_x \neq J_y \neq J_z$ prende il nome di *XYZ-Heisenberg Chain*.

L'hamiltoniana eq. (2.2) è una matrice di dimensione $2^N \times 2^N$ e nell'ottica di voler studiare un sistema per N crescente necessario ricorrere a metodi computazionali per la diagonalizzazione di \mathcal{H}^2 , che tuttavia diventano esponenzialmente complessi con il crescere della dimensione³.

¹Ingmar Saberi. *An introduction to spin systems for mathematicians*. arXiv, 2018. URL: <https://arxiv.org/abs/1801.07270>.

²Rafael I. Nepomechie. *A spin chain primer*. Vol. 13. 24n25. World Scientific Pub Co Pte Lt, 1999, pp. 2973–2985. DOI: 10.1142/s0217979299002800.

³G. et al. Ortiz. *Quantum algorithms for fermionic simulations*. Vol. 64. 2. American Physical Society (APS), 2001. DOI: 10.1103/physreva.64.022319.

2.2 XXX-Heisenberg Chain

Il sistema su cui verrà effettuata la simulazione è descritto da un caso particolare della eq. (2.2), ovvero quando $J_x = J_y = J_z = J$. In linea con la nomenclatura introdotta tale catena prende il nome di XXX-Heisenberg Chain.

In questo modo l'hamiltoniana si riduce a

$$\mathcal{H}_{XXX} = J \cdot \left[\sum_{i,j}^{N, \{x,y,z\}} S_i^j \cdot S_{i+1}^j \right] \quad (2.3)$$

Il sistema in questione è risolvibile analiticamente tramite il l'ansatz di Bethe.⁴

Il segno di J distingue il comportamento del sistema. In particolar modo se $J > 0$ la catena è ferromagnetica, pertanto gli stati di equilibrio massimizzeranno l'allineamento degli spin.

Definita l'hamiltoniana l'evoluzione temporale del sistema quantistico avviene applicando l'*operatore di evoluzione temporale*

$$U(t) = e^{-i\mathcal{H}_{XXX}t} \quad (2.4)$$

su uno stato $|\psi\rangle$ di partenza, che nel caso della tesi viene costruito come prodotto tensoriale degli stati di singolo spin⁵.

Effettuando una simulazione classica del sistema si studia la curva di overlap dello stato $|10\rangle$ in fig. 2.1 che rappresenta la probabilità nel tempo del sistema di trovarsi nello stato.

$$P(t) = |\langle 10 | U(t) | 10 \rangle|^2 \quad (2.5)$$

Trattandosi di una matrice 4×4 il costo computazionale nell'elevazione esponenziale della matrice in $U(t)$ è sufficientemente basso per studiarne l'evoluzione in questo modo.

⁴N. Manojlović e I. Salom. *Algebraic Bethe ansatz for the XXZ Heisenberg spin chain with triangular boundaries and the corresponding Gaudin model*. Vol. 923. Elsevier BV, 2017, pp. 73–106. DOI: 10.1016/j.nuclphysb.2017.07.017.

⁵Daniel S. Abrams e Seth Lloyd. *Simulation of Many-Body Fermi Systems on a Universal Quantum Computer*. Vol. 79. 13. American Physical Society (APS), 1997, pp. 2586–2589. DOI: 10.1103/physrevlett.79.2586.

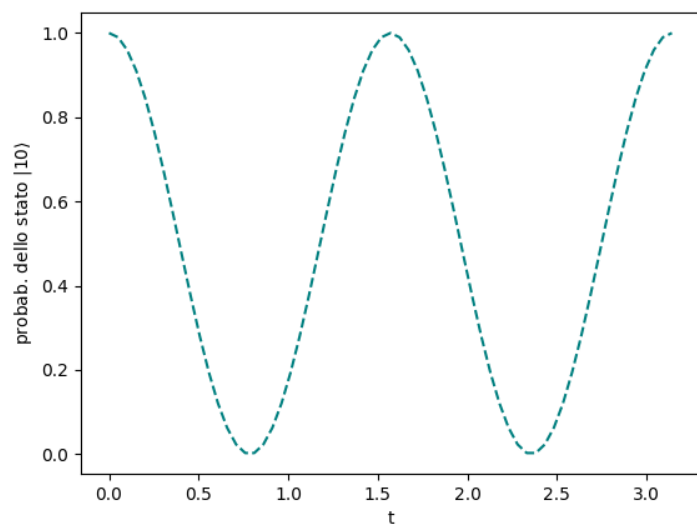


Figura 2.1: Probabilità nel tempo di rimanere nello stato $|10\rangle$.

Lo stesso approccio classico è applicabile anche nel caso $N = 3$ con lo stato $|110\rangle$ ottenendo la curva fig. 2.2

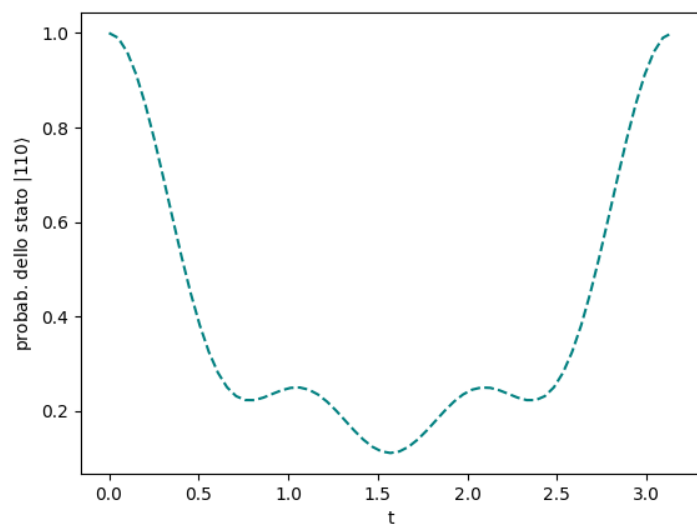


Figura 2.2: Probabilità nel tempo di rimanere nello stato $|110\rangle$.

2.3 Decomposizione di Suzuki-Trotter

La decomposizione di Suzuki-Trotter o *Trotterization*⁶ permette, dati A_i operatori autoaggiunti con $i = 1, \dots, N$, di effettuare la seguente approssimazione⁷ al primo ordine:

$$e^{t \sum_i^N A_i} \approx \prod_i^N e^{t A_i} + O(t^2) \quad (2.6)$$

Formula di Baker-Campbell-Hausdorff Nell'ambito di operatori con commutatore non nullo, la formula di Baker-Campbell-Hausdorff da una stima dell'errore che si compie nello scrivere $e^Z = e^X \cdot e^Y$. In particolare si dimostra⁸ che

$$Z(X, Y) = \log(e^X e^Y) = X + Y + \frac{1}{2}[X, Y] + \dots \quad (2.7)$$

Ciò permette di ipotizzare, come verrà verificato, che nel caso a due spin l'errore algoritmico introdotto sarà nullo, dato che gli operatori considerati commutano.

2.3.1 Trotterization e Heisenberg Chain

Nel caso della Heisenberg Chain poiché nell'hamiltoniana compare la somma degli operatori che rappresentano le interazioni di spin, la decomposizione di Suzuki-Trotter permette, nel caso di $N = 2$ di effettuare la seguente approssimazione

$$U_2(t) \approx e^{-itS_x^{(0)} \cdot S_x^{(1)}} e^{-itS_y^{(0)} \cdot S_y^{(1)}} e^{-itS_z^{(0)} \cdot S_z^{(1)}} \quad (2.8)$$

Se $N > 2$ U_N non può essere direttamente decomposta come nella eq. (2.8). Prendendo in esame il caso con $N = 3$ si scrive l'hamiltoniana dell'interazione globale come una somma di interazioni locali tra i due spin adiacenti $H_2^{(i,j)}$. L'ipotesi su U_3 che viene effettuata è quella di considerare l'evoluzione del sistema come ad una evoluzione temporale di $H_2^{(0,1)}$

⁶H. F. Trotter. *On the Product of Semi-Groups of Operators*. Vol. 10. 4. American Mathematical Society, 1959, pp. 545–551.

⁷Masuo Suzuki. *General theory of fractal path integrals with applications to many-body theories and statistical physics*. Vol. 32. 2. American Institute of Physics, 1991, pp. 400–407. DOI: 10.1063/1.529425.

⁸Jordan C. Moodie e Martin W. Long. *An exact power series representation of the Baker-Campbell-Hausdorff formula*. arXiv, 2018. URL: <https://arxiv.org/abs/1807.07884>.

per un tempo $dt = \frac{t}{n}$, con n *numero di Trotter steps*, seguita da un'evoluzione di $H_2^{(1,2)}$, così facendo

$$U_3(t) \approx \{e^{-i\frac{t}{n}H_2^{(0,1)}} e^{-i\frac{t}{n}H_2^{(1,2)}}\}^n + O(\frac{t^2}{n}) \quad (2.9)$$

In questo modo la generalizzazione per N_{spin} è⁹

$$U_{N_{spin}} \approx \{ \prod_{i=0}^{N_{spin}-1} e^{-i\frac{t}{n}H_2^{(i,i+1)}} \}^n + O(\frac{t^2}{n}) \quad (2.10)$$

⁹Ying Li e Simon C. Benjamin. *Efficient Variational Quantum Simulator Incorporating Active Error Minimization*. Vol. 7. 2. American Physical Society (APS), 2017. DOI: 10.1103/physrevx.7.021050.

2.4 Implementazione Trotterization

Gli esponenziali della eq. (2.8) possono essere implementati come operazioni su due qubit tramite la eq. (1.21). Non è necessario definirli diversamente come gate per il caso con $N > 2$ in quanto l'hamiltoniana in quel caso viene decomposta come interazione tra due spin.

Gate per S_Z Nel caso di 1 qubit la $R_Z(\theta)$ è tale da

$$R_Z(\theta) = \cos(\theta) \cdot I - i \sin(\theta) \cdot Z = \begin{bmatrix} \cos(\theta) - i \sin(\theta) & 0 \\ 0 & \cos(\theta) + i \sin(\theta) \end{bmatrix} \quad (2.11)$$

Si costruisce il gate per la rotazione su 2 qubit R_{ZZ} a partire da $I \oplus R_Z(\theta)$, tramite due CNOT si scambiano gli elementi in terza e quarta posizione lungo la diagonale della matrice, in questo modo

$$R_{ZZ}(\theta) = CNOT_{12} \cdot I \oplus R_Z(\theta) \cdot CNOT_{12} \quad (2.12)$$

In questo modo si ottengono il circuito fig. 2.3 e l'operatore eq. (2.13)

$$R_{ZZ}(\theta) = \begin{bmatrix} \cos(\theta) - i \sin(\theta) & 0 & 0 & 0 \\ 0 & \cos(\theta) + i \sin(\theta) & 0 & 0 \\ 0 & 0 & \cos(\theta) + i \sin(\theta) & 0 \\ 0 & 0 & 0 & \cos(\theta) - i \sin(\theta) \end{bmatrix} \quad (2.13)$$

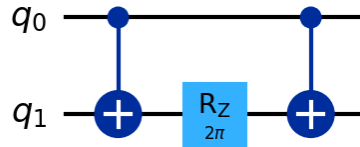


Figura 2.3: Circuito per l'implementazione di una rotazione Z su due qubit.

Gate per S_X Per costruire R_{XX} e R_{YY} serve effettuare un cambio di base in quanto non diagonali sulla base computazionale utilizzando che

$$R_{XX}(\theta) = [R_y(\frac{\pi}{2}) \otimes R_y(\frac{\pi}{2})] \cdot R_{ZZ}(\theta) \cdot [R_y(-\frac{\pi}{2}) \otimes R_y(-\frac{\pi}{2})] \quad (2.14)$$

Si ottiene così il circuito fig. 2.4

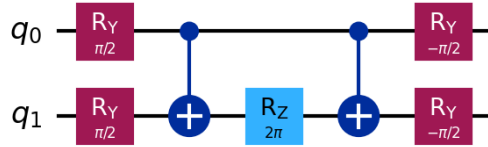


Figura 2.4: Circuito per l'implementazione di una rotazione X su due qubit.

Gate per S_Y Similmente per $R_{XX}(\theta)$ si ottiene il circuito fig. 2.5¹⁰

$$R_{YY}(\theta) = [R_x(\frac{\pi}{2}) \otimes R_x(\frac{\pi}{2})] \cdot R_{ZZ}(\theta) \cdot [R_x(-\frac{\pi}{2}) \otimes R_x(-\frac{\pi}{2})] \quad (2.15)$$

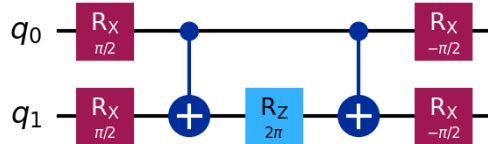


Figura 2.5: Circuito per l'implementazione di una rotazione Y su due qubit.

¹⁰Bo Peng et al. *Quantum time dynamics employing the Yang-Baxter equation for circuit compression*.

Vol. 106. 1. American Physical Society (APS), 2022. DOI: 10.1103/physreva.106.012412.

2.5 Trotterization in un sistema ideale

Tramite l'utilizzo di *Qiskit*, un pacchetto di sviluppo open-source ideato per operare su circuiti quantistici, è stato possibile implementare i gates della Trotterization ed effettuare una simulazione delle curve di overlap delle fig. 2.1 e fig. 2.2 in un sistema ideale, dunque senza la presenza del rumore che influisce sistematicamente sulle misure tipico di un hardware quantistico reale.

Utilizzando una funzione che restituisse un circuito quantistico con i gate per l'hamiltoniana del sistema applicati

```
def Heisenberg_Trotter(num_spins , Trotter_steps , t):  
    #definiamo un time step dt  
    dt = (t)/(Trotter_steps)  
  
    #creiamo il quantumcircuit  
    qc_t = QuantumCircuit(num_spins)  
  
    #iteriamo su ogni time step  
    for i in range(Trotter_steps):  
        #iteriamo su ogni spin  
        for j in range(0, num_spins -1):  
  
            qc_t = qc_t.compose(R_xx(num_spins , dt , j , j+1))  
            qc_t = qc_t.compose(R_yy(num_spins , dt , j , j+1))  
            qc_t = qc_t.compose(R_zz(num_spins , dt , j , j+1))  
  
    return qc_t
```

E implementando l'osservabile da misurare, ovvero $|10\rangle\langle 10|$ e $|110\rangle\langle 110|$ utilizzando che

$$\begin{aligned} |0\rangle\langle 0| &= \frac{1}{2} \cdot (I + Z) \\ |1\rangle\langle 1| &= \frac{1}{2} \cdot (I - Z) \end{aligned} \quad (2.16)$$

```
prj_zero = SparsePauliOp.from_list([("I", 1/2), ("Z", 1/2)])
prj_one = SparsePauliOp.from_list([("I", 1/2), ("Z", -1/2)])
prj = prj_one ^ prj_one ^ prj_zero
```

Si è ottenuta la curva in fig. 2.6

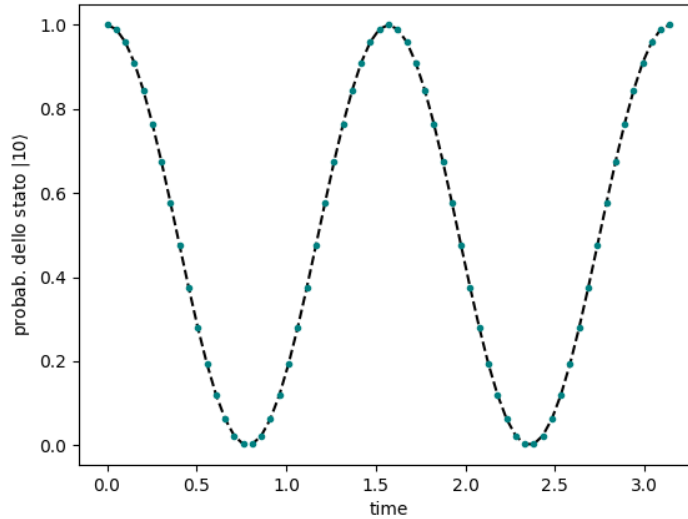


Figura 2.6: Probabilità nel tempo di rimanere nello stato $|10\rangle$ tramite Trotterization.

Nel caso con due spin emerge che la curva con Trotterization coincide con la curva classica. Il motivo di ciò è da ricondursi al fatto che gli operatori su cui avviene la decomposizione commutano e pertanto, poiché l'errore introdotto nella eq. (2.10) dipende anche dal commutatore degli operatori, in questo caso l'errore è nullo.

Per studiare la Trotterization nel caso a 3 spin si confrontano diversi Trotter steps, ottenendo le curve fig. 2.7

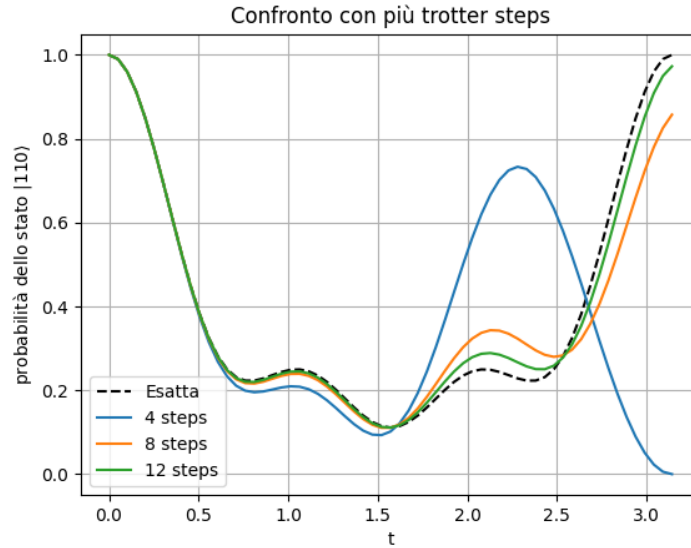


Figura 2.7: Probabilità nel tempo di rimanere nello stato $|110\rangle$ tramite Trotterization con 3 differenti Trotter steps.

In questo caso tutte le curve presentano un errore e nessuna di esse descrive perfettamente la curva esatta.

La curva con 4 Trotter steps diverge dal risultato a metà dell'intervallo temporale. Qualitativamente predice che il sistema alla fine dell'intervallo si trovi fuori dallo stato $|110\rangle$ con probabilità certa.

Le altre due curve si avvicinano al risultato, sebbene anche in questi casi non viene predetto il fatto che alla fine il sistema si trovi in $|110\rangle$, ma qualitativamente ci si avvicinano. Con le curve prese in esame il risultato migliore viene ottenuto con 12 Trotter steps.

Questo problema può essere risolto aumentando il numero di Trotter steps e conseguentemente la lunghezza del circuito. Ad esempio, utilizzando 50 Trotter steps la simulazione approssima perfettamente la curva esatta, come mostrato nella fig. 2.8

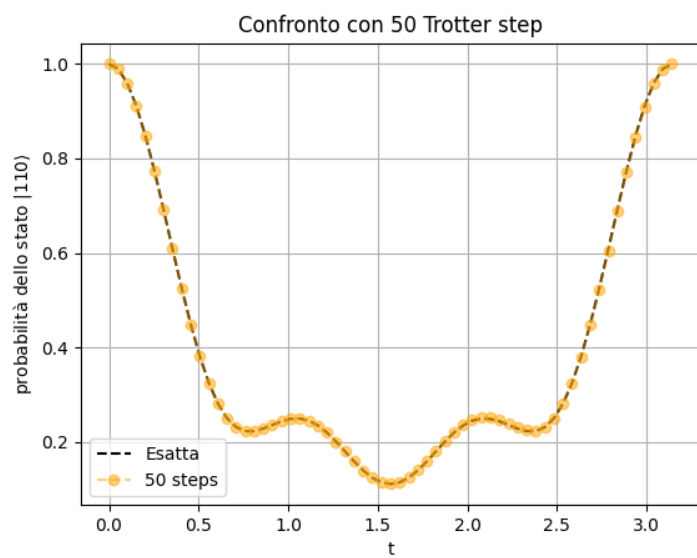


Figura 2.8: Probabilità nel tempo di rimanere nello stato $|110\rangle$ tramite Trotterization con 50 Trotter steps.

2.6 Andamento dell'errore

Dalla eq. (2.10) emerge la dipendenza dell'errore dal numero di Trotter step e dal tempo. E' possibile andare a studiare l'errore relativo tenendo uno dei due valori fisso. Nel caso in cui si faccia variare il numero di Trotter step ad un tempo fissato, ad esempio $t = 1.5$, è atteso un andamento di $\varepsilon = \frac{|true_val - trott_val|}{true_val}$ del tipo $\varepsilon \sim \frac{1}{n}$ come mostrato nella fig. 2.9

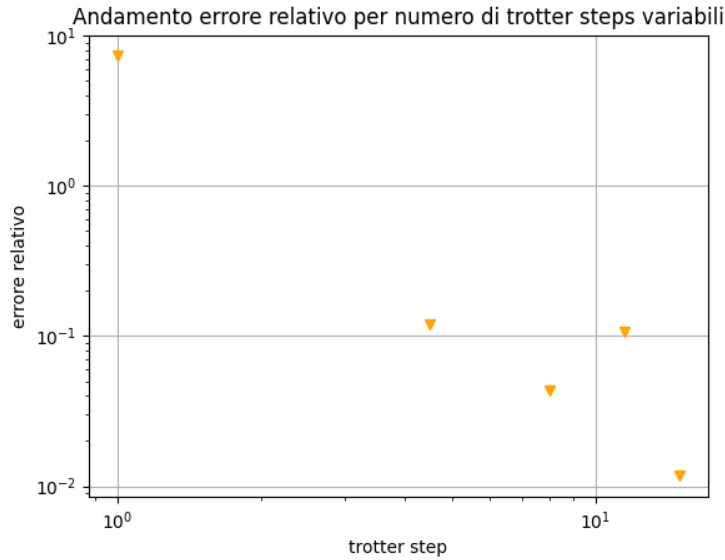


Figura 2.9: Errore relativo con $t = 1.5$ e Trotter steps variabili.

L'andamento di tale errore permette di trovare un valore per n di riferimento. In questo caso con $n = 12$ si nota come l'errore relativo sia abbastanza piccolo al centro dell'intervallo temporale da renderlo un valore accettabile senza dover utilizzare un circuito più lungo.

Per quanto riguarda l'errore con n fissato e facendo variare t si ottiene, in scala logaritmica, i risultati in fig. 2.10.

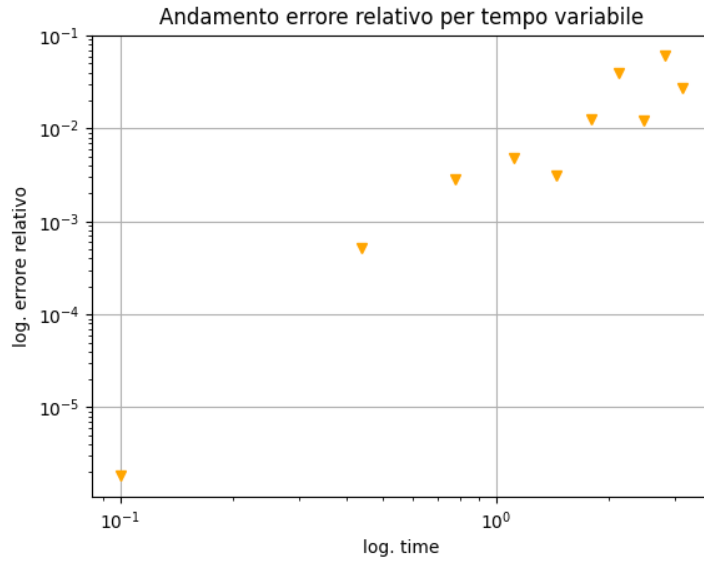


Figura 2.10: Errore relativo con $n = 12$ e tempo variabile in scala logaritmica.

Il circuito che implementa uno step della Trotterization è illustrato in fig. 2.11

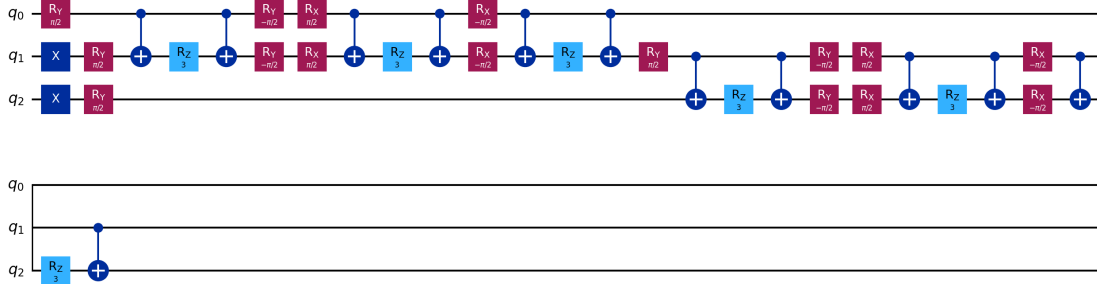


Figura 2.11: Circuito quantistico per l'implementazione di uno step della Trotterization.

2.7 Il rumore nell'hardware

Nell'ottica di implementare il nostro algoritmo su hardware quantistico reale, dobbiamo introdurre il concetto di rumore. Infatti nelle sezioni precedenti si è operato ipotizzando un sistema perfettamente chiuso e senza interazioni con l'esterno. Per quanto questo approccio possa essere utile per trarre alcune conclusioni preliminari, al momento della simulazione su un vero hardware è impossibile ottenere un sistema isolato. Per questo motivo, dobbiamo introdurre il concetto di stati misti. Dato un *ensemble* di stati quantistici $|\psi_i\rangle$ con probabilità p_i con il sistema in esame che si trova in uno degli stati dell'ensemble, si definisce *matrice di densità* ρ come

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i| \quad (2.17)$$

Dato un operatore U , si ottiene che l'effetto dell'operatore sulla matrice di densità ρ è il seguente

$$\sum_i p_i U |\psi_i\rangle \langle \psi_i| U^\dagger = U \rho U^\dagger \quad (2.18)$$

Per studiare come l'ambiente influenza il sistema si sviluppa il formalismo delle *quantum operations*. Data una matrice di densità iniziale ρ , definendo una mappa \mathcal{E} , ovvero una quantum operation, ρ' ovvero la matrice di densità evoluta dopo il processo è dato da

$$\rho' = \mathcal{E}(\rho) \quad (2.19)$$

Un modo per trattare un sistema aperto è considerare lo stato come un prodotto tensoriale di una parte del sistema ρ e una parte data dall'ambiente, ovvero ρ_{env} ¹¹. L'insieme di sistema e ambiente viene considerato un sistema chiuso. Data U la trasformazione del sistema, si ipotizza che l'ambiente non interagisca col sistema trasformato in ρ' . Si può dunque effettuare una traccia parziale rispetto all'ambiente su sistema e ambiente evoluti, ottenendo così uno stato del solo sistema.

$$\mathcal{E}(\rho) = tr_{env}[U(\rho \otimes \rho_{env})U^\dagger] \quad (2.20)$$

¹¹Si tratta di un'approssimazione in quanto nella realtà l'interazione è correlata e non in uno stato puro

Tramite la libreria *Qiskit Aer* è possibile costruire un modello che integri differenti tipi di noise in modo da effettuare delle simulazioni di come il sistema in cui si trova l'hardware possa influenzare le curve ottenute precedentemente in un sistema isolato.¹²

Bit flip e phase flip Il *bit flip* è un noise che introduce una probabilità $(1 - p)$ che lo stato di un qubit venga cambiato da $|0\rangle$ a $|1\rangle$ e viceversa. Il *phase flip* similmente introduce una probabilità che il qubit venga specchiato lungo l'asse z cambiando segno. L'introduzione di entrambi viene detto *bitphase flip*. L'effetto del bitphase flip sulla sfera di Bloch è visualizzato nella fig. 2.12. Pertanto si è introdotto nel modello un bitphase flip secondo questa logica:

- Se un gate agisce sul qubit vi è probabilità di flip $p_{gate} = 0.005$;
- Misurando un qubit vi è probabilità che la misura avvenga sul qubit flippato con $p_{meas} = 0.002$;
- Nella creazione di un qubit questo può essere flippato con probabilità $p_{reset} = 0.003$

ottenendo le curve fig. 2.12

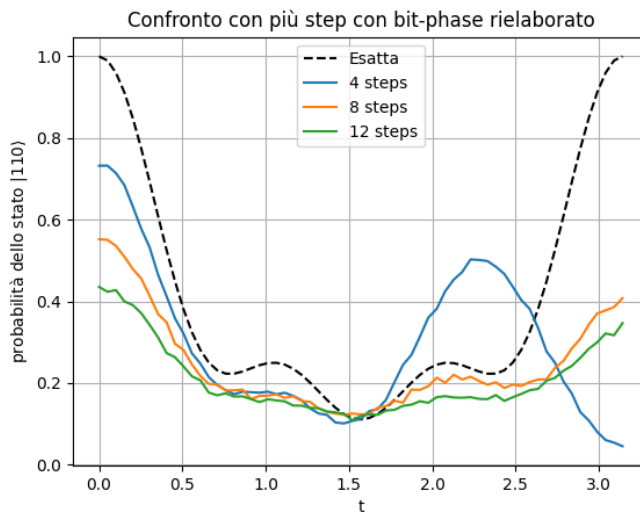


Figura 2.12: Le curve a tre spin con noise bitphase-flip.

¹²Michael A. Nielsen e Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

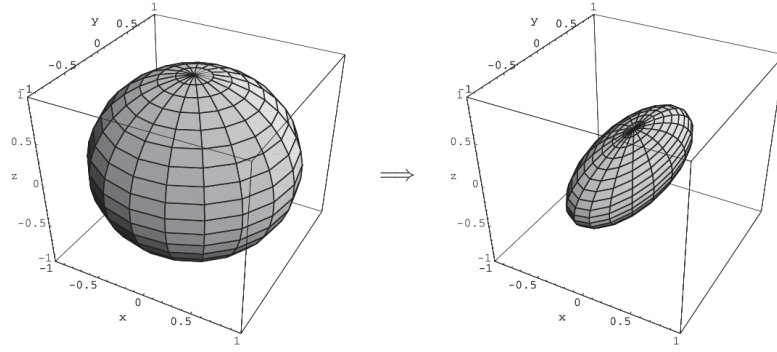


Figura 2.13: Rappresentazione sulla sfera di Bloch di come agisce il bitphase flip sugli stati della sfera. Gli assi x e z sono contratti come funzione della probabilità.

Depolarizing channel Il noise di *depolarizing* introduce sul singolo qubit una probabilità che questo divenga uno stato misto. Rispetto alla sfera di Bloch è la contrazione uniforme rispetto ad ogni direzione come mostrato in fig. 2.14. Si è introdotto nel modello una probabilità di ottenere una depolarizzazione per ogni CNOT pari a 0.001.

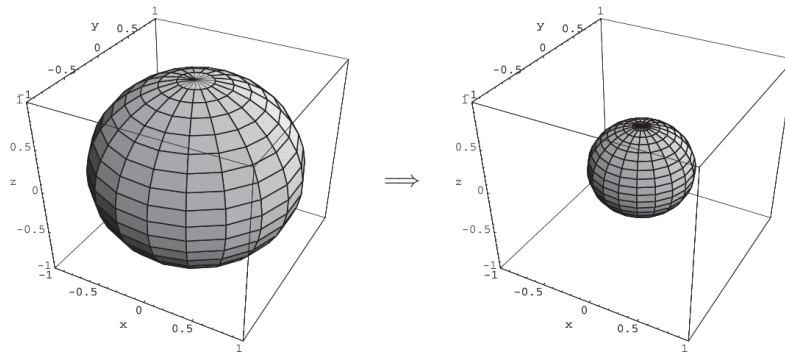


Figura 2.14: Rappresentazione sulla sfera di Bloch della contrazione uniforme dovuta al depolarizing noise.

Amplitude e phase damping L'amplitude damping è un noise che modella gli effetti della dissipazione energetica del sistema con l'ambiente in modo dipendente da un parametro $\gamma = \sin^2(\theta)$ con θ randomico.

Il phase damping è legato invece alla perdita di informazione quantistica senza uno scambio d'energia con l'ambiente ed è legato ad un gate di rotazione $R_z(\theta)$ con θ variabile aleatoria distribuita in modo gaussiano con media in $\theta = 0$. Si sono introdotti nel modello

sia il phase che l'amplitude damping con parametro $\gamma = 0.01$.

Si è costruito dunque il noise model con i tipi di rumore sopracitati e rispettivi parametri. Le curve ottenute con il noise model sono mostrate in fig. 2.15. Qiskit permette la simulazione con dei *fake-provider* che rappresentano un modello quanto più simile a quello di una vera simulazione su QPU, pertanto confrontando la perturbazione delle curve di fig. 2.15 con quelle di qiskit fig. 2.16, si osserva che il Noise Model costruito approssima la simulazione su una QPU senza tecniche di ottimizzazione del circuito e di mitigazione dell'errore.

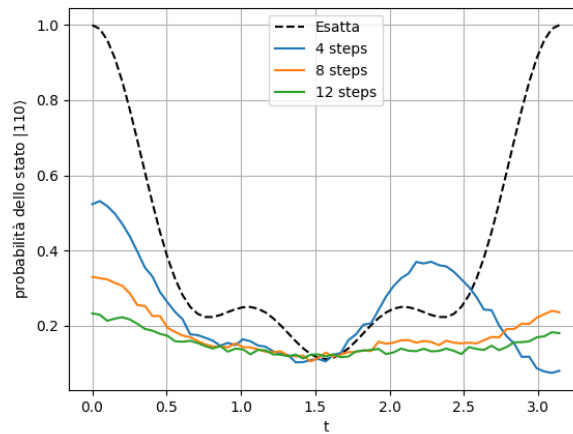


Figura 2.15: Curve per 3 spin con il noise model costruito.

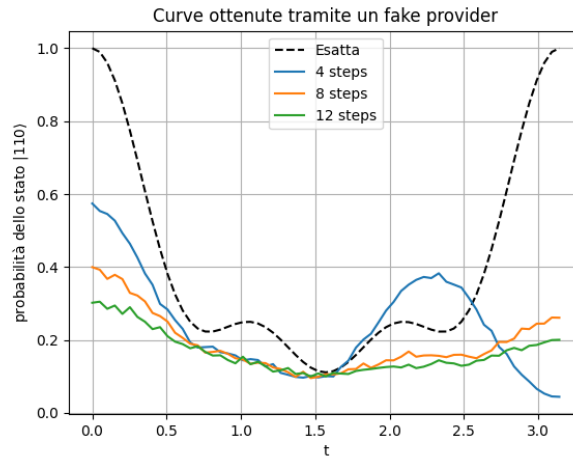


Figura 2.16: Curve per 3 spin con una fake backend di Qiskit.

2.8 Ottimizzazione del circuito

Qiskit permette di effettuare un processo chiamato *transpilation* rispetto ai QPU di *IBM Quantum*. Si tratta di un processo di riscrittura del circuito in input in modo che quest'ultimo possa essere eseguito al meglio sulla topologia del dispositivo quantistico specifico al fine di ridurre gli effetti del noise tramite una riduzione della profondità e della complessità del circuito.¹³

Inoltre fa in modo di far combaciare le istruzioni del circuito con la *Instruction Set Architecture* (ISA) di una particolare backend, ovvero trasformare il circuito in una serie di istruzioni che sono supportate dall'hardware di riferimento rispettandone i limiti e le integrazioni fisiche. Alcuni dei passaggi che effettua sono dunque:

- Inizializzazione delle istruzioni per la conversione dei gate a multi-qubit in gate singoli e a due qubit;
- Mappatura dei qubit virtuali su qubit fisici della QPU in fig. 2.17;
- Compatibilità delle istruzioni del circuito originale rispetto alle istruzioni della QPU;

¹³*Transpilation Guide*. Accessed: 2025-02-14. IBM Quantum, n.d. URL: <https://docs.quantum.ibm.com/guides/transpile>.

- Ottimizzazione nella decomposizione del circuito quantistico finché alcune caratteristiche non vengono soddisfatte, ad esempio una lunghezza fissata del circuito.

L'utente può controllare la fase di ottimizzazione tramite l'uso integrato di tre livelli di ottimizzazione che implementano differenti metodi. Si è effettuato un confronto tra gli errori relativi dei tre livelli di ottimizzazione in fig. 2.18.

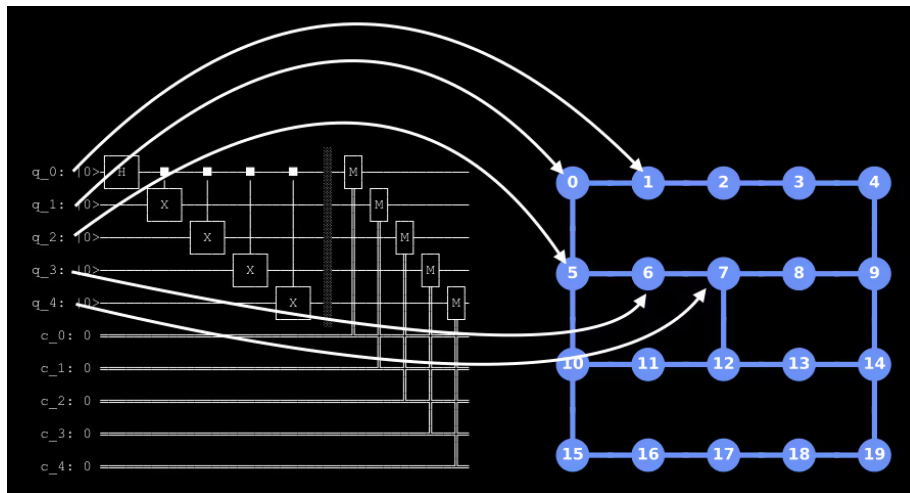


Figura 2.17: Processo di mappatura dei qubit virtuali sull'hardware. Un modo potrebbe essere quello di mappare i numeri dei qubit in modo corrispondente con quelli reali, oppure fare in modo di disporre i qubit in modo tale che i gate a due qubit agiscano su coppie adiacenti.

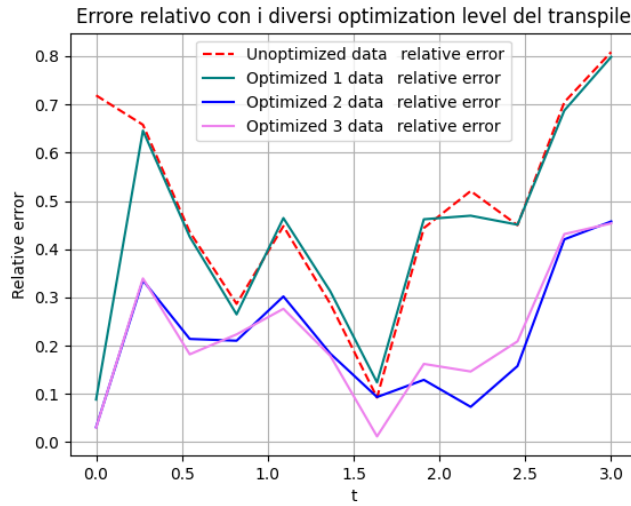


Figura 2.18: Confronto degli errori relativi di ciascuna curva rispetto ai tre livelli di ottimizzazione del transpile e ai dati non ottimizzati.

Dal grafico in fig. 2.18 emerge come il livello di ottimizzazione 2 rende l'errore relativo minore nel nostro circuito, infatti confrontandolo con la curva non ottimizzata si ottiene fig. 2.19.

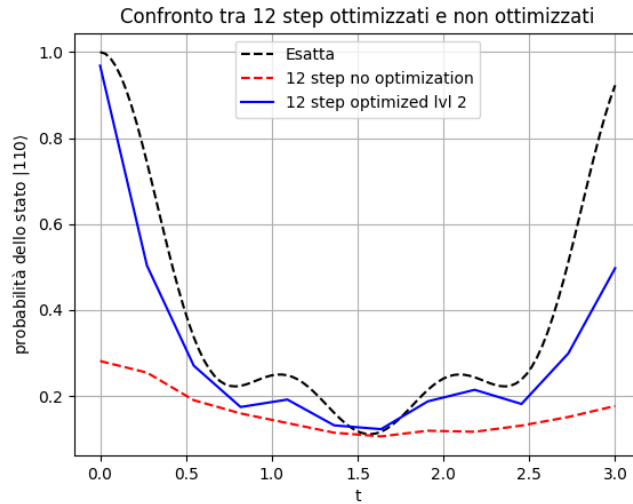


Figura 2.19: Curva a 3 spin con $n = 12$ ottimizzata.

La transpilation opera sulla struttura stessa del circuito che diviene significativamente diverso in fig. 2.20 rispetto a quello senza transpilation della fig. 2.11. La transpilation è

avvenuta rispetto alla fake backend *FakeMontrealV2* di Qiskit.

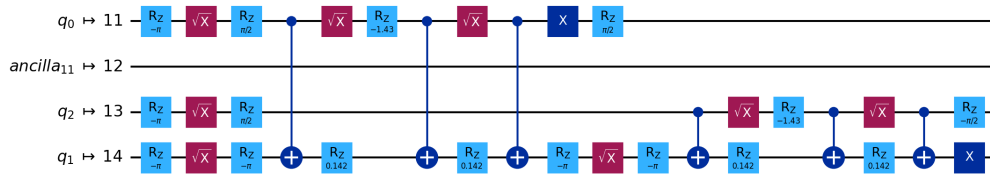


Figura 2.20: Circuito di un singolo Trotter step con $n = 12$ dopo la transpilation su Fake-MontrealV2.

2.9 Variazione dei parametri

Un approccio studiato per la riduzione dell'errore consiste nel variare la lunghezza del circuito al crescere del tempo. Pertanto si è effettuato uno studio sull'andamento dell'errore rispetto a questo metodo di implementazione della Trotterization.

Si sono effettuati differenti confronti, ad esempio confrontare le curve con 12 Trotter steps nel caso in cui la lunghezza sia tenuta fissata da $t = 0$ e nel caso in cui il numero di Trotter step viene aumentato gradualmente, dunque facendo crescere linearmente la lunghezza del circuito fig. 2.21.

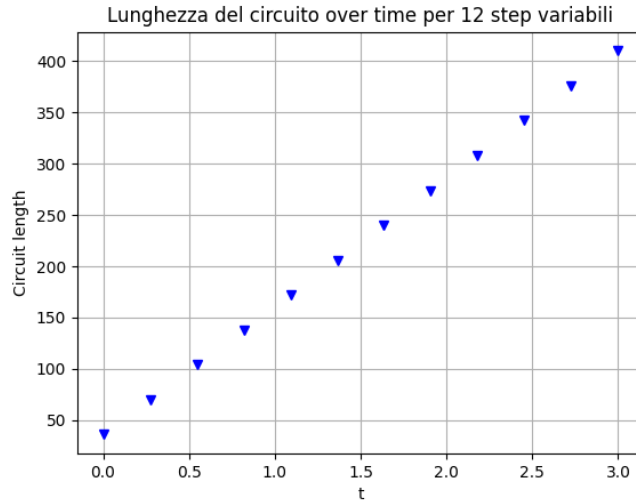


Figura 2.21: Lunghezza del circuito nel caso in cui viene fatto aumentare nel tempo il numero di Trotter steps fino a $n = 12$.

In questo modo si sono ottenute le curve in fig. 2.22.

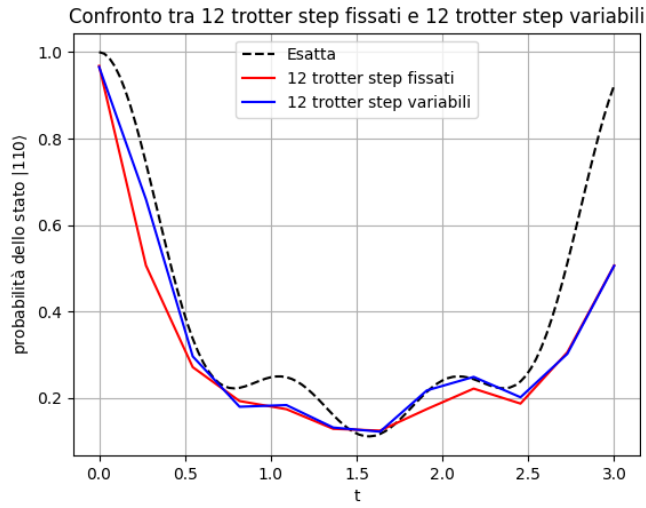


Figura 2.22: Confronto a $n = 12$ delle curve su tre spin con il numero di Trotter steps raggiunto linearmente e fissato a $t = 0$.

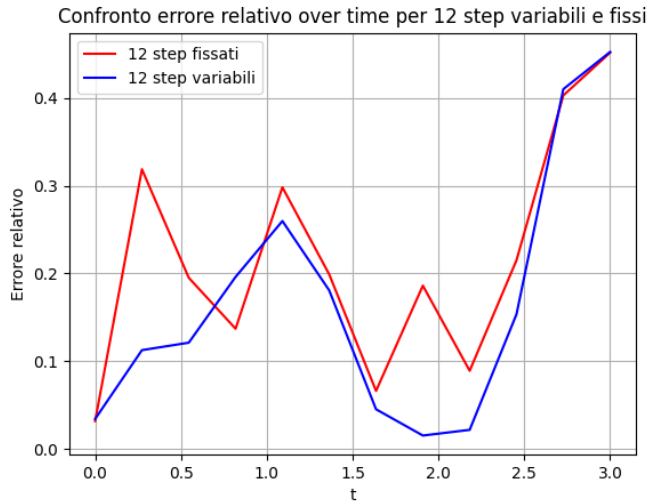


Figura 2.23: Errore relativo con $n = 12$ fissato e raggiunto in modo lineare.

Lo stesso studio è stato effettuato confrontando diverse casistiche, ad esempio confrontare con i due metodi $n = 60$ in fig. 2.24.

Con $n = 60$ la lunghezza del circuito è tale da avere un impatto del noise troppo alto e dunque in entrambi i casi la curva non è significativa, tuttavia il secondo metodo approssima meglio la curva per i tempi piccoli poiché la lunghezza del circuito è ancora tale da

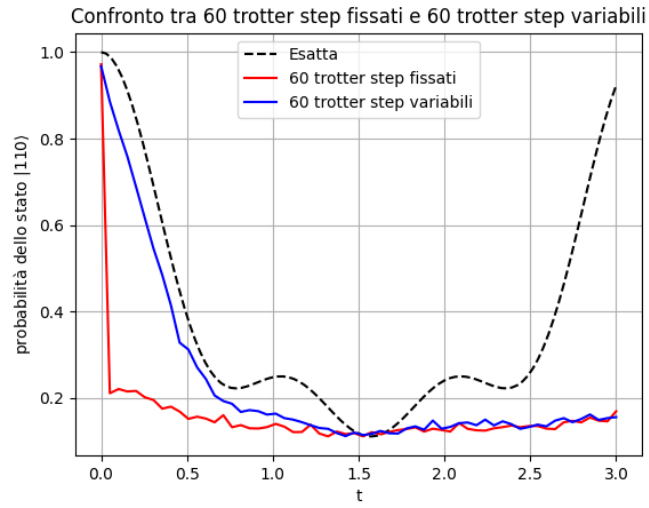


Figura 2.24: Confronto a $n = 60$ delle curve su tre spin con il numero di Trotter steps raggiunto linearmente e fissato a $t = 0$.

non essere influenzata dal noise come nel primo caso.

Aumentare il numero di Trotter steps sopra $n = 12$ non è pertanto un approccio utile al problema in quanto aumenta il noise e la lunghezza del circuito.

2.10 Tecniche di mitigazione dell'errore

La teoria della correzione degli errori nell'informazione quantistica è essenziale per l'applicazione pratica degli strumenti quantistici. Un qubit opera, infatti, nella scala delle lunghezze atomiche e l'accoppiamento con un qualsiasi fenomeno ambientale rischia di distruggerne lo stato. Inoltre il collasso della funzione d'onda impedisce di effettuare una correzione dell'errore che si basi sulla misura del qubit poiché ne altera l'informazione. Un primo approccio consiste nell'utilizzo di qubit definiti *ancelle* che vengono accoppiati tramite l'utilizzo di porte logiche adeguate e che permettono di estrarre delle relazioni sui qubit originari nei limiti del teorema di non clonazione.¹⁴ Le seguenti tecniche di mitigazione sono implementabili nel codice grazie al parametro *resilience_level* dell'Estimator di Qiskit¹⁵.

Gate twirling Il *twirling* è una tecnica utilizzata nella conversione di canali di rumore arbitrari in canali di rumore con strutture specifiche. Nella tesi si è utilizzato il *Pauli twirling*¹⁶, specifico nell'utilizzo di operazioni di Pauli. E' in grado di mitigare il rumore *coerente* in modo tale che si accumuli linearmente e non più in modo quadratico rispetto al numero di operazioni.

La tecnica si basa sull'utilizzare porte di Pauli randomiche da applicare prima e dopo determinati gate del circuito, in particolare su gate a due qubit che sono fonte di errore.

Twirled Readout Error eXtinction La *Twirled readout error extinction* (TREX) mitiga l'effetto degli errori di misura per il valore di aspettazione riguardo osservabili composte da porte di Pauli. Similmente al gate twirling opera sugli operatori di misura sostituendo a quest'ultimi un circuito di X gate, operatore di misura e un gate NOT classico come in

¹⁴N. David Mermin. *Quantum Computer Science: An Introduction*. Cambridge University Press, 2007.

¹⁵*Error Mitigation and Suppression Techniques*. Accessed: 2025-02-14. IBM Quantum, n.d. URL: <https://docs.quantum.ibm.com/guides/error-mitigation-and-suppression-techniques>.

¹⁶Joel J. Wallman e Joseph Emerson. *Noise tailoring for scalable quantum computation via randomized compiling*. Vol. 94. 5. American Physical Society (APS), 2016. DOI: 10.1103/physreva.94.052325.

fig. 2.25. Il readout error in combinazione con gli altri metodi di mitigazione è in grado di ridurre l'errore¹⁷.

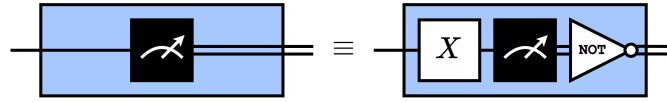


Figura 2.25: Sostituzione di un operatore di misura con la TREX.

Zero-noise extrapolation La *Zero-noise extrapolation* (ZNE) permette di mitigare gli errori sul valore di aspettazione di un osservabile. L'applicazione della ZNE consiste di due passaggi¹⁸

1. Amplificazione del rumore, in cui il circuito originale viene eseguito diverse volte con diversi gradi di rumore.
2. Estrapolazione¹⁹, in cui il risultato ideale viene estrapolato dal valore di aspettazione con il rumore amplificato.

Qiskit effettua l'amplificazione del rumore tramite il *digital gate folding*, in cui le porte a due qubit sono sostituite con una sequenza equivalente di gate e dell'inverso. Sostituire un operatore U con $UU^\dagger U$ amplifica il rumore sul gate di un fattore tre. L'estrapolazione avviene tramite un fit lineare o esponenziale. Il processo è mostrato in fig. 2.26.

Dynamical decoupling Il *dynamical decoupling* (DD) è un ulteriore accorgimento che permette di migliorare²⁰ l'esecuzione dei circuiti sull'hardware quantistico reale imple-

¹⁷Miroslav et al. Urbanek. *Mitigating Depolarizing Noise on Quantum Computers with Noise-Estimation Circuits*. Vol. 127. 27. American Physical Society (APS), 2021. DOI: 10.1103/physrevlett.127.270502.

¹⁸Ryan et al. LaRose. *Mitiq: A software package for error mitigation on noisy quantum computers*. Vol. 6. Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, 2022, p. 774. DOI: 10.22331/q-2022-08-11-774.

¹⁹Vincent Russo e Andrea Mari. *Quantum error mitigation by layerwise Richardson extrapolation*. Vol. 110. 6. American Physical Society (APS), 2024. DOI: 10.1103/physreva.110.062420.

²⁰Petar et al. Jurcevic. *Demonstration of quantum volume 64 on a superconducting quantum computing system*. Vol. 6. 2. IOP Publishing, 2021, p. 025020. DOI: 10.1088/2058-9565/abe519.

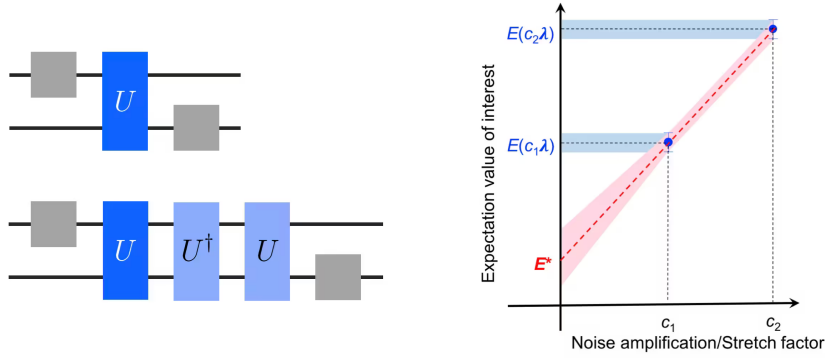


Figura 2.26: Applicazione della ZNE con fase di digital folding ed estrapolazione del dato noiseless.

mentato da Qiskit. Nella mappatura di un circuito quantistico su quantum processor unit, non tutti i gate fisici possono essere eseguiti contemporaneamente e i tempi di esecuzione dei gate possono variare tra le singole operazioni. Inoltre l'architettura dell'hardware utilizzato determina quanti gate fisici possono essere eseguiti contemporaneamente e in quale ordine. Il metodo DD permette di trattare in modo approssimato i gate che sono in attesa di essere eseguiti come operatori identitari tramite l'utilizzo di differenti tecniche²¹.

L'ambito delle tecniche di mitigazione dell'errore è vasto e diverse tecniche di mitigazione richiedono implementazioni più o meno complesse. Per la trattazione del problema della tesi si sono utilizzate le tecniche sopracitate, ma metodi più complessi possono essere utilizzati per affinare ulteriormente la misura²².

²¹Poulami et al. Das. *ADAPT: Mitigating Idling Errors in Qubits via Adaptive Dynamical Decoupling*. ACM, 2021. DOI: 10.1145/3466752.3480059.

²²Ewout et al. van den Berg. *Probabilistic error cancellation with sparse Pauli-Lindblad models on noisy quantum processors*. Vol. 19. 8. Springer Science e Business Media LLC, 2023, pp. 1116–1121. DOI: 10.1038/s41567-023-02042-2.

3. Simulazione su Quantum Processor Unit

Tramite il servizio di *IBM Quantum* è stato possibile effettuare la simulazione su un *Quantum Processor Unit* (QPU). Le QPU di IBM Quantum sono basate su qubit superconduttivi¹. Il design di un qubit superconduttivo permette un'elevata scalabilità, differenti parametri su cui operare e una facilitazione nel creare sistemi accoppiati di qubit. Tuttavia il tempo di coerenza di un qubit superconduttivo è breve, motivo per cui la creazione di computer quantistici presenta ancora delle sfide, ad esempio il mantenimento delle basse temperature che garantiscono la superconduttività tramite coppie di Cooper².

Mettendo in atto le osservazioni compiute sulle simulazioni preliminari è stato possibile giungere alla simulazione sulla QPU avendo chiara l'idea di quanti Trotter steps usare, come gestire il noise e come trattare la lunghezza del circuito per ridurre l'errore.

E' stato significativo una simulazione su QPU senza alcun tipo di osservazione su questi parametri che ha mostrato quanto in una QPU reale il rumore e l'errore rendono inutilizzabili i dati senza gli adeguati accorgimenti, come mostrato in fig. 3.1.

¹*Quantum Processing Units Information*. Accessed: 2025-02-18. IBM Quantum, n.d. URL: <https://docs.quantum.ibm.com/guides/qpu-information>.

²He-Liang Huang et al. *Superconducting quantum computing: a review*. Vol. 63. 8. Springer Science e Business Media LLC, lug. 2020. DOI: 10.1007/s11432-020-2881-9. URL: <http://dx.doi.org/10.1007/s11432-020-2881-9>.

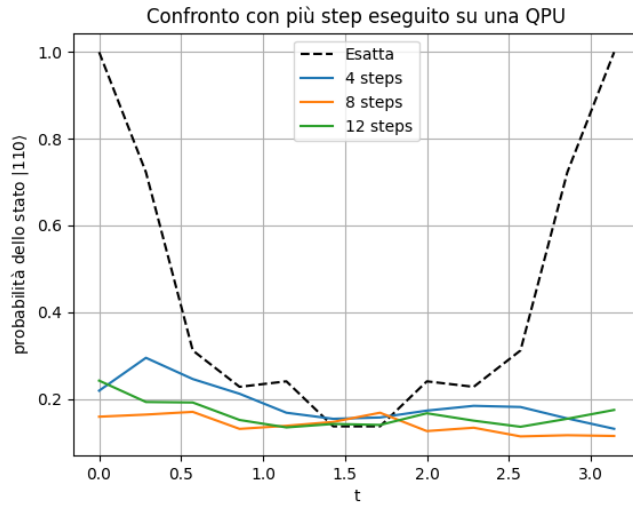


Figura 3.1: Simulazione su QPU senza l'ottimizzazione del circuito e del noise.

Si è effettuata la simulazione con le diverse tecniche analizzate, ottenendo i valori della curva in table 3.1 e le curve simulate di fig. 3.2. Effettuando la simulazione con $n = 12$ e con la transpilation del circuito si è ottenuta una stima dei valori migliore, come previsto. La simulazione su QPU è peggiore della simulazione con fake backend in quanto quest'ultima si basa solo su un'approssimazione del rumore reale.

Andando ad implementare il metodo di incremento lineare della lunghezza del circuito si è ottenuta una curva di migliore approssimazione, specialmente per tempi piccoli dell'intervallo.

Il processo di transpilation è stato utile ad effettuare una simulazione che si avvicinasse alla curva teorica. La simulazione senza la transpilation si allontana dalla curva, presentando un massimo a metà intervallo.

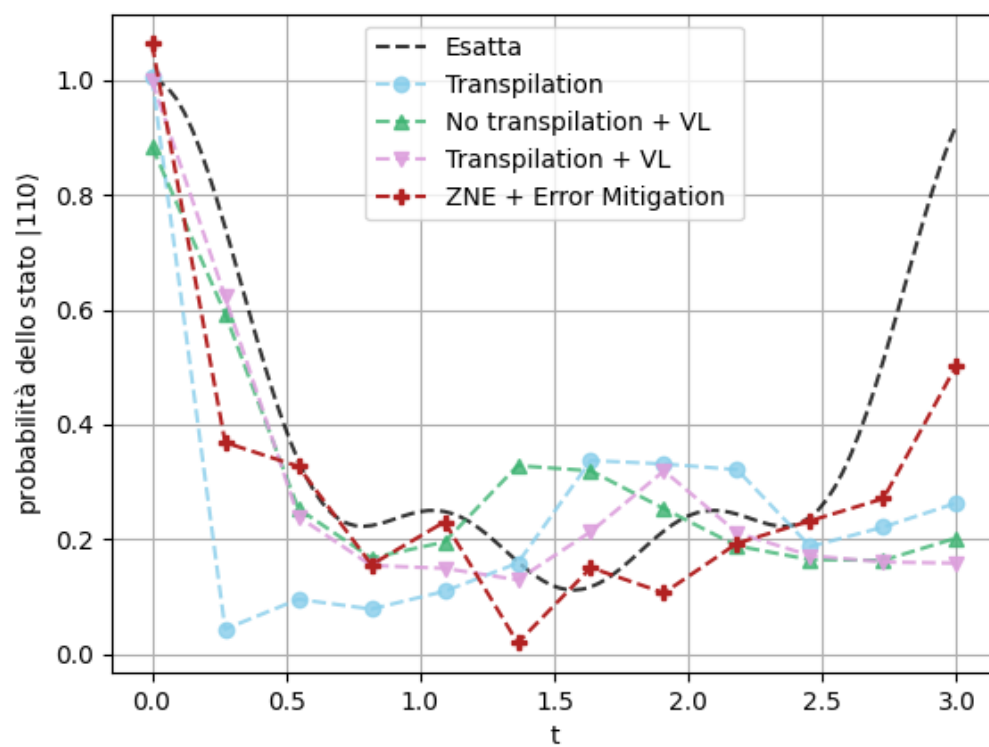


Figura 3.2: Confronto delle curve simulate su QPU **ibm_kyiv** da 127 qubit.

Time	Exact	Transpilation	VL	Transpilation + VL	Error Mitigation
0	1.0	1.005	0.884	0.999	1.063
0.27	0.744	0.043	0.593	0.622	0.368
0.55	0.337	0.095	0.253	0.239	0.327
0.82	0.224	0.078	0.167	0.154	0.155
1.09	0.248	0.109	0.195	0.149	0.230
1.36	0.161	0.158	0.328	0.129	0.019
1.64	0.117	0.337	0.319	0.213	0.150
1.91	0.214	0.331	0.253	0.319	0.106
2.18	0.243	0.321	0.188	0.211	0.193
2.45	0.238	0.187	0.164	0.171	0.232
2.73	0.512	0.221	0.163	0.160	0.271
3	0.993	0.263	0.202	0.158	0.503

Tabella 3.1: Tabella che confronta i valori simulati con quelli esatti nelle tre casistiche, dunque con transpilation, senza transpilation ma con *Lunghezza variabile del circuito* (VL), transpilation + VL e con tecniche di mitigazione dell'errore.

	Transpilation	VL	Transpilation + VL	Error Mitigation
MAE%	22,69%	17,82%	15,90%	13,39%

Tabella 3.2: Tabella degli errori assoluti medi in percentuale *MAE%* dei tre metodi.

Si è studiato il valore assoluto medio delle simulazioni effettuate confrontandone i valori, come in tabella table 3.2. Come previsto il metodo con la transpilation con livello di ottimizzazione 2 e la lunghezza del circuito variabile è il metodo con errore assoluto medio minore 15,90%. Con l'implementazione delle tecniche di mitigazione dell'errore il MAE può essere ulteriormente ridotto al 13.39%.

Conclusioni

La tesi si è posta come obiettivo quello di mostrare quanto la computazione quantistica può rivelarsi uno strumento utile alla simulazione di sistemi quantistici e dei loro fenomeni. In particolare si è studiato come il sistema della catena di Heisenberg con pochi spin, nello specifico nel caso a due e a tre spin, potesse essere simulato con la teoria della computazione quantistica, confrontandolo con una simulazione classica che in questo caso è permessa dal costo computazionale.

Si è dunque proceduto a tradurre l'operatore di evoluzione temporale decomposto tramite il metodo di Trotter-Suzuki in operazioni sui qubit del circuito quantistico sotto forma di porte logiche. La costruzione del circuito quantistico è avvenuto con l'ausilio di *qiskit*, una libreria in python che permette di costruire circuiti e manovrarne il funzionamento compatibilmente con i processori quantistici di *IBM Quantum*.

In una prima fase si sono verificati gli errori algoritmici introdotti nei casi a 2 spin e 3 spin, andando a considerare l'errore rispetto al numero di Trotter step e al tempo al fine di trovare un valore per questi parametri che potesse minimizzare l'errore senza aumentare eccessivamente il costo di computazione. Si è deciso nel caso a 3 spin di effettuare la simulazione della curva dell'overlap dello stato $|110\rangle$ nel tempo con 12 Trotter step.

Dopodiché si è andato a studiare il modo in cui il rumore ambientale impatta sull'hardware nella simulazione costruendo un modello che implementasse alcune delle tipologie di errore che avvengono sui qubit del circuito ad opera dell'ambiente.

La tesi si è poi concentrata nell'analisi dei metodi che vengono utilizzati per ottimizzare il circuito per l'hardware specifico di IBM Quantum, implementati dal metodo della *transpilation* da parte di Qiskit. Si è studiato l'errore relativo dalla curva esatta nei tre livelli di ottimizzazione implementati concludendo che il secondo livello è quello che ri-

duce maggiormente l'errore nel caso specifico del circuito, facendo scendere il numero di CNOT gates da 144 a 72.

Per ridurre ulteriormente l'errore si è utilizzato un approccio *variable length* (VL) che consiste nell'aumentare in modo lineare la lunghezza del circuito nel tempo, e dunque aumentare il numero di Trotter step ad ogni passo temporale. Si è rivelata una tecnica efficace che ha permesso di ridurre l'errore assoluto medio nella simulazione.

Prima di effettuare la simulazione vera e propria su quantum processor unit si sono studiate alcune delle tecniche di mitigazione dell'errore implementabili su un circuito quantistico, ovvero il Gate Twirling, la Twirled Readout Error eXtinction, la Dynamical Decoupling e la ZNE, quest'ultima con un metodo di amplificazione del rumore basato sul digital folding e un'estrapolazione lineare del valore atteso.

Grazie a questi accorgimenti si è potuta effettuare una simulazione con un errore assoluto medio del 13,39%. Si tratta di un risultato che potrà essere affinato con l'utilizzo di tecniche di mitigazione dell'errore più avanzate. Inoltre sviluppando hardware sempre più stabili e meno soggetti ad influenze esterne si potranno utilizzare i vantaggi della computazione quantistica per simulare i fenomeni di reticoli sempre più grandi e in più dimensioni.

Bibliografia

- [1] Daniel S. Abrams e Seth Lloyd. *Simulation of Many-Body Fermi Systems on a Universal Quantum Computer*. Vol. 79. 13. American Physical Society (APS), 1997, pp. 2586–2589. DOI: 10.1103/physrevlett.79.2586.
- [2] Ewout et al. van den Berg. *Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors*. Vol. 19. 8. Springer Science e Business Media LLC, 2023, pp. 1116–1121. DOI: 10.1038/s41567-023-02042-2.
- [3] Poulami et al. Das. *ADAPT: Mitigating Idling Errors in Qubits via Adaptive Dynamical Decoupling*. ACM, 2021. DOI: 10.1145/3466752.3480059.
- [4] *Error Mitigation and Suppression Techniques*. Accessed: 2025-02-14. IBM Quantum, n.d. URL: <https://docs.quantum.ibm.com/guides/error-mitigation-and-suppression-techniques>.
- [5] He-Liang Huang et al. *Superconducting quantum computing: a review*. Vol. 63. 8. Springer Science e Business Media LLC, lug. 2020. DOI: 10.1007/s11432-020-2881-9. URL: <http://dx.doi.org/10.1007/s11432-020-2881-9>.
- [6] Petar et al. Jurcevic. *Demonstration of quantum volume 64 on a superconducting quantum computing system*. Vol. 6. 2. IOP Publishing, 2021, p. 025020. DOI: 10.1088/2058-9565/abe519.
- [7] Ryan et al. LaRose. *Mitiq: A software package for error mitigation on noisy quantum computers*. Vol. 6. Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, 2022, p. 774. DOI: 10.22331/q-2022-08-11-774.
- [8] Ying Li e Simon C. Benjamin. *Efficient Variational Quantum Simulator Incorporating Active Error Minimization*. Vol. 7. 2. American Physical Society (APS), 2017. DOI: 10.1103/physrevx.7.021050.

- [9] Robert Loredó. *Learn Quantum Computing with Python and IBM Quantum Experience*. Packt Publishing, 2020.
- [10] N. Manojlović e I. Salom. *Algebraic Bethe ansatz for the XXZ Heisenberg spin chain with triangular boundaries and the corresponding Gaudin model*. Vol. 923. Elsevier BV, 2017, pp. 73–106. DOI: 10.1016/j.nuclphysb.2017.07.017.
- [11] N. David Mermin. *Quantum Computer Science: An Introduction*. Cambridge University Press, 2007.
- [12] Jordan C. Moodie e Martin W. Long. *An exact power series representation of the Baker-Campbell-Hausdorff formula*. arXiv, 2018. URL: <https://arxiv.org/abs/1807.07884>.
- [13] Rafael I. Nepomechie. *A spin chain primer*. Vol. 13. 24n25. World Scientific Pub Co Pte Lt, 1999, pp. 2973–2985. DOI: 10.1142/s0217979299002800.
- [14] Michael A. Nielsen e Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [15] G. et al. Ortiz. *Quantum algorithms for fermionic simulations*. Vol. 64. 2. American Physical Society (APS), 2001. DOI: 10.1103/physreva.64.022319.
- [16] Bo Peng et al. *Quantum time dynamics employing the Yang-Baxter equation for circuit compression*. Vol. 106. 1. American Physical Society (APS), 2022. DOI: 10.1103/physreva.106.012412.
- [17] *Quantum Processing Units Information*. Accessed: 2025-02-18. IBM Quantum, n.d. URL: <https://docs.quantum.ibm.com/guides/qpu-information>.
- [18] Vincent Russo e Andrea Mari. *Quantum error mitigation by layerwise Richardson extrapolation*. Vol. 110. 6. American Physical Society (APS), 2024. DOI: 10.1103/physreva.110.062420.
- [19] Ingmar Saberi. *An introduction to spin systems for mathematicians*. arXiv, 2018. URL: <https://arxiv.org/abs/1801.07270>.

- [20] Masuo Suzuki. *General theory of fractal path integrals with applications to many-body theories and statistical physics*. Vol. 32. 2. American Institute of Physics, 1991, pp. 400–407. DOI: 10.1063/1.529425.
- [21] *Transpilation Guide*. Accessed: 2025-02-14. IBM Quantum, n.d. URL: <https://docs.quantum.ibm.com/guides/transpile>.
- [22] H. F. Trotter. *On the Product of Semi-Groups of Operators*. Vol. 10. 4. American Mathematical Society, 1959, pp. 545–551.
- [23] Miroslav et al. Urbanek. *Mitigating Depolarizing Noise on Quantum Computers with Noise-Estimation Circuits*. Vol. 127. 27. American Physical Society (APS), 2021. DOI: 10.1103/physrevlett.127.270502.
- [24] Joel J. Wallman e Joseph Emerson. *Noise tailoring for scalable quantum computation via randomized compiling*. Vol. 94. 5. American Physical Society (APS), 2016. DOI: 10.1103/physreva.94.052325.