

Università degli Studi di Milano-Bicocca

Dipartimento di Fisica

Corso di Laurea Triennale



STUDIO DI REAZIONI CHIMICHE SU COMPUTER QUANTISTICI
TRAMITE L'USO DI METODI COUPLED CLUSTER

Relatore:
Prof. Andrea Giachero

Candidato:
Giacomo Fracasso

Correlatore:
Dr. Rodolfo Carobene

Matricola:
863248

Anno Accademico 2023/2024

Dedica?

- *Autore?*

Abstract

La chimica quantistica tradizionale offre una varietà di strategie per descrivere i sistemi molecolari e le loro interazioni, tra questi, i metodi *Coupled Cluster* si sono imposti come *golden standard*. Tuttavia, presentano limitazioni computazionali significative, specialmente in presenza di sistemi fortemente correlati o di grandi dimensioni. Ad esempio, la variante più utilizzata *coupled cluster singles doubles (triples)* CCSD(T), che include eccitazioni singole, doppie e un'approssimazione delle triple, ha un costo computazionale che scala come la settima potenza della dimensione del problema.

Per superare questi limiti, la computazione quantistica si è proposta come una nuova frontiera, promettendo di rappresentare gli stati quantistici in modo più efficiente rispetto agli schemi classici.

Questo lavoro di tesi si concentra sulla simulazione della dissociazione della molecola di idruro di litio (LiH) utilizzando una specifica categoria di funzioni d'onda *Coupled Cluster*: gli ansatze *Unitary Coupled Cluster* (UCC). Questi ultimi hanno acquisito una certa rilevanza grazie alla naturalezza con cui possono essere implementati su processori quantistici (QPU), oltre che per la capacità di modellare le interazioni elettroniche in modo efficiente e accurato. Sebbene la profondità dei circuiti quantistici UCC possa superare ampiamente le capacità dei dispositivi attualmente disponibili, questi metodi rappresentano un buon punto di partenza per analizzare il potenziale delle QPU nel campo della chimica quantistica. Per questa ragione, sono stati impiegati diversi approcci basati su UCC, con l'obiettivo di confrontare le loro prestazioni, sia in termini di energia calcolata che di complessità del circuito quantistico. Nello specifico, si sono analizzate le varianti UCCS, UCCD, UCCSD, in cui S e D stanno per *singles* e *doubles* ed indicano l'ordine delle eccitazioni considerate, e *pair-UCCD* (pUCCD) che considera solamente eccitazioni doppie su elettroni appaiati, riducendo così significativamente le risorse richieste. Infine, è stato considerato l'ansatz *hardware-efficient* EfficientSU(2), per valutare la praticabilità di un'alternativa con una struttura circuitale semplificata. I risultati dell'analisi, eseguite tramite simulatori in locale, evidenziano caratteristiche distintive per ciascun ansatz. Come riferimento per le energie di stato fondamentale e di dissociazione, sono state usate le previsioni di Full Configuration Interaction (FCI), che fornisce un limite inferiore alle forme variazionali UCC.

L'ansatz UCCS si è dimostrato particolarmente inefficace nella stima dell'energia di legame, producendo un risultato praticamente sovrapponibile a quello di Hartree-Fock (-7.862 Ha), lontani dal valore FCI di -7.882 Ha. Nella dissociazione, tuttavia, UCCS è più in linea con UCCD e UCCSD, risultando in -7.782 Ha rispetto al valore FCI di -7.783 Ha. UCCSD, il più oneroso degli approcci, concatena eccitazioni singole e doppie (S e D), aumentando sensibilmente la profondità del circuito (2098 livelli) e il numero di porte CNOT (1616); le energie stimate risultano -7.881 Ha per lo stato fondamentale e -7.782 Ha per la dissociazione.

pUCCD si distingue come compromesso interessante, con una struttura circuitale meno dispendiosa (511 livelli, 384 porte CNOT): l'energia di *ground state* è -7.878 Ha, mentre

quella di dissociazione è -7.778 Ha. Quando viene abbinato a ottimizzazioni orbitali viene detto *orbital optimized* pUCCD (oo-pUCCD) e i suoi risultati a grandi distanze migliorano notevolmente, a fronte di un costo relativamente piccolo in termini di risorse: oo-pUCCD raggiunge i -7.782 Ha.

Infine, EfficientSU2, offre una struttura semplice con 21 livelli e 40 porte CNOT, eppure - almeno a grandi distanze - non mostra discrepanze energetiche più ampie rispetto agli ansatze UCC, con -7.864 Ha per lo stato fondamentale e -7.782 Ha per la dissociazione. Questi risultati confermano che, sebbene gli ansatze basati su UCC possano offrire prestazioni notevoli, per applicazioni nel breve termine rimane necessario ricorrere a delle tecniche approssimative; l'adozione di un ansatz *hardware-efficient* come EfficientSU(2) potrebbe risultare una valida alternativa per utilizzi pratici in cui, per il momento, la riduzione della complessità circuitale è cruciale.

Indice

1	Cenni teorici	6
1.1	Problema elettronico	6
1.1.1	Approssimazione di Born-Oppenheimer	7
1.1.2	Metodo di Hartree	8
1.1.3	Metodo di Hartree-Fock	9
1.1.4	Basi per gli orbitali molecolari	11
1.2	Seconda quantizzazione	12
1.2.1	Spazio di Fock	12
1.2.2	Operatori fermionici di creazione e distruzione	12
1.2.3	Particelle non interagenti	13
1.2.4	Hamiltoniana di seconda quantizzazione	14
1.3	Metodi classici post Hartree-Fock	14
1.3.1	CI: Configuration Interaction	15
1.3.2	CC: Coupled Cluster	16
2	Soluzioni computazionali quantistiche	18
2.1	Introduzione al quantum computing	18
2.1.1	Qubit	18
2.1.2	Operazioni sui qubit	19
2.1.3	Qubit Mapping	23
2.2	Variational quantum algorithms	25
2.2.1	VQE	25
2.3	Quantum Coupled Cluster	26
2.3.1	Implementazione	26
2.3.2	pUCCD: Pair unitary coupled cluster doubles	28
2.4	Hardware efficient ansatzes	29
3	Simulazioni	31
3.1	Introduzione al codice	31
3.1.1	Impostazione del problema	31
3.1.2	Ansatz	34
3.1.3	VQE	35
3.2	LiH - Confronti UCC	37
3.3	Orbital Optimized quantum pair Coupled Cluster Doubles	41
3.3.1	Costruzione del circuito oo-pUCCD	41
3.3.2	oo-VQE	43
3.3.3	Risultati	45
	Conclusioni	47

Elenco delle figure

2.1	Sfera di Bloch [20]	19
2.2	Esponenziazione eccitazioni singole	28
2.3	Esponenziazione eccitazioni doppie	28
3.1	FCI con diverse basi	33
3.2	Circuito UCCS	34
3.3	Circuito EfficientSU(2)	35
3.4	LiH confronto tra ottimizzatori	37
3.5	LiH confronto tra ansatz	38
3.6	Confronto energie e profondità dei circuiti	39
3.7	Processo di ottimizzazione a $d = 1.595 \text{ \AA}$	40
3.8	LiH con ottimizzazioni orbitali	46
3.9	LiH con ottimizzazioni orbitali	47

1. Cenni teorici

Questo capitolo offre una panoramica dei concetti e delle tecniche principali della *chimica computazionale*, un ramo della chimica che utilizza modelli matematici e algoritmi numerici per risolvere problemi legati alla struttura, alle proprietà e al comportamento delle molecole e dei materiali. Con l'avanzamento delle strategie computazionali e l'aumento della potenza di calcolo, la chimica computazionale è diventata uno strumento fondamentale per comprendere e prevedere fenomeni chimici.

In particolare, si affronteranno il metodo di Hartree-Fock e i metodi di correlazione elettronica *Full Configuration Interaction* e *Coupled-Cluster*, che permettono di descrivere le interazioni tra gli elettroni nelle molecole.

1.1 Problema elettronico

Il problema affrontato nel seguente elaborato è trovare soluzioni approssimate all'equazione di Schrödinger stazionaria non-relativistica

$$\hat{H} |\psi(\{\underline{R}_A\}, \{r_i\})\rangle = E |\psi(\{\underline{R}_A\}, \{r_i\})\rangle \quad (1.1)$$

dove \hat{H} è l'hamiltoniana di un sistema di N nuclei e M elettroni, individuati rispettivamente dalle posizioni $\{\underline{R}_A\}_{A=1,\dots,N}$ e $\{r_i\}_{i=1,\dots,M}$. Esplicitando le interazioni tra i componenti del sistema:

$$\hat{H} = \hat{T}_e + \hat{T}_n + U_{nn} + U_{ee} + U_{ne} \quad (1.2)$$

In cui compaiono i seguenti termini:

\hat{T}_e : Operatore energia cinetica degli elettroni

$$\hat{T}_n = \sum_i^M \frac{\hat{p}_i^2}{m_e} \quad (1.3a)$$

\hat{T}_n : Operatore energia cinetica dei nuclei

$$\hat{T}_e = \sum_A^N \frac{\hat{p}_A^2}{M_A} \quad (1.3b)$$

U_{nn} : Energia potenziale coulombiana tra i nuclei

$$U_{nn}(\{\underline{R}_A\}) = \frac{1}{2} \sum_A^N \sum_{B \neq A}^N \frac{1}{4\pi\epsilon_0} \frac{Z_A Z_B e^2}{|\underline{R}_A - \underline{R}_B|} \quad (1.3c)$$

U_{ee} : Energia potenziale coulombiana tra gli elettroni

$$U_{ee}(\{r_i\}) = \frac{1}{2} \sum_i^M \sum_{j \neq i}^M \frac{1}{4\pi\epsilon_0} \frac{e^2}{|r_i - r_j|} \quad (1.3d)$$

U_{en} : Energia potenziale coulombiana tra i gli elettroni e i nuclei

$$U_{en}(\{r_i, \underline{R}_A\}) = - \sum_A^N \sum_i^M \frac{1}{4\pi\epsilon_0} \frac{Z_A e^2}{|\underline{R}_A - r_i|} \quad (1.3e)$$

Dove $\hat{p} = -i\hbar\nabla$ è l'operatore momento lineare, m_e la massa dell'elettrone mentre M_A quella del nucleo A con numero atomico Z_A .

La presenza del termine U_{en} porta subito ad una considerazione: qualsiasi soluzione *esatta* $\Psi(\{r_i\}, \{\underline{R}_A\})$ dell'equazione di Schrödinger non può essere fattorizzata in una parte elettronica e una nucleare [6].

$$\nexists \psi_n, \psi_{el} \text{ t.c. } \Psi(\{x_i\}, \{\underline{R}_A\}) = \psi_n(\{\underline{R}_A\}) \psi_{el}(\{r_i\})$$

Se ciò fosse possibile, il problema degli elettroni e quello dei nuclei potrebbero essere trattati separatamente. Da questa prima osservazione deriva l'utilità di introdurre un'approssimazione che riesca a slegare le due questioni.

1.1.1 Approssimazione di Born-Oppenheimer

Per affrontare il tema presentato alla sezione 1.1, si parte con l'isolare i termini elettronici dal problema globale, costruendo un'hamiltoniana che dipende soltanto parametricamente dalle posizioni dei nuclei.

$$\hat{H}_{el} |\psi_{el}\rangle = \left(\hat{T}_e + U_{ee} + U_{nn} + U_{ne} \right) |\psi_{el}\rangle = E(\{\underline{R}_A\}) |\psi_{el}\rangle \quad (1.4)$$

Gli autostati $|\psi_{el}\rangle$ formano un sistema ortonormale completo, perciò le soluzioni del problema complessivo possono essere espresse come loro combinazioni lineari, pesate da delle funzioni parametriche $\chi(\{\underline{R}_A\})$ [5], che si occupano di modulare le diverse configurazioni elettroniche in relazione alla disposizione spaziale degli atomi nel sistema.

$$\psi = \sum_k \chi_k(\{\underline{R}_A\}) \psi_{elk}(\{r_i\}, \{\underline{R}_A\}) \quad (1.5)$$

Questo rientra nell'approssimazione di Born-Oppenheimer, che separa il problema elettronico da quello nucleare sfruttando la notevole differenza di massa tra nuclei ed elettroni. Essendo questi ultimi molto meno massivi, la loro dinamica avviene su scale temporali estremamente più brevi; ciò permette di assumere che, al variare della configurazione nucleare, gli elettroni adeguino istantaneamente il loro stato, seguendo il moto dei nuclei in modo adiabatico. Secondo questa interpretazione, vengono applicati i seguenti passaggi:

1. La somma 1.5 si riduce ad un solo stato elettronico:

$$\psi \approx \chi(\{\underline{R}_A\}) \psi_{el}(\{r_i\}, \{\underline{R}_A\}) \quad (1.6)$$

Si può quindi riscrivere la eq. (1.1)

$$\left(\hat{T}_n + \hat{H}_e \right) |\chi \psi_{el}\rangle = \left(\hat{T}_n + E_{el}(\{\underline{R}_A\}) \right) |\chi \psi_{el}\rangle = E |\chi \psi_{el}\rangle \quad (1.7)$$

2. Il termine ottenuto applicando l'operatore \hat{T}_n a ψ_{el} diventa trascurabile, rendendo ψ_{el} un semplice fattore moltiplicativo che può essere eliso:

$$(\hat{T}_n + E_{el})\chi = E\chi \quad (1.8)$$

Adottando questo approccio, si ottiene un problema nucleare che dipende dal potenziale efficace $E_{el}(\{R_A\})$, assegnato dalla soluzione del problema elettronico parametrico nelle posizioni dei nuclei e indipendente dal loro moto. In sintesi: l'approssimazione di Born-Oppenheimer permette di studiare un sistema molecolare affrontando separatamente le dinamiche degli elettroni e dei nuclei. Questo lavoro di tesi si concentra prevalentemente sulle tecniche computazionali adottate per la costruzione di $E_{el}(\{R_A\})$.

1.1.2 Metodo di Hartree

Assumendo valida l'approssimazione di Born-Oppenheimer, il tema principale diventa trovare lo stato fondamentale dell'hamiltoniana elettronica (eq. 1.4) per una configurazione nucleare fissata $\{R_A\}$.

Nella risoluzione del problema elettronico - fatta eccezione per pochi casi semplici - ci si trova spesso a considerare problemi a molti elettroni, in cui la presenza del termine d'interazione U_{ee} rende necessario introdurre un nuovo livello di approssimazione.

Un primo approccio è costituito dal metodo di Hartree, in cui è centrale il concetto di *approssimazione di campo medio*. In questo schema, ciascun elettrone viene trattato come se si muovesse all'interno del campo medio generato dagli altri elettroni in *background*, evitando di considerarne individualmente l'influenza. Se si immagina la densità di probabilità $|\psi_i(\underline{r})|^2$ alla stregua di una distribuzione continua di carica, si può definire il *potenziale di Hartree* U_H :

$$\rho_k^{(0)}(\underline{r}) = \sum_{i \neq k} \rho_i^{(0)}(\underline{r}) = \sum_{i \neq k} e |\psi_i^{(0)}(\underline{r})|^2 \quad (1.9)$$

che dà:

$$U_H^k(\underline{r}) = -\frac{e}{4\pi\epsilon_0} \frac{1}{2} \int \frac{\rho_k^{(0)}(\underline{r}')}{|\underline{r} - \underline{r}'|} d\underline{r}' \quad (1.10)$$

A questo punto occorre fare una precisazione: le funzioni d'onda $\psi_i(\underline{r})$, necessarie per determinare il potenziale di Hartree, sono le soluzioni del problema stesso e non è possibile conoscerle a priori. Per questo motivo, il metodo di Hartree rientra nelle cosiddette teorie auto-consistenti di campo medio o *self-consistent field* (SCF) *theories*. In breve, lo schema prevede di inserire inizialmente un *guess* o *ansatz* $\psi_i^{(0)}(\underline{r})$ ¹ per calcolare in prima istanza $U_H^{(0)}$; dopodiché il metodo si sviluppa in maniera iterativa: ad ogni applicazione m -esima, si calcolano nuove soluzioni $\psi_i^{(m+1)}(\underline{r})$ e si determina il nuovo potenziale $U_H^{(m+1)}$, interrompendo il processo quando si raggiunge l'accuratezza desiderata².

¹Tipicamente, se si applica il metodo a sistemi atomici, le funzioni d'onda idrogenoidi

²Nelle righe seguenti si trascurano gli indici d'iterazione (m), poiché si preferisce presentare il metodo di Hartree in maniera sintetica.

In tale impostazione l'hamiltoniana H_{el} diventa separabile [1], perciò può essere scritta come somma delle hamiltoniane di singola particella h_k :

$$H_{el} \approx \sum_k^M h_k \quad (1.11)$$

dove si introduce U_H al posto di U_{ee}

$$U_{ee}^k = \sum_{j \neq k}^M \frac{1}{4\pi\epsilon_0} \frac{e^2}{|r_k - r_j|} \longrightarrow V_H^k \quad (1.12)$$

per cui³:

$$h_k = \frac{|\hat{p}_k|^2}{2m_e} + U_H^k + U_{en}^k \quad (1.13)$$

e si avrà che l'autovalore E_{el} del problema elettronico è somma delle singole energie ϵ_k e la funzione d'onda del sistema ψ_{el} è fattorizzabile nelle specifiche ψ_k

$$E_{el} = \sum_k^M \epsilon_k \quad \wedge \quad \psi_{el} = \prod_k^M \psi_k \quad (1.14)$$

nella letteratura, una ψ_{el} costruita in questo modo è detta di frequente *prodotto di Hartree*.

1.1.3 Metodo di Hartree-Fock

Lo schema di Hartree è alla base dei fondamenti teorici dei metodi SCF ma non è sufficiente per descrivere accuratamente i sistemi elettronici, poiché non tiene in considerazione un aspetto fondamentale della meccanica quantistica: l'*indistinguibilità* delle particelle [8]. Il prodotto di Hartree (eq. 1.14) non soddisfa la condizione di *antisimmetrizzazione*, per la quale è richiesto che ogni funzione d'onda che descrive un sistema di fermioni identici sia antisimmetrica rispetto allo scambio di ciascuna coppia di particelle [4]. La correzione a questo problema fu proposta indipendentemente da Fock e Slater nel 1930 e consiste nell'introdurre una funzione d'onda a molti elettroni espressa tramite un *determinante di Slater* [6]. Da questa idea, combinata l'approssimazione di campo medio ereditata dal metodo di Hartree, nasce il metodo di Hartree-Fock (HF).

Determinante di Slater

Nel seguito della trattazione verrà effettuata una piccola variazione nella notazione: fino ad ora si sono indicati con $\psi(\underline{r})$ gli *orbitali spaziali*; da questo momento in poi, diventa opportuno considerare gli *orbitali di spin* $\phi(\underline{x})$:

$$\phi(\underline{x}) = \psi(\underline{r}) \nu \quad (1.15)$$

in cui $\underline{x} = (\underline{r}, \nu)$ e $\nu \in \{\alpha, \beta\}$ indicizza l'autostato di spin.

³Fissando la posizione dei nuclei il termine U_{nn} diviene una semplice costante che introduce uno slittamento uniforme delle energie, motivo per cui viene omissso nel prosieguo della trattazione.

L'introduzione degli orbitali di spin è propedeutica alla definizione del determinante di Slater, un modo compatto e conveniente per costruire una funzione d'onda Φ totalmente antisimmetrica:

$$\Phi(\underline{x}_1, \dots, \underline{x}_M) = \frac{1}{M!} \begin{vmatrix} \phi_1(\underline{x}_1) & \phi_2(\underline{x}_1) & \dots & \phi_M(\underline{x}_1) \\ \phi_1(\underline{x}_2) & \phi_2(\underline{x}_2) & \dots & \phi_M(\underline{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\underline{x}_M) & \phi_2(\underline{x}_M) & \dots & \phi_M(\underline{x}_M) \end{vmatrix} \quad (1.16)$$

questa scrittura sfrutta le proprietà del determinante di una matrice, il cui segno viene invertito quando vengono scambiate due colonne. Inoltre, affinché il risultato sia diverso da zero, righe e colonne devono essere tutte *linearmente indipendenti*; ciò assicura anche il rispetto del *principio di esclusione di Pauli*, poiché un sistema con due fermioni identici individuati dalle stesse coordinate \underline{x} darebbe luogo a due righe uguali.

Generalmente, si richiede anche che gli orbitali di spin siano ortonormali:

$$\langle \phi_i | \phi_j \rangle = \delta_{ij}, \quad \forall i, j \in [1, M] \quad (1.17)$$

Potenziale di Fock

Il passo cruciale nello schema di Hartree-Fock è supporre che la soluzione del problema elettronico sia un singolo determinante di Slater, su cui l'energia è data da:

$$\langle \Phi | H_{el} | \Phi \rangle \quad (1.18)$$

sinteticamente, si può dire che il metodo di Hartree-Fock aderisce alle stesse logiche dello schema di Hartree (sez 1.1.2) con lo scopo di minimizzare l'energia (eq. 1.18), ma vede l'aggiunta del *potenziale di Fock*:

$$V_F^k(\underline{x}) |\phi_i(\underline{x})\rangle = \left(\frac{e^2}{4\pi\epsilon_0} \int \sum_k^M \frac{\phi_k^*(\underline{x}') \phi_i(\underline{x}')}{|\underline{r}_i - \underline{r}_k|} d\underline{x}' \right) |\phi_k(\underline{x})\rangle \quad (1.19)$$

oltre alla scrittura aggiornata del potenziale di Hartree (eq. 1.10)

$$V_H^k(\underline{x}) |\phi_i(\underline{x})\rangle = \left(\frac{e^2}{4\pi\epsilon_0} \int \sum_k^M \frac{\phi_k^*(\underline{x}') \phi_k(\underline{x}')}{|\underline{r}_i - \underline{r}_k|} d\underline{x}' \right) |\phi_i(\underline{x})\rangle \quad (1.20)$$

infine, spesso si riassume l'equazione agli autovalori - di singola particella - utilizzando l'*operatore di Fock* \hat{F} :

$$\hat{F}^k = h_k + U_H^k - V_F^k \quad (1.21)$$

con cui si scrive:

$$\hat{F}^k |\phi_k\rangle = \epsilon_k |\phi_k\rangle \quad (1.22)$$

1.1.4 Basi per gli orbitali molecolari

Finora, non si è discusso di come esprimere concretamente la funzione d'onda, ma è importante sottolineare che Ψ appartiene allo spazio di Hilbert $\mathcal{H} = L^2(\mathbb{R}^3) \times \mathbb{C}^2$, che è uno spazio infinito dimensionale. Per poterla trattare numericamente, è necessario svilupparla su una base che, pur essendo finita, permetta di ottenerne una rappresentazione approssimata. La selezione della base è dunque essenziale per bilanciare la precisione della descrizione e la complessità computazionale del sistema.

Questo tema fu affrontato inizialmente nel 1951 da Roothaan, che propose di rappresentare gli orbitali molecolari come combinazioni lineari finite di *funzioni di base* $\{\chi_\nu\}$ ⁴, spesso chiamate *orbitali atomici*:

$$\phi_k \approx \sum_{\nu} c_{\nu k} \chi_{\nu} \quad (1.23)$$

dove c_{ν} sono i coefficienti della combinazione lineare. Tipicamente, nell'effettuare calcoli su sistemi atomici, si utilizzano come ansatz le funzioni d'onda idrogenoidi χ_{nlm}^Z , che derivano dalla soluzione dell'unico problema molecolare risolvibile analiticamente. Storicamente, nelle prime versioni della teoria degli orbitali molecolari, furono proprio queste ad essere proposte come orbitali atomici per la costruzione delle funzioni ϕ_k [3]. Diventò consuetudine scegliere funzioni centrate sui nuclei atomici che richiassero le χ_{nlm}^Z e fu in questo spirito che si iniziò ad utilizzare i *Slater-Type Orbitals* (STO) [6]:

$$\chi_{a,b,c,\zeta}^{\text{STO}} = N_{a,b,c,\zeta} x^a y^b z^c e^{-\zeta|r|} \quad (1.24)$$

dove $N_{a,b,c}^{\text{STO}}$ è la costante di normalizzazione, ζ un parametro e a, b, c sono legate al momento angolare. Queste funzioni mostrano gli stessi comportamenti asintotici delle χ_{nlm}^Z a corto e lungo raggio, rispettando le condizioni imposte dalla teoria, tuttavia presentano importanti svantaggi computazionali: gli integrali coinvolti nel calcolo del valore di aspettazione (eq. 1.18) richiedono un grande sforzo in termini di tempo. Ciò rese impossibile l'implementazione delle STO nelle simulazioni di grandi molecole.

La successiva introduzione delle funzioni gaussiane o *Gaussian-Type Orbitals* (GTO) fu un'importante sviluppo poiché, sacrificando alcune proprietà teoriche, permettevano di semplificare, e insieme velocizzare, i calcoli degli integrali.

$$\chi_{a,b,c,\zeta}^{\text{GTO}} = N_{a,b,c,\zeta} x^a y^b z^c e^{-\zeta|r|^2} \quad (1.25)$$

In particolare, grazie al grande vantaggio computazionale, possono essere usate in quantità maggiore per sopperire alla loro inesattezza fisica. Pertanto, per combinare i pregi di STO e GTO, furono introdotte le funzioni gaussiane contratte o *Contracted GTO* (CGTO), che utilizzano un certo numero n di funzioni gaussiane per approssimare una funzione di Slater:

$$\chi_{a,b,c,\zeta}^{\text{CGTO}} = N_{a,b,c,\zeta} x^a y^b z^c \sum_i^n e^{-\zeta_i|r|^2} \quad (1.26)$$

⁴Si adotterà la pratica, comune in letteratura, di associare indici latini agli orbitali molecolari e indici greci a quelli atomici

Infine, da queste funzioni nacque la famiglia di basi STO- n G, in cui ciascuna CGTO è costruita a partire da n GTO. Al crescere del numero di funzioni gaussiane utilizzate aumenta l'accuratezza raggiunta dal calcolo, ma al prezzo di un più elevato tempo di esecuzione. Le simulazioni svolte nel contesto di questo elaborato saranno principalmente orientate ad uno studio qualitativo perciò, nella stragrande maggioranza dei casi, si utilizzerà la base STO-3G.

1.2 Seconda quantizzazione

In questo elaborato, così come nella letteratura, si fa spesso uso del formalismo di *seconda quantizzazione*, una formulazione della meccanica quantistica che facilita la descrizione dei sistemi a molti corpi. Se nella *prima* quantizzazione sono le variabili, come posizione e momento, a essere promosse ad operatori quantistici, nella seconda sono i campi stessi a essere *quantizzati*. Senza addentrarsi ulteriormente nei dettagli, l'idea è di considerare le particelle come eccitazioni fondamentali dei campi stessi; in questo senso, l'occupazione degli orbitali viene rappresentata tramite operatori di creazione e annichilazione, che aggiungono o rimuovono particelle dagli stati. Ne consegue che il numero di particelle nel sistema non è fissato, caratteristica che rappresenta l'elemento di novità fondamentale di questa notazione [12].

1.2.1 Spazio di Fock

Considerando questa descrizione, si introduce lo spazio di Fock \mathcal{F} come una generalizzazione dello spazio di Hilbert, che permette di rappresentare stati con numero di particelle variabile. Si pensi innanzitutto agli elementi di \mathcal{F} : gli stati di Fock $|f\rangle$, che possono essere descritti tramite l'occupazione di ciascuno stato di singola particella. Se $n_\alpha \in \mathbb{N}$ è il numero di particelle in $|\alpha\rangle$ con $\alpha = 1, 2, \dots$ che indicizza lo stato

$$|f\rangle = |n_1 n_2 n_3 \dots\rangle \quad (1.27)$$

Nel caso fermionico, uno stato $|\alpha\rangle$, caratterizzato da un proprio set di numeri quantici, può essere occupato al più da una particella, perciò: $n_\alpha \in \{0, 1\}$. Ad esempio, si considerino gli orbitali di spin (eq. 1.15): per indicare comodamente lo stato di occupazione dei ϕ_i si può adottare la notazione:

$$|jk \dots l\rangle \equiv |\phi_j \phi_k \dots \phi_l\rangle \quad (1.28)$$

Insomma, lo spazio di Fock \mathcal{F} è lo spazio in cui vivono tutti i possibili stati di Fock $|f\rangle$, cioè tutte le configurazioni generabili a partire dal set dei $\{\phi_i\}$; se si hanno $2K$ orbitali di spin, si avrà che $\dim(\mathcal{F}) = 2^{2K}$.

1.2.2 Operatori fermionici di creazione e distruzione

Se con $|0\rangle$ si indica lo *stato di vuoto*, si possono introdurre gli operatori creazione e distruzione:

$$\begin{cases} a_i^\dagger |0\rangle = |i\rangle \\ a_i |j\rangle = \delta_{ij} |0\rangle \end{cases} \quad (1.29)$$

a_i^\dagger agisce sullo stato di vuoto *creando* un elettrone nello stato $|i\rangle$; a_i agisce su uno stato occupato $|j\rangle$ *distruggendo* l'elettrone quando $j = i$, altrimenti restituisce zero.

Equivalentemente, sullo stato $|jk \dots l\rangle$:

$$\begin{cases} a_i^\dagger |jk \dots l\rangle = |ijk \dots l\rangle \\ a_i |ijk \dots l\rangle = |jk \dots l\rangle \end{cases} \quad (1.30)$$

Su particelle di natura fermionica rispettano le condizioni richieste dalla teoria:

1. **Antisimmetria** L'ordine in cui sono applicati due operatori di creazione è determinante:

$$a_i^\dagger a_j^\dagger |k \dots l\rangle = |ijk \dots l\rangle \quad (1.31a)$$

mentre

$$a_j^\dagger a_i^\dagger |k \dots l\rangle = |jik \dots l\rangle = -|ijk \dots l\rangle \quad (1.31b)$$

riassumendo:

$$a_i^\dagger a_j^\dagger |0\rangle = -a_j^\dagger a_i^\dagger |0\rangle = |ij\rangle \quad (1.31c)$$

2. **Esclusione** Non è possibile creare due elettroni nello stesso stato $|i\rangle$

$$a_i^\dagger a_i^\dagger |0\rangle = (a_i^\dagger)^2 |0\rangle = 0 \quad (1.32)$$

Con queste premesse si può definire l'operatore numero:

$$\hat{n}_i = a_i^\dagger a_i \quad (1.33)$$

i cui autovalori sono 0 e 1:

$$\hat{n}_i |ijk \dots l\rangle = |ijk \dots l\rangle \hat{n}_i |jk \dots l\rangle = 0 \quad (1.34)$$

in altri termini, quando \hat{n}_i agisce su uno stato di Fock restituisce zero se lo stato $|i\rangle$ non è occupato, altrimenti restituisce lo stato di Fock stesso.

1.2.3 Particelle non interagenti

Si prenda un sistema di fermioni identici non interagenti. Supponendo di averne M , si può scrivere l'hamiltoniana del sistema

$$\hat{H} = \sum_i^M \frac{|\hat{p}_i|^2}{2m} \quad (1.35)$$

Con il formalismo di seconda quantizzazione è possibile scrivere \hat{H} senza specificare il numero di particelle; si possono ad esempio indicizzare i momenti k , sommando sui livelli energetici $\frac{k^2}{2m}$ pesati dal numero di particelle:

$$\hat{H} = \sum_k \frac{k^2}{2m} a_k^\dagger a_k \quad (1.36)$$

un'altra opzione è considerare l'hamiltoniana di singola particella \hat{h} e supporre che possa essere diagonalizzata in questo modo:

$$\hat{h} = \sum_i \epsilon_i |i\rangle \langle i| \quad (1.37)$$

che permette di scrivere:

$$\hat{H} = \sum_i \epsilon_i a_i^\dagger a_i \quad (1.38)$$

qui $a_i^\dagger a_i$ restituisce l'occupazione dello stato $|i\rangle$. L'hamiltoniana in sostanza è la somma delle possibili energie ϵ_i pesate dal numero di particelle.

1.2.4 Hamiltoniana di seconda quantizzazione

La forma dell'Hamiltoniana elettronica, che descrive il problema di interesse in questo lavoro, deriva direttamente dalle considerazioni esposte finora [1].

$$H_{el} = \sum_{p,q} g_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{p,q,r,s} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \quad (1.39)$$

Dove si sono introdotti due nuovi oggetti:

\mathbf{g}_{pq} è l'integrale a un elettrone, che contiene l'energia cinetica e l'interazione coulombiana con i nuclei

$$g_{pq} \equiv \langle p | \hat{T}_e + U_{en}(\underline{r}) | q \rangle = \int \phi_p^*(\underline{x}) \left(\frac{\underline{p}^2}{2m_e} - \sum_A \frac{Z_A}{|\underline{R}_A - \underline{r}|} \right) \phi_q(\underline{x}) d\underline{x} \quad (1.40a)$$

\mathbf{h}_{pqrs} è l'integrale a due elettroni, dovuto alla repulsione elettrone-elettrone

$$h_{pqrs} \equiv \langle pq | U_{ee}(\underline{r}_1, \underline{r}_2) | rs \rangle = \int \frac{\phi_p^*(\underline{x}_1) \phi_q^*(\underline{x}_2) \phi_r(\underline{x}_1) \phi_s(\underline{x}_2)}{|\underline{r}_1 - \underline{r}_2|} d\underline{x}_1 d\underline{x}_2 \quad (1.40b)$$

Poiché H_{el} contiene somme con fino a quattro operatori creazione e distruzione indicizzati separatamente, se si andasse a costruirla a partire da $2K$ orbitali di spin, si otterrebbe un numero $\mathcal{O}(2K^4)$ di termini.

1.3 Metodi classici post Hartree-Fock

L'approssimazione di campo medio introduce un importante errore sistematico poiché, durante il suo moto, un elettrone non interagisce con una semplice distribuzione uniforme, ma viene istantaneamente influenzato dalla repulsione coulombiana dovuta a ciascuno degli altri. Questo comportamento è noto come *correlazione elettronica* ed è considerato la principale limitazione del metodo di Hartree-Fock (sez. 1.1.3) [8]. La correlazione elettronica porta ad una diminuzione significativa dell'energia repulsiva tra gli

elettroni, determinando un'energia elettronica minore di quella prevista da HF; diventa perciò necessario introdurre delle correzioni che possano prendere in considerazione questo fenomeno.

I metodi sviluppati per descrivere correttamente le interazioni tra elettroni sono detti *post-Hartree-Fock*. Tra questi, *Configuration Interaction* (CI) e *Coupled Cluster* (CC) spiccano, per ragioni diverse, in molte applicazioni della chimica quantistica.

1.3.1 CI: Configuration Interaction

L'idea alla base della *configuration interaction* è migliorare la funzione d'onda HF aggiungendo termini che rappresentano la promozione degli elettroni dagli orbitali occupati a quelli virtuali. Ciascuno dei termini nella funzione d'onda CI rappresenta una configurazione elettronica specifica e la struttura elettronica del sistema viene interpretata come il risultato dell'interazione tra di esse.

Si supponga di studiare un problema molecolare con M elettroni attraverso il metodo di Hartree-Fock, da cui si ottiene il set di $2K$ orbitali di spin $\{|\phi_i\rangle\}$. La funzione d'onda HF $|\Phi_{\text{HF}}\rangle$ è il determinante di Slater costruito a partire dagli M orbitali di spin associati a energie minori, ma combinando diversamente i $\{|\phi_i\rangle\}$ - includendo quelli rimasti liberi - si possono ottenere molti altri determinanti a M particelle, detti *configuration state functions* (CSF). Ad esempio, con $|\Phi_a^r\rangle$ si indica il determinante di singola eccitazione, che differisce da $|\Phi_{\text{HF}}\rangle$ poiché l'orbitale di spin $|\phi_a\rangle$ viene sostituito da $|\phi_r\rangle$. Allo stesso modo, si possono definire il determinante di doppia eccitazione $|\Phi_{ab}^{rs}\rangle$, di terza $|\Phi_{abc}^{rst}\rangle$ e così via gli altri. Insieme, queste funzioni d'onda a molti elettroni possono essere usate per espandere la soluzione esatta del problema elettronico [8]:

$$|\Psi\rangle = c_0 |\Phi_{\text{HF}}\rangle + \sum_{a,r} c_a^r |\Phi_a^r\rangle + \sum_{\substack{a<b \\ r<s}} c_{ab}^{rs} |\Phi_{ab}^{rs}\rangle + \sum_{\substack{a<b<c \\ r<s<t}} c_{abc}^{rst} |\Phi_{abc}^{rst}\rangle + \dots \quad (1.41a)$$

o, sfruttando il formalismo di seconda quantizzazione:

$$|\Psi\rangle = c_0 |\Phi_{\text{HF}}\rangle + \sum_{a,r} c_a^r a_a a_r^\dagger |\Phi_{\text{HF}}\rangle + \sum_{\substack{a<b \\ r<s}} c_{ab}^{rs} a_a a_b a_r^\dagger a_s^\dagger |\Phi_{\text{HF}}\rangle + \sum_{\substack{a<b<c \\ r<s<t}} c_{abc}^{rst} a_a a_b a_c a_r^\dagger a_s^\dagger a_t^\dagger |\Phi_{\text{HF}}\rangle + \dots \quad (1.41b)$$

una volta costruita la $|\Psi\rangle$, parametrica nei coefficienti c , si può calcolare l'energia tramite il principio variazionale. Se tutte le CSF possibili vengono considerate il metodo CI fornisce risultati estremamente accurati e prende il nome *Full Configuration Interaction* (FCI); purtroppo, anche partendo da una base $\{|\phi_i\rangle\}$ incompleta, FCI diventa rapidamente intrattabile al crescere della complessità della molecola, rimanendo applicabile soltanto a sistemi molto semplici. Una comune approssimazione per ridurre il costo computazionale di FCI è la *Configuration Interaction Singles Doubles* (CISD), che consiste nel troncare l'espansione (1.41) alle sole CSF di singola e doppia eccitazione; alcune sue varianti sono CISDT e CISDTQ, che aggiungono a CISD i determinanti di tripla e quadrupla eccitazione [2]. Altre implementazioni frequenti di CI sono i *Multi-Configurational SCF* (MCSCF), in cui si vanno ad ottimizzare anche i coefficienti LCAO

(eq. 1.23), e la variante *Complete Active Space SCF* (CASSCF), in cui si selezionano soltanto gli orbitali molecolari che contribuiscono maggiormente alla correlazione, gli *active orbitals*, per formare un *active space* di dimensioni ridotte.

Pur rappresentando delle alternative valide, le approssimazioni troncate di CI presentano un problema pratico che non può essere eluso: non soddisfano una proprietà fondamentale, la *size consistency*, cruciale per la corretta descrizione dei sistemi complessi.

Size Consistency

Un metodo computazionale è detto *size-consistent* se permette di fattorizzare la funzione d'onda di un sistema - composto da sottosistemi non interagenti - come prodotto delle funzioni d'onda degli elementi che lo compongono, così da poter scrivere l'energia totale come la somma dei singoli frammenti [18]. Ad esempio, per due sistemi A e B infinitamente lontani, come gli atomi dissociati di una molecola, varrebbero le relazioni:

$$\begin{cases} \psi_{AB} = |\psi_A \psi_B\rangle \\ E_{AB} = E_A + E_B \end{cases} \quad (1.42)$$

Dal momento che le tecniche CI appena discusse non godono di questa proprietà, diventa prioritario trovare un metodo con un costo computazionale contenuto che soddisfi anche la condizione di *size consistency*.

1.3.2 CC: Coupled Cluster

I metodi *coupled cluster*, inizialmente introdotti nel 1960, godono della proprietà di *size consistency* e sono attualmente considerati il *golden standard* dei metodi computazionali per via del bilanciamento tra accuratezza e efficienza che offrono [18].

L'idea alla base della teoria CC è descrivere la funzione d'onda di stato fondamentale applicando un operatore d'eccitazione \hat{T} esponenziato su un determinante di riferimento, tipicamente lo stato di Hartree-Fock $|\Phi_{\text{HF}}\rangle$.

$$|\Phi_{\text{CC}}\rangle = e^{\hat{T}} |\Phi_{\text{HF}}\rangle \quad (1.43)$$

\hat{T} è detto *operatore di cluster* e la sua azione, in analogia con il metodo CI, è quella di restituire una combinazione lineare di determinanti. Nello specifico, \hat{T} è la somma degli operatori relativi ai diversi ordini di eccitazione $\hat{T} = \hat{T}_1 + \hat{T}_2 + \dots$, dove:

$$\hat{T}_1 = \sum_{a,r} \theta_a^r a_r^\dagger a_a \quad (1.44a)$$

$$\hat{T}_2 = \sum_{\substack{a < b \\ r < s}} \theta_{ab}^{rs} a_r^\dagger a_s^\dagger a_a a_b \quad (1.44b)$$

Tra gli aspetti vantaggiosi di CC vi è la possibilità di troncature sistematicamente \hat{T} ad ordini minori; a seconda di quanti termini si includono nella somma si ottengono le varianti CCS, CCD, CCSD e così via:

$$|\Phi_{\text{CCS}}\rangle = e^{\hat{T}_1} |\Phi_{\text{HF}}\rangle \quad (1.45a)$$

$$|\Phi_{\text{CCD}}\rangle = e^{\hat{T}_2} |\Phi_{\text{HF}}\rangle \quad (1.45b)$$

$$|\Phi_{\text{CCSD}}\rangle = e^{\hat{T}_1 + \hat{T}_2} |\Phi_{\text{HF}}\rangle \quad (1.45c)$$

I vantaggi della teoria *coupled cluster* hanno un prezzo: il singolo determinante di riferimento rende i metodi CC non variazionali, ma la *size consistency* è ritenuta una questione di maggiore importanza [3].

Vi è una variante significativa di CC, la *Unitary Coupled Cluster* (UCC), che consiste nel modificare la trasformazione $e^{\hat{T}}$ in modo da renderla unitaria:

$$|\Phi_{\text{UCC}}\rangle = e^{\hat{T} - \hat{T}^\dagger} |\Phi_{\text{HF}}\rangle \quad (1.46)$$

quest'ultima è di particolare interesse perché, come verrà trattato in seguito (sez. 2.3), può essere mappata in modo naturale su un circuito quantistico [16].

2. Soluzioni computazionali quantistiche

Dopo aver esplorato il problema elettronico e le tecniche classiche, l'emergere dei computer quantistici apre nuove prospettive per affrontare i sistemi chimici più complessi. In questo capitolo, si introducono i concetti fondamentali della computazione quantistica, come *qubit*, *gates* e *circuiti*, gettando le basi per approcci innovativi come gli algoritmi variazionali quantistici.

2.1 Introduzione al quantum computing

2.1.1 Qubit

Un *qubit*, è un qualsiasi sistema fisico a due livelli, generalmente indicati con $|0\rangle$ e $|1\rangle$ e rappresenta l'unità fondamentale di informazione del quantum computing. Mentre un *bit* classico può assumere soltanto due valori: 0 oppure 1, un *quantum bit* può esistere in una sovrapposizione qualsiasi dei due stati [7, 13]:

$$|q\rangle = \alpha |0\rangle + \beta |1\rangle \quad (2.1)$$

dove $\alpha, \beta \in \mathbb{C}$ sono le ampiezze di probabilità dei due stati e pertanto soddisfano la condizione:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (2.2)$$

Sfera di Bloch

Attraverso semplici manipolazioni algebriche, rispettando la condizione di unitarietà (eq. 2.2) e considerando che una fase globale non ha un effetto fisico visibile, è possibile riscrivere la 2.1:

$$|q\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle \quad (2.3)$$

questa relazione offre una comoda rappresentazione, che permette di visualizzare il qubit in maniera intuitiva: le variabili θ e ϕ individuano un punto nello spazio posizionato sulla superficie di una sfera unitaria, detta *sfera di Bloch*.

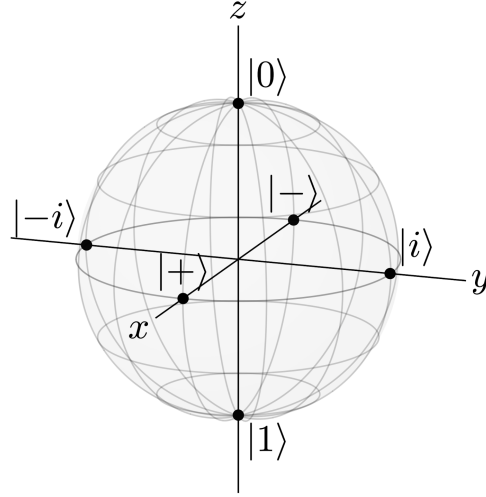


Figura 2.1: Sfera di Bloch [20]

I punti ai poli della sfera sono identificati dagli stati $|0\rangle$, $|1\rangle$ che insieme formano la cosiddetta *base computazionale* $\{|0\rangle, |1\rangle\}$. Allo stesso modo, i punti agli antipodi sugli assi x e y corrispondono ciascuno ad una base bidimensionale, spesso indicate con $\{|+\rangle, |-\rangle\}$ e $\{|+i\rangle, |-i\rangle\}$, i cui elementi possono essere sviluppati sulla base computazionale:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad (2.4a)$$

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \qquad (2.4b)$$

$$|+i\rangle = \frac{|0\rangle + i|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix} \qquad |-i\rangle = \frac{|0\rangle - i|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix} \qquad (2.4c)$$

Informazione di un qubit

Per caratterizzare univocamente uno stato $|q\rangle$ occorrono due numeri reali indipendenti, che lo collocano in uno degli infiniti punti sulla sfera di Bloch; si potrebbe allora pensare che un qubit possa memorizzare una quantità di informazioni virtualmente illimitata. Tuttavia, per la natura quantistica del sistema, non è possibile estrarre i coefficienti α e β con un numero finito di misure: quando si interroga una prima volta lo stato di sovrapposizione $|q\rangle$, esso collassa in uno stato definito, $|0\rangle$ o $|1\rangle$, ciascuno con una probabilità data dal modulo quadro della sua ampiezza, $|\alpha|^2$ o $|\beta|^2$. Da quel momento in poi, ogni misurazione successiva sullo stesso qubit restituirà sempre lo stesso risultato.

2.1.2 Operazioni sui qubit

Sui bit classici le operazioni vengono implementate tramite *logic gates*, quali AND, OR e NOT, e la loro azione viene schematizzata attraverso dei circuiti. Analogamente, le trasformazioni sui qubit sono compiute attraverso *quantum gates*, generalmente indicati con la lettera U , e vengono riassunte con dei *circuiti quantistici*.

$$|q\rangle \longrightarrow \boxed{U} \longrightarrow |q'\rangle$$

Chiaramente, l'applicazione di un gate quantistico U ad uno stato $|q\rangle$ deve restituire uno stato $|q'\rangle$ valido: la condizione 2.2 deve rimanere soddisfatta. U , insomma, agisce sui qubit come un operatore *unitario* ($UU^\dagger = \mathbb{I}$) e può essere anche espresso in forma matriciale. Di seguito vengono esposti alcuni dei principali *quantum gates*.

X gate (NOT)

$$\longrightarrow \boxed{X} \longrightarrow$$

La sua rappresentazione matriciale è:

$$X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (2.5)$$

è considerato l'analogo del NOT gate classico, perché agisce sullo stato $|1\rangle$ mandandolo in $|0\rangle$ e viceversa:

$$|0\rangle \longrightarrow \boxed{X} \longrightarrow |1\rangle$$

$$|1\rangle \longrightarrow \boxed{X} \longrightarrow |0\rangle$$

mentre, nel caso più generale, inverte le probabilità dei due stati:

$$\alpha |0\rangle + \beta |1\rangle \longrightarrow \boxed{X} \longrightarrow \beta |0\rangle + \alpha |1\rangle$$

Y gate

$$\longrightarrow \boxed{Y} \longrightarrow$$

In forma matriciale è:

$$Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad (2.6)$$

ed agisce sul generico stato $|q\rangle$ in questo modo:

$$\alpha |0\rangle + \beta |1\rangle \longrightarrow \boxed{Y} \longrightarrow -i\beta |0\rangle + i\alpha |1\rangle$$

Z gate

$$\longrightarrow \boxed{Z} \longrightarrow$$

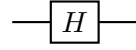
Con forma matriciale:

$$Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (2.7)$$

e azione sul generico stato $|q\rangle$:

$$\alpha |0\rangle + \beta |1\rangle \longrightarrow \boxed{Z} \longrightarrow \alpha |0\rangle - \beta |1\rangle$$

Hadamard gate



Con forma matriciale:

$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (2.8)$$

si distingue dai precedenti tre perché manda i vettori della base computazionale nelle sovrapposizioni $|+\rangle$ e $|-\rangle$ definite nelle relazioni 2.4:

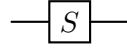
$$|0\rangle \xrightarrow{H} |+\rangle$$

$$|1\rangle \xrightarrow{H} |-\rangle$$

più in generale:

$$\alpha |0\rangle + \beta |1\rangle \xrightarrow{H} \alpha |+\rangle + \beta |-\rangle$$

S gate



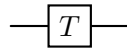
S è spesso indicato come la radice quadrata di Z , poiché vale la relazione $S^2 = Z$. In termini matriciali si scrive:

$$S \equiv \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad (2.9)$$

ed ha il seguente effetto su $|q\rangle$:

$$\alpha |0\rangle + \beta |1\rangle \xrightarrow{S} \alpha |0\rangle + i\beta |1\rangle$$

T gate



T è spesso indicato come la radice quadrata di S , poiché vale la relazione $T^2 = S$. In termini matriciali si scrive:

$$T \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} \quad (2.10)$$

e la sua azione è:

$$\alpha |0\rangle + \beta |1\rangle \xrightarrow{T} \alpha |0\rangle + e^{i\frac{\pi}{4}} \beta |1\rangle$$

CX gate (CNOT)

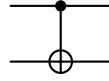
Tutti gli operatori presentati finora agiscono su un singolo qubit, determinandone in qualche modo una rotazione sulla sfera di Bloch (fig. 2.1), ma è possibile introdurre anche trasformazioni che coinvolgono più qubit.

Il gate CX , anche detto *Controlled NOT* (CNOT), è particolarmente rilevante per la sua capacità di sfruttare un fenomeno largamente impiegato nel *quantum computing*: l'*entanglement*.

In breve, CX applica un NOT gate su un qubit soltanto se un secondo qubit, detto di *controllo*, si trova in $|1\rangle$. La sua azione si può allora riassumere con le seguenti relazioni:

$$|00\rangle \rightarrow |00\rangle; |01\rangle \rightarrow |01\rangle; |10\rangle \rightarrow |11\rangle; |11\rangle \rightarrow |10\rangle \quad (2.11)$$


Viene indicato con:



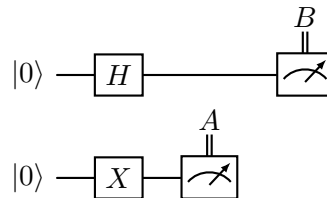
e si può rappresentare con una matrice 4×4 :

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (2.12)$$

Circuiti quantistici

Un *quantum circuit* è una rappresentazione grafica che schematizza le operazioni applicate a un determinato set di qubit. Questi ultimi vengono raffigurati con delle linee orizzontali, che si dipartono da uno stato iniziale posto sulla sinistra; gli operatori sono quindi applicati in ordine procedendo verso destra. L'azione di misura viene indicata con  e il suo risultato può essere registrato su un bit classico legato tramite un *wire*, ritratto come una linea doppia.

Si veda ad esempio il seguente semplice circuito:



Ciascun qubit subisce separatamente una trasformazione e solo successivamente vengono effettuate le misure, i cui risultati sono salvati in due *classical bits* A e B . L'azione di X garantisce che A sia 1, mentre H fa sì che il bit B abbia la stessa probabilità di trovarsi in 0 o in 1. Per ottenere informazioni sulle ampiezze di probabilità finali occorre eseguire più volte il circuito; in questo modo, si osserverà che le combinazioni $[10]$ e $[11]$ compaiono approssimativamente con la medesima frequenza.

La complessità di un circuito quantistico è data dalla sua *profondità* o *depth*, ovvero il numero massimo di operazioni consecutive applicate a un qubit lungo il circuito. Maggiore è la profondità, più tempo richiederà l'esecuzione, aumentando la vulnerabilità del circuito ai fattori di rumore che agiscono durante il calcolo.

Limiti hardware

In linea teorica è possibile progettare un circuito quantistico con centinaia di qubit e migliaia di operazioni, ma le QPU attualmente disponibili, pur avendo superato le centinaia di qubit, soffrono ancora di limitazioni significative in termini di affidabilità e precisione operativa. Per ogni qubit o gate inserito in un circuito viene introdotta una certa probabilità di errore, che può accumularsi rapidamente nei calcoli complessi, riducendo drasticamente la fedeltà del risultato. Il fenomeno della *decoerenza* rappresenta una delle principali fonti di errore: i qubit, durante l'elaborazione, possono perdere informazioni a causa delle interazioni con l'ambiente esterno, provocando la perdita di informazione sulle fasi relative. Anche l'accuratezza dei singoli gate, ovvero la capacità di ogni operazione di riprodurre sistematicamente il medesimo risultato, è un fattore limitante: ogni operazione possiede un margine di errore, legato sia alle imperfezioni nell'hardware, sia a fattori ambientali. Inoltre, l'operazione di misura è intrinsecamente soggetta a rumore, che può compromettere la lettura corretta dello stato finale del qubit. Per tutti questi motivi, è necessario limitare il numero di gate nei circuiti quantistici e utilizzare tecniche di correzione dell'errore per migliorare la precisione dei calcoli, anche se tali metodi aumentano la complessità complessiva dei circuiti. Tra gli obiettivi di questo elaborato vi è proprio il confronto tra diversi circuiti quantistici, ciascuno caratterizzato da una profondità e complessità variabile. L'analisi infatti si concentrerà su come la scelta e la struttura dei *quantum circuits* impiegati influiscano sulla qualità dei risultati, tenendo conto delle limitazioni imposte dagli errori e dalle risorse computazionali disponibili.

2.1.3 Qubit Mapping

Con una maggiore comprensione del problema elettronico e dei principi fondanti del calcolo quantistico, si può ora affrontare il passaggio cruciale per trasporre la descrizione dei sistemi elettronici in termini di qubit. Questo permette di creare un ponte tra il formalismo della chimica quantistica e la struttura dei circuiti quantistici, rendendo possibile la simulazione su hardware quantistico [14, 18].

Per compiere questa traduzione, è necessario un metodo di codifica che colleghi lo spazio di Fock \mathcal{F} del formalismo di seconda quantizzazione (sez. 1.2) allo spazio di Hilbert in cui vivono i qubit. Ricordando che $\dim(\mathcal{F}) = 2^{2K}$, con K numero degli orbitali spaziali, è naturale pensare di utilizzare $2K$ qubit per costruire uno spazio di Hilbert $\mathcal{H}^{\otimes 2K} = \bigotimes_{k=1}^{2K} \mathcal{H}_k$ che possieda lo stesso numero totale di configurazioni possibili. Qui si è indicato con \mathcal{H}_k lo spazio del singolo qubit $|q_k\rangle$ che, se combinato agli altri, dà:

$$|Q\rangle = \bigotimes_{k=1}^{2K} |q_k\rangle = |q_{2K} q_{2K-1} \dots q_1\rangle, \quad \forall |Q\rangle \in \mathcal{H}^{\otimes 2K} \quad (2.13)$$

A questo punto, si può sviluppare un isomorfismo capace di mappare gli stati di Fock $|f\rangle$ in uno stato dello spazio dei qubit $|Q\rangle$:

$$\begin{aligned} \mathfrak{f} : \mathcal{F} &\rightarrow \mathcal{H}^{\otimes 2K} \\ |f\rangle &\mapsto |Q\rangle \end{aligned} \quad (2.14)$$

a buon ragione, una trasformazione di questo tipo è chiamata *qubit mapper*, poiché si occupa sostanzialmente di tradurre gli operatori fermionici a^\dagger e a in operazioni native su un quantum computer, dette *qubit operators*. Esistono diversi sistemi di codifica, di seguito si introduce quello utilizzato nel presente elaborato: la trasformazione di Jordan-Wigner.

Isomorfismo di Jordan-Wigner

Quando si utilizza l'isomorfismo, o trasformazione, di Jordan-Wigner (JW) si conserva l'occupazione di ciascun orbitale di spin all'interno di un qubit: $|q\rangle = |0\rangle$ rappresenta uno stato $|\phi\rangle$ vuoto, $|q\rangle = |1\rangle$ uno occupato. Estendendo ad un sistema di $2K$ orbitali e ricordando che n_k indica il numero di particelle in $|\phi_k\rangle$:

$$|n_{2K} n_{2K-1} \dots n_1\rangle \mapsto |q_{2K} q_{2K-1} \dots q_1\rangle \quad \text{con: } q_p = n_p \in \{0, 1\} \quad (2.15)$$

Gli operatori creazione e distruzione agiscono incrementando o decrementando di 1 il numero di occupazione, introducendo una fase dovuta all'ordine di applicazione (eq. 1.31). Se si riprendono le relazioni 1.29 considerando il singolo qubit:

$$\begin{cases} a^\dagger |0\rangle = |1\rangle \\ a^\dagger |1\rangle = 0 \end{cases} \quad \wedge \quad \begin{cases} a |0\rangle = 0 \\ a |1\rangle = |0\rangle \end{cases} \quad (2.16)$$

è immediato trovare le rappresentazioni matriciali:

$$a^\dagger \longleftrightarrow \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \frac{1}{2}(X - iY) \equiv A^\dagger \quad (2.17a)$$

e

$$a \longleftrightarrow \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \frac{1}{2}(X + iY) \equiv A \quad (2.17b)$$

JW quindi definisce:

$$\begin{cases} a_p = A_p \otimes Z_{p-1} \otimes \dots \otimes Z_1 \\ a_p^\dagger = A_p^\dagger \otimes Z_{p-1} \otimes \dots \otimes Z_1 \end{cases} \quad (2.18)$$

dove, intuitivamente, A e A^\dagger agiscono sull'occupazione del p -esimo qubit, mentre la catena di operatori Z restituisce il fattore di fase.

Si noti che con questo *qubit mapper* è stato possibile rappresentare uno stato con $2K$ orbitali di spin in $2K$ qubit, mentre per una soluzione esatta FCI occorrono $\mathcal{O}(2K^M)$ termini; ciò evidenzia chiaramente perché l'uso dei computer quantistici risulti così promettente nel campo della chimica computazionale.

2.2 Variational quantum algorithms

Applicazioni come la simulazione di sistemi quantistici complessi o la risoluzione di problemi di algebra lineare su larga scala risultano molto impegnative per i computer classici a causa dei costi computazionali elevatissimi. I computer quantistici promettono una soluzione, ma gli attuali dispositivi quantistici sono limitati sia dal numero di qubit che dalla presenza di rumore, che riduce la profondità dei circuiti eseguibili [17].

Dalle necessità legate a queste considerazioni sono emersi i Variational Quantum Algorithms (VQA), algoritmi ibridi che utilizzano un ottimizzatore classico per *addestrare* circuiti quantistici parametrizzati. I VQA rappresentano oggi una delle strategie più promettenti per sfruttare le capacità dei dispositivi quantistici, con applicazioni praticamente in ogni campo previsto per i computer quantistici.

Il primo passo nella costruzione di un VQA è la definizione di una funzione costo che sintetizzi la soluzione del problema; quindi si propone un *ansatz*, un'operazione quantistica che dipende da un certo set di parametri $\{\theta\}$ e che possa essere ottimizzata. Infine, marchio di fabbrica dei VQA, si utilizzano computer quantistici per la stima di una funzione costo $C(\theta)$, combinati ad ottimizzatori classici che ne perfezionano i parametri θ .

2.2.1 VQE

Tra le applicazioni più promettenti dei VQA vi è la stima degli autovalori delle hamiltoniane. Prima della loro introduzione, le soluzioni computazionali proposte richiedevano hardware quantistici al di là di quelli disponibili nell'era NISQ, spesso soggetti a errori di gate e con un numero limitato di qubits, che hanno tempi di coerenza ancora piccoli. Il *Variational Quantum Eigensolver* (VQE) è stato sviluppato per offrire un'alternativa a breve termine per questo compito [9].

Lo scopo di VQE è approssimare autostati e autovalori di una data hamiltoniana H ; ciò viene raggiunto definendo la funzione costo da minimizzare come il valore di aspettazione di quest'ultima su uno stato parametrico $|\psi(\theta)\rangle = U(\theta) |\psi_0\rangle$:

$$C(\theta) = \frac{\langle \psi(\theta) | \hat{H} | \psi(\theta) \rangle}{\langle \psi(\theta) | \psi(\theta) \rangle} \quad (2.19)$$

quando questo algoritmo viene applicato al calcolo dell'energia elettronica di una molecola, problema trattato nel presente elaborato, può essere schematizzato nelle seguenti fasi:

1. Mantenendo fissi i nuclei, si calcola l'hamiltoniana del problema elettronico

2. L'ansatz variazionale $|\psi(\theta)\rangle$ della funzione d'onda del sistema viene espresso tramite un circuito quantistico, composto da rotazioni parametrizzate
3. Il computer quantistico calcola il valore di aspettazione di \hat{H} e le sue derivate rispetto ai parametri θ
4. Un computer classico ottimizza i parametri a partire dai risultati ottenuti al punto 3
5. I punti 3 e 4 vengono reiterati fino al raggiungimento di un minimo dell'energia

Grazie ad algoritmi come VQE per la valutazione dell'energia di un'ampia classe di ansatz di funzioni d'onda, il cui calcolo richiederebbe risorse esponenzialmente crescenti su un dispositivo classico, sono sufficienti solo la preparazione dello stato e la misurazione di operatori di Pauli, operazioni che possono essere realizzate su un processore quantistico in tempo polinomiale [11]. Questo vantaggio potrebbe rendere possibile l'implementazione efficiente di ansatz come *unitary coupled cluster*.

2.3 Quantum Coupled Cluster

Riuscire a costruire un ansatz che approssimi lo stato fondamentale è determinante per il successo di algoritmi variazionali quantistici come VQE, allo stesso tempo rimane l'importanza chiave di costruire la funzione d'onda in modo efficiente.

Il metodo *quantum unitary coupled cluster* (q -UCC), ispirato alla sua controparte classica UCC (sez. 1.3.2), offre una soluzione scalabile, oltre che fisicamente motivata. Inoltre, a differenza dei metodi CC classici, l'energia su uno stato q -UCC può essere determinata variazionalmente [18].

Eccitazioni troncate

In maniera del tutto analoga alla variante classica CC (eq. 1.45), q -UCC permette il troncamento sistematico delle eccitazioni. In questo studio si prenderanno in esame le varianti q -UCCS, q -UCCD e q -UCCSD

$$|\Phi_{q\text{-CCS}}\rangle = e^{\hat{T}_1 - \hat{T}_1^\dagger} |\Phi_{\text{HF}}\rangle \quad (2.20a)$$

$$|\Phi_{q\text{-CCD}}\rangle = e^{\hat{T}_2 - \hat{T}_2^\dagger} |\Phi_{\text{HF}}\rangle \quad (2.20b)$$

$$|\Phi_{q\text{-CCSD}}\rangle = e^{\hat{T}_1 - \hat{T}_1^\dagger + \hat{T}_2 - \hat{T}_2^\dagger} |\Phi_{\text{HF}}\rangle \quad (2.20c)$$

2.3.1 Implementazione

Il primo passo nell'implementazione dell'ansatz q -UCC è la costruzione dell'operatore unitario U , presentato già nell'equazione 1.46:

$$U \equiv e^{\hat{T} - \hat{T}^\dagger} = e^{\sum_k \hat{T}_k - \hat{T}_k^\dagger} \quad (2.21)$$

dove \hat{T} è l'operatore di cluster (eq. 1.44). Quindi, U deve essere tradotto in operazioni eseguibili su un dispositivo quantistico.

Approssimazione di Trotter

I vari ordini di eccitazioni \hat{T}_k non commutano tra loro, perciò è necessario ricorrere a delle approssimazioni; in questo caso, si ricorre alla formula di Trotter $e^{\hat{A}+\hat{B}} \approx e^{\hat{A}}e^{\hat{B}}$:

$$U \approx \prod_k e^{\hat{T}_k - \hat{T}_k^\dagger} = \prod_k U_k \quad (2.22)$$

$$e^{\hat{A}+\hat{B}} \approx \lim_{t \rightarrow \inf} \left(e^{\frac{\hat{A}}{t}} e^{\frac{\hat{B}}{t}} \right)^t \quad (2.23)$$

applicato a U

$$U \approx \left(\prod_i e^{\frac{1}{t} \hat{T}_i - \hat{T}_i^\dagger} \right)^t \quad (2.24)$$

ciò permette di scomporre U in una produttoria di operatori unitari U_k , che poi possono essere espressi in termini di quantum gates in modo efficiente, anche se, in generale, questo processo porta a circuiti piuttosto grandi.

Ad esempio, nel caso q -UCCSD si ha:

$$U_{SD} = e^{\hat{T}_1 - \hat{T}_1^\dagger + \hat{T}_2 - \hat{T}_2^\dagger} \quad (2.25)$$

applicando l'approssimazione di Trotter ed esplicitando gli operatori di eccitazione con le relazioni 1.44:

$$\begin{aligned} U_{SD} &= e^{\hat{T}_1 - \hat{T}_1^\dagger} e^{\hat{T}_2 - \hat{T}_2^\dagger} \\ &= \exp \left(\sum_{a,r} \theta_a^r (a_r^\dagger a_a - a_a^\dagger a_r) \right) \exp \left(\sum_{\substack{a < b \\ r < s}} \theta_{ab}^{rs} (a_r^\dagger a_s^\dagger a_a a_b - a_b^\dagger a_a^\dagger a_s a_r) \right) \end{aligned} \quad (2.26)$$

che si può scomporre con Trotter ulteriormente:

$$U_{SD} = \prod_{a,r} \exp \left(\theta_a^r (a_r^\dagger a_a - a_a^\dagger a_r) \right) \prod_{\substack{a < b \\ r < s}} \exp \left(\theta_{ab}^{rs} (a_r^\dagger a_s^\dagger a_a a_b - a_b^\dagger a_a^\dagger a_s a_r) \right) \quad (2.27)$$

A questo punto si può procedere con la mappatura degli operatori fermionici a *qubit operators*, utilizzando un qubit mapper come le trasformazioni di Jordan-Wigner [10].

Circuiti

I circuiti che ne risultano sono:

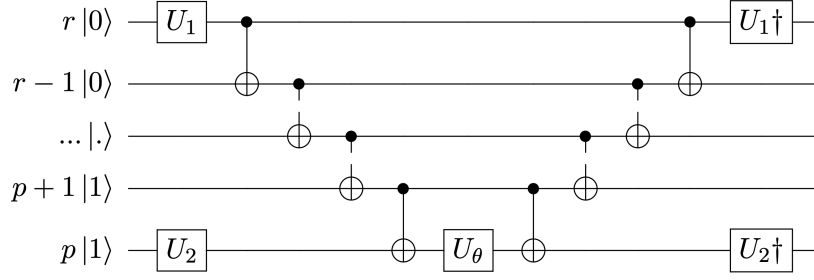


Figura 2.2: Esponenziazione eccitazioni singole

con $(U_1, U_2) \in \{(Y, H), (H, Y)\}$, per l'esponenziazione di $\hat{T}_1 - \hat{T}_1^\dagger$;

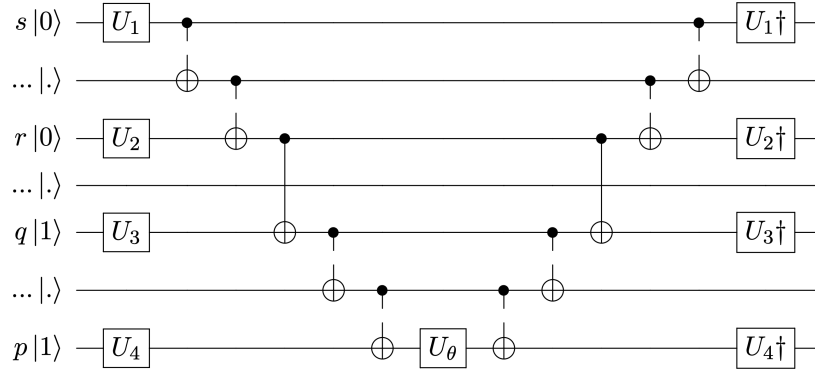


Figura 2.3: Esponenziazione eccitazioni doppie

dove

$$(U_1, U_2, U_3, U_4) \in \{(H, H, Y, H), (Y, H, Y, Y), (H, Y, Y, Y), (H, H, H, Y), \\ (Y, H, H, H), (H, Y, H, H), (Y, Y, Y, H), (Y, Y, H, Y)\}$$

per l'esponenziazione di $\hat{T}_2 - \hat{T}_2^\dagger$.

Per formare l'ansatz UCCSD basterà concatenare questi due circuiti allo stato HF di partenza.

2.3.2 pUCCD: Pair unitary coupled cluster doubles

Nell'approccio pUCCD si selezionano soltanto le eccitazioni doppie e si impone che queste agiscano sempre su coppie di elettroni. Se si studia un sistema con numero pari di elettroni nello stato fondamentale e se ne considerano i K orbitali spaziali, si può definire l'operatore di eccitazione doppia accoppiata:

$$\hat{T}_{\text{pCCD}} = \sum_{a,r} \theta_{a_\alpha a_\beta}^{r_\alpha r_\beta} a_{a_\alpha}^\dagger a_{a_\beta}^\dagger a_{r_\alpha} a_{r_\beta} \quad (2.28)$$

dove a, r stavolta indicizzano gli orbitali spaziali, all'interno dei quali α, β indicano lo spin. La funzione d'onda pUCCD è definita così:

$$|\text{pUCCD}\rangle = e^{\hat{T}_{\text{pCCD}} - \hat{T}_{\text{pCCD}}^\dagger} \quad (2.29)$$

Questo appaiamento determina una sostanziale diminuzione nel numero di parametri variazionali e di gate richiesti, al prezzo di una minore capacità di rappresentare il problema in esame. Esiste tuttavia una strategia correttiva: i risultati ottenuti con pUCCD, in particolare quelli relativi all'energia di dissociazione, migliorano significativamente se corretti tramite *orbital optimization*, con cui si può compensare l'omissione dei termini di eccitazione singola.

oo-pUCCD

L'ottimizzazione orbitale consiste nell'applicare un'operatore di rotazione $\hat{R} = e^{-\mathcal{K}}$, che agisce sugli orbitali di spin $\{\phi_k\}$ [16, 21]

$$|\text{oo-pUCCD}\rangle = e^{-\mathcal{K}} |\text{pUCCD}\rangle \quad (2.30)$$

$\hat{\mathcal{K}}$ è un'operatore *anti-hermitiano* $\hat{\mathcal{K}}^\dagger = -\hat{\mathcal{K}}$, che sostanzialmente rende conto delle eccitazioni singole precedentemente trascurate [15].

$$\hat{\mathcal{K}} = \sum_{p < q} \mathcal{K}_{pq} (a_q^\dagger a_p - a_p^\dagger a_q) \quad (2.31)$$

Il valore di aspettazione diventa:

$$\langle \text{pUCCD} | e^{\mathcal{K}} \hat{H} e^{-\mathcal{K}} | \text{pUCCD} \rangle \quad (2.32)$$

e in seguito alla trasformazione degli orbitali, vengono modificati gli integrali ad uno (eq. 1.40a) e due corpi (eq. 1.40b) utilizzati per generare l'hamiltoniana di seconda quantizzazione:

$$\tilde{g}_{pq} = \sum_{uv} C_{up}^* g_{uv} C_{vq} \quad (2.33a)$$

$$\tilde{h}_{pqrs} = \sum_{uvxy} C_{up}^* C_{vq}^* h_{uvxy} C_{xr} C_{ys} \quad (2.33b)$$

dove si è indicato $C_{pq} = [e^{\hat{\mathcal{K}}}]_{pq}$. Per determinare il set di $\{\mathcal{K}_{pq}\}$ che minimizzano l'energia 2.32 si costruisce il circuito oo-pUCCD concatenando a pUCCD l'operatore $e^{\hat{\mathcal{K}}}$, seguendo gli stessi passaggi descritti alla sezione 2.3. Dunque si procede con la minimizzazione variazionale tramite VQE (sez. 2.2.1) ma stavolta, dopo ciascuna ottimizzazione dei parametri, vengono ricalcolati gli integrali utilizzando le relazioni 2.33 e l'hamiltoniana viene trasformata di conseguenza.

2.4 Hardware efficient ansatzes

UCC rientra nella categoria degli ansatzes *problem inspired*, costruiti a partire da considerazioni sulla fisica del problema da risolvere. Funzioni d'onda di questo tipo promettono di fornire risultati soddisfacenti, ma possono risultare impossibili da implementare, nelle condizioni attuali, per applicazioni al di là delle semplici dimostrazioni [19].

L'obiettivo primario degli ansatze *hardware efficient* è minimizzare profondità e numero di operazioni del circuito per rispondere alle limitazioni dei dispositivi NISQ. Forme variazionali di questo tipo sacrificano l'attinenza con la teoria fisica sottostante, perciò vengono dette *problem agnostic*.

Ansatz rotazionali [22]

Un esempio di forme variazionali *hardware efficient* sono gli *ansatz rotazionali*, così chiamati perché i loro circuiti sono costituiti da operazioni di rotazione su singolo qubit, alternate ad operazioni di entanglement. Un semplice schema per questo ansatz si basa sulle rotazioni parametriche $R_x(\theta)$, $R_y(\theta)$ e $R_z(\theta)$, generate attraverso l'esponenziazione delle matrici X , Y e Z introdotte in 2.1.2

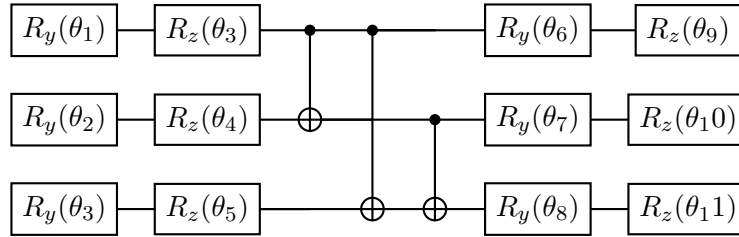
$$R_x \equiv e^{-iX\frac{\theta}{2}} = \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix} \quad (2.34a)$$

$$R_y \equiv e^{-iY\frac{\theta}{2}} = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix} \quad (2.34b)$$

$$R_z \equiv e^{-iZ\frac{\theta}{2}} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \quad (2.34c)$$

poiché queste rotazioni appartengono al gruppo delle matrici speciali unitarie di rango 2 [SU(2)], rappresentate da matrici 2×2 unitarie con determinante 1, gli ansatze rotazionali di questo tipo vengono soprannominati EfficientSU(2).

Un esempio di circuito EfficientSU(2) su 3 qubit che impiega soltanto R_y e R_z potrebbe essere:



Per migliorare la convergenza, è possibile aumentare i parametri circuitali ripetendo questo insieme di rotazioni ed entanglement un numero D arbitrario di volte, al prezzo di una maggiore profondità.

3. Simulazioni

La maggior parte delle simulazioni è fatta sulla molecola di LiH, relativamente semplice, ... si va a calcolare l'energia di stato fondamentale al variare della distanza per tracciare il profilo del potenziale elettronico. Il suo minimo si attesta attorno a 1.595 Å, come corroborato anche da evidenze sperimentali [24].

3.1 Introduzione al codice

In questa sezione viene descritto il codice utilizzato per le simulazioni, con particolare riferimento all'ambiente offerto da Qiskit, una delle piattaforme più diffuse per la programmazione quantistica. Qiskit fornisce strumenti avanzati per la creazione, simulazione e ottimizzazione di circuiti quantistici, fondamentali per affrontare problemi di chimica quantistica come la determinazione degli stati fondamentali e la simulazione di reazioni.

Si presenta una panoramica generale su come realizzare simulazioni tramite il Variational Quantum Eigensolver (VQE) in Qiskit, illustrando i passaggi essenziali e le scelte metodologiche chiave. Inoltre, vengono approfonditi alcuni concetti specifici, come la scelta delle basi e degli ottimizzatori, elementi centrali nella configurazione delle simulazioni quantistiche. La selezione della base influenza direttamente la rappresentazione della molecola, determinando sia il numero di qubit richiesti che l'accuratezza dei risultati. Basi meno complesse, come la STO-3G, vengono spesso utilizzate per ridurre il costo computazionale, mentre basi più articolate, come la 6-31G, possono migliorare la precisione dei risultati a scapito di una maggiore complessità. Gli ottimizzatori, d'altra parte, influenzano il processo di minimizzazione dell'energia e la velocità con cui si raggiunge la convergenza. A seconda del problema e delle caratteristiche della simulazione, è possibile scegliere tra diversi *optimizers* presto implementabili con Qiskit, come SPSA, COBYLA e SPSQL, ciascuno con vantaggi specifici per scenari diversi.

3.1.1 Impostazione del problema

Il primo passo è implementare il sistema molecolare all'interno del codice attraverso `PySCFDriver`, che fa da ponte fra le classi di Qiskit e la libreria di chimica PySCF. Come argomenti si inseriscono gli attributi della molecola, quindi si esegue il metodo `.run()` per estrarre un `ElectronicStructureProblem`: oggetto che rappresenta il problema elettronico.

Dichiarazione molecola

```

1 LiH_geometry = f"""Li 0. 0. 0.; H 0. 0. {distance}"""
2
3 driver = PySCFDriver(
4     atom    = LiH_geometry,
5     basis   = 'sto3g',
6     charge  = 0,
7     spin    = 0,
8     unit    = DistanceUnit.ANGSTROM
9 )
10
11 problem = driver.run()

```

Con `FreezeCoreTransformer` si possono congelare gli orbitali *core* della molecola, il che permette di ridurre la dimensione del problema senza perdere precisione nella descrizione. Il contributo degli orbitali esclusi diventa quindi uno shift all'energia totale.

Congelamento orbitali core e hamiltoniana

```

1 fc_transformer = FreezeCoreTransformer()
2 fc_problem     = fc_transformer.transform(problem)

```

Scelta della base

Ad oggi sono state introdotte numerose basi per rappresentare gli orbitali molecolari, alcune delle quali permettono di ottenere accuratissime elevate, anche se il costo computazionale cresce rapidamente all'aumentare del numero di orbitali atomici. Le basi STO-*n*G vengono principalmente impiegate per studi preliminari e approssimativi, mentre calcoli più dettagliati richiedono basi più articolate. Un esempio di queste è 6-31G, una base di tipo *split-valence* che combina sei funzioni gaussiane per il *core* e due serie di gaussiane per gli elettroni di valenza, migliorando la descrizione degli orbitali.

Di seguito è riportato un grafico che confronta i risultati di FCI nelle basi STO-3G, STO-6G e 6-31G.

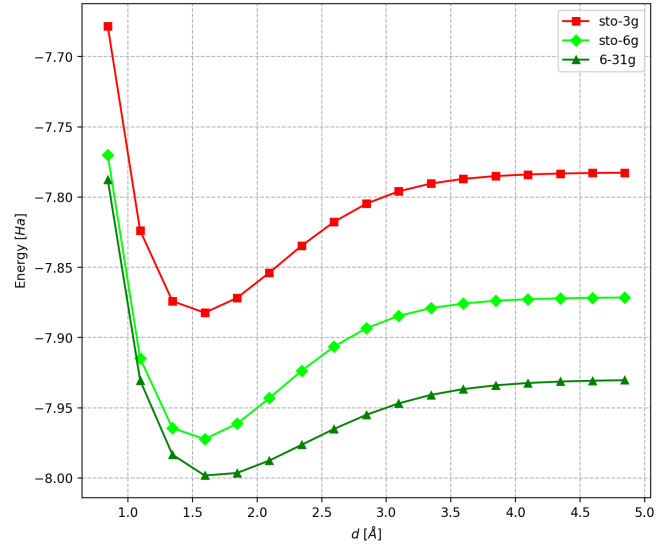


Figura 3.1: FCI con diverse basi

Le STO- n G sovrastimano l'energia in modo marcato ma richiedono tempi di calcolo notevolmente inferiori rispetto a 6-31G. In questo lavoro l'obiettivo è confrontare il costo di diversi circuiti quantistici, per cui è sufficiente lavorare con la base STO-3G.

3.1.2 Ansatz

Qiskit offre un'ampia scelta di ansatz UCC [23], implementabili con poche righe di codice, qui sotto esemplificate per un generico ordine di eccitazioni X . Si veda ad esempio il circuito q -UCCS, riportato qui a destra, generato dalla classe `UCCS` per le analisi della molecola di LiH. Innanzitutto, si può notare che la base STO-3G utilizzata per definire il problema dà luogo ad un circuito con 10 qubit, che per motivi di lunghezza viene suddiviso in quattro sezioni. Come illustrato in 1.3.2, la funzione d'onda *coupled cluster* è costruita a partire da uno stato di Hartree-Fock; i qubit 0 e 5, a cui vengono applicate delle rotazioni U , corrispondono agli stati occupati, mentre gli altri rappresentano orbitali *virtuali*. Dopodiché si può osservare che i gate successivi sono applicati seguendo il pattern presentato in figura 2.2 per riprodurre le eccitazioni singole. È importante sottolineare che, tra i circuiti UCC utilizzati, UCCS è il meno costoso in termini di risorse, con UCCD e UCCSD che possono arrivare a profondità trenta volte maggiori.

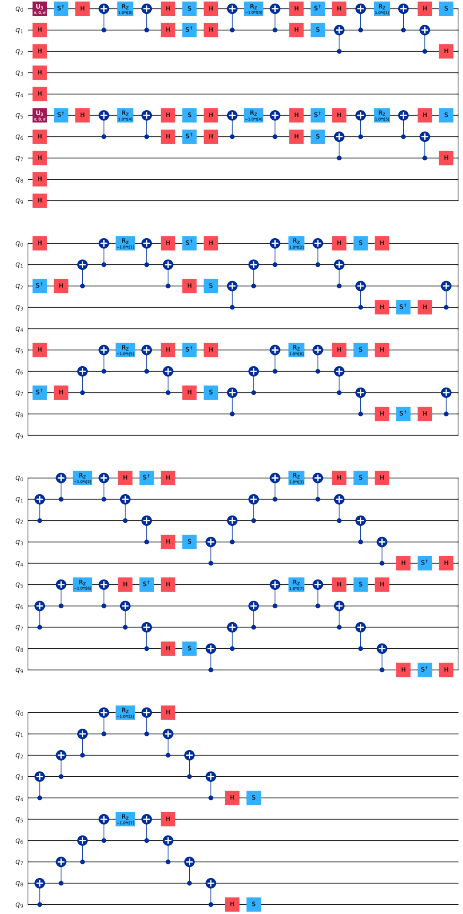


Figura 3.2: Circuito UCCS

Dichiarazione generico ansatz UCCX

```

1 uccx_qc = UCCX (
2     fc_problem.num_spatial_orbitals,
3     fc_problem.num_particles,
4     mapper,
5     excitations = 'x' # e.g.: 's', 'd', 'sd'
6     initial_state=HartreeFock(
7         fc_problem.num_spatial_orbitals,
8         fc_problem.num_particles,
9         mapper,
10    ),
11    )
12
13 bounds = [[-np.pi,np.pi] for _ in range(uccx_qc.num_parameters)]
14 uccx_qc.parameter_bounds = bounds

```

L'ansatz rotazionale *hardware efficient* invece viene implementato tramite la classe `EfficientSU2`, che può essere configurata con molta flessibilità:

Dichiarazione ansatz hardware efficient

```

1 efficient = EfficientSU2 (
2     num_qubits,
3     reps=1,
4     entanglement='reverse_linear',
5     su2_gates=['ry', 'rz']
6     initial_state=HartreeFock(
7         fc_problem.num_spatial_orbitals,
8         fc_problem.num_particles,
9         mapper,
10    ),
11 )
12
13 # stampa il circuito
14 efficient.draw('mpl')

```

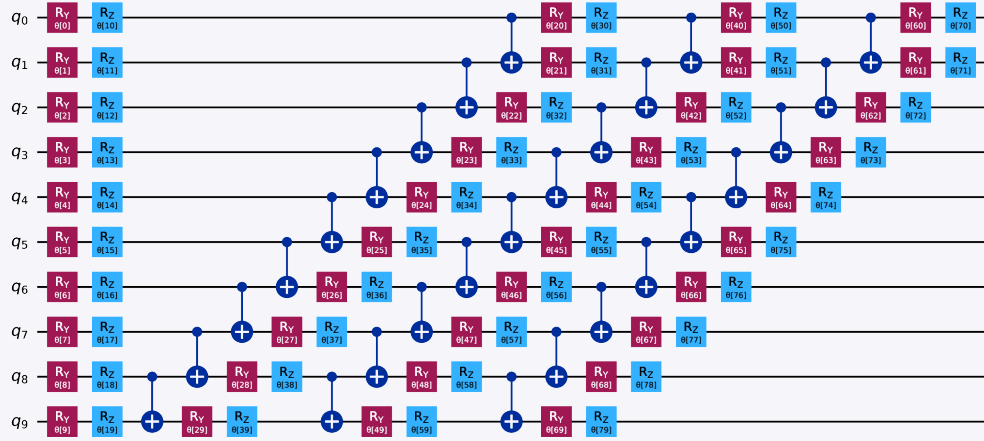


Figura 3.3: Circuito EfficientSU(2)

in cui si sono fornite le seguenti istruzioni:

- `reps` specifica il numero di ripetizioni di ripetizioni del layer di rotazioni e CNOT
- `entanglement` specifica la struttura e l'ordine con cui vengono applicati i CNOT
- `su2_gates` permette di scegliere quali rotazioni applicare a ciascun qubit all'interno di un layer

Il numero di qubit dovrà necessariamente corrispondere al numero di qubit dell'operatore da valutare, che nel caso del LiH è 10.

3.1.3 VQE

Dichiarati il problema e la forma variazionale, si passa all'implementazione degli algoritmi di calcolo, con cui si effettuano le previsioni dei valori di energia. Come introdotto

in 2.2.1, gli ingredienti principali di VQE sono tre: una funzione costo, una forma variazionale e un ottimizzatore. Nelle seguenti linee di codice si va proprio a dichiarare un oggetto `VQE` inserendo l'ansatz e l'*optimizer*, insieme ad un `Estimator`, classe di Qiskit che si occupa di valutare il valore di aspettazione di un'hamiltoniana.

Implementazione VQE

```
1 vqe_solver = VQE(Estimator(), ansatz, opt)
2
3 if ini is None:
4     ini = [0.0] * ansatz.num_parameters
5 else:
6     vqe_solver.initial_point = ini
7
8 calc = GroundStateEigensolver(mapper, vqe_solver)
9
10 res = calc.solve(problem)
```

Il metodo `.solve()` di `GroundStateEigensolver` si occupa di estrarre l'hamiltoniana di seconda quantizzazione del problema elettronico, quindi la dà in pasto a VQE per calcolarne il valore di aspettazione.

Scelta dell'ottimizzatore

VQE è un metodo ibrido, che sfrutta un *optimizer* classico per trovare i migliori parametri per la relativa forma variazionale. Tra questi, quelli più impiegati nel presente lavoro sono COBYLA e SPSQL, ma si è voluto fare un paragone con un terzo ottimizzatore utilizzato diffusamente in letteratura: SPSA.

- SPSA (Simultaneous Perturbation Stochastic Approximation): è un metodo stocastico che approssima le derivate perturbando casualmente i parametri. È progettato per essere robusto in presenza di rumore, risultando efficace su hardware reale dove gli errori sono frequenti, ma può avere prestazioni inferiori in simulazioni ideali prive di rumore.
- COBYLA (Constrained Optimization BY Linear Approximations): un metodo iterativo che ottimizza approssimando linearmente la funzione obiettivo, permettendo anche di impostare vincoli sui parametri. È utile per ottimizzazioni che richiedono stabilità e precisione in assenza di rumore.
- SPSQL: un metodo di ottimizzazione che utilizza le informazioni derivabili dalla funzione per individuare il minimo locale in modo accurato, con buone performance nelle simulazioni ideali.

Di seguito è riportato un grafico che confronta i valori calcolati dai tre ottimizzatori su ansatz pUCCD.

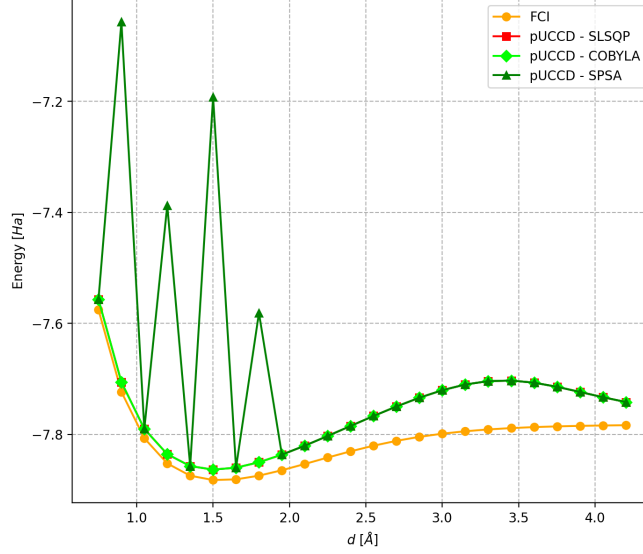


Figura 3.4: LiH confronto tra ottimizzatori

COBYLA e SPSQL convergono a risultati simili nelle energie calcolate, mentre SPSA, a parità di iterazioni, non riesce a raggiungere la convergenza. Questo può dipendere dal fatto che SPSA è ottimizzato per lavorare in condizioni rumorose, dove fornisce maggiore stabilità.

3.2 LiH - Confronti UCC

In questa sezione vengono confrontate le prestazioni di vari ansatz per il calcolo dell'energia dello stato fondamentale della molecola di LiH a diversi valori della distanza internucleare. Gli ansatz analizzati comprendono diverse varianti di q -UCC, tra cui q -UCCS, q -UCCD, q -UCCSD, e q -pUCCD, oltre a un ansatz di tipo hardware-efficient (EfficientSU2). L'obiettivo è valutare le differenze di prestazioni in termini sia di accuratezza energetica sia di complessità circuitale. I calcoli sono stati eseguiti su simulatori ideali, quindi senza rumore hardware, ma possono comunque offrire un'indicazione del potenziale di ciascun metodo per applicazioni pratiche.

Coerentemente con quanto esposto nelle sezioni 2.3 e 2.4, ci si attende di osservare prestazioni migliori in quegli ansatz che considerano un maggior numero di configurazioni: q -UCCSD e q -UCCD. Tuttavia, se da un lato è chiaro che q -UCCSD raggiunga il risultato più accurato, lo stesso non si può dire per le soluzioni intermedie: sarà rilevante osservare fino a che punto possano competere in precisione le alternative meno costose, come q -pUCCD e EfficientSU(2). Quest'ultimo, in particolare, dovrebbe risultare l'ansatz meno costoso in termini di operazioni richieste, data la sua costruzione meno vincolata. Per quanto riguarda q -UCCS, ci si aspetta una *performance* più debole, dal momento che le eccitazioni singole includono un basso numero di configurazioni.

La Figura 3.5 riporta il confronto tra i valori energetici calcolati per ciascun ansatz. Per contestualizzare il miglioramento ottenuto grazie alla considerazione della correlazione

elettronica (si veda 1.3), viene inclusa anche l'energia ottenuta con il metodo di Hartree-Fock (sez. 1.1.3).

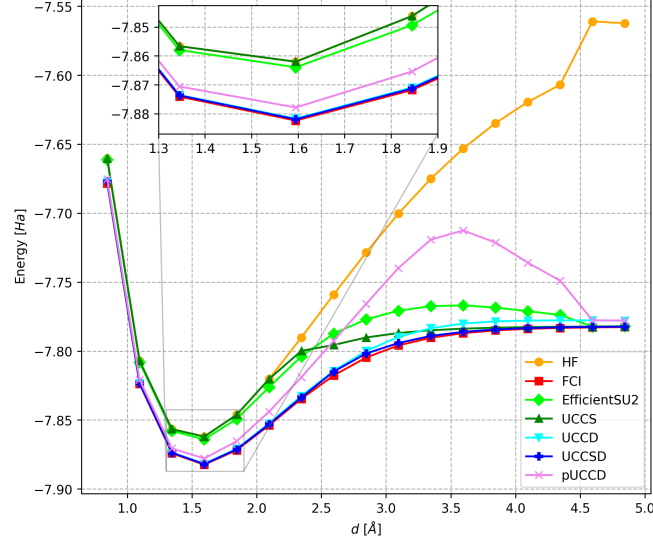


Figura 3.5: LiH confronto tra ansatz

Prima di tutto, si può notare che le previsioni HF si discostano molto dal valore di FCI, specialmente a grandi distanze. È quindi evidente che, per quanto semplice, l'approssimazione di Hartree-Fock non restituisce valori sufficientemente accurati. Gli ansatze q -UCCSD, praticamente sovrapposto a FCI, e q -UCCD si confermano i metodi *coupled cluster* più precisi, con il primo che mostra una distanza massima in energia dal valore di riferimento di soli 0.003 Hartree, e il secondo che si mantiene nello stesso ordine di grandezza con 0.007 Ha. La deviazione di q -UCCS invece è paragonabile a quella di EfficientSU(2), entrambe quasi dieci volte maggiori dei circuiti precedenti. Tuttavia, è interessante notare che, sempre in termini di deviazione massima, l'ansatz *hardware efficient* risulta generalmente più accurato di q -pUCCD, con quest'ultimo che mostra grandi differenze nella descrizione della dissociazione, arrivando a circa 0.07 Ha al di sopra del valore di riferimento. Quindi, in linea con le attese, gli ansatze q -UCC più strutturati tendono a fornire valori di energia più accurati, a fronte di circuiti più profondi. Ad esempio, q -UCCSD ha una *depth* di 2098, con 1616 porte CNOT, risultando il più complesso tra quelli considerati; q -UCCD è simile, essendo fondamentalmente lo stesso circuito privato dei 71 *layers* di q -UCCS. q -pUCCD, avendo un numero significativamente inferiore di configurazioni incluse, scende a 511, un quarto rispetto ai precedenti. D'altra parte, EfficientSU(2) mostra una profondità di solo 21 e nonostante ciò, per lo meno a grandi distanze, non mostra discrepanze energetiche sensibilmente più ampie.

La tabella riporta i valori dell'energia per ciascun ansatz alle distanze di legame E_0 e a lungo raggio E_∞ , fornendo una visione d'insieme sulle differenze tra gli approcci sia in termini di accuratezza che di complessità del circuito.

Metodo	Depth	CNOT	Parametri	E_0 (Ha)	E_∞ (Ha)
FCI	/	/	/	-7.8824	-7.7827
EfficientSU2	21	40	80	-7.8640	-7.7820
q-UCCS	71	80	8	-7.8620	-7.7820
q-UCCD	2032	1536	16	-7.8817	-7.7780
q-UCCSD	2098	1616	24	-7.8820	-7.7823
q-pUCCD	511	384	4	-7.8779	-7.7779

Tabella 3.1: Energie calcolate e dimensioni circuiti

La figura 3.6 illustra il confronto tra le energie stimate e la profondità dei circuiti per ciascun ansatz, sempre alle due distanze E_0 (a) e E_∞ (b). Ogni punto rappresenta un ansatz, con il colore che varia in funzione della profondità del circuito, nonché la complessità di implementazione.

Alla distanza di equilibrio si osserva che i metodi q -UCCSD e q -UCCD si avvicinano maggiormente al valore di riferimento FCI, ma presentano una profondità di circuito significativamente più elevata. All'infinito la situazione arriva quasi all'inversione, con i più semplici EfficientSU(2) e q -UCCS che scendono al di sotto di q -UCCD.

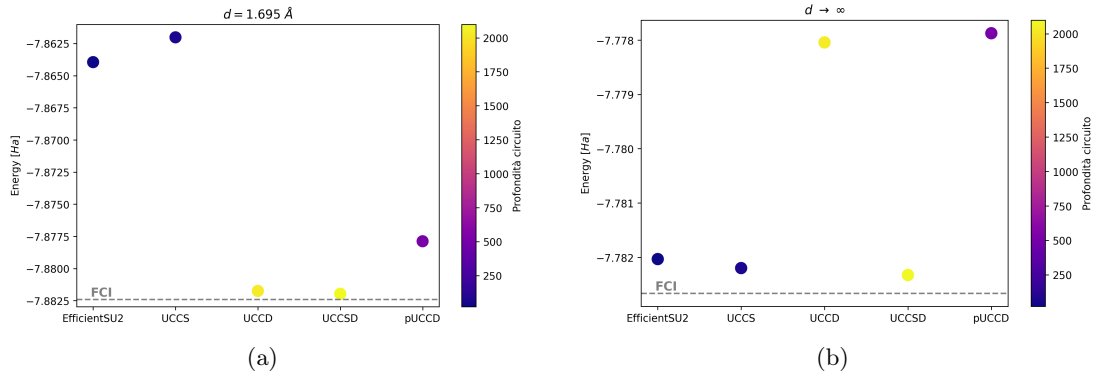


Figura 3.6: Confronto energie e profondità dei circuiti

I comportamenti osservati possono essere spiegati considerando i tipi di correlazioni elettroniche che ciascuna forma variazionale riesce a rappresentare. Alla distanza di equilibrio, dove gli atomi sono vicini e legati, sono richiesti metodi in grado di descrivere con precisione le complesse interazioni elettroniche che caratterizzano il legame chimico. Qui, un ansatz come q -UCCD, che include eccitazioni doppie, cattura meglio le influenze reciproche e pertanto fornisce stime di energia più accurate. In fase di dissociazione, invece, gli elettroni diventano progressivamente più indipendenti. La correlazione elettronica è meno intensa, e un metodo semplice come q -UCCS può descrivere in modo adeguato la situazione, anche se include solo eccitazioni singole. Questo perché, all'infinito, la mancanza di interazione tra gli elettroni fa sì che la descrizione

dell'energia totale non richiede una struttura complessa. q -UCCSD, che comprende eccitazioni singole e doppie, è in grado di bilanciare le esigenze di complessità del legame e la semplicità della dissociazione, combinando i vantaggi degli altri due.

Infine, q -pUCCD mostra contemporaneamente una sovrastima nel valore dell'energia all'equilibrio e, poiché anch'esso trascura le eccitazioni del primo ordine, anche nel valore di E_∞ , al di sopra persino del pessimo risultato di q -UCCD. Tuttavia, esiste una tecnica capace di portare un miglioramento significativo alle stime prodotte da q -pUCCD, soprattutto per quanto riguarda la descrizione della dissociazione: l'*ottimizzazione orbitale*, introdotta in 2.3.2, che va in un certo senso a compensare l'iniziale esclusione delle *singles*.

Analisi del processo di ottimizzazione

Per ogni ansatz, il processo di ottimizzazione dell'energia a una distanza fissa di $d = 1.595 \text{ \AA}$ è stato valutato analizzando:

- la varianza dei valori campionati durante l'ottimizzazione
- la massima distanza tra il valore di convergenza e i valori provati

La 3.7 illustra l'andamento dell'ottimizzazione per ciascun ansatz, mentre i valori numerici sono riportati nella seguente tabella:

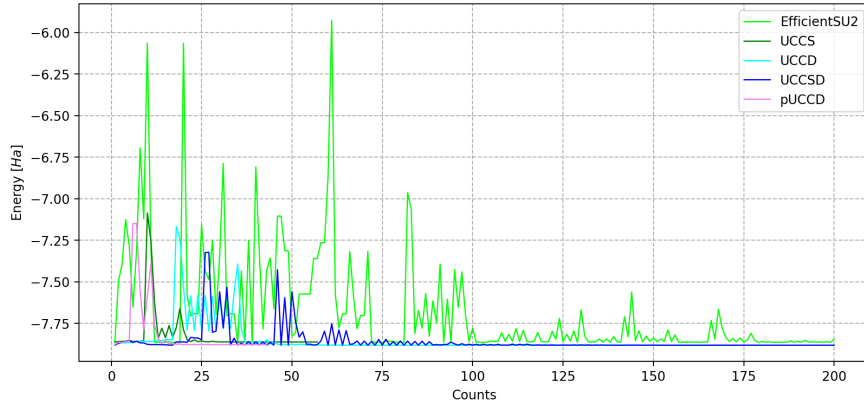


Figura 3.7: Processo di ottimizzazione a $d = 1.595 \text{ \AA}$

Ansatz	Varianza (Ha^2)	Massima ΔE (Ha)
EfficientSU2	0.1	1.9
UCCS	0.02	0.8
UCCD	0.01	0.7
UCCSD	0.006	0.6
pUCCD	0.03	0.7

l'ansatz *hardware efficient* mostra una varianza significativamente più alta (0.1) rispetto alle varianti q -UCC, suggerendo che esso vada ad esplorare un'area più ampia dello spazio dei parametri durante l'ottimizzazione. Data la sua natura *problem agnostic*, ciò

potrebbe significare che nella ricerca vengono incluse anche delle regioni non interessanti in senso fisico, portando a una minore stabilità. Invece, negli ansatz q -UCC, la varianza più bassa rifletterebbe un'esplorazione più mirata che, potenzialmente, li rende più promettenti per applicazioni su hardware con rumore.

Lo stesso comportamento si osserva nella distanza massima tra valore di convergenza e valori provati. In questo caso, l'ansatz EfficientSU(2) presenta una distanza massima molto alta, di circa 2 Ha. Questo è coerente con una maggiore ampiezza nell'esplorazione dello spazio dei parametri, che potrebbe risultare vantaggiosa per evitare minimi locali, ma può aumentare la possibilità di risultati devianti. Al contrario, q -UCCSD presenta la distanza minore (0.6), indicativo di una convergenza più stabile e costante verso il minimo.

In generale, gli ansatze con una varianza più bassa e una distanza massima minore tendono a raggiungere il minimo più rapidamente, mentre EfficientSU(2) potrebbe richiedere più iterazioni a causa della maggiore ampiezza esplorativa. Anche se il numero di iterazioni e il tempo di ottimizzazione dipendono dalla potenza di calcolo disponibile, questo rimane un dato interessante per valutare l'efficienza degli ansatz in contesti pratici.

3.3 Orbital Optimized quantum pair Coupled Cluster Doubles

In questa sezione viene illustrata l'implementazione di un circuito q -oo-pUCCD basato sull'ottimizzazione orbitale. Si parte dalla costruzione teorica dell'operatore di rotazione \mathcal{K} per modificare la base degli orbitali, adattando dinamicamente gli integrali della Hamiltoniana per ogni iterazione del processo VQE. Questo approccio ottimizza direttamente la forma della funzione d'onda e permette di esplorare configurazioni che minimizzano l'energia dell'ansatz scelto.

3.3.1 Costruzione del circuito oo-pUCCD

Per implementare attraverso un circuito l'operatore $e^{\mathcal{K}}$ si parte dal considerare le possibili rotazioni di coppie di orbitali di spin, cioè le entrate non nulle della matrice K anti-hermitiana introdotta in 2.30, qui sotto riportata:

$$\hat{\mathcal{K}} = \sum_{p < q} \mathcal{K}_{pq} (a_q^\dagger a_p - a_p^\dagger a_q)$$

La funzione `create_orbital_rotation_list` genera la lista ordinata delle rotazioni tra orbitali con indici $p < q$, nella forma `[0,1]`, `[0,2]` e così via.

Generazione lista di eccitazioni

```

1 def create_orbital_rotation_list(ansatz) -> list:
2
3 half_as = int(ansatz.num_qubits / 2)
4 orbital_rotations = []
5
6 for i in range(half_as):
7     for j in range(half_as):
8         if i < j:
9             orbital_rotations.append([i, j])
10
11 return orbital_rotations

```

Successivamente, si converte ogni coppia di indici (p, q) in un operatore fermionico $a_q^\dagger a_p$ con la funzione `build_fermionic_excitation_ops`

Costruzione operatore fermionico

```

1 def build_fermionic_excitation_ops(
2     mo_problem,
3     excitations: Sequence,
4     mapper: QubitMapper = JordanWignerMapper()
5 ) -> list[FermionicOp]:
6
7     num_spin_orbitals = 2 * mo_problem.num_spatial_orbitals
8     operators = []
9
10    for exc in excitations:
11        label = []
12        for occ in exc[0]:
13            label.append(f"+_{occ}")
14        for unocc in exc[1]:
15            label.append(f"-_{unocc}")
16        op = FermionicOp(f" ".join(label): 1}, num_spin_orbitals=
17            num_spin_orbitals)
18        op_adj = op.adjoint()
19        # va considerata una fase immaginaria introdotta dalla
20        # approssimazione di
21        # Trotter implementata in Qiskit
22
23        op_minus = 1j * (op - op_adj)
24        operators.append(op_minus)
25
26        # map to qubit operators
27        qubit_ops = mapper.map(operators)
28
29    return qubit_ops

```

Infine, si usa la classe `EvolvedOperatorAnsatz` per costruire il circuito con le eccitazioni esponenziate. Quest'ultimo va poi concatenato a q -pUCCD, passato in argomento come

`initial_state`, per formare q -oo-pUCCD.

Generazione ansatz oo-pUCCD

```

1 def generate_oo_puccd (puccd, mo_problem, excitations):
2
3     qubit_ops = build_fermionic_excitation_ops(
4         mo_problem,
5         excitations
6     )
7
8     oo_puccd = EvolvedOperatorAnsatz(
9         operators=qubit_ops,
10        initial_state=puccd,
11        parameter_prefix='k'
12    )
13
14    bounds = [[-np.pi,np.pi] for _ in range(
15        oo_puccd.num_parameters)]
16    oo_puccd.parameter_bounds = bounds
17
18    return oo_puccd

```

3.3.2 oo-VQE

L'operazione di ottimizzazione orbitale avviene in concomitanza con VQE, motivo per cui spesso ci si riferisce questo algoritmo con la sigla oo-VQE: durante la minimizzazione dell'energia si ricalcola l'hamiltoniana ad ogni ottimizzazione dei parametri. Per poterlo fare, bisogna accedere agli integrali a uno e due corpi, contenuti all'interno dell' `ElectronicStructureProblem` in forma di classe, un oggetto chiamato `ElectronicIntegrals`. Al suo interno, gli integrali sono divisi per spin, perciò occorrono due matrici: K_α e K_β , costruite sempre a partire dalla lista di eccitazioni `orbital_rotations`. La funzione `orbital_rotation_matrix` riceve in argomento i parametri restituiti da ciascuna interazione del ciclo VQE e li inserisce all'interno delle matrici \mathcal{K} , quindi restituisce gli operatori di rotazione e^{K_α} , e^{K_β} :

Costruzione $e^{\hat{K}}$

```

1 def orbital_rotation_matrix(
2     problem,
3     parameters: np.ndarray,
4     rotations: list
5 ) -> Tuple[np.ndarray, np.ndarray]:
6
7     dim_kappa_matrix = problem.num_spatial_orbitals
8
9     k_matrix_alpha = np.zeros((dim_kappa_matrix, dim_kappa_matrix))
10    k_matrix_beta  = np.zeros((dim_kappa_matrix, dim_kappa_matrix))
11
12    for i, exc in enumerate(rotations):
13        k_matrix_alpha[exc[0]][exc[1]] = parameters[i]
14        k_matrix_alpha[exc[1]][exc[0]] = -parameters[i]
15        k_matrix_beta[exc[0]][exc[1]]  = parameters[i]
16        k_matrix_beta[exc[1]][exc[0]]  = -parameters[i]
17
18    matrix_a = expm(k_matrix_alpha)
19    matrix_b = expm(k_matrix_beta)
20
21    return matrix_a, matrix_b

```

dopodiché si può procedere con la modifica degli integrali, utilizzando le relazioni 2.33. Queste vengono implementate utilizzando `numpy` per ottimizzare il calcolo il più possibile, dal momento che il numero di moltiplicazioni richieste scala come $\mathcal{O}(K^8)$, con K numero di orbitali spaziali. In realtà, nello specifico del caso pUCCD, `ElectronicIntegrals` contiene soltanto valori associati a spin α , pertanto `matrix_b` può essere ignorata. Una volta effettuate le trasformazioni degli integrali, si può andare ad estrarre la nuova hamiltoniana di seconda quantizzazione (eq. 1.39) dal problema elettronico ruotato.

Rotazione orbitali

```

1 def rotate_orbitals(
2     problem,
3     one_body,
4     two_body,
5     matrix_a,
6     matrix_b,
7     mapper = JordanWignerMapper()
8 ):
9
10    # Matrice C = exp(K_a)
11    C = matrix_a
12
13    N = len(C)
14
15    # Trasformazione degli integrali a un corpo
16    one_body_transformed = np.einsum('up,uv,vq->pq', C, one_body, C)
17
18    # Trasformazione degli integrali a due corpi
19    eri_transformed = np.einsum('up,vq,xr,ys,uvxy->pqrs', C, C, C,
20    C, eri)
21    eri_transformed = np.zeros((N, N, N, N))
22
23    # Definizione di un oggetto ElectronicIntegrals per modificare
24    # quello contenuto in problem
25    e_int = ElectronicIntegrals.from_raw_integrals(
26        h_transformed,
27        eri_transformed
28    )
29
30    problem.hamiltonian.ElectronicIntegrals = e_int
31
32    # extract rotated operator
33    rotated_operator = problem.hamiltonian.second_q_op()
34    rotated_operator = mapper.map(rotated_operator)
35
36    return rotated_operator

```

Su questo `rotated_operator` sarà calcolato il valore di aspettazione 2.32 nel ciclo successivo di VQE, nel quale saranno ottimizzati ulteriormente i parametri, che poi saranno usati per generare una nuova matrice di rotazione; questo procedimento quindi continua fino a convergenza.

3.3.3 Risultati

In questa sezione si riportano i risultati ottenuti applicando l'ottimizzazione orbitale nella simulazione delle molecole LiH e H₂O, per osservare il comportamento di questo approccio su un sistema più complesso, evidenziando eventuali limiti e vantaggi scalabili a sistemi di dimensioni maggiori.

LiH

Nel caso dell'idruro di litio, il circuito q -oo-pUCCD comporta un aumento in profondità di 'solamente' 155, il che renderebbe questo ansatz estremamente promettente, se fosse anche capace di migliorare sensibilmente i risultati di pUCCD. In realtà, per quanto riguarda l'energia di stato fondamentale, è osservata una diminuzione del valore calcolato, ma non di certo degna di nota. Il valore di energia a grandi distanze invece giova della correzione, compatibilmente con quanto osservato in altri studi [16, 21], anche se non nella stessa misura.

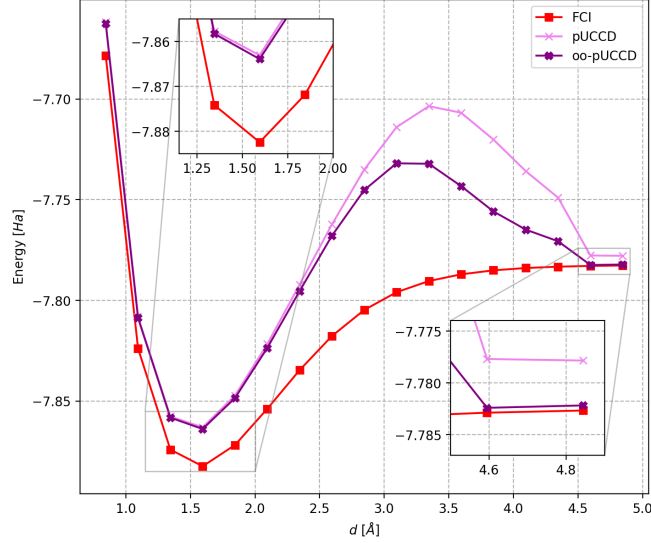


Figura 3.8: LiH con ottimizzazioni orbitali

La differenza nei minimi di energia tra q -pUCCD e q -oo-pUCCD è di appena 0.8 mHa, mentre a grande distanza raggiunge i 15 mHa, abbassando la deviazione dal valore di riferimento FCI a soltanto 0.5 mHa.

H₂O

Per testare ulteriormente le potenzialità dell'algoritmo oo-VQE, si applica ad un sistema più laborioso: la molecola di H₂O. Anche in questo caso si utilizza `FreezeCoreTransformer` per ridurre la dimensione del problema, che stavolta porta all'esclusione di due orbitali *core*. La differenza in profondità tra i circuiti si fa più marcata rispetto a LiH, con q -oo-pUCCD che contiene oltre 250 *layers* in più dei circa 1100 di pUCCD. Tuttavia, si tratta ancora di un circuito piuttosto contenuto, soprattutto considerando che q -UCCSD per lo stesso problema supererebbe una profondità di 10000.

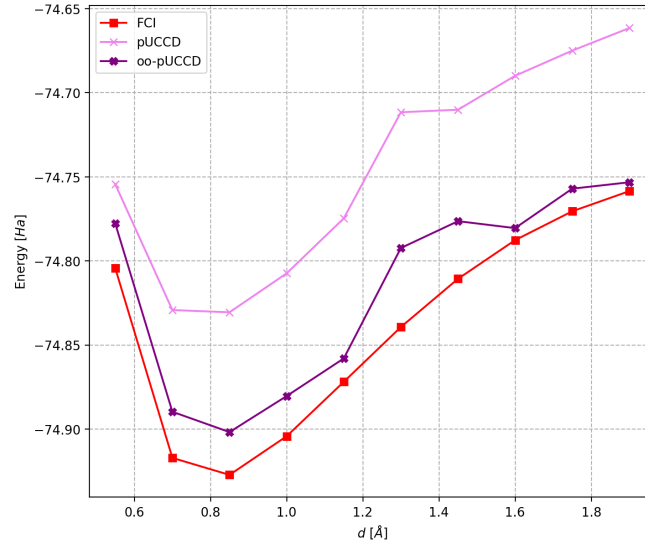


Figura 3.9: LiH con ottimizzazioni orbitali

In questo caso il vantaggio offerto dalle ottimizzazioni orbitali nel calcolo dell'energia è lampante: il minimo diminuisce di oltre 70 mHa, il valore a grandi distanze scende fino a discostarsi di soli 5 mHa dal valore di riferimento.

Riassumendo, questi risultati suggeriscono l'ansatz q -oo-pUCCD può offrire un sensibile miglioramento nella precisione energetica, particolarmente significativo nelle energie a lungo raggio. Allo stesso tempo, l'incremento di profondità si mantiene limitato, soprattutto in confronto agli ansatz UCC più strutturati. In particolare, nel caso di H_2O , si osserva un miglioramento sostanziale sia a grandi distanze che nella zona del minimo, rendendo il metodo interessante per applicazioni future, nelle quali potrebbe costituire un approccio bilanciato fra una buona approssimazione energetica e una complessità circuitale contenuta.

Conclusioni

Commento al lavoro svolto

Nel presente lavoro, è stata esplorata la capacità degli ansatz *quantum Unitary Coupled Cluster* (q -UCC) di riprodurre con precisione le energie fondamentali e di dissociazione della molecola di idruro di litio (LiH), con un focus sulla complessità circuitale richiesta da ciascuna variante. I metodi studiati, inclusi q -UCCS, q -UCCD, q -UCCSD, q -pUCCD e EfficientSU(2), hanno permesso di valutare i compromessi tra accuratezza energetica e profondità circuitale, un aspetto cruciale per una possibile implementazione su *quantum processing units* (QPU) attuali.

L'analisi ha messo in luce le peculiarità di ciascun ansatz: se da un lato q -UCCSD è stato il più accurato, producendo risultati molto vicini alla soglia teorica di Full Configuration Interaction (FCI), dall'altro richiede una profondità circuitale elevata e un numero di porte CNOT significativo, che ne limita la praticità per implementazioni hardware immediate. Al contrario, EfficientSU(2), con una struttura circuitale semplificata, ha dimostrato di poter fornire stime energetiche accettabili per determinate condizioni, nonostante richieda risorse minime. L'introduzione del *quantum pair-UCCD* (q -pUCCD) e delle ottimizzazioni orbitali (q -oo-pUCCD) ha inoltre indicato una promettente via intermedia tra la precisione e la riduzione del costo computazionale, posizionandosi come opzione interessante per simulazioni molecolari in cui le risorse quantistiche siano limitate.

Complessivamente, i risultati ottenuti offrono spunti sul potenziale degli ansatz q -UCC per simulazioni chimiche avanzate, sebbene l'attuale tecnologia richieda la continua ricerca di una maggiore efficienza circuitale. Questo studio potrebbe suggerire che sia possibile ottenere risultati significativi anche con architetture hardware contenute, grazie a scelte ponderate di ansatz e a ottimizzazioni.

Prospettive future

Alla luce di questi risultati, sviluppi futuri potrebbero orientarsi verso la valutazione degli ansatz in contesti più realistici. Un primo passo potrebbe essere quello di testare le configurazioni q -UCC attraverso simulatori quantistici con rumore, che permetterebbero di comprendere meglio l'effetto degli errori di gate e delle fluttuazioni di stato sulle stime energetiche. Questa fase sarebbe cruciale per determinare la robustezza degli ansatz rispetto agli errori di calcolo e per identificare potenziali limiti nelle implementazioni circuitali attuali.

Successivamente, e qualora i risultati su simulatori rumorosi confermassero la stabilità delle soluzioni, si potrebbe prevedere un test diretto su hardware quantistico. Tale verifica consentirebbe di validare le stime energetiche ottenute e di confrontare l'effettiva complessità circuitale con le capacità dei dispositivi fisici, aprendo la strada a simu-

lazioni sperimentali più accurate e, eventualmente, a miglioramenti nelle strategie di ottimizzazione e nella selezione degli ansatz per applicazioni pratiche.

Bibliografia

- [1] Neil S. Ostlund Attila Szabo. *Modern Quantum Chemistry*. Dover Publications Inc., 1989. ISBN: 0486691861.
- [2] C. David Sherrill. *An Introduction to Configuration Interaction Theory*. 1995. URL: <http://vergil.chemistry.gatech.edu/notes/ci.pdf>.
- [3] John A. Pople. *Quantum Chemical Models*. Nobel Lecture, December 8, 1998. Department of Chemistry, Northwestern University, 2145 Sheridan Road, Evanston, Illinois 60208, USA, 1998. URL: <https://iopenshell.usc.edu/chem545/lectures2014/pople-lecture.pdf>.
- [4] C. David Sherrill. *An Introduction to Hartree-Fock Molecular Orbital Theory*. Giu. 2000. URL: <http://vergil.chemistry.gatech.edu/notes/hf-intro/hf-intro.pdf>.
- [5] C. David Sherrill. *The Born-Oppenheimer Approximation*. Mar. 2005. URL: <http://vergil.chemistry.gatech.edu/notes/bo/bo.pdf>.
- [6] Pablo Echenique e J. L. Alonso. «A mathematical and computational review of Hartree-Fock SCF methods in quantum chemistry». In: *Molecular Physics* 105.23–24 (dic. 2007), pp. 3057–3098. ISSN: 1362-3028. DOI: [10.1080/00268970701757875](https://doi.org/10.1080/00268970701757875). URL: <http://dx.doi.org/10.1080/00268970701757875>.
- [7] Michael A. Nielsen e Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [8] Errol G. Lewars. *Computational Chemistry*. A cura di Springer. Springer Cham, 2011. ISBN: 9789048138609.
- [9] Alberto Peruzzo et al. «A variational eigenvalue solver on a photonic quantum processor». In: *Nature Communications* 5.1 (lug. 2014). ISSN: 2041-1723. DOI: [10.1038/ncomms5213](https://doi.org/10.1038/ncomms5213). URL: <http://dx.doi.org/10.1038/ncomms5213>.
- [10] Panagiotis Kl. Barkoutsos et al. «Quantum algorithms for electronic structure calculations: Particle-hole Hamiltonian and optimized wave-function expansions». In: *Physical Review A* 98.2 (ago. 2018). ISSN: 2469-9934. DOI: [10.1103/physreva.98.022322](https://doi.org/10.1103/physreva.98.022322). URL: <http://dx.doi.org/10.1103/PhysRevA.98.022322>.
- [11] Joonho Lee et al. «Generalized Unitary Coupled Cluster Wave functions for Quantum Computation». In: *Journal of Chemical Theory and Computation* 15.1 (nov. 2018), pp. 311–324. ISSN: 1549-9626. DOI: [10.1021/acs.jctc.8b01004](https://doi.org/10.1021/acs.jctc.8b01004). URL: <http://dx.doi.org/10.1021/acs.jctc.8b01004>.
- [12] Gabriel T. Landi. *Second Quantization*. 2019. URL: https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=http://www.fmt.if.usp.br/~gtlandi/courses/second-quantization-4.pdf&ved=2ahUKEwiDmf78KuJAXWn_rsIHc4gCIoQFnoECBcQAQ&usg=AOvVaw35pT3fxU201PVKhfevoyz9.

- [13] Elias Fernandez-Combarro Alvarez. *A practical introduction to quantum computing: from qubits to quantum machine learning and beyond*. Nov. 2020. URL: <https://indico.cern.ch/event/970903/>.
- [14] Sam McArdle et al. «Quantum computational chemistry». In: *Rev. Mod. Phys.* 92 (1 mar. 2020), p. 015003. DOI: [10.1103/RevModPhys.92.015003](https://doi.org/10.1103/RevModPhys.92.015003). URL: <https://link.aps.org/doi/10.1103/RevModPhys.92.015003>.
- [15] Wataru Mizukami et al. «Orbital optimized unitary coupled cluster theory for quantum computer». In: *Phys. Rev. Res.* 2 (3 set. 2020), p. 033421. DOI: [10.1103/PhysRevResearch.2.033421](https://doi.org/10.1103/PhysRevResearch.2.033421). URL: <https://link.aps.org/doi/10.1103/PhysRevResearch.2.033421>.
- [16] Igor O. Sokolov et al. «Quantum orbital-optimized unitary coupled cluster methods in the strongly correlated regime: Can quantum algorithms outperform their classical equivalents?» In: *The Journal of Chemical Physics* 152.12 (mar. 2020). ISSN: 1089-7690. DOI: [10.1063/1.5141835](https://doi.org/10.1063/1.5141835). URL: <http://dx.doi.org/10.1063/1.5141835>.
- [17] M. Cerezo et al. «Variational quantum algorithms». In: *Nature Reviews Physics* 3.9 (ago. 2021), pp. 625–644. ISSN: 2522-5820. DOI: [10.1038/s42254-021-00348-9](https://doi.org/10.1038/s42254-021-00348-9). URL: <http://dx.doi.org/10.1038/s42254-021-00348-9>.
- [18] Abhinav Anand et al. «A quantum computing view on unitary coupled cluster theory». In: *Chem. Soc. Rev.* 51 (5 2022), pp. 1659–1684. DOI: [10.1039/D1CS00932J](https://doi.org/10.1039/D1CS00932J). URL: <http://dx.doi.org/10.1039/D1CS00932J>.
- [19] Kishor Bharti et al. «Noisy intermediate-scale quantum algorithms». In: *Reviews of Modern Physics* 94.1 (feb. 2022). ISSN: 1539-0756. DOI: [10.1103/revmodphys.94.015004](https://doi.org/10.1103/revmodphys.94.015004). URL: <http://dx.doi.org/10.1103/RevModPhys.94.015004>.
- [20] Thomas G. Wong. *Introduction to Classical and Quantum Computing*. Omaha, Nebraska: Rooted Grove, 2022. ISBN: 979-8-9855931-1-2. URL: <https://www.thomaswong.net/introduction-to-classical-and-quantum-computing-1e4p.pdf>.
- [21] Luning Zhao et al. «Orbital-optimized pair-correlated electron simulations on trapped-ion quantum computers». In: *npj Quantum Information* 9.1 (giu. 2023). ISSN: 2056-6387. DOI: [10.1038/s41534-023-00730-8](https://doi.org/10.1038/s41534-023-00730-8). URL: <http://dx.doi.org/10.1038/s41534-023-00730-8>.
- [22] Qiskit Documentation Team. *TwoLocal Circuit*. 2024. URL: <https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.TwoLocal>.
- [23] Qiskit Documentation Team. *UCC Ansatz*. 2024. URL: https://qiskit-community.github.io/qiskit-nature/stubs/qiskit_nature.second_q.circuit.library.UCC.html.
- [24] National Institute of Standards and Technology (NIST). *Experimental data for LiH (Lithium Hydride)*. Accessed: 2024-11-1. URL: <https://cccbdb.nist.gov/exp2x.asp?casno=7580678&charge=0>.