

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

Simulazione di molecole tramite algoritmo Variational Quantum Eigensolver

Relatore:
Prof.ssa Elisa Ercolessi

Presentata da:
Davìd Ferrari

Anno Accademico 2022/2023

Sommario

Lo scopo di questa tesi è quello di introdurre l'algoritmo quantistico **VQE** (Variational Quantum Eigensolver) per il calcolo dell'energia di legame delle molecole H_2 e LiH . Il capitolo 1 riprende i concetti fondamentali della meccanica quantistica, introduce il formalismo dei qubit e descrive le principali caratteristiche dei circuiti quantistici. Il capitolo 2 presenta una trattazione formale dell'algoritmo ibrido VQE, descrivendo come sia possibile rappresentare sistemi fisici tramite un insieme di qubit. Il capitolo 3 presenta il framework `qiskit` per la scrittura di programmi quantistici eseguibili su simulatori, per esempio un laptop, o su dispositivi reali, accessibili in cloud. In particolare, si mostra una semplice applicazione del VQE per le due molecole H_2 , LiH , e una discussione dei risultati ottenuti.

Indice

1	Richiami di computazione quantistica	5
1.1	Spazi di Hilbert	5
1.2	Qubit	6
1.3	Qubit multipli ed entanglement	7
1.4	Modello a circuito della computazione quantistica	8
1.4.1	Porte quantistiche a singolo qubit	9
1.4.2	Porte quantistiche a qubit multipli	10
1.4.3	Operazioni controllate	11
1.5	Misurazione quantistica	12
1.5.1	Misurazione degli operatori di Pauli	12
1.6	Stringhe di Pauli	14
2	Variational Quantum Eigensolver	15
2.1	Descrizione formale del VQE	15
2.2	Rappresentazione dell'Hamiltoniana	16
2.2.1	Prima quantizzazione	17
2.2.2	Seconda quantizzazione	18
2.3	Trasformazione di Jordan-Wigner	18
2.3.1	Trasformazione di parità	20
2.4	Ansatz	21
2.4.1	Ansatz <i>a priori</i>	21
2.4.2	Rotazioni di Givens	22
2.5	Ottimizzazione dei parametri	22
2.5.1	Numero di misurazioni nel circuito quantistico	23
3	Implementazione del VQE mediante Qiskit per molecole	24
3.1	Qiskit	24
3.1.1	Dispositivi IBM	25
3.1.2	Configurazione dell'algoritmo	25
3.2	Simulazione e risultati	28
3.2.1	Energia dello stato fondamentale	28

3.2.2	Energia di legame	34
3.2.3	Risultati numerici	34
3.3	Simulazione su dispositivo reale	37
3.3.1	Energia dello stato fondamentale e fidelity	37
3.3.2	Energia di legame	40
A	Rappresentazione dei circuiti quantistici	42
A.1	Porte logiche a singolo qubit	42
A.2	Porte logiche a qubit multipli	43
A.3	Operazioni controllate	43
A.4	Misurazione e bit classici	43

Introduzione

Nella sua descrizione sulla *macchina universale* per la computazione, Alan Turing scrisse[1]:

È possibile costruire una singola macchina in grado di calcolare qualsiasi sequenza computabile.

In effetti, il progresso tecnologico nel campo della computazione, unito allo sviluppo della teoria dell'informazione ha permesso di raggiungere grandi traguardi in termini umanitari, sociali, e scientifici, inimmaginabili senza l'aiuto di un computer. L'ambizione di affrontare problemi sempre più complessi (simulazione di sistemi fisici senza approssimazioni, crittografia, modelli finanziari, ...) ha però messo in evidenza i limiti, probabilmente invalicabili, di questo paradigma. La ricerca di soluzioni sempre più precise di problemi sempre più complessi, seppur possibile in linea di principio, necessita di hardware dalle prestazioni sempre più elevate, per completare le operazioni in tempi accettabili. Inoltre, la nota legge di Moore sulla possibilità di raddoppiare la potenza di calcolo ogni circa 18 mesi sta raggiungendo il suo limite di validità [2], rendendo impossibile la creazione di computer dalle capacità di calcolo sufficienti per risolvere tali problemi.

La soluzione più studiata per superare i limiti degli attuali metodi computazionali è la **computazione quantistica**, un paradigma di calcolo basato sulla moderna teoria della meccanica quantistica; le tecniche che ne derivano, non rappresentano una novità solo in termini ingegneristici, ma rivoluzionano anche l'intero concetto di informazione: il bit classico, deterministico e governato dall'algebra discreta booleana, viene sostituito dal qubit, rappresentato da una sovrapposizione di stati appartenenti ad uno spazio continuo di Hilbert. Questa proprietà, insieme ad altri fenomeni di natura puramente quantistica come l'entanglement degli stati, conferisce a questo nuovo paradigma un potenziale di calcolo immensamente maggiore rispetto a quello classico, portando alla ricerca di algoritmi quantistici che possano risolvere problemi intrattabili, non risolvibili in tempi accettabili con gli attuali computer.

Tuttavia, questo vantaggio rispetto ai computer classici è difficilmente sfruttabile nella pratica, sia per motivi puramente implementativi, sia per motivi della teoria stessa, in quanto, ad esempio, nonostante un qubit esista in una sovrapposizione di stati e possa trasportare più informazione rispetto alla controparte classica, la misura dello stato del

qubit ne provoca il collasso della funzione d'onda trasformando l'informazione quantistica in informazione classica in maniera irreversibile. Lo studio di metodi in grado di sfruttare la potenza dei computer quantistici è attualmente una delle principali frontiere di ricerca della teoria, e ad oggi sono pochi gli algoritmi noti per risolvere problemi intrattabili classicamente.

Gli attuali limiti della teoria, uniti ai limiti tecnologici, hanno portato allo sviluppo di algoritmi che sfruttano solo parzialmente il potere di calcolo di hardware quantistici, facendoli lavorare accoppiati ad una componente classica ottimizzata. Tali algoritmi *ibridi* sono tipicamente implementati per determinare il minimo di una funzione obiettivo, mediante metodi variazionali; la componente quantistica ha il compito di preparare lo stato iniziale dell'algoritmo e di valutare il valore della funzione, mentre la parte classica ha l'obiettivo di ottimizzare i parametri variazionali ad ogni iterazione; il **Variational Quantum Eigensolver** è un esempio di un siffatto algoritmo, usato tipicamente per il calcolo dello stato fondamentale di molecole, ma applicabile ad una vasta gamma di problemi scientifici. La descrizione e l'applicazione di questo algoritmo a semplici molecole rappresentano il principale argomento di questa tesi.

L'avvento della computazione quantistica ha spinto grandi aziende come IBM a mettere a disposizione risorse hardware e software per chiunque sia interessato a sperimentare i concetti della computazione quantistica. La *IBM Quantum Platform* è una piattaforma online utilizzata per l'esecuzione di algoritmi su simulatori e dispositivi reali, e per la formazione sui concetti fondamentali della computazione quantistica; gli algoritmi sono direttamente implementabili tramite un framework in Python chiamato `qiskit`, mantenuto dalla IBM stessa e da una community open source. Tale piattaforma è stata utilizzata in questa tesi, che è strutturata nel seguente modo.

Il capitolo 1 introduce il concetto di qubit e il modello computazionale dei circuiti quantistici per l'implementazione di algoritmi. Il capitolo 2 descrive formalmente l'algoritmo ibrido VQE, trattando i singoli step che lo compongono e mostrando come sia possibile descrivere un sistema fisico su hardware quantistico. Infine, il capitolo 3 presenta il framework `qiskit` e descrive i risultati ottenuti dall'esecuzione dell'algoritmo su un simulatore e un dispositivo reale per le molecole H_2 e LiH .

Capitolo 1

Richiami di computazione quantistica

In questo capitolo viene presentata la base formale su cui si fonda la teoria della computazione quantistica. Si richiamano i postulati fondamentali presentando la formulazione astratta del concetto di Qubit e delle porte logiche quantistiche. Si conclude con la presentazione del modello a circuito quantistico. La discussione seguente ha come riferimento il testo [3].

1.1 Spazi di Hilbert

Le proprietà di linearità discendenti dall'equazione di Schrödinger si traducono in una naturale trattazione dei sistemi quantistici mediante l'algebra lineare. Si richiamano qui le definizioni fondamentali di prodotto interno e di spazio di Hilbert.

Definizione 1.1 (Prodotto interno) *Sia H uno spazio vettoriale complesso. Si definisce prodotto interno un'applicazione $\phi : H \times H \rightarrow \mathbb{C}$ tale che $\forall x, y, z \in H$ e $\forall \lambda \in \mathbb{C}$:*

- $\phi(x, x) \geq 0$, con $\phi(x, x) = 0 \Leftrightarrow x = 0$
- $\phi(x + y, z) = \phi(x, z) + \phi(y, z)$
- $\phi(\lambda x, y) = \lambda \phi(x, y)$
- $\phi(x, y) = \overline{\phi(y, x)}$

Nel seguito verrà adottata la notazione $\langle x, y \rangle$ per indicare il prodotto interno tra i vettori $x, y \in H$.

Dato un prodotto interno \langle, \rangle si può definire la *norma* di un vettore dello spazio $x \in H$ tramite la relazione:

$$\|x\| := \sqrt{\langle x, x \rangle} \quad (1.1.1)$$

La definizione della funzione norma permette di introdurre un concetto fondamentale alla base dell'intera teoria della meccanica quantistica: lo **spazio di Hilbert**.

Definizione 1.2 (Spazio di Hilbert) *Sia H uno spazio vettoriale complesso, dotato di prodotto interno \langle, \rangle . Se lo spazio risulta completo rispetto alla norma indotta da \langle, \rangle si dice che H è uno spazio di Hilbert.*

L'intera meccanica quantistica si fonda sul seguente postulato:

Postulato 1 *Ad ogni sistema quantistico chiuso corrisponde uno spazio di Hilbert \mathcal{H} che ne descrive univocamente lo stato tramite un vettore di norma unitaria. Tale spazio viene detto **spazio degli stati**.*

1.2 Qubit

Il **qubit** è l'unità fondamentale dell'intera teoria della computazione quantistica, analogamente al bit per la teoria classica. Formalmente, un qubit è interpretato come un sistema quantistico a due stati, per il quale sono ammessi solo due stati indipendenti, sicché lo spazio di Hilbert associato è bidimensionale. Indicando con $\{|0\rangle, |1\rangle\}$ una base ortonormale di \mathcal{H} si può esprimere il generico vettore di stato $|\psi\rangle$ nella forma:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad \alpha, \beta \in \mathbb{C} \quad (1.2.1)$$

dove la condizione di normalizzazione del vettore $|\psi\rangle$ impone la relazione:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (1.2.2)$$

A differenza di quanto avviene per un bit classico che può trovarsi solo nello stato 0 o 1, un qubit dalla (1.2.1) si troverà in una **sovrapposizione** degli stati $|0\rangle$ e $|1\rangle$ e in seguito ad una misura, la funzione collasserà nello stato $|0\rangle$ con probabilità $|\alpha|^2$ o nello stato $|1\rangle$ con probabilità $|\beta|^2$. La base $\{|0\rangle, |1\rangle\}$ dello spazio di Hilbert è detta *base computazionale*.

Sfruttando la relazione di Eulero per i numeri complessi si può riscrivere la (1.2.1) come :

$$|\psi\rangle = |\alpha| \exp(i\phi_a) |0\rangle + |\beta| \exp(i\phi_b) |1\rangle, \quad \phi_a, \phi_b \in [0, 2\pi) \quad (1.2.3)$$

Inoltre, dalla (1.2.2) si può imporre:

$$|\alpha| = \cos\left(\frac{\theta}{2}\right), \quad |\beta| = \sin\left(\frac{\theta}{2}\right) \quad \theta \in [0, \pi] \quad (1.2.4)$$

da cui segue:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + \cos\left(\frac{\theta}{2}\right) \exp(i\gamma) |1\rangle \quad (1.2.5)$$

con $\gamma = \phi_b - \phi_a$, dove si è trascurato il fattore di fase globale $\exp(i\phi_a)$. Dalla (1.2.5) si deduce che esiste una biiezione tra il generico vettore di stato di un sistema a due livelli e i punti della sfera unitaria tridimensionale, chiamata *sfera di Bloch* nel contesto della computazione quantistica. I parametri θ e ϕ costituiscono un sistema di coordinate polari sferiche e rappresentano rispettivamente la coordinata latitudinale e longitudinale su tale sfera, detta di Bloch.

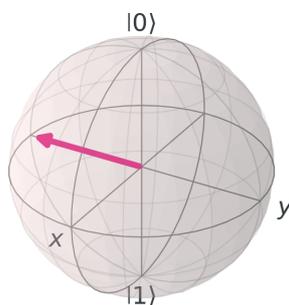


Figura 1.1: Rappresentazione di un generico stato del qubit sulla sfera di Bloch, disegnato con `qiskit`. Nell'immagine sono mostrati anche gli stati della base computazionale, coincidenti con i poli della sfera.

1.3 Qubit multipli ed entanglement

In generale, dato un sistema fisico costituito da due o più sottosistemi distinti, vale il seguente postulato per la costruzione dello stato:

Postulato 2 *Lo spazio degli stati di un sistema fisico composto è il prodotto tensoriale dello spazio degli stati dei sistemi costituenti. Inoltre, numerando i sistemi da 1 a n e preparando ogni sistema nello stato $|\psi_i\rangle$, lo stato del sistema totale è $|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_n\rangle$.*

Questo fatto permette di definire la base computazionale associata ad un insieme di N qubit, e di conseguenza il vettore di stato. Dato un registro di n qubit, la base computazionale è costruita dai vettori

$$|\phi_1 \dots \phi_n\rangle = |\phi_1\rangle \otimes \dots \otimes |\phi_n\rangle \quad (1.3.1)$$

dove $\phi \in \{0, 1\}$ per ogni $i = 1 \dots n$.

Il vettore di stato è quindi definito da 2^n coefficienti complessi $\alpha_j \in \mathbb{C}$, tali che $\sum_{j=1}^{2^n} |\alpha_j|^2 =$

1.

Il postulato 2 introduce anche il concetto di **entanglement**: un qualunque stato di un sistema composto che non possa essere ottenuto come prodotto tensoriale dei suoi sottosistemi è detto stato **entangled**. Un esempio è lo stato di *Bell*:

$$|\Psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad (1.3.2)$$

nel quale non esistono qubit singoli $|\psi_1\rangle, |\psi_2\rangle$ tali che $|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$.

1.4 Modello a circuito della computazione quantistica

Nella computazione classica, un circuito è rappresentato tramite *fili e porte logiche*, dove i primi rappresentano conduttori fisici, il cui voltaggio definisce lo stato del bit. Un circuito quantistico utilizza la stessa notazione (fig. 1.2) interpretando i fili con stati di qubit, e le porte logiche con **gate quantistici**: operatori unitari che manipolano lo stato dei qubit in maniera reversibile, a differenza del caso classico. I gate quantistici sono tipicamente rappresentati come scatole etichettate dal nome dell'operatore associato, con un certo numero di qubit di input e di output.

Infine, i circuiti quantistici introducono gli operatori di misurazione, indicati col simbolo di un metro digitale, la cui azione sfrutta il principio di collasso della funzione d'onda per fornire un risultato memorizzabile in un bit classico, rappresentato tramite un doppio filo. In appendice A sono riportati tutti i simboli utilizzati per la discussione seguente

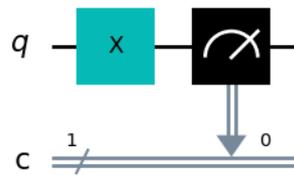


Figura 1.2: Esempio di circuito quantistico. In figura sono mostrati i simboli utilizzati per la rappresentazione di un qubit (filo singolo), del gate quantistico \hat{X} agente sul qubit (scatola etichettata con il nome del gate), operatore di misurazione (metro digitale), e bit classico (filo doppio). Il circuito applica il gate quantistico \hat{X} al qubit q , opera una misurazione sullo stato e memorizza il valore ottenuto sul bit classico c .

1.4.1 Porte quantistiche a singolo qubit

Le più semplici porte logiche costruibili sono quelle agenti su un singolo qubit, per il quale la bidimensionalità dello spazio di Hilbert \mathcal{H} implica che l'algebra degli operatori sia quadridimensionale. Ogni operatore è quindi rappresentabile con una matrice hermitiana 2×2 .

Dato un operatore \hat{U} l'unico vincolo imposto perché sia utilizzabile come gate quantistico è quello di conservare la norma del vettore di stato sul quale agisce, dal quale si deriva la nota proprietà di unitarietà:

$$\hat{U}\hat{U}^\dagger = \hat{I} \quad \Leftrightarrow \quad \hat{U}^{-1} = \hat{U}^\dagger \quad (1.4.1)$$

Dalla (1.4.1) si evince come il gate \hat{U} abbia sempre un inverso $\hat{U}^{-1} = \hat{U}^\dagger$ che conferisce alle porte logiche la proprietà di *invertibilità* non presente nel caso classico.

Tra le principali porte logiche a singolo qubit si trovano le matrici di Pauli, la cui rappresentazione nella base computazionale è :

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (1.4.2)$$

che risultano particolarmente utili per rappresentare rotazioni del vettore di stato nella sfera di Bloch. Infatti, calcolando l'esponenziale delle matrici di Pauli e sfruttando la relazione di Eulero si ottengono gli operatori:

$$\begin{aligned} \hat{R}_x(\theta) &= \exp(-i\hat{X}\theta/2) = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \\ \hat{R}_y(\theta) &= \exp(-i\hat{Y}\theta/2) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \\ \hat{R}_z(\theta) &= \exp(-i\hat{Z}\theta/2) = \begin{pmatrix} \exp(-i\theta/2) & 0 \\ 0 & \exp(i\theta/2) \end{pmatrix} \end{aligned} \quad (1.4.3)$$

che possono essere interpretati come operatori di rotazione attorno all'asse x, y , e z nella sfera di Bloch.

In generale, dato un gate a singolo qubit \hat{U} , è sempre possibile decomporlo in termini di rotazione del tipo (1.4.3), per esempio:

$$\hat{U} = \exp(i\alpha)\hat{R}_z(\beta)\hat{R}_y(\gamma)\hat{R}_z(\delta) \quad (1.4.4)$$

dove $\alpha, \beta, \gamma, \delta \in \mathbb{R}$.

Un'altra porta diffusamente utilizzata è la porta di Hadamard \hat{H} :

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (1.4.5)$$

utile per creare gli stati di sovrapposizione:

$$\begin{aligned} |+\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \hat{H} |0\rangle \\ |-\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \hat{H} |1\rangle \end{aligned} \quad (1.4.6)$$

e può essere interpretata come una rotazione di $\frac{\pi}{2}$ attorno all'asse y , seguita da una rotazione di π attorno all'asse x .

1.4.2 Porte quantistiche a qubit multipli

Dati N qubit $|\phi_1\rangle \dots |\phi_N\rangle$, il postulato 2 fornisce la struttura dello spazio degli stati in termini di prodotto tensoriale dei singoli spazi:

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \dots \otimes \mathcal{H}_N \quad (1.4.7)$$

Considerando un insieme di operatori a singolo qubit $\hat{A}_1, \dots, \hat{A}_N$ agenti rispettivamente su $|\phi_1\rangle \dots |\phi_N\rangle$ si definisce l'operatore lineare $\hat{U} = \hat{A}_1 \otimes \dots \otimes \hat{A}_N$ agente su $|\phi_1\rangle \otimes \dots \otimes |\phi_N\rangle$ come:

$$\begin{aligned} \hat{U}(|\phi_1\rangle \otimes \dots \otimes |\phi_N\rangle) &= (\hat{A}_1 \otimes \dots \otimes \hat{A}_N)(|\phi_1\rangle \otimes \dots \otimes |\phi_N\rangle) \\ &= \hat{A}_1 |\phi_1\rangle \otimes \dots \otimes \hat{A}_N |\phi_N\rangle \end{aligned} \quad (1.4.8)$$

La definizione di \hat{U} si estende poi a tutti gli elementi di \mathcal{H} per garantire la linearità:

$$\hat{U} \left(\sum_{i=1}^N \alpha_i |\phi_1\rangle \otimes \dots \otimes |\phi_N\rangle \right) = \sum_{i=1}^N \alpha_i \hat{U}(|\phi_1\rangle \otimes \dots \otimes |\phi_N\rangle) \quad (1.4.9)$$

dove $\alpha_i \in \mathbb{C}$.

Tramite le (1.4.8), (1.4.9) è possibile estendere il concetto di gate quantistico ad un sistema di N qubit partendo da quelli introdotti per il singolo. Ad esempio, l'azione globale degli operatori mostrati in figura (FIG...) può essere espressa nella base computazionale del sistema a due qubit sottoforma di *prodotto di Kronecker*:

$$\hat{U} = \hat{X} \otimes \hat{H} = \begin{pmatrix} 0 \cdot H & 1 \cdot H \\ 1 \cdot H & 0 \cdot H \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix} \quad (1.4.10)$$

In generale, non tutte le porte a qubit multipli possono essere espresse come prodotto; tuttavia, è sempre possibile esprimere un tale gate come combinazione lineare di prodotti di Kronecker tra due o più gate a singolo qubit (ref [3]).

1.4.3 Operazioni controllate

Analogamente al caso classico, le operazioni condizionate del tipo **if - then** sono uno dei fondamenti della computazione quantistica. Considerando un gate a singolo qubit \hat{U} , si definisce formalmente l'operatore \hat{U} -controllato $C\hat{U}$ come un operatore a due qubit nella forma:

$$C\hat{U} = |0\rangle\langle 0| \otimes \hat{I} + |1\rangle\langle 1| \otimes \hat{U} \quad (1.4.11)$$

Sviluppando il prodotto di Kronecker si ottiene la rappresentazione:

$$C\hat{U} = \begin{pmatrix} 1 \cdot \hat{I} & 0 \\ 0 & \hat{U} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \langle 0|\hat{U}|0\rangle & \langle 0|\hat{U}|1\rangle \\ 0 & 0 & \langle 1|\hat{U}|0\rangle & \langle 1|\hat{U}|1\rangle \end{pmatrix} \quad (1.4.12)$$

Dalla (1.4.12) risulta evidente come l'operazione controllata $C\hat{U}$ applica il gate \hat{U} al secondo qubit quando il primo è nello stato $|1\rangle$, e coincide con l'identità quando questo è nello stato $|0\rangle$. Il primo qubit è detto **qubit di controllo**, mentre il secondo è detto **qubit obiettivo**.

L'operatore controllato più comune è il *CNOT*, detto anche $C\hat{X}$:

$$C\hat{X} = |0\rangle\langle 0| \otimes \hat{I} + |1\rangle\langle 1| \otimes \hat{X} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (1.4.13)$$

La sua azione consiste nell'invertire lo stato del qubit obiettivo, quando il qubit di controllo è attivo e si può notare come questa operazione sia una generalizzazione del gate classico *XOR*. Infatti è evidente che dati i due qubit di input $|\phi_1\rangle, |\phi_2\rangle$:

$$C\hat{U} |\phi_1, \phi_2\rangle = |\phi_1, \phi_2 \oplus \phi_1\rangle \quad (1.4.14)$$

Più in generale, data una porta logica \hat{U} a k qubit $|\phi_1 \dots \phi_k\rangle$, si definisce l'operazione di controllo $C^n(\hat{U})$ tramite la relazione:

$$C^n(\hat{U}) |x_1 x_2 \dots x_n\rangle |\phi_1 \dots \phi_k\rangle = |x_1 x_2 \dots x_n\rangle U^{x_1 x_2 \dots x_n} |\phi_1 \dots \phi_k\rangle \quad (1.4.15)$$

dove il termine $x_1 x_2 \dots x_n$ all'esponente di U indica il prodotto booleano dei bit x_1, x_2, \dots, x_n . La (1.4.15) implica che l'operazione $C^n(\hat{U})$ sia una porta logica a $k+n$ qubit che applica l'operatore \hat{U} al vettore $|\phi_1 \dots \phi_k\rangle$ costituito dagli ultimi k qubit, se i primi n qubit sono nello stato $|1\rangle$; altrimenti il vettore rimane imperturbato.

1.5 Misurazione quantistica

Come discusso nella sezione 1.2 una misura dello stato $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ di un qubit fa collassare la funzione d'onda nello stato $|0\rangle$ con probabilità $|\alpha|^2$, oppure nello stato $|1\rangle$ con probabilità $|\beta|^2$. Si può descrivere formalmente questo processo, detto *misurazione nella base computazionale*, tramite i due operatori hermitiani **di proiezione**:

$$\hat{M}_0 = |0\rangle\langle 0|, \quad \hat{M}_1 = |1\rangle\langle 1| \quad (1.5.1)$$

che rispettano la proprietà di idempotenza $\hat{M}_0^2 = \hat{M}_0$, $\hat{M}_1^2 = \hat{M}_1$ dalla quale si deduce:

$$\hat{M}_0^\dagger \hat{M}_0 + \hat{M}_1^\dagger \hat{M}_1 = \hat{M}_0 + \hat{M}_1 = \hat{1} \quad (1.5.2)$$

Gli operatori (1.5.1) permettono di scrivere la probabilità dei risultati della misura sullo stato $|\psi\rangle$ come:

$$\begin{aligned} P(0) &= \langle \psi | \hat{M}_0^\dagger \hat{M}_0 | \psi \rangle = \langle \psi | \hat{M}_0 | \psi \rangle = |\alpha|^2 \\ P(1) &= \langle \psi | \hat{M}_1^\dagger \hat{M}_1 | \psi \rangle = \langle \psi | \hat{M}_1 | \psi \rangle = |\beta|^2 \end{aligned} \quad (1.5.3)$$

Più in generale, il processo di misurazione quantistica è descritto dal seguente postulato:

Postulato 3 *Il processo di misurazione quantistica di un sistema è descritto da un insieme di operatori di misura $\{\hat{M}_i\}$ agenti nello spazio del sistema, dove i etichetta il risultato della misura, che rispettano la proprietà di completezza:*

$$\sum_i \hat{M}_i^\dagger \hat{M}_i = \hat{I} \quad (1.5.4)$$

Dato uno stato $|\psi\rangle$, la probabilità che si ottenga il risultato i è data da:

$$p(i) = \langle \psi | \hat{M}_i^\dagger \hat{M}_i | \psi \rangle \quad (1.5.5)$$

e la funzione d'onda collassa nello stato:

$$\frac{\hat{M}_m |\psi\rangle}{\sqrt{\langle \psi | \hat{M}_m^\dagger \hat{M}_m | \psi \rangle}} \quad (1.5.6)$$

1.5.1 Misurazione degli operatori di Pauli

L'operazione di misura in un circuito quantistico consiste tipicamente nella misurazione dello stato nella base computazionale tramite operatori di proiezione del tipo (1.5.1). È evidente dalla definizione (1.4.2) che una proiezione sulla base computazionale restituisca sempre un autostato dell'operatore \hat{Z} , con una data probabilità. Da ciò consegue che

tramite misure nella base computazionale sia possibile ottenere il valore di aspettazione di \hat{Z} su qualsiasi stato $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$:

$$\langle\phi|\hat{Z}|\phi\rangle = |\alpha|^2 - |\beta|^2 \quad (1.5.7)$$

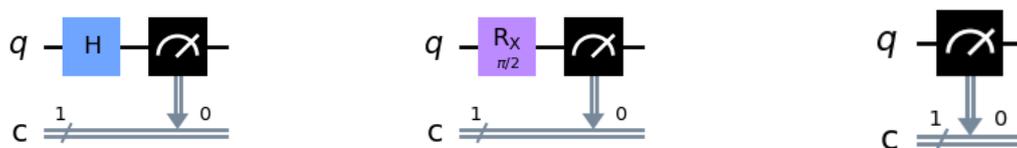
Tuttavia, come mostrato in seguito, per alcuni problemi è necessario accedere al valore di aspettazione di altri operatori di Pauli, ottenibili effettuando misure in basi diverse da quella computazionale. Tale processo può essere eseguito applicando una trasformazione unitaria \hat{T} che mappi la base di interesse nella base computazionale e successivamente eseguendo una misura proiettiva.

Come esempio, si consideri il problema di calcolare il valore di aspettazione dell'operatore di Pauli \hat{X} su un generico stato $|\phi\rangle$. È immediato verificare che gli autovalori dell'operatore siano $\{+1, -1\}$ con autostati dati rispettivamente dagli stati (1.4.6). Da ciò consegue che la trasformazione unitaria \hat{T} per mappare la base di autostati di \hat{X} nella base computazionale coincida con l'operatore di Hadamard (1.4.5); la ripetizione di misure su stati risultanti dall'applicazione del gate \hat{H} sullo stato $|\phi\rangle$ può dunque essere utilizzata per determinare il valore di aspettazione dell'operatore \hat{X} .

Con un ragionamento analogo è possibile ottenere le trasformazioni unitarie per effettuare misure nella base di autostati dell'operatore di Pauli \hat{Y} .

I circuiti mostrati in figura 1.3 implementano una possibile configurazione per la misurazione nella base di autostati dei tre operatori di Pauli. Tali configurazioni non sono univoche e possono essere modificate in funzione delle necessità del problema e dalla disponibilità dell'hardware.

Il processo discusso in questo paragrafo può essere esteso ad un qualunque operatore agente a singolo qubit, oppure a qubit multipli.



(a) Misurazione nella base di \hat{X} (b) Misurazione nella base di \hat{Y} (c) Misurazione nella base di \hat{Z}

Figura 1.3: Circuiti per effettuare la misurazione nella base degli autostati degli operatori di Pauli, disegnati con Qiskit. Il circuito 1.3a utilizza un gate \hat{H} come trasformazione unitaria. Il circuito 1.3b utilizza una rotazione attorno all'asse x di $\pi/2$ nella forma (1.4.3). Il circuito 1.3c non utilizza nessuna trasformazione poiché la base computazionale coincide con la base di autostati di \hat{Z}

1.6 Stringhe di Pauli

Una stringa di Pauli è un operatore \hat{P} a qubit multipli definito dal prodotto tensoriale di operatori di Pauli a singolo qubit (ref. [4]):

$$\hat{P} = \bigotimes_j \sigma_j \quad (1.6.1)$$

dove $\sigma_j \in \{\hat{X}, \hat{Y}, \hat{Z}, \hat{I}\}$ è un operatore di Pauli agente sul j -esimo qubit.

L'utilità di tali operatori risiede nel fatto che qualsiasi osservabile \hat{O} possa essere decomposta in stringhe di Pauli (ref. [5]):

$$\hat{O} = \sum_{\alpha} w_{\alpha} \hat{P}_{\alpha} \quad (1.6.2)$$

dove $w_{\alpha} \in \mathbb{R}$.

È immediato calcolare il valore di aspettazione dell'osservabile \hat{O} :

$$\langle \hat{O} \rangle = \sum_{\alpha} w_{\alpha} \langle \hat{P}_{\alpha} \rangle \quad (1.6.3)$$

dove ogni valore di aspettazione $\langle \hat{P}_{\alpha} \rangle$ può essere ottenuto estendendo il procedimento descritto nel paragrafo 1.5.1 ad ogni operatore della stringa \hat{P}_{α} .

Le stringhe di Pauli rappresentano dunque un utile metodo per la valutazione su hardware quantistico di una qualsiasi osservabile; nel seguito viene discusso come tali operatori vengano utilizzati per la rappresentazione dell'Hamiltoniana molecolare.

Capitolo 2

Variational Quantum Eigensolver

In questo capitolo si introducono gli aspetti formali dell'algorithmo VQE e dei metodi utilizzati per la simulazione di una molecola. La discussione seguente ha come riferimento gli articoli [6, 7, 8, 9].

2.1 Descrizione formale del VQE

Il VQE si basa sul metodo variazionale di Rayleigh-Ritz (ref. [10]), il quale permette di determinare una stima superiore dell'energia dello stato fondamentale di una data Hamiltoniana \hat{H} . In particolare, data \hat{H} e un generico stato $|\psi\rangle$, l'energia dello stato fondamentale E_0 è sempre limitata superiormente dal valore di aspettazione di \hat{H} su $|\psi\rangle$:

$$E_0 \leq \frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle} \quad (2.1.1)$$

L'algorithmo parte da uno stato di prova parametrizzato $|\psi\rangle$, o *ansatz*, e ha l'obiettivo di minimizzare il valore $\frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle}$, ottimizzando iterativamente i parametri, idealmente raggiungendo il valore E_0 .

Per l'implementazione su hardware quantistico è necessario realizzare un circuito che permetta la rappresentazione dell'ansatz tramite qubit, tale da permettere la misurazione del valore medio della Hamiltoniana. Per farlo si introduce un insieme di porte logiche parametrizzate la cui azione globale è espressa da un generico operatore $\hat{U}(\vec{\theta})$, detto *forma variazionale*, con $\vec{\theta} = (\theta_0, \theta_1, \dots, \theta_K)$, dove per ogni $i = 1, \dots, K$ si ha $-\pi < \theta_i < \pi$. L'ansatz è dunque rappresentato dall'applicazione dell'operatore $\hat{U}(\vec{\theta})$ su un registro di N qubit $|\psi_1 \dots \psi_N\rangle$, inizializzato per rappresentare uno stato di prova iniziale (ad esempio lo stato di Hartree-Fock $|\psi_{HF}\rangle$) (ref. [10]).

Il problema si riduce dunque alla ricerca del minimo della funzione obiettivo:

$$E(\vec{\theta}) = \langle \psi(\vec{\theta}) | \hat{H} | \psi(\vec{\theta}) \rangle \quad (2.1.2)$$

dove $|\psi(\vec{\theta})\rangle = \hat{U}(\vec{\theta}) |\psi_1 \dots \psi_N\rangle$.

Per poter determinare il valore di aspettazione della Hamiltoniana, questa deve essere espressa in una forma misurabile su computer quantistico; sfruttando il formalismo di seconda quantizzazione (Sez. 2.2) e la trasformazione di Jordan-Wigner (Sez. 2.3), si può espandere \hat{H} come somma di stringhe di Pauli, discusse nel paragrafo (1.6.1):

$$\hat{H} = \sum_{\alpha} w_{\alpha} \hat{P}_{\alpha} = \sum_{\alpha} w_{\alpha} (\bigotimes_j \hat{\sigma}_j)_{\alpha} \quad (2.1.3)$$

dove $w_{\alpha} \in \mathbb{R}$ e $\hat{\sigma}_j$ operatore di Pauli agente sul j -esimo qubit. Mediante la (2.1.3) si può riscrivere la (2.1.2) come:

$$E(\vec{\theta}) = \langle \psi(\vec{\theta}) | \sum_{\alpha} w_{\alpha} \hat{P}_{\alpha} | \psi(\vec{\theta}) \rangle = \sum_{\alpha} w_{\alpha} \langle \psi(\vec{\theta}) | \hat{P}_{\alpha} | \psi(\vec{\theta}) \rangle \quad (2.1.4)$$

che rende evidente la natura ibrida del VQE: il valore di aspettazione delle stringhe di Pauli $\langle \hat{P}_{\alpha} \rangle_{\vec{\theta}} = \langle \psi(\vec{\theta}) | \hat{P}_{\alpha} | \psi(\vec{\theta}) \rangle$ può essere misurato con un circuito quantistico come descritto nel paragrafo 1.6, mentre la sommatoria e la variazione dei parametri vengono eseguite su hardware classico.

2.2 Rappresentazione dell'Hamiltoniana

Il sistema da simulare è tipicamente una molecola a più elettroni interagenti tra loro. e un metodo per la sua rappresentazione è quello *ab initio*, nel quale la Hamiltoniana è costruita partendo dalla configurazione atomica e dalla posizione relativa tra i nuclei. Trascurando gli effetti relativistici e utilizzando l'approssimazione di Born-Oppenheimer, l'Hamiltoniana per gli elettroni è espressa come:

$$\hat{H} = -\frac{1}{2} \sum_i \frac{\hbar^2}{m_e} \nabla_i^2 - \sum_{i,j} \frac{e^2}{4\pi\epsilon_0} \frac{Z_k}{|\hat{\mathbf{r}}_i - \hat{\mathbf{R}}_j|} + \frac{1}{2} \sum_{i \neq j} \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j|} \quad (2.2.1)$$

con $\hat{\mathbf{R}}_j$ operatore posizione del j -esimo nucleo, $\hat{\mathbf{r}}_i$ operatore posizione dell' i -esimo elettrone, m_e ed e massa e carica dell'elettrone, \hbar costante di Planck ridotta, ϵ_0 costante dielettrica del vuoto.

Evidentemente, per eseguire la simulazione su un computer quantistico è necessario approssimare \hat{H} con una Hamiltoniana \hat{H}_Q per N qubit interagenti; in altre parole, è necessario trasformare lo spazio di Hilbert \mathcal{H} infinito dimensionale in uno spazio \mathcal{H}_Q finito dimensionale. Questo risultato si ottiene troncando lo spazio degli stati di un singolo elettrone, in un insieme generato da N funzioni di base $\{\phi_1, \dots, \phi_N\}$, dette orbitali di spin. Ad esempio, ogni ϕ_i può essere espressa come una combinazione lineare di Gaussiane centrate sui nuclei, ognuna con orientazione di spin fissata. Va osservato che le

funzioni così costruite differiscono dai noti orbitali atomici che possono essere occupati da due elettroni con spin opposto; in questo caso ogni orbitale è costruito partendo da uno spin fissato, sicché può essere occupato da un solo elettrone. L'utilità di questa rappresentazione sarà evidente nel passaggio al sistema di qubit.

Una volta costruita la base a singolo elettrone, le funzioni d'onda Ψ a più elettroni, chiamate **orbitali molecolari**, sono ottenute come prodotti tra queste, imponendo l'antisimmetria richiesta dal teorema spin-statistica.

La costruzione della Hamiltoniana può avvenire infine tramite due differenti formalismi: **prima quantizzazione**, nel quale si costruisce \hat{H} direttamente tramite proiezione sulle funzioni di base; **seconda quantizzazione**, dove si rappresenta \hat{H} tramite operatori di creazione e annichilazione agenti sullo spazio dei numeri di occupazione.

2.2.1 Prima quantizzazione

Nel formalismo di prima quantizzazione, dato un sistema di K elettroni, le funzioni di base Ψ si costruiscono direttamente combinando gli orbitali di spin $\{\phi_1 \dots \phi_N\}$. Per garantire l'antisimmetria, è utile definire Ψ tramite il determinante di Slater:

$$\Psi(x_1, x_2, \dots, x_K) = \frac{1}{\sqrt{K!}} \det \begin{pmatrix} \phi_1(x_1) & \phi_1(x_2) & \dots & \phi_1(x_K) \\ \phi_2(x_1) & \phi_2(x_2) & \dots & \phi_2(x_K) \\ \vdots & & & \vdots \\ \phi_K(x_1) & \phi_K(x_2) & \dots & \phi_K(x_K) \end{pmatrix} \quad (2.2.2)$$

Lo spazio di Hilbert $\mathcal{H}_{K,N}$ generato dalle funzioni (2.2.2) è detto *spazio delle configurazioni di interazione* ed è costruito distribuendo i K elettroni negli N orbitali disponibili, sicché ha dimensione $\binom{N}{K}$ e si può identificare con il sottospazio totalmente antisimmetrico di $(\mathbb{C}^N)^{\otimes K}$.

Proiettando l'Hamiltoniana elettronica sullo spazio $\mathcal{H}_{K,N}$ si ottiene l'espressione:

$$\hat{H} = \sum_{i=1}^K \sum_{p,q=1}^N h_{pq} |\phi_p^{(i)}\rangle \langle \phi_q^{(i)}| + \frac{1}{2} \sum_{i=1, i \neq j}^K \sum_{p,q,r,s=1}^N h_{pqrs} |\phi_p^{(i)} \phi_q^{(j)}\rangle \langle \phi_r^{(i)} \phi_s^{(j)}| \quad (2.2.3)$$

dove

$$\begin{aligned} h_{pq} &= \left\langle \phi_p \left| -\frac{1}{2} \sum_i \frac{\hbar^2}{m_e} \hat{\nabla}^2 - \sum_{i,j} \frac{e^2}{4\pi\epsilon_0} \frac{Z_k}{|\hat{\mathbf{r}}_i - \hat{\mathbf{R}}_j|} \right| \phi_q \right\rangle \\ h_{pqrs} &= \left\langle \phi_p \phi_q \left| \frac{1}{2} \sum_{i \neq j} \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j|} \right| \phi_r \phi_s \right\rangle \end{aligned} \quad (2.2.4)$$

sono gli integrali di interazione elettronica ad uno e due elettroni.

La (2.2.3) è l'Hamiltoniana di prima quantizzazione e può essere mappata nella Hamiltoniana \hat{H}_Q utilizzando $K \log_2 N$ qubit.

2.2.2 Seconda quantizzazione

Nel formalismo di seconda quantizzazione l'antisimmetria degli stati è garantita dalla struttura degli operatori, piuttosto che dalle funzioni d'onda. Questa caratteristica si riflette sulle proprietà degli operatori utilizzati per costruire l'Hamiltoniana.

Data una base di orbitali di spin $\{\phi_1 \dots \phi_N\}$, per ogni orbitale ϕ_j si introducono gli operatori di **creazione fermionica** \hat{a}_j^\dagger e **annichilazione fermionica** \hat{a}_j che rispettano le seguenti proprietà di anticommutazione:

$$\{\hat{a}_j, \hat{a}_j^\dagger\} = \delta_{ij}, \quad \{\hat{a}_j^\dagger, \hat{a}_i^\dagger\} = \{\hat{a}_j, \hat{a}_i\} = 0 \quad (2.2.5)$$

o analogamente:

$$[\hat{a}_i, \hat{a}_j] = -2\hat{a}_j\hat{a}_i, \quad [\hat{a}_i, \hat{a}_j^\dagger] = \delta_{ij} - 2\hat{a}_j\hat{a}_i \quad (2.2.6)$$

Introducendo la base dei numeri di occupazione $\{n_1 \dots n_N\}$, dove per ogni $i = 1 \dots N$ si ha $n_i = 0$ se l'orbitale ϕ_i non è occupato e $n_i = 1$ se è occupato, le funzioni di base della molecola si esprimono come $|\Psi\rangle = |n_1 \dots n_N\rangle = (\hat{a}_1)^{n_1} \dots (\hat{a}_N)^{n_N} |0\rangle$.

Con queste definizioni si può scrivere l'azione degli operatori $\hat{a}_j, \hat{a}_j^\dagger$ sugli stati $|\Psi\rangle$ come:

$$\hat{a}_j^\dagger |\Psi\rangle = \hat{a}_j^\dagger |n_1 n_2 \dots n_N\rangle = \begin{cases} 0, & n_j = 1 \\ p_j |n_1 n_2 \dots 1_j \dots n_N\rangle & n_j = 0 \end{cases} \quad (2.2.7)$$

$$\hat{a}_j |\Psi\rangle = \hat{a}_j |n_1 n_2 \dots n_N\rangle = \begin{cases} 0, & n_j = 0 \\ p_j |n_1 n_2 \dots 0_j \dots n_N\rangle & n_j = 1 \end{cases}$$

dove $p_j = (-1)^{\sum_{i=1}^{j-1} n_i}$ rappresenta la parità del j -esimo orbitale: positiva quando ci sono un numero pari di orbitali occupati tra ϕ_1 e ϕ_{j-1} , negativa se sono in numero dispari.

Gli operatori fermionici consentono di riscrivere l'Hamiltoniana di prima quantizzazione (2.2.3) come:

$$\hat{H} = \sum_{p,q=1}^N h_{pq} \hat{a}_p \hat{a}_q + \frac{1}{2} \sum_{p,q,r,s=1}^N h_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s \quad (2.2.8)$$

Nonostante i due formalismi siano equivalenti, la seconda quantizzazione è attualmente la più utilizzata per la simulazione di sistemi a molti elettroni su computer quantistici, in quanto fornisce una trattazione più semplice e naturale del problema. Nel seguito si considera solo il caso di Hamiltoniane nel formalismo di seconda quantizzazione.

2.3 Trasformazione di Jordan-Wigner

Come discusso nella sezione 2.2.2, il formalismo di seconda quantizzazione è preferito in quanto garantisce un minor numero di qubit necessari per rappresentare l'Hamiltoniana.

Tuttavia, poiché i qubit sono sistemi a spin $\frac{1}{2}$, gli unici operatori direttamente misurabili su un computer quantistico sono quelli agenti su uno spazio di Hilbert bidimensionale. Ciò implica la necessità di trasformare l'Hamiltoniana (2.2.8) in una combinazione lineare di operatori di spin, tipicamente costruiti dagli operatori di Pauli. Inoltre, gli operatori di Pauli non rispettano le regole di anticommutazione (2.2.5) imposte dalla seconda quantizzazione, sicché è necessario definire operatori ad hoc con suddette proprietà.

La tecnica più utilizzata per questa costruzione è la trasformazione di Jordan-Wigner, nella quale si crea una mappa diretta tra i numeri di occupazione $|n_1 \dots n_N\rangle$ di N orbitali di spin in N qubit. Lo stato $|0\rangle_j$ corrisponde all'orbitale ϕ_j non occupato, mentre $|1\rangle_j$ corrisponde a ϕ_j occupato.

Considerando il j -esimo qubit, si possono associare gli operatori $\hat{a}_j^\dagger, \hat{a}_j$ ad operatori di Pauli, sfruttando le (2.2.7). L'azione degli operatori fermionici diventa

$$\begin{aligned}\hat{a}_j^\dagger &\Leftrightarrow |1\rangle_j \langle 0|_j \equiv \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}_j = \frac{\hat{X}_j - i\hat{Y}_j}{2} \\ \hat{a}_j &\Leftrightarrow |0\rangle_j \langle 1|_j \equiv \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}_j = \frac{\hat{X}_j + i\hat{Y}_j}{2}\end{aligned}\tag{2.3.1}$$

dove $|0\rangle_j, |1\rangle_j$ indicano gli stati del qubit e \hat{X}_j, \hat{Y}_j operatori di Pauli agenti sul j -esimo qubit¹.

Gli operatori (2.3.1) non rispettano le proprietà di anticommutazione (2.2.5). Infatti, per ogni $i, j = 1 \dots N$ si ha che $\hat{X}_i \hat{X}_j = \hat{X}_j \hat{X}_i$ e $\hat{Y}_i \hat{Y}_j = \hat{Y}_j \hat{Y}_i$, da cui: $[\hat{X}_i + i\hat{Y}_i, \hat{X}_j + \hat{Y}_j] = [\hat{a}_i, \hat{a}_j] = 0$.

Per garantire l'antisimmetria si introduce una stringa di operatori $\hat{Z}, \hat{Z}_0 \otimes \hat{Z}_1 \otimes \dots \otimes \hat{Z}_{j-1}$ agente sui primi $j - 1$ qubit:

$$\begin{aligned}\hat{a}_j^\dagger &\Leftrightarrow \hat{Z}_0 \otimes \dots \otimes \hat{Z}_{j-2} \otimes \hat{Z}_{j-1} \otimes \frac{\hat{X}_j - i\hat{Y}_j}{2} \\ \hat{a}_j &\Leftrightarrow \hat{Z}_0 \otimes \dots \otimes \hat{Z}_{j-2} \otimes \hat{Z}_{j-1} \otimes \frac{\hat{X}_j + i\hat{Y}_j}{2}\end{aligned}\tag{2.3.2}$$

Si osserva infatti che l'operatore $\hat{Z}_{j-1} \otimes \hat{Z}_{j-2} \otimes \dots \otimes \hat{Z}_0$ ha autovalore 1 per un numero pari di qubit con stato $|1\rangle$ e -1 per un numero dispari, garantendo la struttura di parità richiesta.

¹Formalmente, nella (2.3.1) andrebbe specificata l'azione su ogni qubit del registro. In particolare, poiché gli operatori $\hat{a}_j^\dagger, \hat{a}_j$ modificano solamente lo stato del qubit j -esimo, l'azione globale sul registro è data da: $\hat{a}_j^\dagger \Leftrightarrow \hat{I}_0 \otimes \dots \otimes \frac{\hat{X}_j - i\hat{Y}_j}{2} \otimes \dots \otimes \hat{I}_N$, $\hat{a}_j \Leftrightarrow \hat{I}_0 \otimes \dots \otimes \frac{\hat{X}_j + i\hat{Y}_j}{2} \otimes \dots \otimes \hat{I}_N$. Nel seguito si omettono tutti gli operatori identità, sottintendendo che lo stato dei qubit su cui agiscono resta imperturbato.

2.3.1 Trasformazione di parità

Per la simulazione di sistemi che conservano il numero di particelle è possibile utilizzare un altro tipo di trasformazione per costruire le stringhe di Pauli, detta trasformazione di parità. In particolare, sfruttando tale mappatura è possibile ridurre il numero di qubit necessari per la rappresentazione.

La trasformazione consiste nel mappare la parità degli N orbitali $\{\phi_1 \dots \phi_N\}$, discussa nel paragrafo 2.2.2 negli N qubit. Il j -esimo qubit ha stato $|0\rangle_j$ se il numero di orbitali occupati fino al j -esimo, incluso, è pari, mentre ha stato $|1\rangle_j$ se è dispari.

Dato uno stato con numeri di occupazione $|n_1 \dots n_N\rangle$ si può costruire un registro di N qubit $|q_1 \dots q_N\rangle$ dove:

$$q_j = \sum_{i \leq j} n_i \pmod{2} \quad (2.3.3)$$

L'azione degli operatori fermionici sul singolo qubit j -esimo diventa²:

$$\begin{aligned} \hat{a}_j^\dagger &\Leftrightarrow |01\rangle_{j,j+1} \langle 00|_{j,j+1} - |10\rangle_{j,j+1} \langle 11|_{j,j+1} \equiv \hat{Z}_j \otimes \left(\frac{\hat{X}_{j+1} - i\hat{Y}_{j+1}}{2} \right) \\ \hat{a}_j &\Leftrightarrow |00\rangle_{j,j+1} \langle 01|_{j,j+1} - |11\rangle_{j,j+1} \langle 10|_{j,j+1} \equiv \hat{Z}_j \otimes \left(\frac{\hat{X}_{j+1} + i\hat{Y}_{j+1}}{2} \right) \end{aligned} \quad (2.3.4)$$

D'altra parte, poiché l'effetto di aggiungere o rimuovere un elettrone nell'orbitale ϕ_j modifica la parità di tutti gli orbitali successivi, è necessario invertire lo stato dei qubit che li codificano. Tale effetto si può ottenere tramite l'azione di una stringa di operatori \hat{X} agenti sui qubit $|q_{j+2} \dots q_N\rangle$. La mappatura è quindi data dalla seguente trasformazione:

$$\begin{aligned} \hat{a}_j^\dagger &\Leftrightarrow \hat{Z}_j \otimes \left(\frac{\hat{X}_{j+1} - i\hat{Y}_{j+1}}{2} \right) \otimes \hat{X}_{j+2} \otimes \hat{X}_{j+3} \otimes \dots \otimes \hat{X}_N \\ \hat{a}_j &\Leftrightarrow \hat{Z}_j \otimes \left(\frac{\hat{X}_{j+1} + i\hat{Y}_{j+1}}{2} \right) \otimes \hat{X}_{j+2} \otimes \hat{X}_{j+3} \otimes \dots \otimes \hat{X}_N \end{aligned} \quad (2.3.5)$$

Dalla definizione (2.3.3) segue che, se due qubit successivi $|q\rangle_j, |q\rangle_{j+1}$ sono nello stesso stato, l'orbitale ϕ_{j+1} non è occupato, mentre se sono in due stati diversi l'orbitale ϕ_{j+1} è occupato. In particolare, considerando sistemi che conservano il numero di particelle, è evidente che il qubit $|q\rangle_N$ ha sempre stato $|M \pmod{2}\rangle_N$, dove M è il numero di particelle del sistema. Questa proprietà permette di conoscere a priori lo stato di tale qubit, note le condizioni iniziali del problema, sicchè è possibile eliminarlo (*qubit tapering*).

Tipicamente, oltre al numero di particelle è noto anche lo spin totale, per cui è possibile mappare separatamente gli orbitali di spin up e down, garantendo la possibilità di eliminare due qubit seguendo il ragionamento precedente.

²Analogamente al caso della (2.3.1) non sono riportati gli operatori identità agenti sui qubit imperturbati.

2.4 Ansatz

Uno step fondamentale dell'intero algoritmo è la costruzione di un ansatz appropriato per il sistema in esame. Infatti, una scelta accurata è necessaria per ottenere un valore dell'energia effettivamente vicino a quello reale. Dato un registro iniziale di N qubit è necessario costruire un circuito quantistico parametrizzato che permetta un'efficiente misurazione del valore di aspettazione dell'Hamiltoniana bilanciando due aspetti fondamentali:

- *Espressibilità*: dato uno stato iniziale $|\Psi(\vec{\theta})\rangle$ è necessario che lo spazio di Hilbert accessibile variando i parametri sia sufficientemente grande da contenere la soluzione;
- *Trattabilità*: la soluzione deve essere valutabile in un tempo trattabile.

Evidentemente, ansatz con alta espressibilità portano a tempi di calcolo non trattabili. D'altra parte, scegliendo strutture trattabili, ad esempio polinomiali, è impossibile generare l'intero spazio di Hilbert.

Un approccio possibile per la scelta dell'ansatz è detto *a priori* e si basa sulla conoscenza del sistema da simulare per imporre vincoli sugli stati accessibili e sulla struttura degli operatori utilizzabili.

2.4.1 Ansatz *a priori*

In linea di principio, qualsiasi operatore unitario può essere utilizzato per manipolare i qubit iniziali; tuttavia, i sistemi simulati sono tipicamente molecole con numero di particelle fissato, permettendo di definire vincoli sugli stati accessibili e sulla struttura degli operatori. Ad esempio, data una molecola con quattro orbitali di spin popolati da un elettrone, una possibile soluzione dell'equazione di Hartree-Fock per lo stato di prova è $|\Psi_{HF}\rangle = |0001\rangle$, dove l'elettrone occupa l'orbitale ad energia minima con spin $s = +\frac{1}{2}$. Imponendo la conservazione delle particelle, il generico stato ottenibile da $|\Psi_{HF}\rangle$ è:

$$|\Psi\rangle = c_1 |0001\rangle + c_2 |0010\rangle + c_3 |0100\rangle + c_4 |1000\rangle \quad (2.4.1)$$

con $c_1, c_2, c_3, c_4 \in \mathbb{C}$.

Inoltre, una qualunque porta logica applicabile allo stato dovrà conservare il numero di particelle; in altre parole dovrà essere un endomorfismo sul sottospazio di Hilbert $\mathcal{H}_{4,1}$ generato dagli stati a $N = 4$ orbitali occupati da $K = 1$ elettroni. Queste osservazioni permettono la costruzione di ansatz basati sulla struttura del problema in esame che limitano il numero di parametri necessari.

2.4.2 Rotazioni di Givens

Data una coppia di qubit, si definisce rotazione di Givens l'operatore rappresentato dalla seguente matrice nella base computazionale (ref [11]):

$$\hat{G}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta/2) & -\sin(\theta/2) & 0 \\ 0 & \sin(\theta/2) & \cos(\theta/2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.4.2)$$

Essa rappresenta una rotazione interna al sottospazio di Hilbert bidimensionale generato dagli stati $\{|01\rangle, |10\rangle\}$. Scrivendo in maniera esplicita l'azione di $\hat{G}(\theta)$ su questi stati si ottiene:

$$\begin{aligned} \hat{G}(\theta) |01\rangle &= \cos\left(\frac{\theta}{2}\right) |01\rangle - \sin\left(\frac{\theta}{2}\right) |10\rangle \\ \hat{G}(\theta) |10\rangle &= \sin\left(\frac{\theta}{2}\right) |01\rangle + \cos\left(\frac{\theta}{2}\right) |10\rangle \end{aligned} \quad (2.4.3)$$

che rappresenta la sovrapposizione di due stati ottenuti scambiando una particella tra due orbitali. Una tale trasformazione si può interpretare come una **eccitazione singola** dell'elettrone dal primo qubit al secondo, e viceversa.

In generale, su un registro di N qubit, le rotazioni \hat{G} permettono di creare stati entangled di qubit presi a due a due che differiscono per una singola eccitazione.

Un ragionamento analogo può essere applicato per generare stati che differiscono per una **doppia eccitazione**, ad esempio:

$$\begin{aligned} \hat{G}(\theta) |0011\rangle &= \cos\left(\frac{\theta}{2}\right) |0011\rangle - \sin\left(\frac{\theta}{2}\right) |1100\rangle \\ \hat{G}(\theta) |1100\rangle &= \sin\left(\frac{\theta}{2}\right) |0011\rangle + \cos\left(\frac{\theta}{2}\right) |1100\rangle \end{aligned} \quad (2.4.4)$$

L'utilità delle trasformazioni di Givens risiede nella loro versione controllata. Infatti, è dimostrato che le rotazioni di Givens a singola eccitazione controllate $CG(\hat{\theta})$ sono un insieme universale per gli operatori che conservano le particelle: qualunque ansatz può essere convenientemente costruito con tali operatori, eventualmente accoppiati a trasformazioni non controllate.

2.5 Ottimizzazione dei parametri

L'ultimo step del VQE è la ricerca del minimo della funzione obiettivo (2.1.2), mediante la variazione dei parametri $\vec{\theta}$ eseguita su computer classico. L'ottimizzazione dei parametri presenta due problemi da tenere in considerazione:

- *Rumore dell'hardware quantistico*: ogni computer quantistico presenta un rumore intrinseco che può portare a risultati molto diversi dal valore atteso;
- *Numero di misure della funzione obiettivo*: ogni algoritmo di ottimizzazione necessita di un certo numero di valutazioni della funzione obiettivo, a volte proporzionale alla cardinalità dell'insieme dei parametri $\vec{\theta}$.

Una scelta popolare ed intuitiva è quella di *discesa del gradiente*, nella quale si aggiornano i parametri in direzione della massima variazione ΔE dell'energia; tuttavia, questa tecnica presenta il rischio di determinare un minimo locale della funzione, portando a risultati inesatti, oltre che ad un alto costo algoritmico.

Nel seguito verrà utilizzato un algoritmo più appropriato per il VQE, chiamato **COBYLA** (Constrained Optimization by Linear Approximation). Tale algoritmo esegue l'ottimizzazione tramite un'approssimazione lineare richiedendo un'unica valutazione della funzione obiettivo, rendendolo indipendente dal numero di parametri (ref. [12]).

2.5.1 Numero di misurazioni nel circuito quantistico

Un altro fattore che può influenzare la performance dell'algoritmo è il numero di misurazioni che vengono eseguite ad ogni iterazione per determinare il valore di aspettazione dell'Hamiltoniana, nel seguito chiamati *shots* (ref [4]). Infatti, richiamando la (2.1.4), la determinazione del valore medio dell'energia si riduce al calcolo del valore medio delle stringhe di Pauli in cui è stata decomposta l'Hamiltoniana. Tale valore viene determinato ripetendo la misurazione sullo stato per calcolare empiricamente le probabilità di ogni autovalore. Evidentemente, un maggiore numero di shots consegue in un valore medio più accurato per ogni stringa di Pauli. Nel seguito si discuterà come cambia la performance variando il numero di shots.

Capitolo 3

Implementazione del VQE mediante Qiskit per molecole

Questo capitolo tratta l'implementazione del VQE per la simulazione delle molecole H_2 e LiH tramite SDK Qiskit (<https://qiskit.org>). Nella prima parte si introduce il framework, discutendo come permetta di configurare l'algoritmo per la simulazione locale e l'esecuzione su dispositivo reale. Successivamente, si mostrano i risultati ottenuti dalla esecuzione sia su simulatore che su dispositivo reale, utilizzato tramite cloud. Il codice descritto nelle sezioni seguenti è disponibile su GitHub al repository https://github.com/ferraridavid-hub/VQE_LiH_H2

3.1 Qiskit

Qiskit (Quantum Informatin Software Kit) è un SDK open source sviluppato in Python per la produzione di programmi quantistici eseguibili su simulatori locali o dispositivi reali implementati secondo il modello a circuito quantistico, accessibili in cloud tramite la piattaforma IBM Quantum Composer.

Il kit è in costante fase di sviluppo, e ad oggi conta numerose librerie per l'implementazione di algoritmi quantistici applicabili a diverse categorie di problemi. Per lo scopo di questa simulazione si sono utilizzati i seguenti moduli:

- **qiskit**: il modulo `qiskit` fornisce un API per la costruzione e la gestione di qubit, gate e circuiti quantistici. Il listato mostra un semplice esempio di implementazione di circuito;
- **Algorithms**: il modulo `qiskit.algorithm` contiene oggetti e funzioni per implementare diversi algoritmi quantistici; nel caso specifico di questa simulazione è stato sfruttato per la configurazione degli ottimizzatori e del simulatore.

- **Qiskit Nature:** `qiskit_nature` è un modulo indipendente che mette a disposizione librerie per la risoluzione di problemi scientifici di natura quantistica, come il calcolo degli stati eccitati elettronici e vibrazionali di una molecola, o la soluzione dei modelli a reticolo. Il framework contiene oggetti e funzioni utili per l'implementazione dei vari step del VQE.

```

1 from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
2 from matplotlib import pyplot as plt
3
4 # Build a quantum circuit
5 circuit = QuantumCircuit(1, 1)
6
7 # Place Hadamard gate
8 circuit.h(0)
9 circuit.measure(0, 0)
10
11 # Show circuit
12 circuit.draw('mpl')
13 plt.show()

```

Listing 3.1: Codice per l'implementazione del circuito 1.3a per la misurazione nella base dell'operatore \hat{X}

3.1.1 Dispositivi IBM

IBM mette a disposizione diversi simulatori e dispositivi quantistici reali, accessibili in cloud, per eseguire il codice scritto con Qiskit. Per gli scopi di questa tesi, oltre al simulatore locale, si sono utilizzati dispositivi reali che mettersero a disposizione almeno 2 qubit. Per velocizzare l'esecuzione, nel codice è presente una funzione che determina il dispositivo con il minor numero di processi in attesa e mette in coda l'esecuzione. Nel seguito verrà specificato quale dispositivo è stato utilizzato per le varie esecuzioni.

3.1.2 Configurazione dell'algoritmo

Le sezioni seguenti illustrano come sia stato utilizzato il framework Qiskit per configurare l'algoritmo ed eseguirlo in locale e su dispositivi reali.

Costruzione dell'Hamiltoniana

Il primo step dell'algoritmo è quello di costruire l'Hamiltoniana agente nello spazio dei qubit \hat{H}_Q della molecola di interesse. Come detto nel paragrafo 2.2 l'Hamiltoniana si ottiene partendo dalla configurazione relativa tra i nuclei e scegliendo una base finito-dimensionale per le funzioni d'onda degli elettroni. Successivamente, utilizzando il formalismo di seconda quantizzazione e una mappatura a qubit si ottiene l'operatore ha-

miltoniano espresso come combinazione lineare di stringhe di Pauli. Per la molecola di H_2 è stata utilizzata la trasformazione di Jordan-Wigner, mentre per la molecola di LiH è stata sfruttata la trasformazione di parità per diminuire il numero di qubit necessari. Ai fini della simulazione è stata scritta la funzione:

```
def get_qubit_hamiltonian(distance, mapper) (3.1.1)
```

la quale partendo dalla distanza in Angstrom, e dalla mappatura a qubit passate come argomento esegue la seguente procedura:

- *Costruzione del driver rappresentante la molecola:* tramite la classe `PySCFDriver` fornita da `qiskit_nature` istanzia un oggetto per la rappresentazione della molecola, utilizzando la distanza tra i nuclei passata come argomento e la base *sto3g* per gli orbitali;
- *Costruzione dell'hamiltoniana di seconda quantizzazione:* l'oggetto istanziato possiede un metodo che restituisce la rappresentazione dell'Hamiltoniana di seconda quantizzazione. Inoltre, in questa fase vengono anche calcolati i qubit necessari per la simulazione;
- *Costruzione dell'operatore agente nello spazio dei qubit:* tramite la mappatura passata come argomento la funzione restituisce l'operatore \hat{H}_Q agente sui qubit.

Inoltre, la funzione descritta restituisce anche un oggetto del tipo `ElectronicStructureProblem` generato per la costruzione dell'Hamiltoniana, contenente tutte le informazioni della molecola necessarie agli step successivi.

La configurazione descritta in precedenza necessita di 4 qubit per la molecola di H_2 , garantendo la possibilità di simulare la molecola anche su un semplice laptop. Nel caso della molecola di LiH, è stato necessario ridurre il numero di qubit per poter effettuare la simulazione. Oltre alla trasformazione di parità è stata sfruttata l'approssimazione di *freeze-core* (ref [13]), considerando solo gli elettroni più esterni per l'implementazione dell'algoritmo e considerando costante il contributo di quelli più interni. Inoltre, sono stati trascurati gli orbitali non occupati più esterni per ridurre il numero di qubit a 6.

Ansatz

Il secondo oggetto necessario alla simulazione è l'ansatz per la costruzione del circuito parametrizzato. In questo caso è stata scritta la funzione

```
get Ansatz(mol_electronic_structure, mapper) (3.1.2)
```

che prende come argomenti l'oggetto rappresentante la molecola restituito dalla funzione (3.1.1) e un oggetto per la mappatura. Tale funzione è semplicemente un wrapper per

la costruzione dell'ansatz generato dal costruttore della classe `UCCSD` fornita dal modulo `qiskit_nature`. L'istanziamento avviene passando al metodo il numero di particelle della molecola, il numero di orbitali, la funzione di mappatura, e la configurazione iniziale degli orbitali. In questo caso, si è scelta la soluzione a campo medio di Hartree-Fock per lo stato iniziale dei qubit. In figura 3.1 è mostrato il circuito parametrizzato ottenuto dalla configurazione descritta per la molecola di H_2 con una distanza di 0.735 Angstrom.

Ottimizzatori e risolutore esatto

L'ultimo elemento per l'esecuzione dell'algoritmo è l'ottimizzatore. Il modulo `qiskit_algorithms` mette a disposizione diversi ottimizzatori utili in diverse situazioni, e per diversi dispositivi.

Per la simulazione locale si sono utilizzati entrambi gli ottimizzatori SLSQP e COBYLA mettendo a confronto i risultati. Mentre le esecuzioni su dispositivo reale sono state fatte solo con ottimizzatore COBYLA.

Infine, per valutare la bontà dei risultati ottenuti dalla simulazione si è utilizzato un risolutore esatto dello stato fondamentale delle molecole in esame fornito dalla classe `NumPyMinimumEigensolver` anch'essa contenuta nel modulo `qiskit_algorithms`.

Risolutore VQE

L'esecuzione dell'algoritmo viene compiuta da un'istanza della seguente classe del modulo `qiskit_algorithms`:

$$\text{VQE}(\text{estimator}, \text{ansatz}, \text{optimizer}, *, \text{gradient}=\text{None}, \text{initial_point}=\text{None}, \text{callback}=\text{None}) \quad (3.1.3)$$

La (3.1.3) mostra i diversi parametri che possono essere utilizzati per istanziare l'oggetto risolutore, dipendenti dal problema in esame. Per questa simulazione l'unico parametro

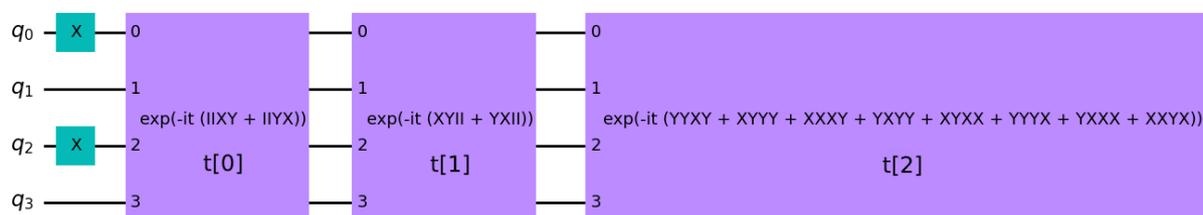


Figura 3.1: Circuito parametrizzato utilizzato come ansatz UCCSD, disegnato tramite il modulo `QuantumCircuit`. In figura i parametri variazionali sono indicati come $t[0], t[1], t[2]$.

variabile è l'oggetto `estimator`, il quale incapsula le operazioni per la valutazione dell'energia media ad ogni iterazione, ad esempio gli shots, ed è implementato diversamente nel caso di simulatore locale e dispositivo reale.

Il parametro `callback` viene sfruttato passando come argomento la funzione:

```
def store_intermediate_data(n_iter, params,
                            energy, meta) (3.1.4)
```

Tale funzione ha lo scopo di memorizzare ad ogni iterazione dell'algoritmo l'indice dell'iterazione stessa ed il valore attuale dei parametri e dell'energia. Tali valori sono necessari per la valutazione della performance del VQE.

3.2 Simulazione e risultati

In questa sezione sono mostrati i risultati ottenuti dalla simulazione del VQE per le molecole H_2 e LiH . Per entrambe le molecole, la simulazione avviene in due fasi: nella prima si esegue l'algoritmo ad una data distanza atomica analizzando il comportamento dei valori calcolati per diversi valori di shots; nella seconda si utilizza un intervallo discreto di distanze per ognuna delle quali si esegue una simulazione completa, ottenendo una distribuzione dell'energia in funzione della distanza. In entrambe le fasi si utilizza l'ottimizzatore COBYLA

Il lavoro seguente è una applicazione per le molecole H_2 e LiH di quello discusso in [4].

3.2.1 Energia dello stato fondamentale

La prima fase dell'algoritmo è stata implementata fissando la distanza tra i nuclei ai valori sperimentali di 0.735 \AA e 1.375 \AA , rispettivamente per H_2 e LiH , e chiamando la funzione (3.1.1) per costruire l'Hamiltoniana.

Successivamente si è utilizzata la funzione (3.1.2) per la costruzione dell'ansatz iniziando tutti i parametri al valore arbitrario 0. Inoltre, la chiamata alla funzione (3.1.1) permette di estrarre il valore numerico della repulsione atomica e dell'energia degli elettroni di core nel caso di LiH , non conteggiate durante le iterazioni del VQE; tali valori sono necessari nella presentazione dei risultati finali in quanto devono essere sommati ad ogni valore intermedio.

Per l'effettiva esecuzione si è istanziato un oggetto risolutore della classe (3.1.3) per eseguire la simulazione con l'ottimizzatore COBYLA con un certo numero di shots; tale istanza esegue l'algoritmo e memorizza i valori e le informazioni calcolate nell'oggetto di ritorno. Per entrambe le molecole, e per ogni valore del numero di shots, si è eseguito il VQE dieci volte calcolando la media e la deviazione standard dei valori dell'energia risultanti. I risultati sull'energia sono riassunti in fig. 3.1 per H_2 e in fig. 3.2.

Infine, si sono confrontati i risultati ottenuti dalla simulazione con il caso di un risolutore esatto discusso nel paragrafo 3.1.2.

In figura 3.2 e 3.3 è mostrato l'andamento del valore dell'energia calcolato ad ogni step dell'algoritmo, per diversi valori di shots. In entrambi i casi è evidente come il numero di shots influisca notevolmente sull'efficacia dell'ottimizzatore: un basso numero di shots causa una maggiore imprecisione nella misurazione del valore medio delle stringhe di Pauli, con conseguente andamento fluttuante dei valori dell'energia calcolati e distanza del valore finale da quello esatto. Aumentando il numero di shots, le fluttuazioni diminuiscono e il valore finale tende a convergere sempre di più verso al valore esatto.

Fidelity

Per confrontare l'autostato calcolato dall'algoritmo con quello esatto si è utilizzato il concetto di *fidelity*, che nel caso di stati quantistici si riduce al quadrato del prodotto scalare tra essi.

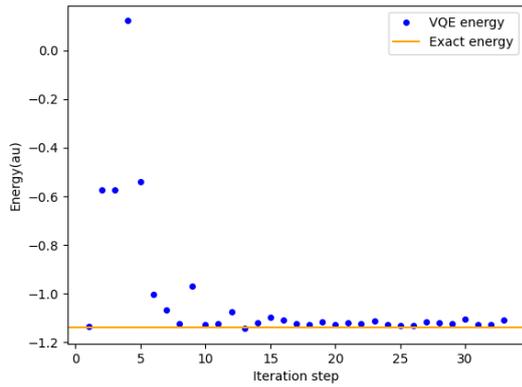
Per la seguente simulazione la *fidelity* F è definita come:

$$F = \langle \psi_0(\vec{\theta}_f) | \psi_f \rangle^2 \quad (3.2.1)$$

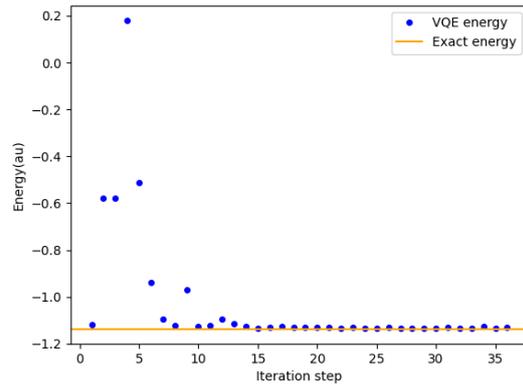
dove $\psi_0(\vec{\theta}_0)$ indica l'autostato ottenuto dal circuito parametrizzato con i parametri al valore ottimo, e ψ_0 indica l'autostato ottenuto dal risolutore esatto.

Per effettuare il precedente calcolo è stata eseguita una iterazione su tutti i valori intermedi dei parametri memorizzati durante l'algoritmo. Ad ogni step è stato costruito un circuito temporaneo partendo da quello parametrizzato assegnando ai parametri il valore intermedio attuale; per determinare lo stato finale si è eseguita la misura dello stato dei qubit uscenti dal suddetto circuito, a diversi valori di shots. Infine, si è calcolato il prodotto scalare (3.2.1) ad ogni iterazione, mediante modulo `numpy`. Anche in questo caso si è ripetuto il procedimento descritto per dieci esecuzioni di VQE, per ogni valore di shots, e si è calcolata la media e la deviazione standard dei valori ottenuti. Le tabelle 3.1, 3.2 mostrano questi risultati per entrambe le molecole.

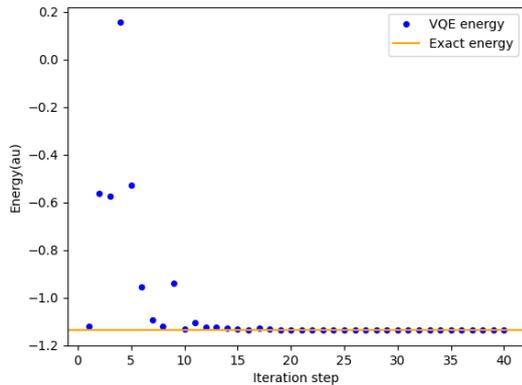
Le figure 3.4 e 3.5 mostrano i valori intermedi della fidelity durante l'esecuzione del VQE per le due molecole H_2 e LiH . Analogamente al caso dell'energia, si vede come il numero di shots influisca sui valori finali: in questo caso bassi valori di shots portano ad un offset della fidelity, particolarmente evidente nel caso di LiH (fig. 3.5); all'aumentare del numero di shots tale distanza si riduce e la fidelity tende a convergere al valore esatto 1.



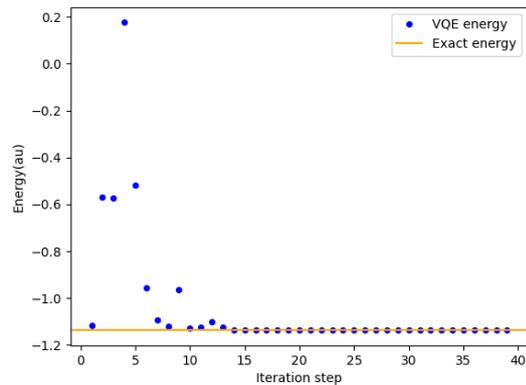
(a) 256 shots



(b) 1024 shots

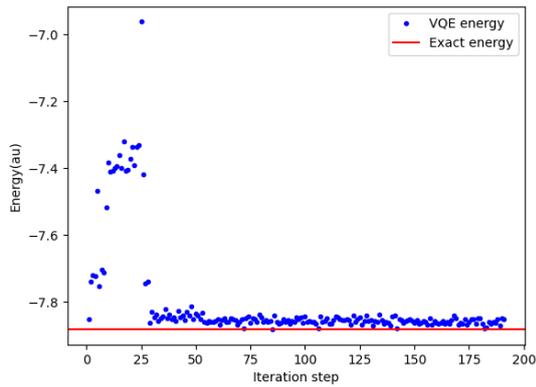


(c) 4096 shots

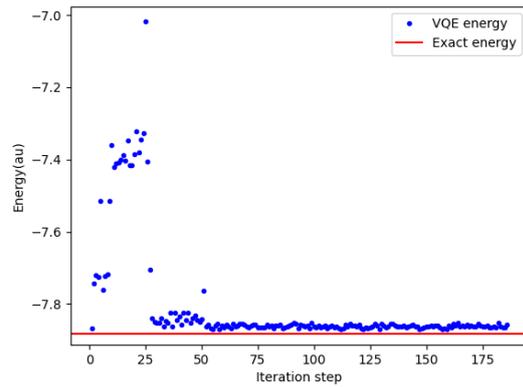


(d) 16384 shots

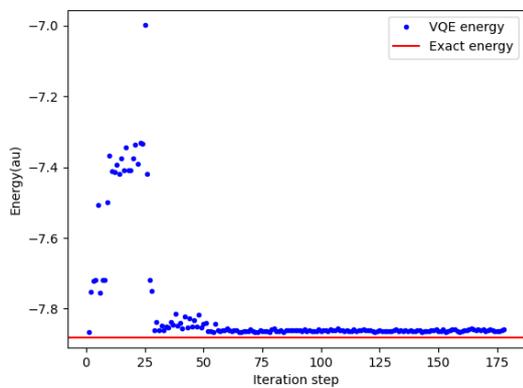
Figura 3.2: Esecuzione del VQE per la molecola H_2 a diversi valori di shots. Dai grafici è evidente come all'aumentare del numero di shots l'andamento è meno fluttuante e il valore di convergenza è sempre più prossimo al valore esatto. In tutti i grafici la deviazione standard non è apprezzabile in quanto la sua dimensione è minore delle dimensioni del punto.



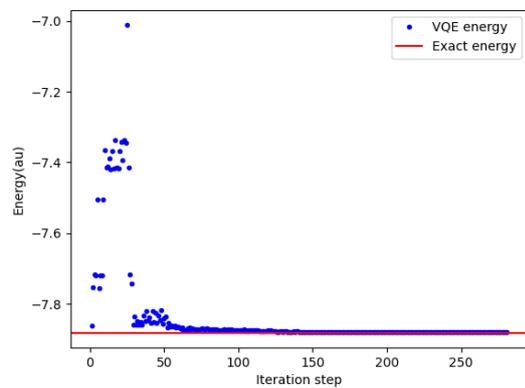
(a) 256 shots



(b) 1024 shots

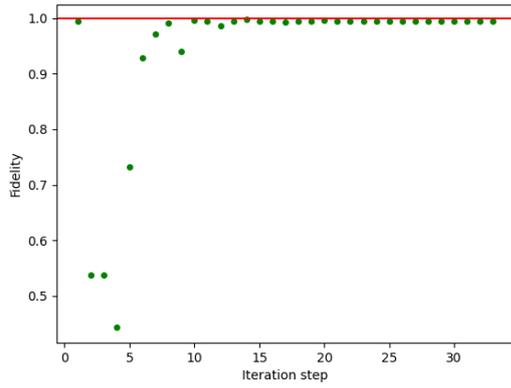


(c) 4096 shots

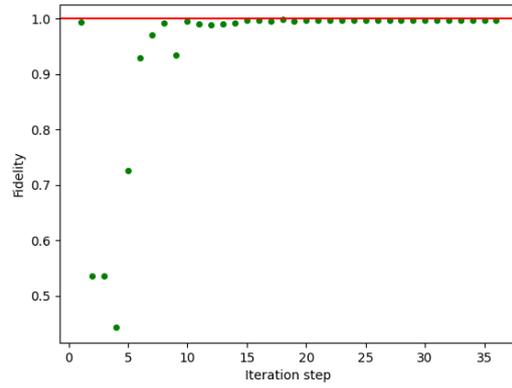


(d) 16384 shots

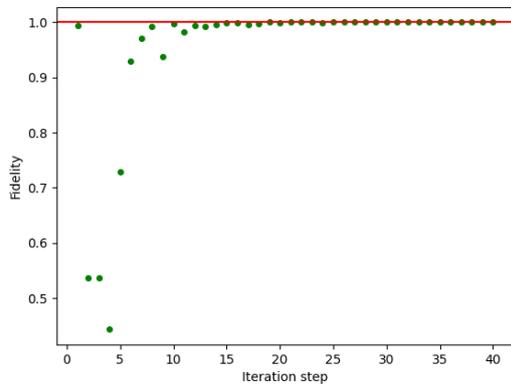
Figura 3.3: Esecuzione del VQE per la molecola LiH a diversi valori di shots. Come nel caso della figura 3.2 all'aumentare degli shots l'andamento è meno fluttuante e il valore finale è più vicino al valore esatto. In tutti i grafici la deviazione standard non è apprezzabile in quanto la sua dimensione è minore delle dimensioni del punto.



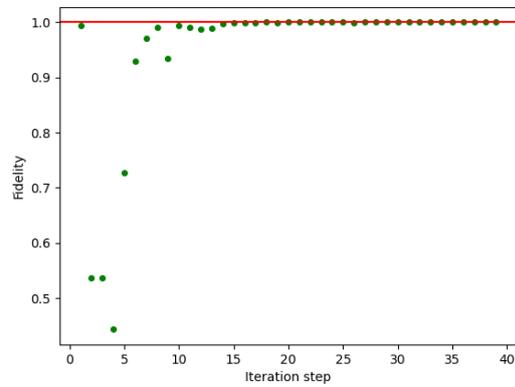
(a) 256 shots



(b) 1024 shots

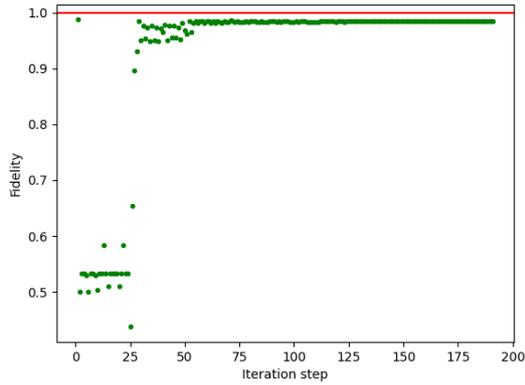


(c) 4096 shots

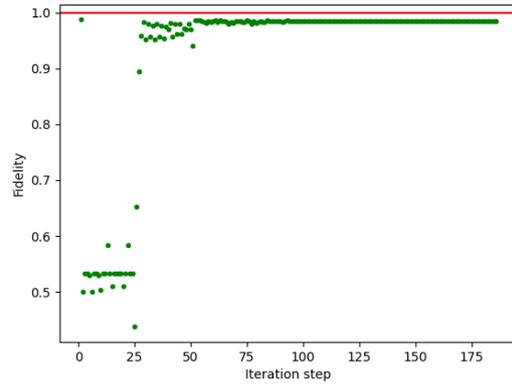


(d) 16384 shots

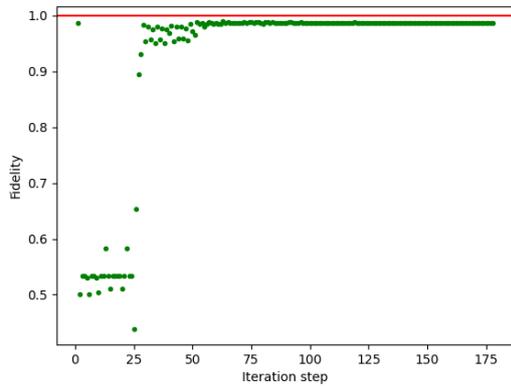
Figura 3.4: Andamento della fidelity per la molecola H_2 in funzione delle iterazioni dell'algoritmo, a diversi valori di shots. Dai grafici si nota come all'aumentare degli shots la fidelity tenda al valore esatto 1. Analogamente al caso dell'energia, la deviazione standard non è apprezzabile in quanto la sua dimensione è minore di quella del punto.



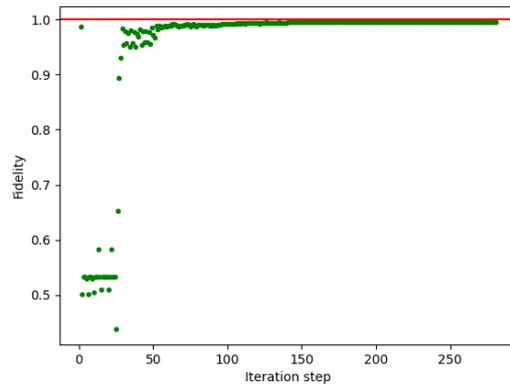
(a) 256 shots



(b) 1024 shots



(c) 4096 shots



(d) 16384 shots

Figura 3.5: Andamento della fidelity per la molecola LiH durante l'esecuzione del VQE. Analogamente a quanto mostrato in figura 3.4 all'aumentare degli shots la fidelity tende al valore esatto 1. Inoltre, in questo caso è più evidente come a bassi shots lo stato finale si discosti maggiormente da quello esatto. Analogamente al caso dell'energia, la deviazione standard non è apprezzabile in quanto la sua dimensione è minore di quella del punto.

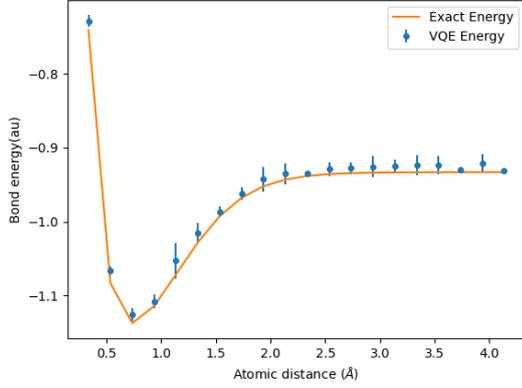
3.2.2 Energia di legame

La seconda fase della simulazione consiste nel chiamare la funzione (3.1.1) per un range di distanze interatomiche prefissato costruendo una Hamiltoniana per ogni geometria così definita. Per ogni operatore si sono eseguite dieci simulazioni complete del VQE per diversi valori di shots, e una risoluzione esatta. Le figure 3.6 e 3.7 mostrano l'andamento dell'energia di legame per le molecole H_2 e LiH ottenute dal procedimento appena descritto. In ogni grafico sono rappresentate, per ogni distanza, la media e la deviazione standard derivanti dalle dieci esecuzioni del VQE. Nel caso di H_2 si osserva come con un basso numero di shots (fig.3.6a) l'algoritmo porti a risultati distanti da quello esatto, in analogia con quanto discusso nel paragrafo 3.2.1. Aumentando il numero di shots tale effetto si placa rapidamente, portando ad un evidente accordo con l'andamento esatto. I grafici dell'energia di legame della molecola LiH mostrano una maggiore influenza del numero di shots sull'andamento. Con un basso numero (fig.3.7a) si vede un accordo solo per distanze minori di quella di legame con una divergenza evidente nel caso di distanze maggiori. In particolare, osservando le dispersioni, si evince come, anche ripetendo più volte l'esecuzione, i valori dell'energia si discostino da quello esatto. All'aumentare del numero di shots (figg.3.6b, 3.6c), si vede come l'andamento dei punti tenda ad avvicinarsi a quello esatto, pur mantenendo un offset sistematico tra le varie esecuzioni, evidenziato dalle dispersioni con valore assoluto molto minore del valore medio. Solo nel caso di elevato numero di shots (fig.3.6d) si vede come l'energia tenda a coincidere con il caso esatto.

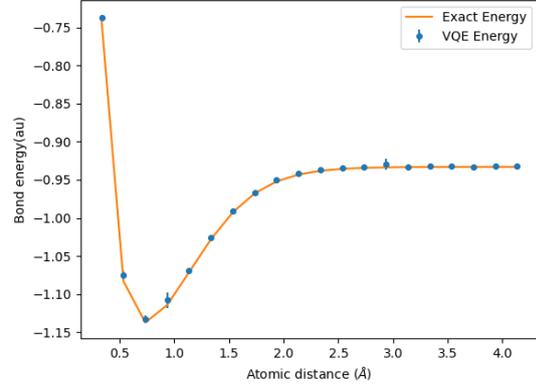
3.2.3 Risultati numerici

Le tabelle 3.1, 3.2 mostrano i valori numerici dei risultati ottenuti dalla simulazione, confrontati con quelli esatti; anche i valori numerici confermano quanto discusso nelle sezioni precedenti; il numero di shot influisce sul risultato finale dell'algoritmo, portando risultati sempre più convergenti a quelli esatti aumentando il numero di shots. In particolare, si vede che per un valore molto elevato, i risultati sono in accordo fino alla terza cifra significativa in entrambi i casi.

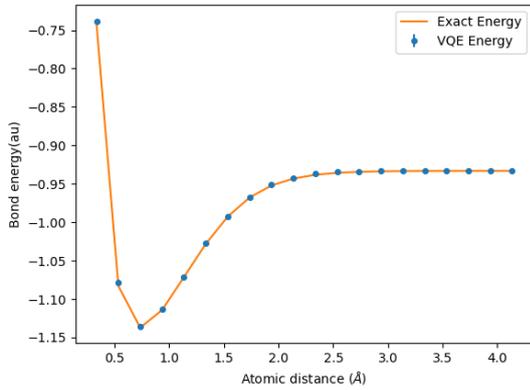
Tuttavia, la procedura descritta in questa sezione è ideale in quanto avviene tenendo conto solo del *rumore statistico* derivato dalla valutazione dei valori medi dei gates tramite un numero finito di misurazioni, senza considerare gli effetti presenti sui dispositivi reali. L'accordo dei risultati dà quindi solo conferma della correttezza teorica dell'algoritmo. Nella sezione successiva si discutono i risultati ottenuti nel caso di un hardware reale.



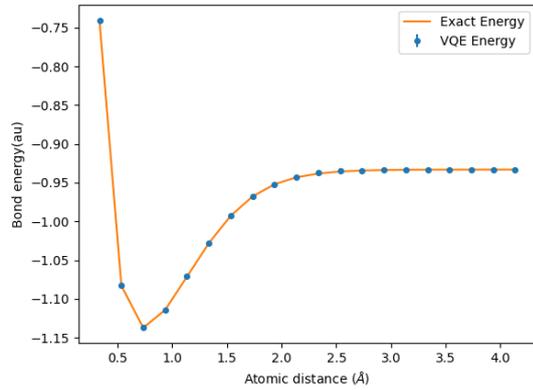
(a) 256 shots



(b) 1024 shots



(c) 4096 shots

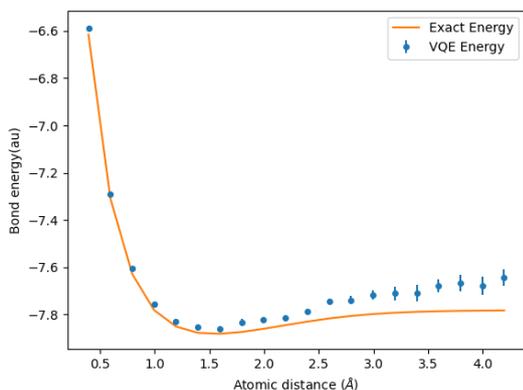


(d) 16384 shots

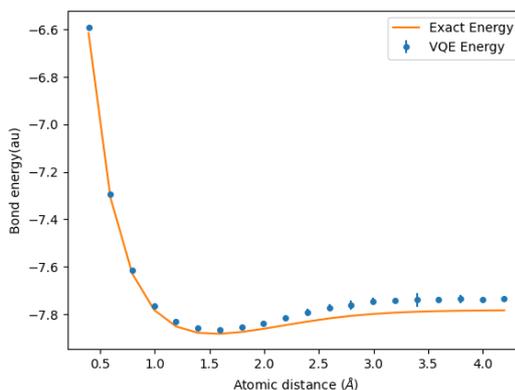
Figura 3.6: Energia di legame per la molecola H_2 calcolata mediante VQE. Ogni grafico rappresenta la media e la deviazione standard calcolate, ad ogni distanza, dal risultato di dieci esecuzioni complete di VQE ognuna con un numero differente di shots. Ad eccezione del grafico 3.6a, dove si vede come con un basso numero di shots l’algoritmo possa portare a risultati divergenti da quello atteso, l’andamento riproduce fedelmente quello esatto.

Shots	256	1024	4096	16384
Energia(au)	-1.126 ± 0.008	-1.133 ± 0.004	-1.1356 ± 0.0019	-1.1370 ± 0.0004
Fidelity	0.993 ± 0.003	0.998 ± 0.003	0.9993 ± 0.0006	0.99995 ± 0.00004

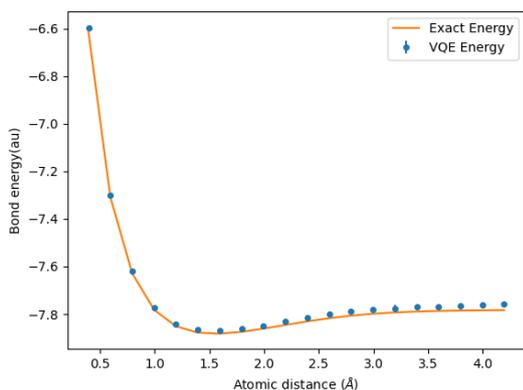
Tabella 3.1: Risultati numerici per la molecola H_2 . Ogni valore rappresenta la media, con associata deviazione standard, dei risultati ottenuti da dieci iterazioni di VQE. È evidente come all’aumentare del numero di shots aumenti l’accordo coi valori esatti di energia e fidelity calcolati con NumPySolver, pari rispettivamente a -1.137306036 , e 1.



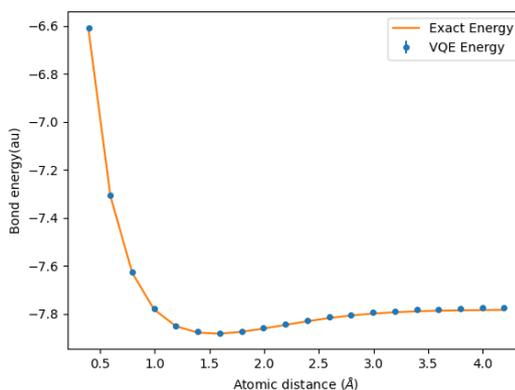
(a) 256 shots



(b) 1024 shots



(c) 4096 shots



(d) 16384 shots

Figura 3.7: Energia di legame per la molecola LiH calcolata mediante VQE. Ogni grafico rappresenta la media e la deviazione standard calcolate, ad ogni distanza, dal risultato di dieci esecuzioni complete di VQE ognuna con un numero differente di shots. A differenza del caso della molecola H_2 (3.6), l'effetto del numero di shots sull'andamento è più evidente: escludendo il caso delle distanze più basse, si vede come, a distanze maggiori di quella di legame, l'algorithm tenda a divergere dal risultato atteso anche ripetendo più volte l'esecuzione. Tale effetto si attenua aumentando il numero di shots, raggiungendo un accordo soddisfacente solo nel caso di un numero di shots elevato (3.7d)

Shots	256	1024	4096	16384
Energia(au)	-7.861 ± 0.010	-7.864 ± 0.007	-7.870 ± 0.006	-7.8789 ± 0.0012
Fidelity	0.973 ± 0.011	0.984 ± 0.004	0.989 ± 0.006	0.9957 ± 0.0013

Tabella 3.2: Risultati numerici per la molecola LiH. Ogni valore rappresenta la media, con associata deviazione standard, dei risultati ottenuti da dieci iterazioni di VQE. È evidente come all’aumentare del numero di shots aumenti l’accordo coi valori esatti di energia e fidelity calcolati con NumPySolver, pari rispettivamente a -7.877972888 , e 1.

Shots	128	512	2048	8192
Energia(au)	-0.94846309	-1.09533681	-1.12502105	-1.11964071
Fidelity	0.92053789	0.98857915	0.99412787	0.99535250

Tabella 3.3: Risultati numerici per la molecola H₂ calcolati su dispositivo IBMQ Lima. Dai valori della fidelity si vede come all’aumentare delle misurazioni lo stato tenda a convergere a quello esatto, mantenendo un offset nell’energia causato dal rumore.

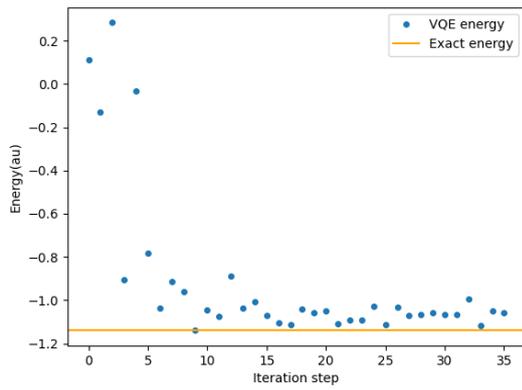
3.3 Simulazione su dispositivo reale

Nelle sezioni precedenti si è discussa la performance dell’algoritmo nelle condizioni ideali di un simulatore locale senza rumore; per valutare il VQE in condizioni più realistiche sono stati utilizzati i dispositivi reali messi a disposizione da IBM, in particolare il computer IBMQ Lima. In questo caso si è eseguito il calcolo dell’energia della molecola H₂, utilizzando la mappatura di parità per ottimizzare l’esecuzione e diminuire i tempi di attesa. Anche su dispositivo reale si utilizzato l’ansatz UCCSD e l’ottimizzatore COBYLA configurato per eseguire al massimo 1000 iterazioni.

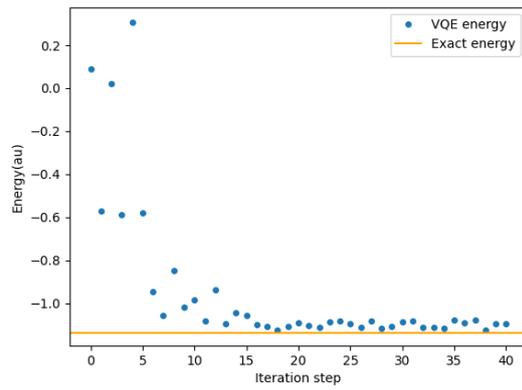
3.3.1 Energia dello stato fondamentale e fidelity

Analogamente al caso locale, si è fissata la distanza interatomica al valore sperimentale di 0.735 \AA e si è generato un valore casuale per il valore iniziale dei parametri.

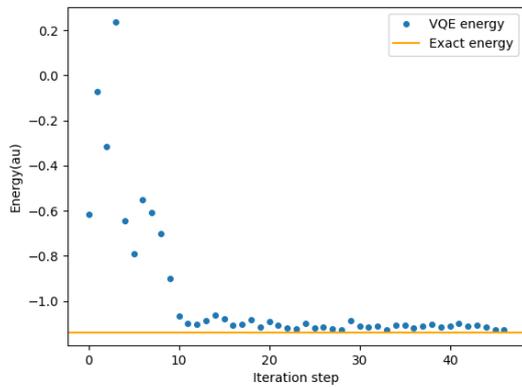
Le figure 3.8, 3.9 mostrano l’andamento dell’energia e della fidelity in funzione delle iterazioni dell’algoritmo per differenti valori di shots. In tutti i grafici è evidente come il valore di convergenza sia diverso da quello atteso; inoltre, analogamente al caso del simulatore ideale, all’aumentare degli shots l’oscillazione dei valori diminuisce, mantenendo però l’offset di energia. D’altra parte, come mostrato in figura 3.9 e in tabella 3.3, all’aumentare degli shots la fidelity tende a convergere maggiormente al valore esatto. La tabella 3.3 mostra i risultati ottenuti dall’esecuzione per i diversi valori di shots.



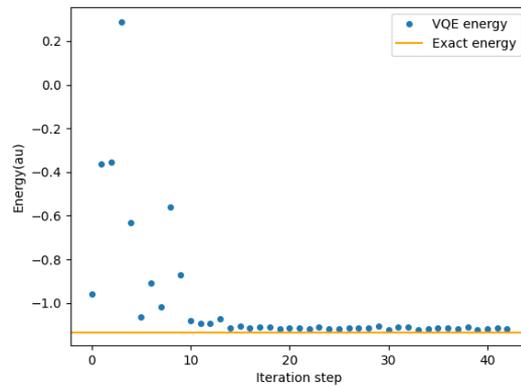
(a) 128 shots



(b) 512 shots

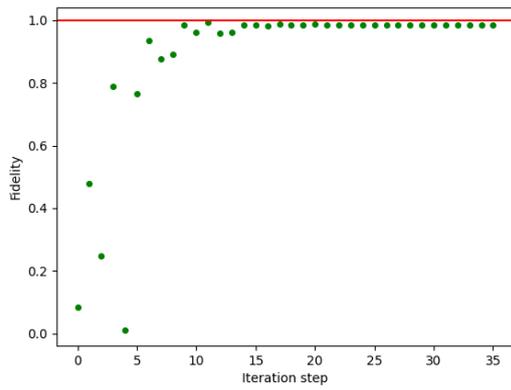


(c) 2048 shots

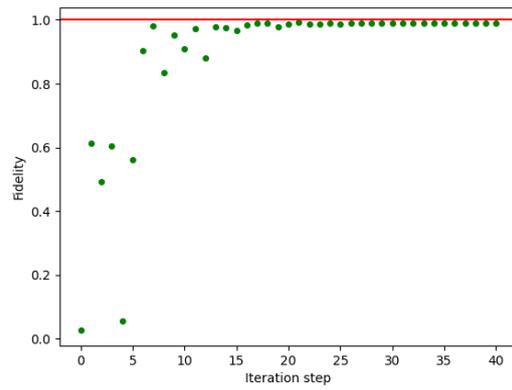


(d) 8192 shots

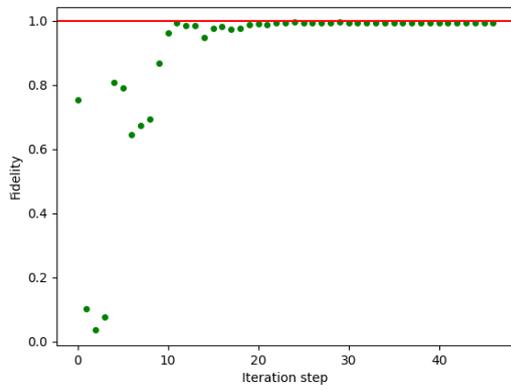
Figura 3.8: Andamento dell'energia dello stato fondamentale per diversi valori del numero di shots calcolata su dispositivo IBMQ Lima.



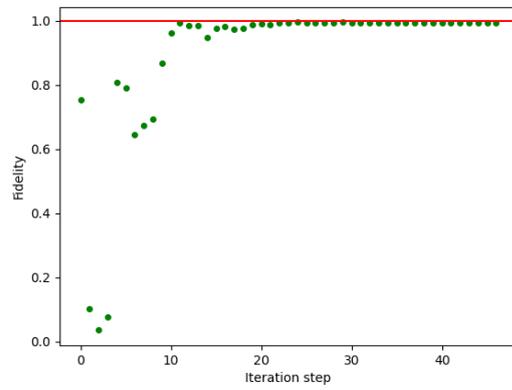
(a) 128 shots



(b) 512 shots



(c) 2048 shots



(d) 8192 shots

Figura 3.9: Andamento della fidelity dello stato fondamentale per diversi valori del numero di shots calcolata su dispositivo IBMQ Lima.

3.3.2 Energia di legame

L'ultima analisi è stata fatta per l'andamento dell'energia di legame, fissando il numero di misurazioni a 8192. In figura 3.10 è mostrato il grafico ottenuto confrontato con il caso esatto, analogamente a quanto fatto nel simulatore ideale. Anche in questo caso si nota un valore dell'energia che si discosta da quello esatto, ma l'andamento della curva è quello atteso per l'energia di legame.

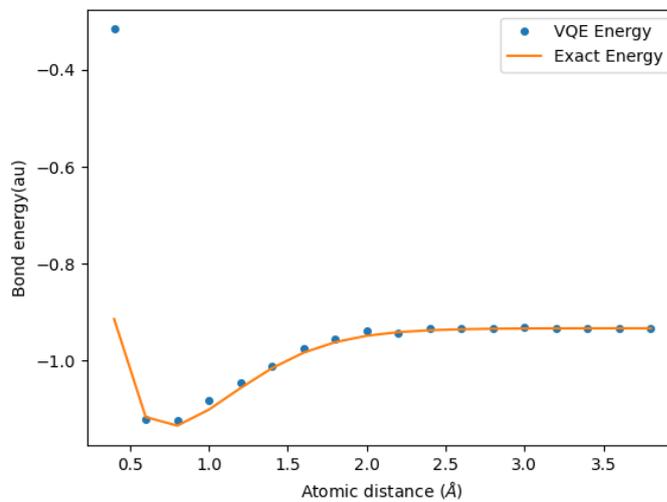


Figura 3.10: Profilo dell'energia di legame della molecola H_2 calcolata su dispositivo IBMQ Lima per un valore di shots configurato a 8192.

Conclusioni

L'obiettivo di questa tesi era quello di presentare una descrizione dell' algoritmo ibrido VQE per mostrare il formalismo che si può utilizzare per codificare e trattare problemi di natura fisica sui computer quantistici. L'analisi fatta, seppur introduttiva, ha confermato la correttezza teorica dell' algoritmo fornendo risultati in linea con quelli attesi, aprendo alla possibilità di analizzare le possibili ottimizzazioni applicabili all' algoritmo per trattare problemi più complessi e ottenere risultati più fedeli. Sfruttando diverse combinazioni di mappature a qubit, ansatz e ottimizzatori è possibile costruire varianti del VQE adeguate a problemi di diversa natura; le fonti [4, 9] trattano in maniera formale ed approfondita alcune di queste possibilità.

D'altra parte, l'esecuzione dell' algoritmo sul dispositivo reale ha messo in luce gli attuali difetti ancora presenti in questo tipo di tecnologia, derivanti soprattutto da limiti ingegneristici come, per esempio, la difficoltà di creare sistemi quantistici perfettamente isolati; infatti, come noto dalla teoria della meccanica quantistica, l'interazione con l'ambiente (*quantum decoherence*) si manifesta come un processo di misura che influisce sullo stato finale dei qubit, influenzando sulla correttezza dei risultati.

Infine, `qiskit` si è rivelato un utile strumento per la comprensione dei principi della computazione quantistica e per l'implementazione dell' algoritmo. Attualmente, il modulo `qiskit_nature` contiene numerose librerie per l'esecuzione del VQE (oltre che di numerosi altri algoritmi) con diverse configurazioni, come discusso in precedenza; inoltre, la community open source e la stessa IBM sono continuamente al lavoro per ottimizzare e arricchire l'intero SDK, portando a possibilità sempre crescenti per la produzione di algoritmi quantistici.

Appendice A

Rappresentazione dei circuiti quantistici

Come introdotto nel paragrafo 1.4 i circuiti quantistici sono rappresentati da fili e porte logiche, rappresentanti rispettivamente qubit (e bit classici) e gate quantistici. Di seguito sono descritti i simboli utilizzati per la rappresentazione dei circuiti quantistici, disegnati con il modulo `qiskit`.

A.1 Porte logiche a singolo qubit

Le porte logiche a singolo qubit sono rappresentate come scatole etichettate dal nome dell'operatore associato, con un singolo filo entrante ed uscente. Di seguito sono riportati i simboli per le porte logiche discusse nel paragrafo 1.4.1 e la rappresentazione di un generico gate \hat{U} .

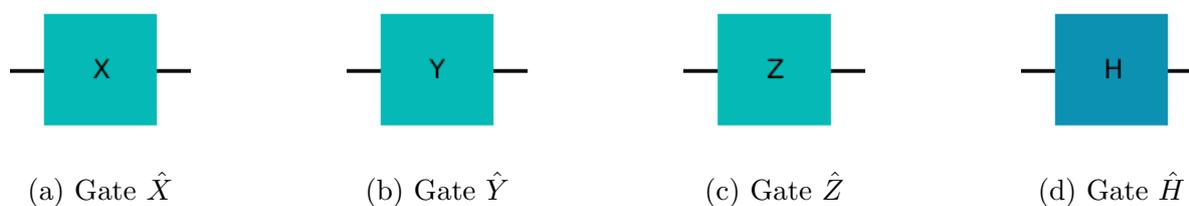


Figura A.1: Rappresentazione dei gate quantistici a singolo qubit descritti nel paragrafo 1.4.1. Ogni porta è etichettata dal nome dell'operatore associato, e possiede un filo (qubit) di input e di output.



Figura A.2: Rappresentazione di un generico gate \hat{U} a singolo qubit.

A.2 Porte logiche a qubit multipli

Le porte logiche a qubit multipli sono rappresentate da scatole etichettate, con più fili entranti ed uscenti. Va notato che il numero di fili entranti è sempre uguale al numero di fili uscenti, in quanto ogni operatore associato ad un gate deve essere reversibile. Di seguito è riportato il simbolo associato ad un generico gate a due qubit.

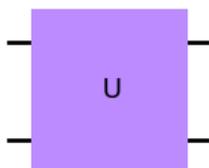


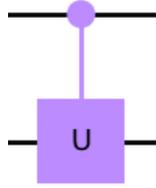
Figura A.3: Rappresentazione di un generico gate a due qubit descritto nel paragrafo 1.4.2. In questo caso il gate è etichettato dal nome generico \hat{U} , e possiede due fili entranti ed uscenti. In generale, dati N qubit, saranno presenti N fili entranti ed uscenti.

A.3 Operazioni controllate

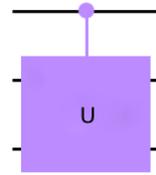
Le operazioni controllate sono un caso particolare di porte logiche a qubit multipli; tuttavia, esiste una notazione specifica per rappresentarle, nella quale si mettono in evidenza i qubit di controllo, collegati alla porta logica associata all'operazione. Di seguito sono mostrate due operazioni controllate, rispettivamente a singolo qubit e a qubit multipli. Tipicamente, l'operazione controllata $C\hat{X}$ possiede una rappresentazione particolare, differente da quella generica, mostrata in figura A.5

A.4 Misurazione e bit classici

Come discusso nel paragrafo 1.4, la misurazione è rappresentata da un metro digitale; i bit classici sui quali si memorizzano i risultati vengono rappresentati da un doppio filo.



(a) Gate controllato a singolo qubit



(b) Gate controllato a due qubit

Figura A.4: Rappresentazione dei gate quantistici controllati, rispettivamente a singolo qubit (A.4a) e a qubit multipli (A.4b). I qubit di controllo vengono rappresentati tramite cerchi collegati alle porte logiche che controllano. In generale, data un'operazione con k qubit di controllo, la rappresentazione avrà un cerchio per ogni qubit di controllo, ognuno collegato alla porta logica.

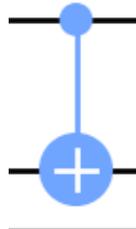


Figura A.5: Rappresentazione dell'operazione controllata $C\hat{X}$, anche detta $CN\hat{O}T$

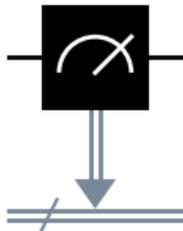


Figura A.6: Rappresentazione della misura dello stato di un qubit (filo singolo), con memorizzazione in un bit classico (filo doppio).

Bibliografia

- [1] A. M. Turing, *On computable numbers, with an application to the entscheidungsproblem*, Proceedings of the London Mathematical Society s2 42 (1937) 230, <https://doi.org/10.1112/plms/s2-42.1.230>.
- [2] S. Kumar, *Fundamental Limits to Moore's Law*, arXiv:1511.05956 (2015).
- [3] M. A. Nielsen, I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 10th Anniversary Edition (2010).
- [4] M. F. Ramôa, *Ansätze for Noisy Variational Quantum Eigensolvers*, arXiv:2212.04323 (2022).
- [5] C. Fontana, *Projection Method for the Decomposition of Hermitian Matrices into Pauli Matrices*, ResearchGate, 10.13140/RG.2.2.13529.29285/3 (2023).
- [6] D. A. Fedorov, B. Peng, N. Govind, Y. Alexeev, *VQE method: a short survey and recent developments*, arXiv:2103.08505 (2022).
- [7] B. Bauer, S. Bravyi, M. Motta, G. K. Chan, *Quantum Algorithms for Quantum Chemistry and Quantum Materials Science*. Chem. Rev. 120 22 (2020) 12685, <https://doi.org/10.1021/acs.chemrev.9b00829>.
- [8] Y. Cao, J. Romero J. P. Olson, and M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, A. Aspuru-Guzik, *Quantum Chemistry in the Age of Quantum Computing*. Chem. Rev. 119 19 (2019) 10856, <https://doi.org/10.1021/acs.chemrev.8b00803>.
- [9] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, J. Tennyson, *The Variational Quantum Eigensolver: A review of methods and best practices*, Phys. Rep. 986 (2022) 1, <https://doi.org/10.1016%2Fj.physrep.2022.08.003>.
- [10] B. H. Bransden, C. J. Joachain, *Physics of Atoms and Molecules*, Prentice Hall, 2nd Edition (2003).

- [11] J. M. Arazzola, *Givens rotations for quantum chemistry* pennylane.ai (2021), https://pennylane.ai/qml/demos/tutorial_givens_rotations#givens-rotations-for-quantum-chemistry.
- [12] Qiskit webpage: qiskit.org, <https://qiskit.org>.
- [13] M. Jackson, R. P. McEachran, M. Cohen, *The frozen-core approximation for diatomic molecules: the lowest $1 \Sigma +$ states of H_2* , J. Phys. B: Atom. Mol. Phys. 5 (1972) 927, <https://dx.doi.org/10.1088/0022-3700/5/5/013>.