ABSTRACT

Title of Dissertation: OPTIMIZED SIMULATIONS OF FERMIONIC SYSTEMS ON A QUANTUM COMPUTER

Qingfeng Wang
Doctor of Philosophy, 2024

Dissertation Directed by: Professor Christopher Monroe
Department of Physics

Quantum computing holds promise for simulating microscopic phenomena, offering profound implications across disciplines such as chemistry, condensed matter physics, and high-energy physics, particularly in the accurate simulation of fermions. However, practical implementation requires the optimization of quantum programs to mitigate quantum noise and decoherence effects. Given the constraints of near-term quantum computers, the Variational Quantum Eigensolver (VQE) emerges as a key approach for estimating molecular ground state energies, crucial for determining chemical properties. This work aims to present advancements in optimizing VQE simulations to minimize quantum computational resources. Specifically, this work explores various optimization strategies, including the utilization of second-order perturbation correction to recover additional energy beyond VQE estimates and select critical ansatz terms. Additionally, circuit optimization techniques are investigated, focusing on achieving shorter equivalent ansatz circuits, particularly for physically-inspired VQE ansatz, through methods such as

generalized fermion-to-qubit transformations and Pauli string orderings. Furthermore, this work demonstrates the advantage of a better initial state on a trapped-ion quantum computer.

OPTIMIZED SIMULATIONS OF FERMIONIC SYSTEMS ON A QUANTUM COMPUTER

by

Qingfeng Wang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2024

Advisory Committee:
Professor Christopher Monroe, Chair/Advisor
Professor Alexey Gorshkov
Professor Qudsia Quraishi
Professor Xiaodi Wu
Professor Christopher Jarzynski, Dean's representative

Dedication

To my family.

Acknowledgments

I extend my heartfelt gratitude to my advisor, Dr. Christopher Monroe, for accepting me as a graduate student despite my background as a computational chemist with minimal knowledge of quantum computing. Dr. Monroe provided me with unique research opportunities not only within his research group at the University of Maryland, College Park, and later at Duke University, but also through collaboration with industry leaders such as IonQ.

I am particularly indebted to the former IonQ employees Yungseong Nam and Ming Li, whose collaboration made possible the early stages of my PhD research [1]. I am also grateful to Ze-Pei Cian and Igor Markov for their contributions, which eventually led to the publication of [2]. Additionally, I had the privilege of collaborating with other members of the research group and external theorists on projects such as “Interactive protocols for classically-verifiable quantum advantage”, “Cross-platform comparison of arbitrary quantum states”, “Phase transition in magic with random quantum circuits” and “Enhancing the electron pair approximation with measurements on trapped-ion quantum computers". I extend my sincere thanks to all these collaborators, including Daiwei Zhu, Gregory D Kahanamoku-Meyer, Laura Lewis, Crystal Noel, Or Katz, Bahaa Harraz, Andrew Risinger, Lei Feng, Debopriyo Biswas, Laird Egan, Alexandru Gheorghiu, Thomas Vidick, Umesh Vazirani, Norman Y Yao, Marko Cetina, Yingyue Zhu, Alaina M Green, C Huerta Alderete, Nhung H Nguyen, Andrii Maksymov, Norbert M Linke, Mohammad Hafezi, Pradeep Niroula, Christopher David White, Sonika Johri, Michael J. Gul-

lans, Luning Zhao, Joshua Goings, Kyujin Shin, Woomin Kyoung, Seunghyo Noh, Young Min Rhee and Kyungmin Kim.

During the latter half of my PhD journey, I relocated to the Duke Quantum Center, where I had the invaluable opportunity to work directly with quantum circuits. I am particularly grateful to postdoc Liudmila Zhukas for her assistance in maintaining the hardware of the EURIQA-Blue system and the accompany of countless overnight experiment runs. Through hands-on experience, I contributed to the calibration of trapped-ion quantum machines and collaborated with Gokul Subramanian Ravi, Fred Chong, and Qiang Miao to successfully demonstrate CAFQA.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ADAPT-VQE	Adaptive Derivative-Assembled Pseudo-Trotter ansatz Variational Quantum Eigensolver
BOA	Born-Oppenheimer approximation
BK	Bravyi-Kitaev (transformation)
CAFQA	Clifford Ansatz For Quantum Accuracy
CCSD	Coupled Cluster with Single and Double excitation
FCI	Full Configuration Interaction
FT	Fault-Tolerant
pre-FT	pre-Fault-Tolerant
GA	genetic algorithm
GC	General Commuting
GTO	Gaussian-Type Orbital
GTSP	Generalized Travelling Salesman Problem
HF	Hartree Fock
HMP2	Hybrid second order Møller-Plesset perturbation
JW	Jordan-Wigner (transformation)
LCAO	Linear Combination of Atomic Orbital
LK	Lin-Kernighan
MO	Molecular Orbital
MP2	second order Møller-Plesset perturbation
PSO	Particle Swarm Optimization
QCC	Qubit Coupled-Cluster
QSR	Quibt Space Reduction
QWC	Quibt-Wise Commuting
SCF	Self-Consistent Field

SD	Slater Determinant
SO	Spin Orbital
SPSA	Simultaneous Perturbation Stochastic Approximation
STO	Slater-type orbital
STO-nG	Slater-Type Orbital simulated by n Gaussian-Type Orbitals
TSP	Travelling Salesman Problem
UCC	Unitary Coupled Cluster
UCCGSD	Unitary Coupled Cluster with Generalized Single and Double excitation
UCCSD	Unitary Coupled Cluster with Single and Double excitation
VQE	Variational Quantum Eigensolver

Chapter 1: Introduction

1.1 Overview

Simulating microscopic phenomena is anticipated to be the epitome of quantum-computing applications. Chemistry, condensed matter, high energy physics and other fields stand to benefit from accurate quantum simulations of *fermions*, particles ubiquitous in nature. To make this endeavor practical, quantum programs simulating fermionic matter must be optimized, or else quantum noise and decoherence may overwhelm quantum calculations. Due to scarce computational resources of a near-term quantum computer [8], variational quantum eigensolver (VQE) [9] is widely used to estimate the *ground state energy* of a molecule, critical for determining its chemical properties, using relatively short programs.

This thesis aims to present the advancements achieved in facilitating VQE implementation with minimal quantum computational resources. The structure of the thesis unfolds as follows.

In Chapter 2, we delve into the fundamentals of electronic structure theory in quantum chemistry. Here, we elucidate several key concepts prevalent in quantum computing literature, including the STO-3G basis set, the Hartree-Fock method, spin orbitals, and more.

Subsequently, Chapter 3 provides a comprehensive overview of quantum computing, portraying it as an efficient and controllable unitary evolution apparatus. We elucidate the process of mapping a quantum chemistry problem onto a quantum computing framework. Moreover, we

introduce the VQE as the foremost hybrid method for tackling quantum chemistry problems on noisy quantum computers.

After all basics have been explained, we detail the optimization opportunities as the following:

1. *Select the most relevant ansatz terms.* We introduce a perturbation-based approach in Chapter 4, which inherently guides the selection of variational parameters to be incorporated into the ansatz for a given VQE circuit. This method allows for systematic construction of an expanding VQE circuit in an iterative manner, progressively advancing towards a more accurate estimation of the ground-state energy of the simulated system.

2. *Recover extra energy beyond VQE energy.* Recognizing that the VQE energy can be expressed as the sum of zeroth and first-order perturbation energies, we naturally extend beyond this to recover second-order perturbation energy, as elaborated upon in Chapter 4.

3. *Optimize ansatz circuit to equivalently shorter forms.* In Chapter 5, focusing on UCC-type VQE ansatz, we explore various opportunities for circuit optimization, including generalized fermion-to-qubit transformations and Pauli string orderings. These techniques are further refined and integrated more extensively to achieve enhanced optimizations, as discussed in Chapter 6.

4. *Start with a better initialization closer to global minimum.* Finally, in Chapter 7, we elucidate how a superior initialization using CAFQA can accelerate VQE convergence. Additionally, we showcase the advantages of CAFQA on a trapped-ion quantum computer in Section 7.3.

The material presented in Chapter 4, Chapter 5 are largely unmodified work from my publication:

Resource-optimized fermionic local-hamiltonian simulation on a quantum computer for quantum chemistry

Qingfeng Wang, Ming Li, Christopher Monroe, and Yunseong Nam

Quantum, 5:509, 2021

while content for Chapter 6 is from

Ever more optimized simulations of fermionic systems on a quantum computer

Qingfeng Wang, Ze-Pei Cian, Ming Li, Igor L. Markov, and Yunseong Nam

2023 60th ACM/IEEE Design Automation Conference (DAC), 1-6, 2023.

Modifications have been made as necessary to seamlessly integrate it into the thesis. The content in Chapter 7 involves experimental work currently undergoing preparation for publication.

Chapter 2: Quantum chemistry preliminaries

2.1 Overview

In quantum chemistry computation, the primary objective often revolves around employing numerical methods to investigate both static and dynamic properties. In this thesis, our emphasis is specifically directed towards electronic theory, which seeks to compute the ground state of molecular systems.

The material discussed herein is commonly addressed in standard quantum chemistry textbooks that delve into electronic structure theory [10–12]. It is not our intention to offer an exhaustive overview of quantum chemistry; rather, our aim is to provide a more cohesive framework for understanding the myriad concepts encountered throughout the remainder of the thesis.

2.2 The goal and challenge

While this characterization may not be entirely precise, it is widely acknowledged within the scientific community that many intriguing properties of molecules, as well as advanced applications of quantum chemistry, hinge on the capacity to derive the ground state energy of the

molecule as described by the time-independent Schrödinger equation

$$H|\psi\rangle = E|\psi\rangle \tag{2.1}$$

where E is the eigenstate energy, $|\psi\rangle$ is the eigenstate wavefunction. H is the molecular Hamiltonian expressed as the

$$\begin{aligned} H &= T + V \\ &= (\sum_{N} T_N + \sum_{e} T_e) + (\sum_{A>B} V_{N_A N_B} + \sum_{Ne} V_{eN} + \sum_{i>j} V_{e_i e_j}) \end{aligned} \tag{2.2}$$

where e stands for electron and N stands for nuclei. Also,

$$T_i = -\frac{1}{2m_i}\nabla_i^2 \tag{2.3}$$

is kinetic energy for particle i with mass m_i, and

$$V_{ij} = \frac{Z_i Z_j}{r_{ij}} \tag{2.4}$$

is Coulomb potential energy between two charged particles with charge $Z_{i/j}$ and distance r_{ij}.

The Hamiltonian of a molecular system comprises the sum of its kinetic energy and potential energy components. The kinetic energy encompasses both the nuclei and electrons of the molecule, while the potential energy encompasses interactions such as nuclei-nuclei repulsion, electron-electron repulsion, and electron-nuclear attraction.

However, the intricate interplay of these particle interactions renders analytical solutions nearly unattainable. Consequently, the problem necessitates numerical solutions, introducing challenges associated with achieving precise results within finite computational resources. This constitutes a fundamental challenge in modern electronic structure theory.

The challenge stems from the curse of dimensionality, particularly evident when employing the position basis and approximating each spatial degree of freedom of $|\Psi\rangle$ with k points. Considering a system with N particles, each contributing three dimensions, the state $|\Psi\rangle$ necessitates approximately k^{3N} points. This exponential scaling with the number of particles renders computations infeasible for molecules containing numerous particles.

Consequently, the practical solution involves employing approximations to address this computational complexity.

2.3 Born-Oppenheimer approximation

The Born-Oppenheimer Approximation (BOA) stands as one of the most effective, robust, and widely employed approximations in quantum chemistry, often assumed if not explicitly stated otherwise.

BOA capitalizes on the substantial mass disparity between nuclei and electrons, with nuclei being approximately 2000 times heavier than electrons. This significant difference enables the treatment of nuclei and electrons as separate entities, facilitating numerical tractability.

Intuitively, one can conceive of electrons as lightweight entities moving considerably faster than nuclei. Consequently, electrons can swiftly adapt and equilibrate around new configurations of nuclei. Meanwhile, on the timescale relevant to nuclei motion, they navigate within the av-

eraged electronic potential field. This conceptualization underscores the rationale behind BOA's effectiveness in decoupling the dynamics of electrons and nuclei in molecular systems.

In BOA, the initial step involves treating the locations of nuclei, denoted by R, as fixed, while allowing the electrons to move freely. Consequently, the Hamiltonian $H(r, R)$ that we seek to solve is effectively reduced to the

$$H(r, R) = H_{el}(r; R) + H_{nu}(R) \tag{2.5}$$

$$H_{el}(r; R) = T_e + V_{ee} + V_{eN} \tag{2.6}$$

$$H_{el}(r; R)|\psi(r; R)\rangle = E_{el}|\psi(r; R)\rangle \tag{2.7}$$

where $(r; R)$ implies electron locations r at fixed nuclei locations R. Once we have solved for E_{el} for many R locations, we effectively can construct a potential energy field (PES) and dynamics can be computed based on PES. For example, the work [13] computed a large number of electronic energies at different physically reasonable configurations (geometries) and fit them for a PES and used it to compute spectrum.

Throughout the thesis, our main focus is to compute the ground state energy of electronic Hamiltonian given a certain bond length. On a high level, the general practice is then to parameterize the wavefunction with a basis set, and then minimize the energy with variational theorem. In Section 2.4 we explain how to represent the molecular wavefunction with a basis set and later in Section 2.5 we explain the popular algorithm to minimize energy – the Hartree-Fock method.

2.4 The choice of electronic wavefunction

2.4.1 Single-electron atomic orbital and basis

These are so-called hydrogen-like atomic species and can be solved analytically. The solution form is often captured as Slater-type orbital (STO). [14] However, such a form is difficult to calculate integrals. A computationally cheaper but less accurate form is called Gaussian-type orbital (GTO). [15]

To combine the benefit of both, we now use n parametrized GTOs to mimic the STO, which is called STO-nG basis. For example, the fit for s orbital take the form

$$\psi_{\mathrm{STO}-n\mathrm{G}}(r) = \sum_{i=1}^{n} c_i \phi_i(r) \tag{2.8}$$

where n usually takes 2-6, and

$$\phi_i(i) = \left(\frac{2\alpha_i}{\pi}\right)^{3/4} e^{-\alpha_i r^2} \tag{2.9}$$

are the GTOs, where r is the location for the electron.

Notice each STO-nG has n α_i and n c_i unknowns which are determined by fitting into different STO such as $1s, 2s, 2p$, etc. Those coefficients are tabulated for easy use. [16, 17]

2.4.2 Single-electron molecular orbital and LCAO

Molecular species such as H_2^+ have two nuclei sharing the same electron. At ground state, the electron is not centered around any of the H atoms but roaming around the entire H−H molecule structure. Apparently, the single atomic orbitals, which are centered around single atoms, can not capture the roaming feature of the electron. To make the wavefunction to spread the spatial probability to the entire molecule we can use a linear combination of atomic orbital (LCAO). For example, suppose the wavefunction centered around two H atoms are $\phi_{r;R_A}$ and ϕ_{r,R_B}, then the new combined orbital is

$$\psi(r; R_A, R_B) = c_A\phi_A(r; R_A) + c_B\phi_B(r; R_B). \tag{2.10}$$

Since $\psi(r; R_A, R_B)$ describes the electron roaming the entire molecule, we call it the molecular orbital (MO). The coefficients c_A and c_B can be determined by minimizing the electronic energy. Linear combination of N atomic orbitals will result in N new molecular orbitals.

2.4.3 Basis set for multi-electron atoms

From hydrogen-like analytical solutions, we know the shape of the electrons in excited states, such as $2s, 2p, 3s$, etc. Populating more than one electron introduces the interaction between electrons and makes the problem unsolvable analytically, as a result wavefunctions such as $2s, 2p, 3s$ are no longer accurate descriptions of actual electrons.

However, due to their clear and intuitive physical picture, we can still approximately describe the wavefunction using STO or other basis set as the starting point by ignoring the dynamic

interactions between electrons. Most popular in quantum computing literature, people use STO-3G basis set to approximate the atomic orbitals.

We introduce the STO-nG basis set as the minimal basis set that describes each of the atomic orbitals. STO-nG is a minimal basis set meaning it uses only a single function Eq. (2.8) to represent one atomic orbital such as $1s, 2s$ or $2p_{x,y,z}$. There are other basis sets that use more than one function to represent a single atomic orbital to make it more expressive but also more computationally expensive.

In addition, STO-nG basis set usually provides orbitals for the entire valence shell if that shell hosts any electron. For example, H and Li require a single STO-3G orbital for $1s$, while Li-Ne requires five STO-3G orbitals containing $1s, 2s$ and $2p_{x,y,z}$.

STO-3G basis set is one of the most widely used in the quantum community for its simplicity. However, in order to get high quality computation, a larger basis set is often required.

2.4.4 Wavefunction for many-electron molecules

We now consider a molecular system containing many identical electrons. Following the logic of the previous sections, we can use MO to approximate the probability distribution of each electron.

However, for real molecule, there are two important rules we also need to consider:

1. Each electron has spin degree of freedom (σ), with either spin up ($\sigma = \alpha$) or spin down ($\sigma = \beta$);

2. As electrons are indistinguishable fermions, the wavefunction must exhibit global anti-symmetry when particles are exchanged.

The first rule can be satisfied by multiplying the spin to each of MO to result in a spin orbital (SO) denoted as

$$\chi(\mathbf{x_i}) = \psi(r_i; R)\sigma. \tag{2.11}$$

The second rule can be naturally realized by Slater determinant (SD) [18]

$$|\psi\rangle = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(\mathbf{x_1}) & \chi_2(\mathbf{x_1}) & \cdots & \chi_N(\mathbf{x_1}) \\ \chi_1(\mathbf{x_2}) & \chi_2(\mathbf{x_2}) & \cdots & \chi_N(\mathbf{x_2}) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(\mathbf{x_N}) & \chi_2(\mathbf{x_N}) & \cdots & \chi_N(\mathbf{x_N}) \end{vmatrix} \tag{2.12}$$

for wavefunction of N electrons.

Notice this wavefunction is antisymmetric and each term contains the product of N single-electron spin orbitals. This SD form has an intuitive physical picture and can be used to describe wavefunction of ground state as well as excited states, depending on the placement of the electrons. Starting from a ground state reference SD $|\psi_0\rangle$, an excited state with occupied orbital $i, j, k, \cdots$ excited to empty orbitals $a, b, c, \cdots$ can be represented with excitation operator T as

$$|\psi_{ijk\cdots}^{abc\cdots}\rangle = T_{ijk\cdots}^{abc\cdots}|\psi_0\rangle. \tag{2.13}$$

However, there is still one problem left. Remember that spin orbitals χ_i are like MO in Eq. (2.10), which is the linear combination of the atomic orbitals (with spin), we don't know in

advance what these coefficients are. Solving for these coefficients $\{\chi_i\}$ is the task of Hartree-Fock method in Section 2.5.

Wavefunctions in SD form naturally ignore interaction between electrons, therefore deviating from the actual system. To account for the real system, we can use SD as basis and linearly combine them which leads to the multi-reference method including the most complete one called full configuration interaction (FCI) introduced in Section 2.6.

2.5 Hartree-Fock method

2.5.1 Overview

The Hartree-Fock (HF) is the starting point of many modern electronic structure theories. It translates the problem of many-electron Hamiltonian Eq. (2.6) into a problem single-electron Hamiltonian (Fock operator F) in which interactions from other electrons are expressed as an effective averaged potential, as known as mean field. As a result, HF lacks specific electron-electron correlation interactions.

Suppose the many-electron Hamiltonian is H_{el}, then the total energy with the SD wavefunction Eq. (2.12) is also defined as the Hartree-Fock energy,

$$E_{el} = \langle\psi|H_{el}|\psi\rangle \equiv E_{\mathrm{HF}}. \tag{2.14}$$

Limited by the SD form, ψ in general is not the eigenstate of H_{el}, therefore $E_{el}(E_{\mathrm{HF}})$ computed this way is not the true ground state energy.

Also remember that ψ is parameterized due to the undecided coefficients during LCAO

when forming molecular orbitals. If we determine the coefficients in χ by minimizing E_{el} via variational theorem, in the end we obtain the minimized Hartree-Fock energy

$$E_{\text{HF}} = \min_{\{c_{ij}\}} \frac{\langle\psi(\{c_{ij}\})|H_{el}|\psi(\{c_{ij}\})\rangle}{\langle\psi(\{c_{ij}\})|\psi(\{c_{ij}\})\rangle} \tag{2.15}$$

where c_{ij} is the component of j-th atomic orbital for the i-th electron. Next we briefly describe the general procedure for the HF method.

2.5.2 Express HF energy as one- and two-electron integrals

Since electrons are indistinguishable, the energy in Eq. (2.14) expressed as the Hamiltonian of all electrons integrated over the SD, with math tricks not shown here, can break down to one- and two-electrons integrated over spin orbitals as

$$E_{el} = \sum_i \langle i|h|i\rangle + \frac{1}{2}\sum_{ij}\left([ii|jj] - [ij|ij]\right) \tag{2.16}$$

where i, j runs over all possible spin orbitals with a given basis set. The electronic energy now is the sum of one-electron integrals

$$\langle i|h|j\rangle = \int d\mathbf{x}\, \chi_i^*(\mathbf{x}) \left(-\frac{1}{2}\nabla^2 - \sum_A \frac{Z_A}{|\mathbf{x} - \mathbf{x_A}|}\right) \chi_j(\mathbf{x}) \tag{2.17}$$

and two-electron integrals

$$[ij|kl] = \int d\mathbf{x_1} d\mathbf{x_2}\, \chi_i^*(\mathbf{x_1})\chi_j^*(\mathbf{x_1}) \frac{1}{|\mathbf{x_1} - \mathbf{x_2}|} \chi_k(\mathbf{x_2})\chi_l(\mathbf{x_2}). \tag{2.18}$$

2.5.3 HF equation obtained from Lagrange multipliers

The minimized HF energy Eq. (2.16) should be stationary for infinitesimal spin orbital changes $\chi_i \to \chi_i + \delta\chi_i$, and also the spin orbitals should be orthogonal to each other. The cost function can be expressed in Lagrange multiplier method with multiplier ϵ_i

$$L(\{\chi_i\}) = \left(\sum_i \langle i|h|i\rangle + \frac{1}{2}\sum_{ij}\left([ii|jj] - [ij|ij]\right)\right) - \sum_i \epsilon_i(\langle i|i\rangle - 1). \tag{2.19}$$

Then, applying $\chi_i \to \chi_i + \delta\chi_i$ and solving for

$$\delta L = 0 \tag{2.20}$$

leads to the *Hartree-Fock equation* for a single electron located at $\mathbf{x}$

$$F(\mathbf{x})\chi_i(\mathbf{x}) = \epsilon_i\chi_i(\mathbf{x}). \tag{2.21}$$

The Fock operator of the single electron located at $\mathbf{x_1}$ is

$$F(\mathbf{x_1}) = h(\mathbf{x_1}) + \sum_j \left(\mathcal{J}_j(\mathbf{x_1}) - \mathcal{K}_j(\mathbf{x_1})\right) \tag{2.22}$$

where the sum is over all participating spin orbitals. Here

$$\mathcal{J}_j(\mathbf{x_1}) = \int d\mathbf{x_2}\,|\chi_j(\mathbf{x_2})|^2 \frac{1}{|\mathbf{x_1} - \mathbf{x_2}|} \tag{2.23}$$

is the *Coulomb term* where $\mathbf{x_2}$ is the location of the second electron. Another term $\mathcal{K}_j$ will turn

χ_i into χ_j, therefore called *exchange term*:

$$\mathcal{K}_j(\mathbf{x_1})\chi_i(\mathbf{x_1}) = \left[\int d\mathbf{x_2}\,|\chi_j(\mathbf{x_2})|^2 \frac{1}{|\mathbf{x_1}-\mathbf{x_2}|}\right]\chi_j(\mathbf{x_1}). \tag{2.24}$$

Since the Fock operator Eq. (2.22) is the sum of all spin orbitals, the HF equation in Eq. (2.21) becomes non-linear. In practice, the HF equation in Eq. (2.21) has to be solved by providing an initial guess of spin orbitals, and then updating the spin orbitals iteratively. As a result, the HF method is also called the self-consistent field (SCF) method.

2.6 FCI: Exact solution within basis set

Due to the intrinsic limitation of the SD wavefunction form used for HF, the energy obtained in Eq. (2.16) does not include all interaction between electrons. At the end of HF, it obtained a set of optimized and orthonormal spin orbitals $\{\chi_i\}$. This naturally leads to a set of SD basis $\{T_{ij\cdots}^{ab\cdots}|\mathrm{HF}\rangle\}$ in which electrons can live in any of the spin orbital in $\{\chi_i\}$, corresponding to different configurations.

For a N_e number of electrons and a basis set resulting N_{so} spin orbitals, the total number of possible SD, i.e., configurations, are

$$\begin{pmatrix} N_{so} \\ N_e \end{pmatrix} = \frac{N_{so}!}{N_e!(N_{so}-N_e)!}. \tag{2.25}$$

Linearly combine all possible SDs results in the FCI wavefunction expression

$$|\Psi\rangle_{\mathrm{FCI}} = |\mathrm{HF}\rangle + \sum_{i<a} c_i^a T_i^a |\mathrm{HF}\rangle + \sum_{i<a,j<b} c_{ij}^{ab} T_{ij}^{ab} |\mathrm{HF}\rangle + \cdots . \tag{2.26}$$

This can be solved exactly by computing the numerical value of matrix elements of H_{el} integrated on basis set $\{T_{ij\cdots}^{ab\cdots}|\mathrm{HF}\rangle\}$ and directly diagonalize the H_{el}. FCI gives the exact solution within a basis, however the cost is too expensive because of the fast growing number of SD basis.

Chapter 3: Quantum computing for quantum chemistry

3.1 Overview

In this chapter, we first explain in Section 3.2 the quantum computer model used throughout the thesis. Then we explain in general how the quantum chemistry problem is mapped to a quantum computer in Section 3.3. With fundamentals being explained, in Section 3.4 we discuss some of the latest methods for fermionic simulation using a quantum computer. In Section 3.5, we explain VQE and the ansatz used, which is the main algorithm used in the rest of the thesis. Lastly, we introduced the standard procedure to implement ansatz circuit through Trotterization in Section 3.6.

3.2 Quantum computing in a nutshell

3.2.1 Quantum computing model

The notion of employing a quantum computer to simulate inherently quantum processes was initially introduced by Richard Feynman in 1982 [19]. Since then, quantum computing and quantum information have emerged as rapidly growing research fields.

In this thesis, we focus on utilizing the circuit model of quantum computing. For practical experimental setups, DiVincenzo's criteria [20] serve as widely accepted guidelines:

1. A scalable physical system with well-characterized qubit;

2. The ability to initialize the state of the qubits to a simple fiducial state;

3. Long relevant decoherence times;

4. A "universal" set of quantum gates;

5. A qubit-specific measurement capability.

In the context of this thesis, a quantum computer can be conceptualized as an advanced black box capable of executing arbitrary and controllable unitary evolution, denoted as U, on a prepared initial state vector, represented as $|\psi\rangle$. Following the application of the evolution operator U to $|\psi\rangle$, users can ascertain the probability distribution of the basis states by conducting repeated measurements of the experiment's outcome.

State vectors within this framework are depicted as tensor products of qubits, which constitute the fundamental units spanned by two orthonormal bases within a linear vector space

$$|0\rangle = \begin{bmatrix} 1 \\ \\ 0 \end{bmatrix}; \quad |1\rangle = \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix}. \tag{3.1}$$

Numerous qubit implementations have been achieved using various technologies, including trapped-ion qubits [21–25], superconducting qubits [26–30], neutral atom qubits [31–33], photonic qubits [34, 35], and others. However, these quantum hardware platforms are susceptible to various types of noise, limiting the number of operations feasible on a quantum computer. This period, termed the pre-fault-tolerant regime, persists until a fault-tolerant quantum computer is realized. Tremendous efforts have been dedicated to mitigating errors and noise in hardware

through error-correction codes and fault-tolerant implementations [36, 37], with several demonstrations across different platforms [38–41]. Nonetheless, achieving fully fault-tolerant quantum computing is widely regarded as still years, if not decades, away.

Beyond quantum simulation of fermionic systems, quantum computers hold promise in various other fields, including quantum cryptography facilitated by Shor's algorithm [42] and search problems addressed by Grover's algorithm [43].

3.2.2 Commonly used quantum gates

The arbitrary unitary evolution U can be expressed as a sequence of standard unitary operations known as quantum logical gates. In this paper, we primarily utilize single-qubit and two-qubit gates, each applying unitary transformations to specified single qubits or pairs of qubits.

Commonly used single-qubit gates are:

$$R_x(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \tag{3.2}$$

$$R_y(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \tag{3.3}$$

$$R_z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \tag{3.4}$$

$$\sigma_x \equiv X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{3.5}$$

$$\sigma_y \equiv Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{3.6}$$

$$\sigma_z \equiv Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3.7}$$

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \tag{3.8}$$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{3.9}$$

$$H = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{3.10}$$

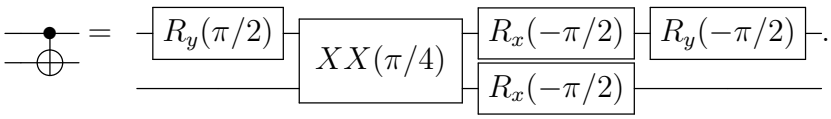
Commonly used two-qubit gates are:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.11}$$

$$XX(\theta/2) \equiv R_{xx}(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & 0 & 0 & -i\sin(\frac{\theta}{2}) \\ 0 & \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) & 0 \\ 0 & -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) & 0 \\ -i\sin(\frac{\theta}{2}) & 0 & 0 & \cos(\frac{\theta}{2}) \end{pmatrix}. \tag{3.12}$$

A set of quantum gates is called universal if any U can be decomposed as the finite sequence of the gate set. For example, the one of the most commonly used universal gate sets for trapped-ion quantum computers is $\{R_{XX}, R_x, R_z\}$.

When transpiling quantum circuits for specific quantum hardware, it is crucial to adapt the circuit from a given gate set to the native gates supported by the hardware. A common adaptation is the translation of CNOT gates into the native XX gate sequence, which is typical in systems using the Mølmer-Sørensen protocol in trapped-ion quantum computers, see more about the physics behind in Appendix C.2. A CNOT to XX translation can be done via the following [44]



(3.13)

3.3 Connecting quantum chemistry and quantum computing with second quantization

3.3.1 Wavefunction in second quantized form

The wavefunction of the SD form in Eq. (2.12) explicitly tracks the location of each electron. After HF, the coefficients within the spin orbital basis set $\{\chi_i\}$ are finalized, and we can then express the wavefunction more conveniently as Fock state as we only need to keep track whether a spin orbital χ_i is populated by an electron. If the electron is present in χ_i then it is a $|1_i\rangle$ otherwise it is a $|0_i\rangle$ state. This binary Fock state is then naturally connected with the qubit representation in Eq. (3.1). Examples of two electron configurations of a H_2 molecule with STO-3G basis are shown in Fig. 3.1.

3.3.2 State transition in second quantized form

Wavefunction represented in the Fock state can change between states through creation $a_j^\dagger$ (annihilation a_j) operators. The creation operators change the state as

$$a_j^\dagger|n_1,\cdots,0_j,\cdots,n_N\rangle = (-1)^{\sum_{k<j} n_k}|n_1,\cdots,1_j,\cdots,n_N\rangle \tag{3.14}$$

$$a_j^\dagger|n_1,\cdots,1_j,\cdots,n_N\rangle = 0 \tag{3.15}$$

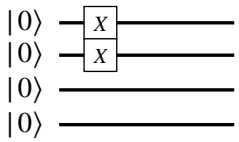
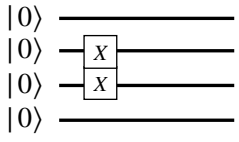
MO diagram	First quantization SD wavefunction	Second quantization wavefunction (Fock state)	Quantum circuit wavefunction
$\|\psi_0\rangle$	$\frac{1}{\sqrt{2}}\begin{vmatrix}\chi_0(\mathbf{x_1}) & \chi_1(\mathbf{x_1})\\ \chi_0(\mathbf{x_2}) & \chi_1(\mathbf{x_2})\end{vmatrix}$	$\|\psi_0\rangle = \|1100\rangle$	
 $\|\psi_0^2\rangle$	$\frac{1}{\sqrt{2}}\begin{vmatrix}\chi_1(\mathbf{x_1}) & \chi_2(\mathbf{x_1})\\ \chi_1(\mathbf{x_2}) & \chi_2(\mathbf{x_2})\end{vmatrix}$	$\|\psi_0^2\rangle = \|0110\rangle$	

Figure 3.1: The figure provides an example of how the same wavefunction is expressed in different representations. In MO diagram, $\uparrow$ ($\downarrow$) represents the α (β) spin.

while the annihilation operator change the state via

$$a_j|n_1, \cdots, 1_j, \cdots, n_N\rangle = (-1)^{\sum_{k<j} n_k}|n_1, \cdots, 0_j, \cdots, n_N\rangle \tag{3.16}$$

$$a_j|n_1, \cdots, 0_j, \cdots, n_N\rangle = 0 \tag{3.17}$$

where the phase $(-1)^{\sum_{k<j} n_k}$ is the consequence of the anti-commutation relation that fermions (electrons) need to obey. The anti-commutation relation states

$$\{a_j, a_k\} = 0\ , \quad \{a_j^\dagger, a_k^\dagger\} = 0\ , \quad \{a_j, a_k^\dagger\} = \delta_{jk}\mathbf{1}\ , \tag{3.18}$$

where $\{A, B\}$ denotes the anti-commutator $AB + BA$, δ_{jk} is the Kronecker delta, and $\mathbf{1}$ is the identity operator.

On the other hand, qubit states can be updated via the following operators

$$\sigma^+ = |1\rangle\langle 0| = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} = \frac{X - iY}{2} \tag{3.19}$$

$$\sigma^+|0\rangle = |1\rangle \tag{3.20}$$

$$\sigma^+|1\rangle = 0 \tag{3.21}$$

and

$$\sigma^- = |0\rangle\langle 1| = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} = \frac{X + iY}{2} \tag{3.22}$$

$$\sigma^-|1\rangle = |0\rangle \tag{3.23}$$

$$\sigma^-|0\rangle = 0, \tag{3.24}$$

while the phase can also be extracted by applying Z operators

$$Z|0\rangle = |0\rangle \tag{3.25}$$

$$Z|1\rangle = -|1\rangle. \tag{3.26}$$

Then, with the following transformation

$$\begin{aligned} a_j^\dagger &= 1^{\otimes n-j-1} \otimes \sigma_+ \otimes \sigma_z^{\otimes j} \\ a_j &= 1^{\otimes n-j-1} \otimes \sigma_- \otimes \sigma_z^{\otimes j} \end{aligned} \tag{3.27}$$

we are able to perform the same operation on qubits to change the qubit states as what we did to fermions on Fock space.

For a concrete example, suppose we have a second-quantized operator $a_1^\dagger a_0$, then after the transformation, it becomes

$$\begin{aligned} a_1^\dagger a_0 &= (Z_0 \otimes \sigma_1^+)(\sigma_0^- \otimes I_1) \\ &= (Z_0\, \sigma_0^-) \otimes (\sigma_1^+ I_1) \\ &= \sigma^- \otimes \sigma^+ \\ &= \frac{X + iY}{2} \otimes \frac{X - iY}{2} \\ &= \frac{1}{4}(XX - iXY + iYX + YY). \end{aligned} \tag{3.28}$$

If the operator shows up in a Hamiltonian, which is real, then it usually was taken by the real part, which can be obtained by

$$a_1^\dagger a_0 + c.c. = \frac{1}{2}(XX + YY) \tag{3.29}$$

while if the operator serves as the unitary evolution showing up in the exponent, then we need the imaginary part, which is

$$a_1^\dagger a_0 - c.c. = \frac{1}{2}(iYX + iXY). \tag{3.30}$$

The straightforward transformation in Eq. (3.27) is nothing but the well-known, popular choice Jordan-Wigner (JW) transformation [45]. Other transformations [45–47] are also available and finding a transformation that can help minimize the number of two-qubit gates is one of the

major tasks in Chapter 5 and Chapter 6.

3.3.3 Molecular Hamiltonian in second quantized form

The second quantized molecular Hamiltonian can be expressed as

$$H = \sum_{p,q} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{p,q,r,s} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s, \tag{3.31}$$

where h_{pq} and and h_{pqrs} are the single-electron integral and two-electron integrals defined in Eq. (2.17) and Eq. (2.18). Those coefficients are outcomes of the classical HF calculation. The Hamiltonian in Eq. (3.31) after proper transformation will result in linear summation of Pauli strings. For example, H_2 at bond length 1.5 Å with STO-3G basis set and parity transformation and two-qubit reduction based on symmetry will result in the two-qubit Hamiltonian

$$\begin{aligned} H = &- 1.0096446943601909\, II \\ &+ 0.1291013128871107\, IZ \\ &- 0.1291013128871106\, ZI \\ &- 0.0041889582600267\, ZZ \\ &+ 0.2295359360597018\, XX. \end{aligned} \tag{3.32}$$

The correctness of the Hamiltonian can be verified by expanding the Pauli string into the matrix form and then directly diagonalizing it. The lowest eigenvalue is the true ground state energy and should match the FCI energy computed by other quantum chemistry software calculations with the same basis set. Since the dimension of the matrix resulting from expanding

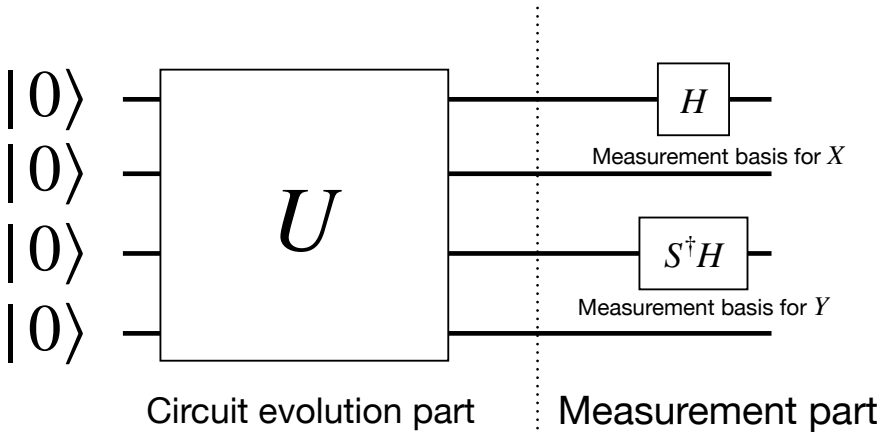


Figure 3.2: This figure shows how to measure $\langle XIYZ \rangle$ after circuit execution U. Measurement bases are appended at the end of the circuit so that the measurement is always on computational (Z) basis .

the Pauli string is $2^n \times 2^n$ for n spin orbitals, this numerical diagonalization becomes intractable rapidly. As a result, this verification is only used as a sanity check for small systems.

3.3.4 Measurement of Hamiltonian

The last piece of the puzzle of a complete mapping of a chemistry problem into a quantum computing problem is the measurement. Since by default the quantum computer measures only on computational Z basis, a proper basis transformation needs to be applied to the circuit before the final measurement. For $X(Y)$ basis, $H(S^\dagger H)$ basis transformations are required while Z, I basis do not need additional basis transformations. An example is shown in Fig. 3.2.

3.4 Simulating fermionic system with quantum computer overview

Simulating fermionic matter on a quantum computer has recently been receiving much attention. Already available in the literature are various chemistry and nuclear physics simulation results [4,48–52], performed across multiple quantum computing platforms, including supercon-

ducting [48, 50, 51] and trapped-ion [4, 49, 52] based approaches. The spotlight on the simulation of fermionic systems on a quantum computer is not accidental. Simulations of these systems are useful for furthering fundamental science and practical engineering [53, 54], and quantum computers are expected to enable the quantum simulations of fermions undergoing local interactions [19, 55], a task believed to be classically difficult to scale.

Broadly speaking, simulations of fermionic systems on a quantum computer may be classified into two categories: a variational, quantum-classical hybrid simulation [9], suitable for imperfect, pre-fault tolerant (pre-FT) quantum computers, and a Hamiltonian dynamics simulation based on pure quantum simulation algorithms [56], typically considered in fault-tolerant (FT) quantum computers. In the context of estimating the ground-state energy of a fermionic system, the former leverages efficient preparation of ansatz states and evaluation of operator expectation values, both enabled by quantum computers. The latter leverages the ability of a quantum computer to efficiently simulate evolution of quantum systems with a local Hamiltonian, which, when combined with quantum phase estimation [57], allows us to evaluate the ground-state energy of the system.

3.4.1 Fermonic simulation in Fault-Tolerant regime

Although our main focus in this thesis was to optimize the circuit for pre-FT regime, for completeness, we also explain briefly how to obtain the ground state energy once we have a FT quantum computer. In the case where we consider a quantum dynamics simulation approach

more suitable for the FT regime, we aim to implement the evolution operator

$$U_{\text{evo}} = e^{-iHt} \tag{3.33}$$

on a quantum computer, where H is the system Hamiltonian and t is the duration by which we desire to evolve the system forward in time. Roughly, once the initial state $|\psi_{\text{init}}\rangle$, sufficiently close to the ground state $|\psi_{\text{ground}}\rangle$, is evolved to

$$|\psi(t)\rangle \approx e^{-iE_{\text{ground}}t}|\psi_{\text{init}}\rangle \tag{3.34}$$

up to the closeness, quantum Fourier transform may be used to estimate the phase angle $E_{\text{ground}}t$, thus extracting the ground-state energy E_{ground}. Detailed discussion on the circuit layout that implements the quantum phase estimation algorithm is available in [58]. Note, an efficient implementation of the quantum Fourier transform on a FT quantum computer is known [59]. Therefore, the problem of an efficient energy-spectra calculation on a quantum computer in the FT regime boils down to the problem of efficiently implementing real-time quantum dynamics simulations.

3.4.2 Fermionic Simulation of pre-FT regime

In the case where we consider a quantum-classical hybrid approach more suitable for the pre-FT regime, we consider the widely-adopted technique of the variational quantum eigensolver (VQE) [9]. Specifically, we aim to implement a unitary ansatz evolution operator U_{ansatz} on a quantum computer, an example of which is the well-established unitary coupled cluster (UCC) ansatz [60, 61]. By tuning the variational parameters in U_{ansatz}, in a typical VQE approach,

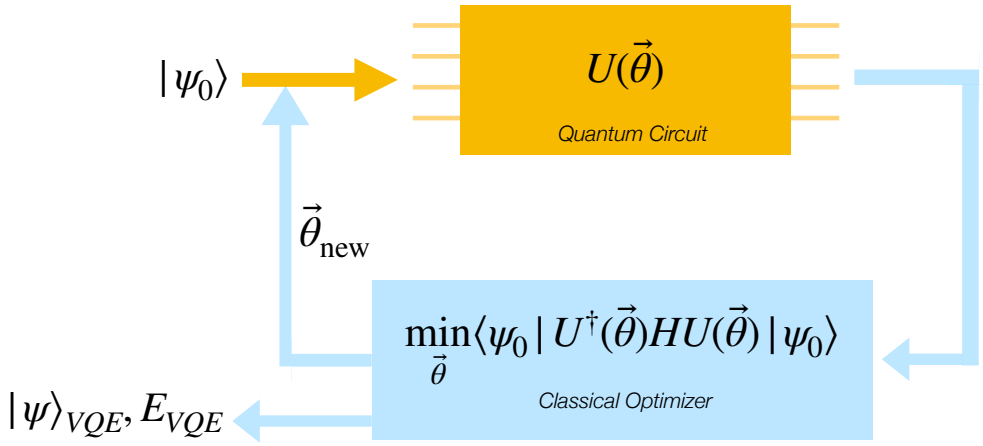


Figure 3.3: The most basic VQE cycle. $|\psi_0\rangle$ is the initial state, usually the Hartree-Fock state. We start by a controllable unitary evolution $U(\vec{\theta})$ performed by a quantum computer to obtain $U(\vec{\theta})|\psi_0\rangle$. By measurement procedure detailed in Section 3.3.4, eventually we obtain the expectation energy. The classical optimizer then suggests the next parameters to try or decide to terminate.

$\langle\psi_0|U^\dagger_{\text{ansatz}} H U_{\text{ansatz}}|\psi_0\rangle$ is minimized, where $|\psi_0\rangle$ is an initial state that is assumed to be close to the target ground state of H and can readily be prepared on a quantum computer. The goal of this hybrid approach is then to estimate the ground state energy of the fermionic system with the Hamiltonian of the form Eq. (3.31).

3.5 Variational Quantum Eigensolver

3.5.1 Basic VQE

In the pre-FT regime, we aim to calculate the ground state energy of the system whose Hamiltonian is given in the form Eq. (3.31). This is typically achieved by the VQE approach. [9, 62, 63] In this approach, by iteratively calling the quantum computer to compute the energy of parametrized ansatz states, one aims to minimize the energy and variationally obtain the ground state of the target system. A diagram in Fig. 3.3 shows the most basic VQE framework. We

consider in this thesis a type of ansatz states $|\Psi_{\text{ansatz}}\rangle$ that are transformed from an initial state $|\Psi_0\rangle$ using a parametrizable unitary ansatz evolution operator U_{ansatz}, as $|\Psi_{\text{ansatz}}\rangle = U_{\text{ansatz}} |\Psi_0\rangle$.

To be complete, $|\Psi_0\rangle$ can be a ground-state Hartree-Fock wavefunction, multiconfigurational self-consistent field [64], or density-matrix renormalization group calculations [65]. Many choose to work with the former for its simplicity, including [1, 4].

For brevity, we drop the subscript in U_{ansatz} in the following text. Since U is a unitary operator, it can always be written in the exponential form $e^{Z-Z^\dagger}$ and Z is in turn parametrized. By varying the parameters in Z, the ansatz state energy $\langle\Psi_{\text{ansatz}}| H |\Psi_{\text{ansatz}}\rangle = \langle\Psi_0| U^\dagger H U |\Psi_0\rangle$ is variationally minimized. With a proper fermion to qubit basis transformation the energy expectation value can be evaluated efficiently on a quantum computer, where the quantum resource cost of this hybrid approach is the implementation cost of the operator U. We note that the procedures proposed in this section can be implemented with any ansatz with a unitary evolution operator U.

3.5.2 UCCSD ansatz

As a concrete example for discussion, we use the widely-adopted UCC ansatz with single and double excitations (UCCSD), a systematic method that is universally applicable to any quantum hardware backend. Other ansatzes, for instance the k-UpCCGSD ansatz of Lee et al. [66] or the UCCGSD ansatz of Grimsley et al. [67], can also be readily applied.

The UCCSD method starts with a ground state Hartree-Fock wavefunction $|\Psi_0\rangle$ that can be easily calculated on a classical computer and implemented on a quantum computer. It then

evolves $|\Psi_0\rangle$ with a unitary operator

$$U = e^{Z_{\mathrm{UCCSD}} - Z_{\mathrm{UCCSD}}^{\dagger}} \tag{3.35}$$

where

$$Z_{\mathrm{UCCSD}} = \sum_{l=1}^{2} Z_l \tag{3.36}$$

is the so-called cluster operator. $Z_{1,2}$ are the single and double excitation operators and are written in second quantization as

$$\begin{aligned} Z_1 &= \sum_{\substack{p\in\mathrm{virt} \\ r\in\mathrm{occ}}} t_{pr} a_p^{\dagger} a_r, \\ Z_2 &= \sum_{\substack{p,q\in\mathrm{virt} \\ r,s\in\mathrm{occ}}} t_{pqrs} a_p^{\dagger} a_q^{\dagger} a_r a_s, \end{aligned} \tag{3.37}$$

where t_{pr} and t_{pqrs} are variational parameters and "virt" and "occ" denote virtual and occupied levels respectively. Notice that for the UCCSD ansatz, to minimize the size of the circuit that implements U, not all terms in the cluster operator are needed to achieve a certain precision. In other words, only a subset of all the parameters t_{pq} and t_{pqrs} need to be included in the VQE approach, while the rest can be set to zero. In fact, it is very much the name of the game in the pre-FT regime to find a minimal set of variational parameters so that the final ground state energy satisfies certain error criteria.

3.6 Ansatz circuit implementation with Trotterization

There is no straightforward way to implement a generic unitary in the form Eq. (3.35). The standard procedure is to break down the unitary through Trotterization into standard unitary units which can be implemented with known standard procedure.

For completeness, we start by introducing the $2k$th order PF algorithm. Assuming the JW transformation has been performed on the Hamiltonian H in Eq. (3.31), the unitary evolution operator we aim to implement may be written as

$$\exp\left(-i\sum_{j=1}^{L}\theta_j\hat{\sigma}^{(j)}\right) \approx [S_{2k}(\lambda)]^r, \tag{3.38}$$

where $\lambda := 1/r$, $\hat{\sigma}^{(j)} = \bigotimes_i \tilde{\sigma}^{(i,j)} + \text{h.c.}$, where $\tilde{\sigma}^{(i,j)} \in \{\sigma_+, \sigma_-, \sigma_z\}$ and h.c. denotes the Hermitian conjugate operator, and

$$\begin{aligned}
S_1(\lambda) &:= \prod_{j=1}^{L}\exp(-i\theta_j\hat{\sigma}^{(j)}\lambda), \\
S_2(\lambda) &:= \prod_{j=1}^{L}\exp(-i\theta_j\hat{\sigma}^{(j)}\lambda/2)\prod_{j=L}^{1}\exp(-i\theta_j\hat{\sigma}^{(j)}\lambda/2), \\
S_{2k}(\lambda) &:= [S_{2k-2}(p_k\lambda)]^2 S_{2k-2}((1-4p_k)\lambda)[S_{2k-2}(p_k\lambda)]^2,
\end{aligned} \tag{3.39}$$

with $p_k := 1/(4-4^{1/(2k-1)})$ for $k > 1$ [68]. Inspecting (3.38) and (3.39), we observe that the individual exponential terms, hereafter referred to as a Trotter term, are of the form

$$\exp\left[-i\theta'_j(\bigotimes_i \tilde{\sigma}^{(i,j)} + \text{h.c.})\right], \tag{3.40}$$

where θ'_j is a suitably scaled θ_j. A standard circuit that implements a Trotter term is readily available in [57], also see template in Fig. 6.4 (b). Employing the circuit tricks in [4], in Fig. 3.4, we show an optimized quantum circuit that implements an example Trotter term $e^{-i\theta/2(\sigma_+\sigma_+\sigma_-\sigma_- + \text{h.c.})}$ resulting from a four-qubit UCC ansatz $e^{\theta a_0^\dagger a_1^\dagger a_2 a_3 - h.c.}$ with a double excitation term $a_0^\dagger a_1^\dagger a_2 a_3$.

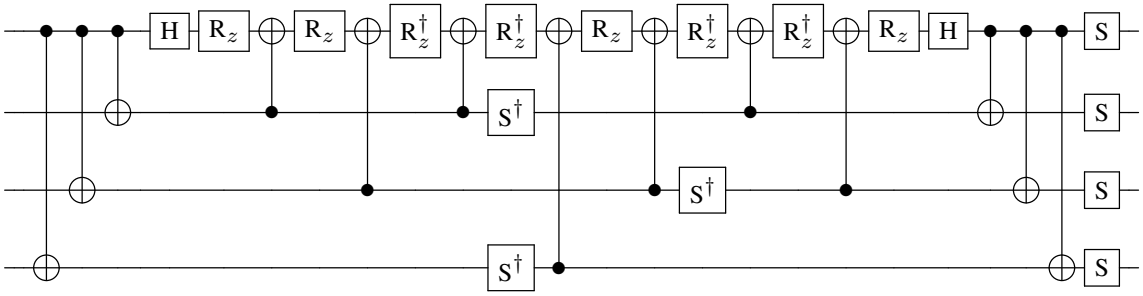


Figure 3.4: Standard two-body interaction circuit that implements $\exp[-i(\theta/2)\sigma_+ \otimes \sigma_+ \otimes \sigma_- \otimes \sigma_- + \text{h.c.}]$. Following the steps detailed in [4] closely, expanding $\sigma_+\sigma_+\sigma_-\sigma_- + \text{h.c.}$ (we suppress $\otimes$ hereafter whenever contextually clear) into the particular ordering of $\sigma_x\sigma_x\sigma_x\sigma_x$, $\sigma_x\sigma_x\sigma_y\sigma_y$, $\sigma_x\sigma_y\sigma_y\sigma_x$, $\sigma_x\sigma_y\sigma_x\sigma_y$, $\sigma_y\sigma_y\sigma_x\sigma_x$, $\sigma_y\sigma_x\sigma_x\sigma_y$, $\sigma_y\sigma_x\sigma_y\sigma_x$, and $\sigma_y\sigma_y\sigma_y\sigma_y$, and implementing them one after another with the last qubit as the target qubit, we obtain the circuit shown in this figure after applying the circuit optimization routines detailed in [4, 5].

Chapter 4: Perturbation assisted quantum simulation

4.1 Overview

In this chapter, we describe a general framework that leverages the power of perturbation theory to optimize VQE-based quantum simulations by predicting the subset of variational parameters to include in the ansatz and correcting the VQE result via post processing. The framework optimizes both the total number of VQE executions as well as the size of the ansatz state preparation circuits used to reach convergence in the ground-state energy estimate.

In Section 4.2 we outline the framework. The derivation of a simple perturbation scheme that can be straightforwardly implemented in the framework is shown in Section 4.3, which we hereafter refer to as a hybrid second order Møller-Plesset perturbation (HMP2) method. In Section 4.5 we present a classically simulated comparison between the HMP2 scheme and a more conventional VQE approach [4], by computing the ground state energy of a water molecule using the UCCSD ansatz.

4.2 Perturbative predictor and corrector

We detail, in this section, a general, systematic framework of using perturbation methods to improve quantum resources required for the VQE approach. Specifically, we aim to rapidly

converge to the ground state energy with small quantum circuits.

Any general (not specific to any system) unitary ansatz operator can be written as

$$U = e^{Z - Z^\dagger} \ , \ Z = \sum_{\{\alpha\}} f_\alpha(\{t_\beta\}) D_\alpha, \tag{4.1}$$

where the summation is over a set of orbital sequences $\{\alpha\}$, f_α are real functions of the set of variational parameters $\{t_\beta\}$, and D_α is a generic orbital substitution operator that substitutes the orbitals in a wavefunction according to the orbital sequence α. For instance, a single orbital substitution operator $D_{p^\dagger q} = a_p^\dagger a_q$ substitutes orbital q with p. We can then define the orbital substituted wavefunction

$$|\Psi_{D_\alpha}\rangle = D_\alpha \, |\Psi_0\rangle \, . \tag{4.2}$$

For a specific general ansatz approach there is a full set of variational parameters $T_{\text{full}} = \{t_\beta\}$ that can be used to parametrize the ansatz evolution operator. In the case of UCCSD, by comparing Eq. (3.37) with Eq. (4.1), we note that

$$T_{\text{full}} = \{t_{pr}, t_{pqrs} \mid p, q \in \text{virt}; \ r, s \in \text{occ}\}, \tag{4.3}$$

$$\{\alpha\} = \{p^\dagger r, p^\dagger q^\dagger r s \mid p, q \in \text{virt}; \ r, s \in \text{occ}\}, \tag{4.4}$$

and $f_\alpha = t_\alpha$.

Note further that we can generate a set of ansatz evolution operators $\{U_{T_i}\}$, each element of which is the unitary operator that induces an ansatz state with a subset T_i of the full variational

parameters T_{full} ($T_i \subset T_{\text{full}}$). The rest of the parameters in the complement set $T_{\text{full}} \setminus T_i$ are fixed to an ordered set of default values $T_{\text{full}}^{(0)}$. Without loss of generality, we assume the default values are absorbed into f_α in Eq. (4.1) and the elements in the complement set $T_{\text{full}} \setminus T_i$ are simply set to zero. For every U_{T_i} in $\{U_{T_i}\}$, the ansatz state energy can be variationally minimized as

$$E_{T_i} = \min_{\text{elements}(T_i)\in\mathbb{R}} \langle\Psi_0| U_{T_i}^\dagger H U_{T_i} |\Psi_0\rangle , \tag{4.5}$$

where $\text{elements}(T_i)$ denotes the elements of the set T_i. For an application of a particular general ansatz through the VQE approach in the pre-FT regime, one aims to find a U_{T_i} that has the least circuit complexity while satisfying

$$|E_{T_i} - E_{T_{\text{full}}}| < \varepsilon \tag{4.6}$$

with ε a pre-determined error criterion.

A concrete strategy to reach convergence with minimal circuit complexity is to start with an operator with a small set of variational parameters that is known to be necessary (but not necessarily sufficient) to achieve the error threshold and iteratively grow it. Going from the mth iteration to the $(m+1)$th one, the set of variational parameters in the operators grows and satisfies $T_m \subset T_{m+1}$. A critical part in ensuring a rapid convergence is the selection of the additional variational parameter(s) to include between iterations. In the case of UCCSD, they directly correspond to additional individual excitation terms in Eq. (3.37) used to prepare the ansatz state. Previous works have, e.g., used the full configuration interaction (FCI) results [4] in the case the system is sufficiently small to be simulatable on a classical computer, and have

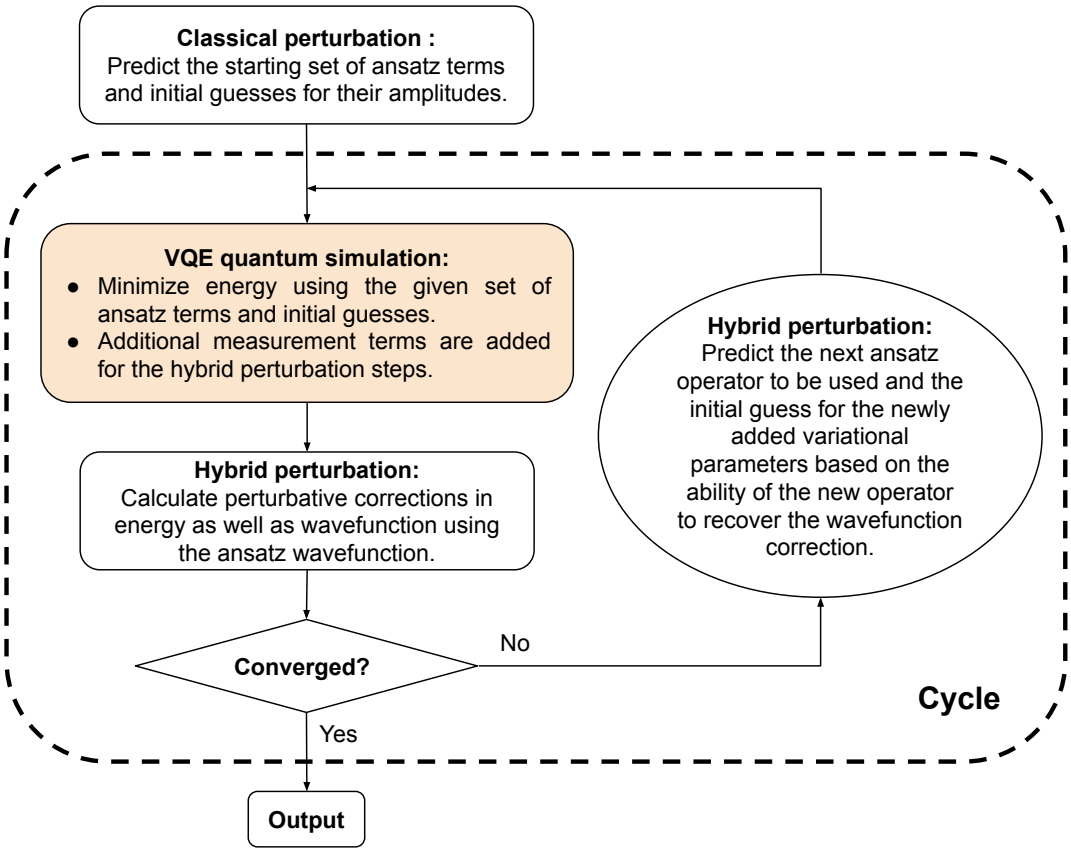


Figure 4.1: Flow diagram for the proposed framework incorporating perturbation methods to VQE simulations. See main text for detailed description.

demonstrated the significance of the excitation term selection for the resource requirement. Here, we propose to iteratively select the next variational parameters to include in the ansatz state preparation based on the size of the perturbatively predicted wavefunction correction amplitude for a given ansatz state whose variational parameters are already optimized via the conventional means of VQE. Our strategy also evaluates a perturbative energy correction in addition to the conventional VQE approach, directly contributing to the fast convergence, while incurring no overhead in the pre-FT regime resource requirement.

Fig. 4.1 shows the flow diagram of the proposed framework. As in the conventional VQE approach for fermionic simulations, we start with the ground state Hartree-Fock wavefunction of

the single particle Hamiltonian as $|\Psi_0\rangle$. The first step is to determine the initial evolution operator U_{T_1} to use for the first iteration of VQE. As an example, using the two-particle Hamiltonian as perturbation, energy and wavefunction corrections can be calculated using classical algorithms with relative ease. From the perturbed wavefunction, we can extract the amplitudes of individual excited states in the single-particle Hamiltonian basis. These amplitudes will serve as the initial guesses of the variational parameters for the first round of VQE simulation, which could significantly reduce the number of evaluations of the quantum circuit comparing to all-zeros or random initial guesses [69]. If we demand the energy convergence criteria of δ_E, then the entire simulation is considered to be converged when the magnitude of energy change associated with an addition of any extra ansatz term is smaller than δ_E. We include in the initial ansatz set the ansatz terms whose contribution to the correlation energy is greater than $f(\delta_E)$, where a standard choice of function $f(\delta_E)$ may simply be δ_E.

With the initial set of ansatz terms and their initial variational parameter values, we run the first round of VQE simulation to minimize the energy

$$E_{T_1} = \langle \Psi_0 | \, U_{T_1}^\dagger H U_{T_1} \, | \Psi_0 \rangle \, . \tag{4.7}$$

Once the energy is minimized and the ansatz state converges, we proceed to compute the expectation values of a set of operators that were chosen in advance to inform us about the perturbation around the converged ansatz state. We refer to this method as a hybrid perturbation since unlike the conventional perturbation where we know the unperturbed wavefunction in advance, we start the perturbation from the converged VQE ansatz state.

Based on the additional measurements performed on a quantum computer, we may now

use the hybrid perturbation method to calculate corrections to the correlation energy and the wavefunction. The resulting total energy of this cycle is the summation of the energy correction and the VQE energy. Before starting the next cycle, we first identify a pool of operators that can be the immediate successor of the current operator. Then we evaluate how much each operator in the pool can account for towards the perturbative wavefunction correction. The operator that can recover the wavefunction correction the most gets the nod for the next cycle and the way it recovers the correction informs the initial guesses of the newly added variational parameters in the next cycle. A more detailed illustration of such a procedure is given in the next section. The initial variational parameter values of the ansatz terms that continue to be a part of the next VQE simulation may simply be imported from the cycle before.

The cycle iterates the outlined procedure of running the VQE simulation and the hybrid perturbation calculation, until convergence of the total energy is achieved. We next detail the steps of a simple implementation of our hybrid perturbation framework that is inspired by the classical MP2 method.

4.3 HMP2 method

We derive in this section our hybrid perturbation method, inspired by the Møller-Plesset [70] perturbation theory, applied to the VQE simulation. Our hybrid MP2-based perturbation method, HMP2, aims to help each VQE cycle in terms of both improving the resulting energy via a perturbative energy correction and optimizing the ansatz operator to use in the next cycle through a perturbative wavefunction correction.

In the mth VQE cycle, we can write the energy of the ansatz state E_{T_m} in the context of

first-order perturbation theory, i.e.,

$$\begin{aligned} E_{T_m} &= \langle \Psi_0 | \, U_{T_m}^\dagger H U_{T_m} \, | \Psi_0 \rangle \\ &= \langle \Psi_0 | \, F + (U_{T_m}^\dagger H U_{T_m} - F) \, | \Psi_0 \rangle \\ &= \langle \Psi_0 | \, F \, | \Psi_0 \rangle + \langle \Psi_0 | \, U_{T_m}^\dagger H U_{T_m} - F \, | \Psi_0 \rangle \\ &= E_0 + (E_{T_m} - E_0) \\ &= E^{(0)} + E_{T_m}^{(1)}, \end{aligned} \tag{4.8}$$

where F is the Fock operator, E_0 is the sum of orbital energies, $E^{(0)} = E_0$ is the zeroth-order energy, and $E_{T_m}^{(1)} = E_{T_m} - E_0$ is the first-order correction energy. Based on perturbation theory, a second-order correction to the energy can now be written as

$$E_{T_m}^{(2)} = \sum_{\{\alpha'\}} \frac{\left| \left\langle \Psi_{D_{\alpha'}} \right| V_{T_m} \left| \Psi_0 \right\rangle \right|^2}{\Delta E_{D_{\alpha'}}}, \tag{4.9}$$

where $V_{T_m} = U_{T_m}^\dagger H U_{T_m} - F$. The set of orbital sequences $\{\alpha'\}$ does not necessarily need to be the same as the set $\{\alpha\}$ used in Eq. (4.1) and should be chosen according to accuracy requirement and resource constraints. See Section 5.6 for more detailed discussion. In our case study using the UCCSD ansatz, we choose $\{\alpha'\} = \{\alpha\}$. The energy $\Delta E_{D_{\alpha'}}$ is defined as the orbital energy difference of the orbital substitutions. For instance, $\Delta E_{D_{p^\dagger q}} = E_q - E_p$, where E_p and E_q are the orbital energies of the pth and the qth orbitals, respectively. Inserting $V_{T_m} = U_{T_m}^\dagger H U_{T_m} - F$ in

Eq. (4.9), the numerator becomes

$$\begin{aligned}\left|\left\langle\Psi_{D_{\alpha'}}\right| V_{T_m}\left|\Psi_0\right\rangle\right|^2 &= \left|\left\langle\Psi_{D_{\alpha'}}\right| U_{T_m}^\dagger H U_{T_m} - F\left|\Psi_0\right\rangle\right|^2 \\ &= \left|\left\langle\Psi_{D_{\alpha'}}\right| U_{T_m}^\dagger H U_{T_m}\left|\Psi_0\right\rangle - \left\langle\Psi_{D_{\alpha'}}\right| F\left|\Psi_0\right\rangle\right|^2 \\ &= \left|\left\langle\Psi_{D_{\alpha'}}\right| U_{T_m}^\dagger H U_{T_m}\left|\Psi_0\right\rangle\right|^2,\end{aligned} \tag{4.10}$$

where we used $\left\langle\Psi_{D_{\alpha'}}\right| F\left|\Psi_0\right\rangle = 0$.

In order to apply the VQE results, especially the ansatz wavefunction $\left|\Psi_{\text{ansatz}}^{T_m}\right\rangle$ directly to the perturbation calculation, we proceed as follows. For brevity, we introduce $\tilde{Z}_{T_m} = Z_{T_m} - Z_{T_m}^\dagger$ with $U_{T_m} = e^{\tilde{Z}_{T_m}}$ and $\tilde{D}_{\alpha'} = D_{\alpha'} - D_{\alpha'}^\dagger$. Inserting these into Eq. (4.10), we obtain

$$\begin{aligned}\left|\left\langle\Psi_{D_{\alpha'}}\right| U_{T_m}^\dagger H U_{T_m}\left|\Psi_0\right\rangle\right|^2 &= \left|\left\langle\Psi_0\right| \tilde{D}_{\alpha'}^\dagger U_{T_m}^\dagger H U_{T_m}\left|\Psi_0\right\rangle\right|^2 \\ &= \left|\left\langle\Psi_{\text{ansatz}}^{T_m}\right| U_{T_m}\tilde{D}_{\alpha'}^\dagger U_{T_m}^\dagger H\left|\Psi_{\text{ansatz}}^{T_m}\right\rangle\right|^2.\end{aligned} \tag{4.11}$$

Next we Taylor expand the $e^{\pm\tilde{Z}_{T_m}}$ in $e^{\tilde{Z}_{T_m}}\tilde{D}_{\alpha'}^\dagger e^{-\tilde{Z}_{T_m}} H$ up to first order in $\tilde{Z}_{T_m}$. This is consistent with our choice of the first-order PF algorithm for the implementation of the ansatz. We obtain

$$\begin{aligned}&\left|\left\langle\Psi_{\text{ansatz}}^{T_m}\right| e^{\tilde{Z}_{T_m}}\tilde{D}_{\alpha'}^\dagger e^{-\tilde{Z}_{T_m}} H\left|\Psi_{\text{ansatz}}^{T_m}\right\rangle\right|^2 \\ =&\left|\left\langle\Psi_{\text{ansatz}}^{T_m}\right| (1+\tilde{Z}_{T_m}+\cdots)\tilde{D}_{\alpha'}^\dagger(1-\tilde{Z}_{T_m}+\cdots)H\left|\Psi_{\text{ansatz}}^{T_m}\right\rangle\right|^2 \\ \approx&\left|\left\langle\Psi_{\text{ansatz}}^{T_m}\right| \tilde{D}_{\alpha'}^\dagger H - \tilde{D}_{\alpha'}^\dagger\tilde{Z}_{T_m}H + \tilde{Z}_{T_m}\tilde{D}_{\alpha'}^\dagger H\left|\Psi_{\text{ansatz}}^{T_m}\right\rangle\right|^2.\end{aligned} \tag{4.12}$$

With Eq. (4.9) and Eq. (4.12) we can obtain the energy correction to the VQE result.

We note that $\tilde{D}_{\alpha'}^\dagger H - \tilde{D}_{\alpha'}^\dagger\tilde{Z}_{T_m}H + \tilde{Z}_{T_m}\tilde{D}_{\alpha'}^\dagger H$ is a sum of products of Pauli operators $\sum_j \epsilon_j\hat{\sigma}^{(j)}$ in the qubit basis, after applying a suitable fermion to qubit basis transformation. Since

$\hat{\sigma}^{(j)}$ have eigenvalues $+1$ or -1, denoting $\left|\Psi_j^{(p)}\right\rangle$ and $\left|\Psi_j^{(m)}\right\rangle$ as the eigenvectors with the respective eigenvalues, we may then write

$$\begin{aligned}
&\left|\left\langle \Psi_{\text{ansatz}}^{T_m} \left| \sum_j \epsilon_j \hat{\sigma^{(j)}} \right| \Psi_{\text{ansatz}}^{T_m} \right\rangle\right|^2 \\
&= \left|\left\langle \Psi_{\text{ansatz}}^{T_m} \left| \sum_j \epsilon_j \left(\sum_p \left|\Psi_j^{(p)}\right\rangle \left\langle \Psi_j^{(p)}\right| - \sum_m \left|\Psi_j^{(m)}\right\rangle \left\langle \Psi_j^{(m)}\right| \right) \right| \Psi_{\text{ansatz}}^{T_m} \right\rangle\right|^2 \\
&= \left| \sum_j \epsilon_j \left(\sum_p \left|\left\langle \Psi_{\text{ansatz}}^{T_m} \mid \Psi_j^{(p)} \right\rangle\right|^2 - \sum_m \left|\left\langle \Psi_{\text{ansatz}}^{T_m} \mid \Psi_j^{(m)} \right\rangle\right|^2 \right) \right|^2 .
\end{aligned} \tag{4.13}$$

Note that Eq. (4.13) requires only a simple projection of $|\Psi_{\text{ansatz}}\rangle$ onto $\left|\Psi_j^{(p)}\right\rangle$ or $\left|\Psi_j^{(m)}\right\rangle$. Thus the second order correction energy may be obtained without any quantum resource overhead in the circuit size.

To optimize the choice of the ansatz evolution operator to use in the next cycle, we turn to the wavefunction correction. The ansatz state wavefunction can be considered as the zeroth-order wavefunction $\left|\Psi^{(0)}\right\rangle$ and a first order correction is given by

$$\left|\Psi_{T_m}^{(1)}\right\rangle = \sum_{\{\alpha\}} \frac{\left\langle \Psi_{D_\alpha}\right| V_{T_m} \left|\Psi_0\right\rangle}{\Delta E_{D_\alpha}} \left|\Psi_{D_\alpha}\right\rangle . \tag{4.14}$$

We then use the following procedure to determine which ansatz evolution operator to use next in the $(m+1)$th round. In the mth round we have variationally determined a set of values for the parameters in the set T_m, denoted as S_m. We then look for a set $R_m = \{U_{T_i}\}$ of all the evolution operators U_{T_i} that satisfies $T_m \subset T_i$ and

$$n(T_i \setminus T_m) = \min_{j \neq m} [n(T_j \setminus T_m)], \tag{4.15}$$

where $n(A)$ denotes the number of elements in set A. R_m is then the pool of operators from which we are going to choose the next ansatz evolution operator. For each U_{T_i} in R_m, we can then compute the overlap

$$F_m^i = 1/w_m^i \sum_{t_\beta \in T_i \setminus T_m} \left| \left\langle \Psi_{t_\beta}^{m,i} | \Psi_{T_m}^{(1)} \right\rangle \right|, \tag{4.16}$$

where

$$\left| \Psi_{t_\beta}^{m,i} \right\rangle = \left(\left. \frac{\partial U_{T_i}}{\partial t_\beta} \right|_{S_m} \right) |\Psi_0\rangle \ . \tag{4.17}$$

Here we use w_m^i to capture the additional cost associated with implementing U_{T_i} instead of U_{T_m}. For example, it can be the number of additional two-qubit gates needed to compute $\langle \Psi_0 | U^\dagger H U | \Psi_0 \rangle$ for $U = U_{T_i}$ instead of $U = U_{T_m}$. For our concrete example, we simply take w_m^i to be the number of additional fermionic operators in T_i comparing to T_m. Using the identity $\left\langle \Psi_{D_\alpha} | \Psi_{D'_{\alpha'}} \right\rangle = \delta_{\alpha\alpha'}$, it is straightforward to evaluate all of the F_m^i using Eq. (4.9), Eq. (4.12), and Eq. (4.17). Then we simply pick the U_{T_i} out of R_m with the largest F_m^i to be the ansatz evolution operator to use in the next cycle. We can also use the wavefunction correction to guess the initial values for an additional variational parameter $t_\beta \in T_{m+1} \setminus T_m$ as $\pm \left\langle \Psi_{t_\beta}^{m,m+1} | \Psi_{T_m}^{(1)} \right\rangle$.

In the case of UCCSD, each U_{T_i} in R_m only has one more D_α term in Z_{T_i} than Z_{T_m} of U_{T_m}, which corresponds to an incremental change in the terms included in the ansatz operator Z. Using

$$e^{\tilde{Z}_{T_i}} \approx 1 + \tilde{Z}_{T_i}, \tag{4.18}$$

F_m^i is approximately the perturbative amplitude $\left| \langle \Psi_{D_\alpha} | V_{T_m} | \Psi_0 \rangle / \Delta E_{D_\alpha} \right|$ of the corresponding

additional D_α.

All the procedures introduced above for the mth round of a VQE cycle can also be used before the first cycle by simply taking $m = 0$. In this case, all the calculations are classical and the energy correction is simply the classical MP2 energy correction. The wavefunction correction however can inform the first operator to be tried and suggest the initial values for its variational parameters.

4.4 HMP2 cost analysis

In principle, the number of individual Pauli-string operators in Eq. (4.13) whose expectation values are to be evaluated using a quantum computer scales as $O(n^4)$ per UCCSD ansatz where n is the number of qubits. This is so since H consists of $O(n^4)$ terms, while, for a particular ansatz, Z is fixed and provides a constant prefactor. Thus the predictive feature using Eq. (4.14) requires $O(n^8)$ Pauli-string measurements because there are $O(n^4)$ elements in $\{\alpha\}$. The perturbative correction using Eq. (4.9) can reuse all the measurement results from the predictive feature because we choose $\{\alpha'\} = \{\alpha\}$.

While this may appear challenging, in practice, a series of techniques can be applied to significantly reduce the number of evaluations. First, we can take advantage of the fact that only the qubits that are coupled via a chosen ansatz are entangled. By carefully choosing the qubits to use to create the ansatz state, we can treat the qubits that have not been operated on classically. A straightforward extension to small, disjoint sets of qubits with set sizes that admit inexpensive classical postprocessing may be considered as well, although we do not consider such an aggressive optimization in this chapter. In addition, if we have two qubits that represent

the same spatial orbital with two opposite spins, in the case where they are not distinguishable in their energy due to a particular choice of the ansatz (see Section 5.2 and [4] for more details), the two qubits would encode a redundant piece of information, and this allows us to encode the information using just one qubit. We call this optimization technique qubit space reduction (QSR) (see Appendix A for the implementation details).

Secondly, it is possible to measure multiple Pauli strings simultaneously to reduce the total number of measurements provided that these Pauli strings commute with each other. For instance, two methods, the general commuting (GC) partition and the qubit-wise commuting (QWC) partition, have been proposed to enable such simultaneous measurements [71]. Generally speaking, GC can reduce the number of measurements significantly but could incur additional cost in terms of additional two-qubit gates while QWC can reduce the number of measurements, though to a lesser degree than GC, without increasing the number of two-qubit gates. If using GC, it is beneficial to use QSR beforehand to reduce the size of the qubit space needed to be partitioned, which in turn reduces the number of extra two-qubit gates incurred from using the GC partition.

4.5 Comparison to prior state-of-the-art

To demonstrate our framework of perturbation assisted quantum simulation using HMP2 method, we performed classically emulated VQE calculations with the UCCSD ansatz of the ground state energy of a water molecule at its equilibrium geometry. Using the STO-3G basis, the calculation contains 14 qubits in total.

Table 4.1 shows the incremental changes of the UCCSD correlation energy as well as the HMP2 correction as more ansatz terms are included according to our framework. Note that our

Ansatz		E_{UCCSD}	$E^{\mathrm{corr}}_{\mathrm{P_D}}$	$E_{\mathrm{UCCSD+HMP2}}$
Ansatz terms in HMP2 order	HF+1	-74.9749	-0.0249	-74.9998
	HF+2	-74.9781	-0.0220	-75.0001
	HF+4	-74.9854	-0.0170	-75.0024
	HF+5	-74.9881	-0.0155	-75.0036
	HF+6	-74.9909	-0.0139	-75.0048
	HF+8	-74.9966	-0.0091	-75.0057
	HF+9	-74.9996	-0.0068	-75.0063
	HF+11	-75.0038	-0.0039	-75.0077
	HF+12	-75.0047	-0.0034	-75.0081
	HF+14	-75.0074	-0.0023	-75.0098
	HF+16	-75.0091	-0.0015	-75.0106
	HF+17	-75.0100	-0.0009	-75.0109
	HF+19	-75.0102	-0.0007	-75.0109
	HF+21	-75.0107	-0.0004	-75.0111
	HF+23	-75.0111	-0.0002	-75.0113
	HF+24	-75.0113	-0.0001	-75.0114
	HF+26	-75.0113	-0.0001	-75.0114
	HF+28	-75.0113	-0.0001	-75.0114
E_{HF}				-74.9624
E_{MP2}				-74.9977
E_{CCSD}				-75.0114
E_{FCI}				-75.0116

Table 4.1: Ground state energy calculations for a water molecule using STO-3G basis. HF+N represents a VQE cycle with the UCCSD ansatz with N D_α terms, as described in Section 4.2. $E^{\mathrm{corr}}_{\mathrm{P_D}}$ is the HMP2 correction based on Eq. (4.12). $E_{\mathrm{UCCSD+HMP2}} = E_{\mathrm{UCCSD}} + E^{\mathrm{corr}}_{\mathrm{P_D}}$ is the total energy obtained in one VQE cycle. For the classical optimization step in VQE, we used L-BFGS-B optimizer [3], and we assumed perfect measurement with an infinite number of circuit repetitions to suppress the effect of finite measurement. The classically computed energies E_{HF}, E_{MP2}, E_{CCSD}, and E_{FCI} for the water molecule with the same geometry and basis set are also listed for comparison. All energies are in units of Hartree.

choice of $\{\alpha'\}$ in Eq. (4.9) for this example guarantees that the final energy reaches the UCCSD energy when all the ansatz terms are included in Eq. (3.37). To compare, we also listed the classically calculated HF, MP2, CCSD, and FCI energies since it is difficult to obtain the UCCSD energy via classical algorithms. With the HMP2 correction, the total energy $E_{\text{UCCSD+HMP2}}$ descends quickly towards FCI energy.

Fig. 4.2 shows the convergence of the ground state energies using different approaches as the number N of D_α terms included in the UCCSD ansatz operator increases. For the conventional approach [4], the insertion order of the ansatz terms is obtained from the prior knowledge of the order of contribution of determinants in a classical FCI calculation, which closely mimics the ideal case but is rarely realistic to obtain. For the UCCSD energies obtained using the proposed HMP2, we bootstrapped the ordering of the ansatz terms as detailed in Section 4.3. Comparing the convergence of the UCCSD energies, we find that the HMP2-bootstrapped ordering effectively captures the major ansatz terms. This is confirmed by the good agreement between the FCI ordering and the HMP2 ordering shown in Fig. 4.2. We also observe that the HMP2 correction to the ground-state energy helps accelerate the energy convergence towards the FCI energy significantly. Provided that implementation of each additional ansatz term leads to a substantial accumulation of noise in the pre-FT hardware, we find the rapid energy convergence enabled by the HMP2 method to be particularly useful for the near-term quantum computers.

In addition to focusing on reducing the circuit complexity, which is the most important cost for pre-FT era applications, we examine the total number of measurements as a secondary cost using the water molecule calculation as an example. Fig. 4.3 (a) shows the number of measurements using various optimization schemes discussed in Section 4.3. Comparing to the number of measurements optimized using only QSR, the number optimized using the GC partition method

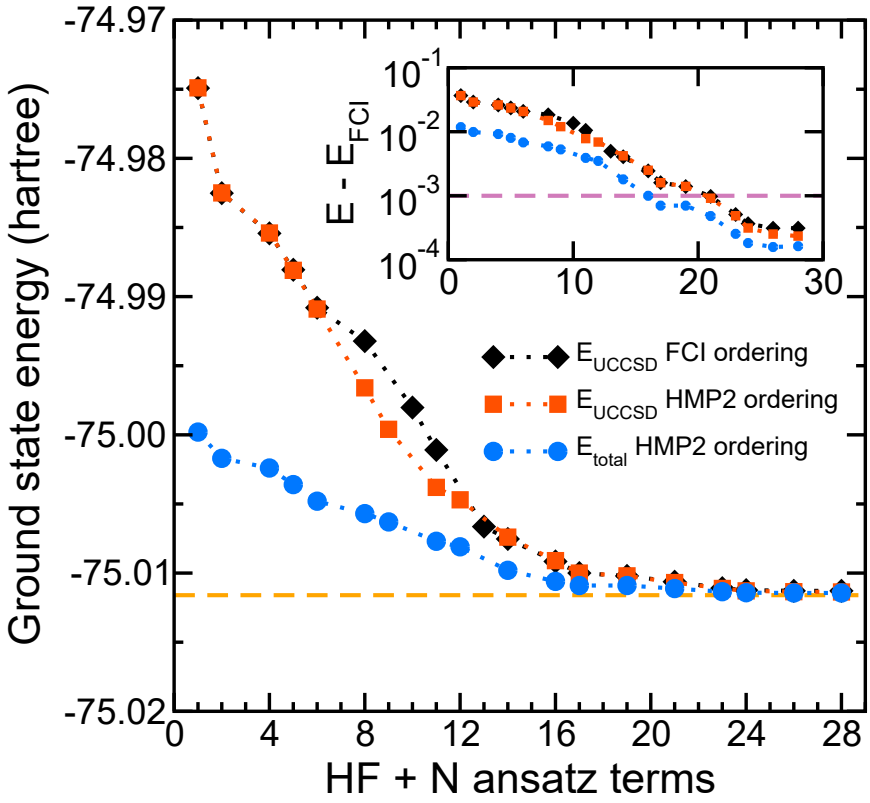


Figure 4.2: Comparison of the ground state energies of a water molecule at its equilibrium geometry using the STO-3G basis set, calculated by various methods as a function of N, the number of D_α terms included in the HF+N ansatz. The orange dashed line is the FCI energy calculated using the PSI4 package [6], which serves as the benchmark. The black diamonds connected by the dotted lines are the UCCSD energies E_{UCCSD}, calculated using different numbers of ansatz terms ordered by the contribution of corresponding determinants to the FCI energy. The red squares and blue circles connected by the dotted lines are the ground-state energies computed according to the proposed framework with the HMP2 ordering, with the circles containing the additional HMP2 energy correction at each VQE cycle. The inset shows in semi-log the differences between the energies obtained by the aforementioned methods and the FCI energy as a function of N. The purple dashed line shows the chemical accuracy given by 10^{-3} hartree. As stated in the caption of Table 4.1 as well, for the classical optimization step in VQE, we used L-BFGS-B optimizer [3], and we assumed perfect measurement with an infinite number of circuit repetitions to suppress the effect of finite measurement.

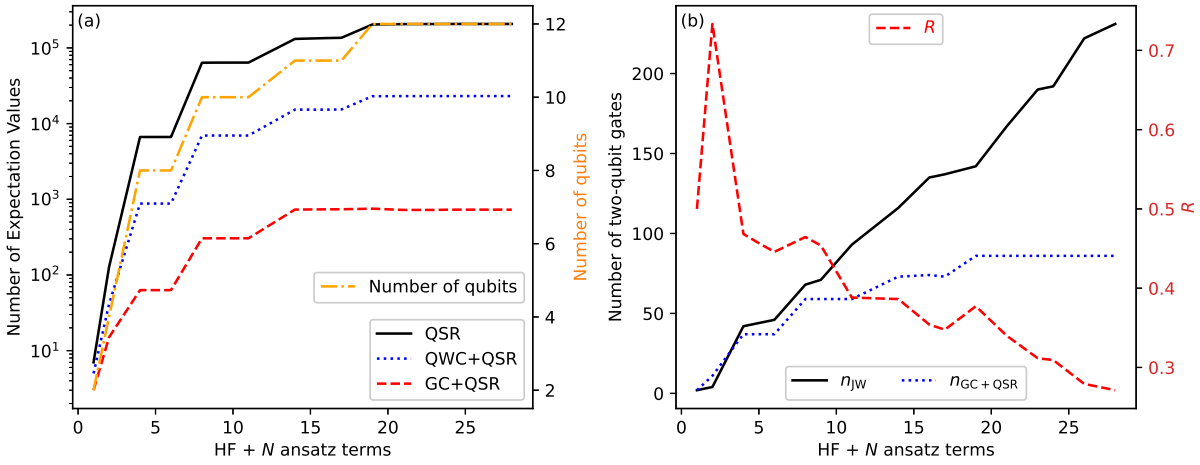


Figure 4.3: Demonstration of the reduction of the number of measurements as a secondary cost for the water molecule calculation using various optimization schemes. The calculation is carried out using the HMP2 framework with the UCCSD ansatz. The total number of expectation values to be evaluated on a quantum computer as a function of the number N of D_α terms in the UCCSD ansatz operator for a VQE cycle is shown in (a) for three different optimization schemes. The black solid line represents the reduction using only QSR. The blue dotted line shows the reduction using the QWC partition and QSR at the same time. The red dashed line illustrates the most reduction, using the GC partition along with QSR. Note the plateau-like behaviors that make the lines to appear like staircases are due to our QSR technique (orange, dot-dashed line with the y axis on the right), i.e., the jumps in the number of expectation values to be measured per VQE iteration occur when the QSR technique makes a transition from one space requirement to another. The average number of additional two-qubit gates incurred by using the GC partition along with QSR, $n_{\mathrm{GC+QSR}}$, is shown in (b) as the the blue dotted line. This may be compared with the black solid line, representing the total number of two-qubit gates n_{JW} required to induce UCCSD ansatz states, implemented with the JW transformation. The distribution of $n_{\mathrm{GC+QSR}}$ is given in the Appendix B. The red dashed line with an alternative y-axis label on the right shows the ratio $R = n_{\mathrm{GC+QSR}}/(n_{\mathrm{JW}} + n_{\mathrm{GC+QSR}})$.

and QSR is more than two orders of magnitude smaller while the number is about one order of magnitude smaller using the QWC partition method along with QSR. Fig. 4.3 (b) shows the additional cost in terms of the number of two-qubit gates associated with using the GC partition method along with QSR comparing to the total number of two-qubit gates using the JW transformation. The ratio between the two quickly reduces as the number of ansatz terms increases. Thus one should decide whether to use GC or QWC along with QSR after weighing their benefits against their cost on a case-by-case basis.

4.6 Discussion

For the pre-FT regime VQE simulations, we have proposed a general framework that leverages the predictive and corrective power of perturbation theory. The predictive feature of our framework resembles that of the ADAPT-VQE method [67], which aims to iteratively add terms to a generalized Trotterized exponential ansatz by following the steepest descent based on the calculations of numerical gradients. Despite trying to solve the same problem, our framework adopts a different philosophy and computes an approximate (perturbative) correction to the wavefunction and chooses the following operator based on how much of a correction an operator could potentially provide. Due to the two completely different approaches to the problem, the two methods could have vastly different use cases for which they are suitable respectively. To identify the difference in use cases, further thorough investigation into different areas of applications are needed which is beyond the scope of this thesis.

Additionally, our framework is exceedingly flexible and provides means to balance between resource constraints and accuracy requirement. First of all, our framework not only works with

the UCCSD ansatz, but any unitary ansatz, including the generalized Trotterized exponential ansatz used in the ADAPT-VQE method. Second, the predictive feature and the corrective feature can be used independently from each other. For instance, the perturbative corrections can be used along with the ADAPT-VQE method, which provides the ansatz term prediction. Note that doing so, according to our framework, does not incur additional cost in terms of circuit complexity; it incurs overhead only in terms of the number of measurements. Our specific examples described in this chapter results in smaller number of measurements required, since both the prediction and the correction are obtained by the same set of measurements, as opposed to two different set of measurements that would otherwise be required, had we combined ADAPT-VQE with our perturbative correction. Third, the choice of the set of orbital sequence $\{\alpha'\}$ in Eq. (4.9) can be leveraged to balance between accuracy requirement and resource constraints. On one hand, the size of the set can be decreased to reduce the number of measurements needed. On the other hand, it is also possible to include orbital sequences outside of the set $\{\alpha\}$ for the ansatz operator in order to achieve better accuracy without sacrificing circuit complexity. For instance, one could include triple excitation sequences in $\{\alpha'\}$ along with the UCCSD ansatz to provide a perturbative "T" correction to the final ground state energy. In this case the increased accuracy does not correspond to increased circuit complexity but only incurs overhead in terms of the number of measurements.

Chapter 5: Ansatz optimization basic techniques

5.1 Overview

Depending on the target gate to be optimized, the optimization strategy for ansatz could be vastly different. In the pre-FT regime, quantum computational cost is dominated by the use of multi-qubit gates. In the FT regime, where quantum circuits are typically composed of gates in the Clifford+T gateset, quantum computational cost is dominated by the use of T gates, many of which are used in the FT implementation of $\mathrm{R}_z(\theta)$ gates (see, e.g., [72]). In this thesis, we present approaches that optimize quantum simulations of fermionic systems in the pre-FT regime, reducing the number of multi-qubit gates in the simulation circuit.

Specifically, the number of two-qubit CNOT gates used is reduced. Among the de-facto quantum gateset of CNOT+single-qubit gates, effectuating high-quality CNOT gates is the most challenging for physical implementation, both time- and fidelity-wise. Indeed, not only is the computational efficiency improved, but the fidelity of quantum computation also boosted, as the CNOT count is reduced.

In this chapter, we propose a series of techniques to reduce circuit complexity of the UCC-based VQE ansatz, represented by the total number of two-qubit gates. To be concrete with our example, we use the first-order PF algorithm to implement the UCCSD ansatz, although extensions to other ansatz or higher order PF algorithms are straightforward.

In Section 5.2, we extend the well-known transformations to generalized transformation. Next in Section 5.3, we provide a heuristic that can take advantage of the generalized transformation. We then compare the heuristic with the prior art, with results shown in Section 5.4. Many technical details mentioned in this section are further explained in Section 5.5. We last conclude in Section 5.6 with discussion and conclusion.

5.2 Generalized transformations for fermion to qubit operator

It is important to note that all of the transformations are equivalent and thus the resulting quantum circuits for each transformation implements exactly the same fermion simulation. Therefore, the resource savings we obtain in this section are independent of the accuracy of the simulations, in general.

Well-known fermion to qubit transformations, such as Jordan-Wigner (JW) [45] or Bravyi-Kitaev (BK) [46] transformations, map fermionic creation (annihilation) operators to Pauli strings. However, there exist numerous other transformations available for use. Below, we introduce a generalized transformation (GT) method (see also [47]), of which the JW and BK transformations are a part. We show that, when used with the PF algorithm as a concrete example for the implementation of the UCCSD ansatz, significant quantum resource savings may be achieved by a suitable choice of the mapping for a given cluster operator input, together with a carefully chosen sequence of heuristic optimization methods.

All transformations in the GT method must respect relations specified in Eq. (3.18). This may be achieved by considering the following invertible, upper-triangular basis-transformation

matrix β, which transforms the occupation number basis to a GT basis according to

$$\beta = \begin{pmatrix} \beta_{n-1,n-1} & \beta_{n-1,n-2} & \cdots & \beta_{n-1,0} \\ 0 & \beta_{n-2,n-2} & \cdots & \beta_{n-2,0} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \beta_{0,0} \end{pmatrix}, \tag{5.1}$$

where $\beta_{i,j} \in \{0, 1\}$ for $i > j$, $\beta_{i,i} = 1$, and n is the number of qubits involved in the transformation. Following closely the notations used in [73], we define the following sets of indices for convenience, which we detail below. We note that all matrix operations are performed in modulo-2 space and the main diagonal elements are excluded when generating these sets.

- Update set $U(j)$: elements of this set are the row indices with non-zero entries in column j of the basis-transformation matrix β.

- Parity set $P(j)$: elements of this set are the column indices with non-zero entries in row j of the matrix $(\pi\beta^{-1} - \beta^{-1})$, where $\pi_{i,j} = 1$ if $i \leq j$, otherwise 0.

- Remainder set $R(j)$: elements of this set are the column indices with non-zero entries in row j of the matrix $\pi\beta^{-1}$.

The GT-based creation and annihilation operators are then

$$\begin{aligned} a_j^\dagger &\equiv \left[\sigma_x^{U(j)} \otimes \sigma_x^j \otimes \sigma_z^{P(j)} - i\sigma_x^{U(j)} \otimes \sigma_y^j \otimes \sigma_z^{R(j)}\right]/2, \\ a_j &\equiv \left[\sigma_x^{U(j)} \otimes \sigma_x^j \otimes \sigma_z^{P(j)} + i\sigma_x^{U(j)} \otimes \sigma_y^j \otimes \sigma_z^{R(j)}\right]/2, \end{aligned} \tag{5.2}$$

which can straightforwardly be shown to satisfy Eq. (3.18). We note that, e.g., the JW transformation is a special case of $\beta = \mathbf{1}$.

5.3 Optimization heuristic based on generalized transformation

In this section, we investigate how a variety of fermion to qubit transformations may be used to reduce the quantum resource requirements for the pre-FT fermion simulations. The state of the art implementation of the UCCSD ansatz [4] relies on the JW transformation and it considers a series of heuristics that were chosen carefully to optimize the resulting quantum circuits. To enable a proper comparison of those with our results then, it is necessary for us to consider a similar series of heuristics as well. Details of the heuristics we consider are provided in the Section 5.5.1–Section 5.5.6. Below, we briefly outline the steps in the order of applications. The cost function we consider for our concrete example is the number of two-qubit gates.

The outer-most loop of our approach considers different transformation matrices β. For a given mapping matrix β, we execute the following routines to construct and optimize our circuit that implements the UCCSD ansatz. The routines are designed to repeatedly call a suite of dedicated, automated circuit optimization tools, whose technical details may be found in [4, 5]. The efficiency of these tools allows us to quickly evaluate the cost function for different cases we consider in each of the subroutines.

Routine 1 *Fermionic level labeling:* Unlike in the JW transformation where $U(j)$ is an empty set and $P(j) = R(j)$, in the GT approach, myriads of combinations of sets $U(j)$, $P(j)$, and $R(j)$ are possible. To best take advantage of this, it is critical to carefully select which fermion level is mapped onto which qubit index. Exploring all possi-

ble mappings is however computationally prohibitively expensive. We thus resort to a simple greedy approach, whereby we explore one permutation at a time from a given fermion-level to qubit-index mapping. Specifically, from the given mapping, we apply the permutation that results in the most reduction in the quantum resource requirements. We iterate this process until no single permutation results in the reduction of quantum resource requirements. See Section 5.5.1 for detail. See **Subroutine 1** below for the cost function evaluation for each permutation.

Routine 2 *Inter-Trotter term ordering:* Demonstrated in [4, 74–78] was that ordering the Trotter terms appropriately can lead to large savings in the two-qubit gate counts due to gate cancellations between the neighboring Trotter term circuits. While a similar approach is indeed possible – and we base our method on the work reported in [4] – in our GT approach, a non-trivial modification to a method reported in [4] needs to be made. Specifically, we need to preprocess each Trotter term to determine its eligibility for being classified under the same equivalence class, the elements of which have the opportunities for resource savings when placed next to one another on a quantum circuit via the aforementioned gate cancellations. The eligibility criteria are straightforward for the JW transformation, considered in [4]. See Section 5.5.3 for details for the GT method. Once the equivalence classes according to the eligibility of each Trotter term is determined, we use a simple greedy approach to order the Trotter-term elements of each equivalence class to reduce the two-qubit gate counts.

Subroutine 1 *Intra-Trotter term ordering:* For a given fermion-label to qubit-index mapping,

an efficient method to implement a single- or a double-fermion excitation Trotter term is known [57] in the case where the JW transformation is used, and an aggressively optimized circuit construction that leverages full qubit-to-qubit connectivity is available in [4]. The method in [4] relies on a careful ordering of the intra-Trotter operator implementation, where, for instance, an example double-fermion Trotter term $e^{t_{pqrs}(a_p^\dagger a_q^\dagger a_r a_s - h.c)}$ is expanded to $\sigma_{x,y,z}$-based intra-Trotter terms. To enable an efficient implementation in other transformations used in our GT approach, we compute the cost function for every possible permutation of the intra-Trotter terms (see Section 5.5.2 for detail). We choose an ordering with the least cost.

The resulting optimized circuit implements the UCCSD ansatz in the chosen transformation basis defined by β. However, with the exception of the JW transformation where $\beta = \mathbf{1}$, the GT β matrix requires us to implement the initial mapping of the basis at the beginning of the circuit. This incurs an overhead $O(n^2/\log(n))$ in the two-qubit gate counts [79]. To obtain the final quantum resource requirement, we call an automated optimizer with the input quantum circuit that consists of the prefix subcircuit that implements β and the postfix subcircuit that implements the β-basis UCCSD ansatz.

We note that in [4] the concept of "bosonic" excitations is discussed. Effectively, in the JW transformation, whenever a pair of neighboring qubits, whichever appropriate fermion levels they correspond to, are excited to yet another pair of neighboring qubits that denote another set of fermion levels, the circuit that implements such an excitation term can be dramatically simplified to require only two two-qubit gates, while requiring only half the number of qubits that would

otherwise be required (see Section 5.5.4 for the cases where the pairs do not neighbor). To take advantage of this, we use a juxtaposition of the bosonic circuit written according to the JW transformation and non-bosonic circuit written according to our GT approach. We note in passing that, to return from the half-qubit space of the bosonic circuit to the full-qubit space of the non-bosonic circuit, at most $n/2$ CNOT gates are expended. All our circuit metrics appropriately reflect this.

5.4 Compare to prior art

Table 5.1 shows circuit metrics, measured according to the number of two-qubit gates used to implement the UCCSD ansatz circuit, for different molecules of our choice. We show the results for the JW, BK, and the best GT transformations that our heuristic toolchain specified above found for comparison. To find the best GT transformations, we used a particle swarm optimization, as detailed in Section 5.5.5 and Section 5.5.6. The advantages offered by the GT transformations vary in the suite of molecules we consider, ranging from 1.44% to 21.43%. This demonstrates the capability of our heuristics that it is indeed possible to further optimize the quantum circuits over the previous state of the art obtained via the JW transformation by considering GT transformations, custom selected for different input cases.

We make an open-ended note here on the savings ratio obtained. The savings ratio for the GT method is obtained by comparison to the JW transformation, which often admits better circuit optimization over the BK transformation in the instances we considered, especially around the interesting regimes where the ansatz circuit is large enough to recover, e.g., chemical accuracy, but is sufficiently small to merit VQE implementations on a near-term, noisy device. In the

Molecule	n	NE	JW	BK	GT	Improve(%)
HF	6	3	30	29	25	16.67
LiH	6	3	30	29	25	16.67
BeH_2	8	9	70	71	60	14.29
NH_3	14	52	485	607	478	1.44
$H_2O(4)$	8	4	42	50	33	21.43
$H_2O(5)$	8	5	44	52	35	20.45
$H_2O(6)$	8	6	46	47	37	19.56
$H_2O(8)$	10	8	68	88	63	7.35
$H_2O(9)$	10	9	71	89	66	7.04
$H_2O(11)$	10	11	93	110	87	6.45
$H_2O(12)$	10	12	95	112	89	6.32
$H_2O(14)$	10	14	114	140	111	2.63
$H_2O(16)$	10	16	135	166	131	2.96
$H_2O(17)$	10	17	137	168	133	2.91

Table 5.1: Number of two-qubit gates required for the VQE simulation of different molecules with different fermion to qubit transformations. n is the dimension of the β matrix. NE is the number of excitation terms considered in the UCCSD ansatz. JW/BK are the number of two-qubit gates with JW or BK transformations. GT is the number of two-qubit gates given by the best β other than JW or BK. All molecules use the STO-3G basis, and ansatz terms are determined according to the HMP2 ordering. (Top): Here, we report the results for the cases where the HMP2 method was used to reach chemical accuracy. (Bottom): Here, we report the results for the HMP2 progression for a water molecule. The number in the parentheses next to H_2O indicates the total number of excitation terms (D_α) considered for the UCCSD ansatz. The cutoff of $H_2O(17)$ is chosen based on the recovery of chemical accuracy (see Fig. 4.2).

asymptotic limit, the BK transformation, one of the GTs, is known to incur $O(\log(n))$ overhead, an exponential advantage over the JW transformation that incurs $O(n)$ overhead [3]. Provided that, in the small instances, the savings ratio can be larger than 20%, a question arises as to how the ratio will behave in the intermediate-sized, aforementioned interesting regime. We mention here a couple possibilities. It may be that the improved circuit efficiency by the GT method for a particular excitation term could be offset by the increased cost of implementing other excitation terms. In this case, the ratio is expected to drop. It could also be that the JW transformation, due to its lack of nimbleness in the definition of transformation to help reduce potentially large cost of adding the excitation terms to the ansatz circuit, incurs a large overhead, thus resulting in the more significant savings ratio. Further research is required to provide better predictions. The low reduction rate observed for larger sized circuits in Table 5.1 may in part be due to the ineffectiveness of the classical optimization strategy and the limited classical computational resources used, which is further discussed in Section 5.5.6.

5.5 Technical details

5.5.1 Fermionic level labeling

In this section, we detail the heuristic described in **Routine 1** in Section 5.2 which aims to reduce the number of two-qubit gates by exploiting different mappings from labels of fermion levels to qubit indices. Note that the Pauli strings for an excitation operator, after applying the fermion $\mapsto$ qubit transformation of choice, depends on the qubit index values according to Eq. (5.2). A better optimized mapping from labels of fermion levels to qubit indices results in fewer Pauli matrices in a given Pauli string which leads to a smaller number of two-qubit gates.

Notice that such a mapping is equivalent to maintaining a simple mapping from any fermion level k to qubit index k while rearranging the labeling of the fermion levels which is entirely arbitrary up until this point. Thus we may use the freedom of fermion level labeling to reduce the number of two-qubit gates required to implement the one- and two-body Trotter terms. For an n-qubit system representing n fermion levels, exploring all possible permutations of fermion level labeling is not computationally scalable as the total number of permutations is $n!$. We thus resort to a simple greedy approach as follows.

1. We first generate a set of unique permutation matrices $\{P_i\}$ for all possible k-swap operations. Here we define a k-swap operation as swapping k unique ordered pairs of indices $(2l_j, 2l_j + 1)$ with k other unique ordered pairs of indices $(2m_j, 2m_j + 1)$ with the same subscript index $j \in \{0, \cdots, k-1\}$ where $l_j \in \{0, \cdots, n/2 - 2\}$ and $m_j \in \{l_j + 1, \cdots, n/2 - 1\}$. We use $k = 2$ for all of the examples in the main text.

2. Denoting the initial fermion labels as a column vector L_0, we can then go through all the permutation matrices in the first round and compute the number of two-qubit gates that corresponds to each instance of relabeled fermion levels $P_i L_0$ by simply adding the number of two-qubit gates required for each excitation term obtained from the intra-Trotter term reordering subroutine. When going through the permutation matrices, if any matrix results in a lower two-qubit gate count than all the previous matrices, we record that two-qubit gate count as N_1 and the corresponding relabeled level vector as L_1.

3. We proceed with the greedy approach iteratively following the first round. Denoting the resulting relabeled fermion level vector from the mth round as L_m and the corresponding two-qubit gate count as N_m, we again go through all the permutation matrices and compute

the corresponding two-qubit gate counts for each instance of relabeled levels $P_i L_m$. Taking $N_{m+1} = N_m$ initially, every time a matrix results in a two-qubit count lower than N_{m+1}, we record that gate count as N_{m+1} and the matrix as L_{m+1}. If there is not any matrix that results in a lower two-qubit count than N_m, we terminate the iteration and return L_m as the optimal labeling. Otherwise we proceed to the next round.

We note that once the labels are determined, they are applied to all relevant fermionic operators, including the molecular Hamiltonian, to be consistent throughout our simulation.

5.5.2 Intra-Trotter term ordering

In this section, we provide the details of **Subroutine 1** described in Section 5.2 which aims to reduce the number of two-qubit gates by optimizing the ordering of the Pauli strings transformed from a two-body operator.

We start by briefly noting that each one-body term leads to only one ordering, up to inversion. This is so, because it contains only two Pauli strings. Therefore, the one-body term does not require a specific ordering.

We next consider two-body operators. A single two-body operator, after applying proper transformation and PF algorithm, contains eight subterms. Specifically, we have

$$U^{\text{two-body}} = \prod_{j=0}^{7} e^{-i\theta \otimes_v \sigma^{(j,v)}/2}, \tag{5.3}$$

where j denotes the intra-Trotter term index, v denotes the qubit index, and $\sigma \in \{\mathbf{1}, \sigma_x, \sigma_y, \sigma_z\}$. Each of the intra-Trotter terms $e^{-i\theta \otimes_v \sigma^{(j,v)}/2}$ can be readily translated into a standard circuit as described in Eq. (8) in Sec. Methods H of [4]. In a naive implementation of U that uses an

arbitrary fermion to qubit transformation, each intra-Trotter term results in $2(N_j - 1)$ number of CNOT gates, where N_j is the number of non-identity $\sigma^{(j,v)}$ in the jth Pauli string.

Compared to the JW transformation, in the GT method, it is useful to have a more concrete set of rules to determine which qubit may be used as a target qubit t, in relation to the realization of U as a quantum circuit (see, e.g., Eq. 8 (b) in Sec. Methods H of [4], for the JW transformation). In the JW transformation, any choice of t such that $\sigma^{(j,t)}$ are either σ_x or σ_y, i.e., the qubits that correspond to the fermion labels in the two-body term, are good choices. However, if $\sigma^{(j,t)} = \mathbf{1}$ in any one of the eight Pauli strings in U above, which frequently occur in the GT method, the intended target qubit index t is unusable. Therefore, in the GT method, qubit choices t that lead to $\sigma^{(j,t)} = \mathbf{1}$ are removed from the set of eligible targets that start with all qubits that correspond to the fermion labels that appear in the two-body term of interest and will subsequently not be used in the inter-Trotter term reordering subroutine detailed in Section 5.5.3.

Now, for a chosen target qubit of choice t, we consider the ordering of intra-Trotter terms to maximize the CNOT reduction. In particular, we only consider the CNOT reduction between adjacent Pauli string circuits. For each of the adjacent intra-Trotter terms, i.e., $e^{-i\frac{\theta}{2}\otimes_v \sigma^{(j,v)}}$ and $e^{-i\frac{\theta'}{2}\otimes_v \sigma^{(j+1,v)}}$, where $j \in [0, 6]$, we enumerate v through all non-target qubits. We then compare $\sigma^{(j,v)}$ and $\sigma^{(j+1,v)}$. For a particular control qubit v, if neither of the two aforementioned Pauli matrices $\sigma^{(j,v)}$ and $\sigma^{(j+1,v)}$ is $\mathbf{1}$, then the circuit can be expressed as Fig. 5.1. In this particular circuit, if we have $M_0 = M_2$, there is a two CNOT reduction, while, if we have $M_0 \neq M_2$, there is a one CNOT reduction, according to the circuit identity discussed in [4] Sec. Method H. Suppose now there are $m_j^{(t)}$ two-CNOT reductions and $n_j^{(t)}$ one-CNOT reductions for the chosen target qubit t. Then, the total number of CNOT reductions is $2m_j^{(t)} + n_j^{(t)}$ for the target qubit of choice t. The total optimized number of CNOT for a chosen target t of a certain ordering is thus

$N^{(t)} = \sum_{j=0}^{7} 2(N_j - 1) - \sum_{j=0}^{6}(2m_j^{(t)} + n_j^{(t)})$.

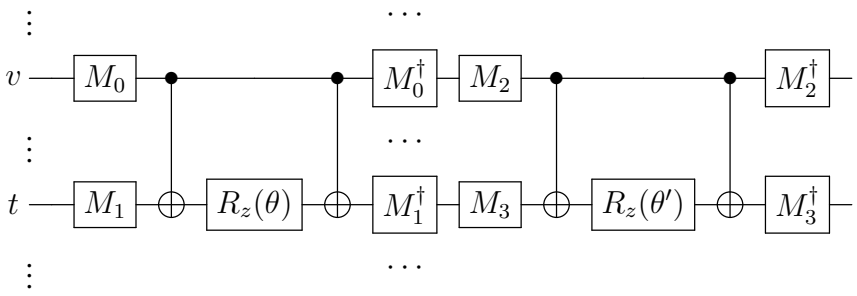


Figure 5.1: Example circuit for adjacent Pauli string circuits $e^{-i\frac{\theta}{2}\sigma_0^v\otimes\sigma_1^t\otimes\cdots}e^{-i\frac{\theta'}{2}\sigma_2^v\otimes\sigma_3^t\otimes\cdots}$. Subscripts l to Pauli operators are introduced to conveniently label the associated operators M_l shown in the figure. t denotes the target qubit, v denotes the control qubit, $\sigma_l \in \{\sigma_x, \sigma_y, \sigma_z\}$, and $M_l \in \{\mathrm{H}, \mathrm{S}^\dagger\mathrm{H}, \mathbf{1}\}$. If $\sigma_l = \sigma_x$, $M_l = \mathrm{H}$. If $\sigma_l = \sigma_y$, $M_l = \mathrm{S}^\dagger\mathrm{H}$. If $\sigma_l = \sigma_z$, $M_l = \mathbf{1}$.

At this point, we are equipped to compute and compare $N^{(t)}$ among all possible orderings and all possible targets to determine which one of the target qubit sets and the intra-Trotter orderings result in the minimum number of CNOTs. We note that for a certain ordering, there likely is a degeneracy in the choice of target qubit t that results in the same CNOT counts. All of these degenerate cases with the optimal resources are returned to the Inter-Trotter term ordering routine, detailed in Section 5.5.3, for use.

5.5.3 Inter-Trotter term ordering

In this subsection, we detail a heuristic to reduce the number of two-qubit gates by taking advantage of the freedom to arbitrarily order the Trotter terms. This is justified since any orderings respect the error bound provided by the PF algorithm. Denoted as **Routine 2** in Section 5.2, this subroutine requires us to first run a preprocessing step based on the information passed from the intra-Trotter term optimization step described in Section 5.5.2. Specifically, for operators $a_i^\dagger a_j$ and $a_i^\dagger a_j^\dagger a_k a_l$, we check if any one of the indices i and j for the former, and i, j, k, and l for

the latter, may or may not be used as a target qubit in the circuit implementation of the operators in the standard compilation (see, e.g., Fig. 2d of [4] for a two-body term using the JW transformation). See Section 5.5.2 for details. We flag the qubit indices that cannot be used as a target as ineligible.

After all of the ineligibilities have been determined, we proceed according to a simple greedy approach. We first identify the most frequently eligible index, say p, across all terms. Then, we group all terms with the index p as an eligible target qubit and classify them under the equivalence class $[p]$. We next remove all the group elements from the list of one- and two-body operators. We repeat the procedure from the identification of the most frequent eligible index until no more operators are left in the list.

Note that the quantum resource cost reduction will likely now result in between the circuit representation of the elements of the same equivalence class, since the target qubit of two-qubit gates from each element of the same class is the same. Therefore, once all of the equivalence classes are specified, we consider permuting the orderings by which the elements are implemented on a quantum circuit. Considering all permutations can be prohibitively expensive. We thus use a simple greedy approach once more, first starting out with two of the elements that result in the most resource cost reduction. Then, we concatenate a next element, identified from the set of elements that have not been implemented in the circuit, based on the resource cost reduction. We repeat the concatenation process until no more element is left in the set. We note that each trial of testing out which element may be the best for the given iteration consists in general of four cases. This is so, since the circuit concatenation may be performed as a prefix or suffix, and the element to be concatenated can be considered in its original intra-term order or the reverse.

5.5.4 Generalized bosonic terms

The correspondence between Generalized bosonic terms and fermionic terms can be established as follows. A fermionic double excitation term $\theta_{pqrs}(a_p^\dagger a_q^\dagger a_r a_s - h.c.)$, when transformed by the JW transformation, turns into $\theta_{pqrs}(\sigma_+^p \sigma_+^q \sigma_-^r \sigma_-^s \otimes_k \sigma_z^k - h.c.)$. If p and q belong to the same spatial orbital and r and s belong to yet another same spatial orbital, assuming no other terms that break the symmetry between p and q or r and s have been considered in the circuit, p and q levels may be encoded by a single qubit and likewise for the r and s levels. In this case, using p and r as representatives, the qubit-space operator can be simplified to $\theta_{pqrs}(\sigma_+^p \sigma_-^r \otimes_{k'} \sigma_z^{k'} - h.c.)$, where k' runs over the set of qubits σ_z operator needs to be applied for the given excitation term in the appropriately reduced space. The appropriately reduced space may include the single qubits that each denotes the reduced, symmetric levels and those that are not reduced. To illustrate, if two of the levels k_1 and k_2 in the original space require σ_z and if k_1 and k_2 are encoded into a single index k', we simply call $\sigma_Z^{k'}$ twice, one each for k_1 and k_2. This for instance amounts to identity, which results in resource savings.

5.5.5 Binary particle swarm optimization

In this section we detail the procedure for using binary particle swarm optimization (PSO) [80] to optimize the binary matrix β for GT. To encode the problem, we map the upper triangular entries in an $n \times n$ binary matrix β to a one dimensional binary vector $X \in \{0,1\}^d$ where size $d = n(n-1)/2$. Vector X then serves as the location vector for PSO. The cost function for PSO at a location X mapped from a β matrix is then defined as the number of two-qubit gates needed to implement a UCCSD evolution operator using GT with the β matrix and subsequently a series

of heuristics described in Section 5.2.

We start by creating a swarm of particles, each with an initial location $X^i(t=0)$ mapped from a β matrix sampled from the set of upper triangular matrices whose diagonal elements are ones and off-diagonal elements are zeros except for k of them. The non-negative integer t represents the time step and $t = 0$ is the beginning of the optimization. The total number of particles in the swarm is $\sum_{j=1}^{k} \binom{d}{j}$ where $\binom{\cdot}{\cdot}$ denotes a binomial coefficient. Specifically for our examples, we vary k from 1 to $k_{\max}$ and we have $k_{\max} = 6$ for $n \leq 8$ and $k_{\max} = 3$ for the rest of the cases. Each particle also has a velocity vector $V^i(t)$ of the same dimension as $X^i(t)$. The initial matrix elements of the velocity vectors are taken to be zero.

At every time step t, we also keep track of a local optimal location vector $L^i(t)$ and the corresponding local optimal cost $s^i(t)$ for each particle i and a global optimal location vector $G(t)$. The local optimal location vector $L^i(t)$ is defined as the location vector with the lowest cost function value $s^i(t)$ across all the locations particle i has traveled to up to the time step t. The global location vector is the $L^i(t)$ with the lowest $s^i(t)$ among all the particles. The velocity vectors are updated from the tth step to the $(t+1)$th step according to

$$V_j^i(t+1) = wV_j^i(t) + c_1(L_j^i(t) - X_j^i(t)) + c_2(G_j(t) - X_j^i(t)) \,, \tag{5.4}$$

where w is the inertia parameter, c_1 is the cognitive parameter, and c_2 is the social parameter. For the examples here, we use various parameters with $w \in [-4, 4]$, $c_1 \in [0, 2]$, and $c_2 \in [0, 2]$. The subscript j denotes the jth element of the corresponding vector. The elements of the location vectors $X_j^i(t+1)$ are then updated to 0 if $\mathrm{rand}() \leq 1/[1+e^{-V_j^i(t+1)}]$ or 1 otherwise. Here $\mathrm{rand}()$ is a real quasirandom number generating function that samples from the uniform distribution on

$[0.0, 1.0]$.

In practice, binary PSO is usually terminated when t reaches a predetermined maximal time $t_{\max}$. A particular particle is also stopped when its location vector oscillates between two vectors for more than δt_{osc} times. To further save computation time spent on binary PSO, we impose a third rule that we stop a particle if its location vector maintains a Hamming distance larger than s from its initial location without reaching a lower two-qubit gate counts for more than δt_s steps. Such a termination condition allows similar search space sizes for β matrices of different sizes and optimization runs with different initial conditions. For our examples, we use $\delta t_{\text{osc}} = 10$, $s = 6$, $\delta t_s = 10$, $t_{\max} = 10000$ for $n \leq 8$, and $t_{\max} = 100$ for the rest of the cases. In most runs in our examples, our choice of $t_{\max}$ is large enough that the optimization starts to oscillate before t reaches $t_{\max}$.

5.5.6 General transformation results using binary particle swarm optimization

In this section we present and discuss examples of GT results using binary PSO with various amounts of classical computing resources. Specifically, we are interested in the fractional improvement in two-qubit gate count, ρ, defined according to

$$\rho = \frac{f_{\text{GT}}(R)}{f_{\text{GT}}(R) - f_{\text{JW}}} , \tag{5.5}$$

where f_{GT} and f_{JW} are the numbers of two-qubit gates needed to implement a UCCSD evolution operator using GT with binary PSO or JW, respectively. f_{GT} depends on the classical computing resources used by binary PSO which are determined mostly by the number of particles used since s and δt_s are the same for all optimization runs and $t_{\max}$ is always set to be sufficiently large for

our examples as discussed in Section 5.5.5. To contrast the amount of classical computing resources used with what is maximally possibly needed, we define a fractional classical computing resource metric as the ratio between the number of particles used, N, and the total number of possible variations of the β matrices, which can be written as

$$R = \frac{N}{2^{n(n-1)/2}}, \tag{5.6}$$

where n is the number of diagonal elements of the β matrices. Since the number of particles increase with k as discussed in Section 5.5.5, we change k in our examples to effectively change the amount of classical computing resources used for binary PSO.

Fig. 5.2 shows examples of the fractional improvement, ρ, as a function of the fractional classical computing resources used, R, for a few different molecules as well as for different numbers of ansatz terms used in the simulation of the water molecule. All the cases are simulated with the STO-3G basis set. We observe that ρ increases monotonically with R in all cases simulated. In some cases, $i.e.$ hydrogen fluoride (HF), beryllium hydride (BeH_2), and water molecules with four to six ansatz terms included where n is small enough that we are able to explore a relatively large portion of possible variations of β matrices ($R > 10^{-3}$), we are able to obtain significant fractional improvement ranging from $\approx 14\%$ to over 20%. In other cases where n is large and we are only able to sample exponentially small portions of the location vector space in PSO, the fractional improvement obtained is much smaller and on the single digit percentage level. Thus we expect, based on the set of numerical results, that more cleverly exploring a large search space is likely a key to increasing the value of ρ; Note the cost function $f_{\mathrm{GT}}(\cdot)$ landscape is likely complex, making it tricky for binary PSO to avoid being trapped in local minima – We indeed

observed many PSO jobs undergoing oscillatory behaviors.

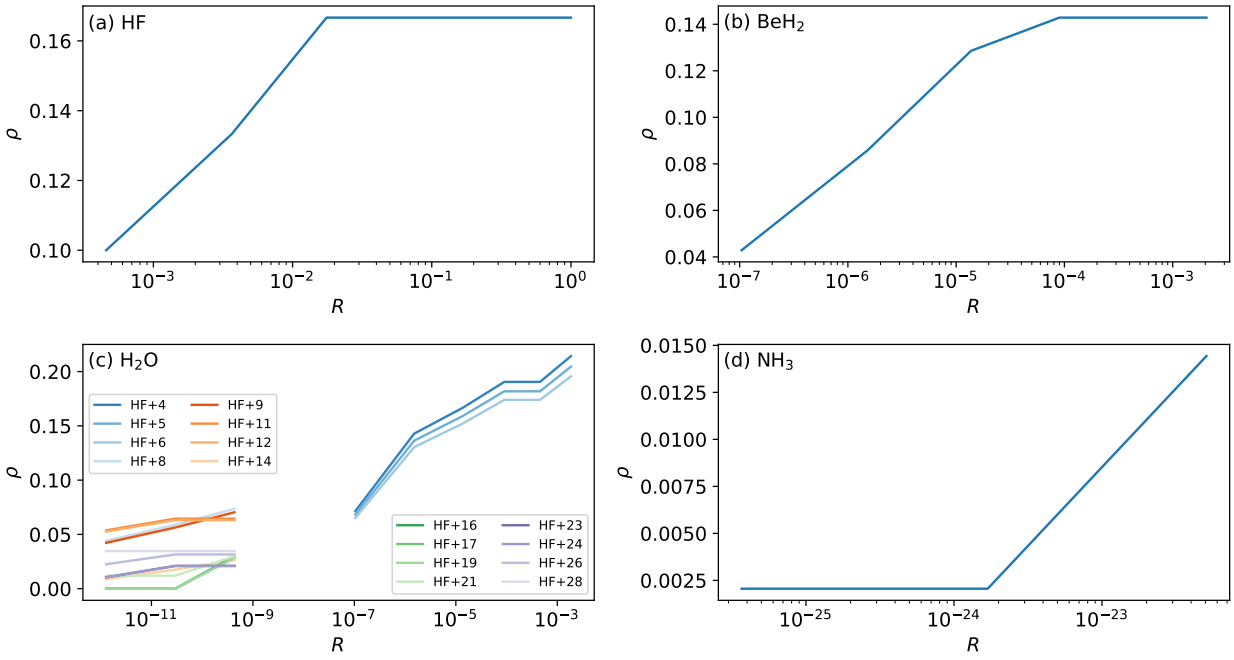


Figure 5.2: Fractional improvement ρ as a function of R, the fractional classical computing resources used for binary PSO. (a) HF, (b) BeH_2, (c) H_2O, and (d) NH_3. All molecules are simulated with the STO-3G basis set using the HMP2 method with UCCSD ansatz. For HF, BeH_2, and NH_3, the number of excitation terms included in the final UCCSD ansatz operators are 3, 9, and 52, respectively, which result in their respective ground state energy estimates within chemical accuracy from their known ground state energies. The size of the β matrices given by their respective number of diagonal terms n are shown in Table 5.1.

5.6 Discussion and conclusion

We note that, extending and generalizing the framework used for considering the bosonic terms in the JW transformation and the non-bosonic terms in the GT transformation, the use of multiple fermion to qubit transformations β for a given set of excitation terms could be of value in reducing the overall resource requirement. Drawing from the fact that different β transformations result in different resource requirements in implementing the excitation terms of a target UCCSD ansatz circuit, it is reasonable to expect that a certain subset of the excitation terms may

be more efficiently implemented by one β transformation and some other excitation terms may be implemented more efficiently by yet another β transformation. Thus, dividing the set of excitation terms required for preparing a UCCSD ansatz state into subsets of excitation terms that may be more efficiently implemented by respective, appropriate choices of β transformations for each of the subsets may prove to be more advantageous in the quantum resource requirement. Optimizing over the tug of war expected between the overhead cost incurred due to the switching of the transformations and the savings obtained via the tailor-made choices of the transformations remains as future work.

Quantum simulations performed by quantum computers have long been thought to be one of the most promising quantum applications that will prove advantageous over classical computers. Despite the recent technological advancements made by the community, there still remains a gap between what a quantum device can realistically achieve and what is required to demonstrate a practical advantage in running a quantum simulation on a quantum device. In an attempt to bridge this gap, in this thesis, we focused on optimizing the quantum resource requirements for pre-FT regime quantum simulations of fermionic systems. We considered GT methods to obtain significant quantum resource savings, upwards of 20% in some instances, over the conventional JW or BK transformations in practice. While these results were applied to a specific set of fermionic systems in this thesis as a concrete example, we expect similarly-spirited works could leverage our methodologies. We believe the savings in the quantum resource requirement demonstrated by our approaches help bring the day of solving practical problems using a quantum computer closer.

Chapter 6: Integrated ansatz optimization

6.1 Overview

In the preceding chapter, we explored methods for optimizing the ansatz. However, many of the techniques mentioned are disjointed, resulting in unnecessary complexity. The objective of this chapter is to consolidate these techniques in a manner that facilitates their integration.

Relevant to this chapter, references [5, 81] focus on the quantum circuit optimization with qubit-to-qubit connectivity and hardware constraints in mind. Reference [78] leverages execution sequences of quantum subroutines to minimize algorithmic errors incurred. References [82, 83] exploit various symmetries of chemistry problems to better optimize quantum circuits. Most closely related to our work are [1, 4], where a complete-graph connectivity between qubits, available in trapped-ion [84] or neutral-atom [85] quantum computers, is considered. Explored and exploited include quantum subroutine execution sequence degrees of freedom and quantum data compression, among others, to significantly optimize chemistry simulations on a near-term quantum machine.

Now with the integration, we make the following contributions:

- For several state-of-the-art simulation methods where circuits are designed hierarchically, we streamline and reduce the hierarchy with high-level optimization methods, to reduce

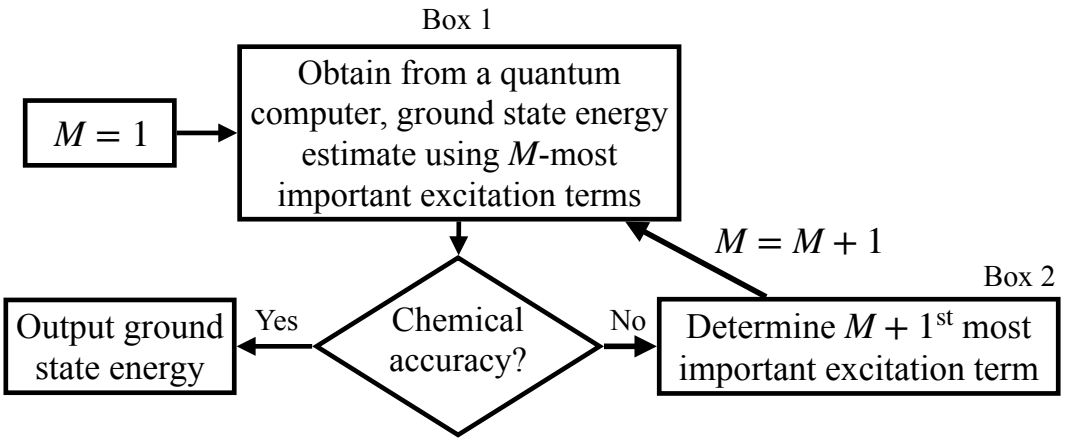


Figure 6.1: Typical VQE cycle. M is a parameter that denotes the size of the ansatz, such as the number of excitation terms. The expressivity of the ansatz is increased as the size becomes larger. Quantum computers here provide the ground-state energy estimate, given the ansatz specification. The ansatz size is gradually increased until it is sufficiently large to allow for the energy estimate to be within a pre-specified threshold from the ground state energy, such as chemical accuracy. Other termination criteria, including requiring a minimum energy-estimate improvement, can be considered as well.

gate counts.

- We perform high-level circuit optimizations using advanced computational tasks (such as the Travelling Salesman Problem and Graph Coloring) that we solve with high performance solvers.

- For simulations of near-term interest, we reduce published gate counts by 3.5-24%.

- We demonstrate optimizations applicable to wider classes of near-term simulations and real-time dynamics simulations.

6.2 Baseline optimization

Fig. 6.1 illustrates the computational loop of the updated Variational Quantum Eigensolver (VQE) algorithm, where the number of ansatz terms is now increased iteratively.

Focusing on Box 1 of Fig. 6.1, [1, 4] investigated the compilation and optimization of the UCCSD ansatz, where, by use of the first-order *Trotter formula* [68], the circuit implemented was of the form $\prod_k e^{O_k}$, where each O_k denotes individual summands of $Z - Z^\dagger$. At a high level, (overly-)complicated optimization techniques used in [1, 4] can be summarized as:

"Bosonic" encoding – Exploiting quantum data compression, when the state of an electron pair is symmetric, e.g., a superposition of $|00\rangle$ and $|11\rangle$, one can use only a single qubit, not two, to encode the state of the pair on a quantum computer. This also compresses the quantum operations that involve the pair. Previously, the compression was used only when two such symmetric pairs were provided simultaneously[1].

Fermion-to-qubit transformation matrix – Operators that create and annihilate fermions obey certain rules. On the other hand, operators that excites ($|0\rangle \to |1\rangle$) and de-excites ($|1\rangle \to |0\rangle$) a qubit obey different rules. Therefore, fermionic operators need to be properly transformed to qubit operators (e.g., Pauli matrices X, Y, Z), prior to their implementation on a quantum computer. Here, the choice of transformation drastically changes the computational resource requirement. Previously, the space of an $n \times n$ *upper-triangular* reversible binary matrix, which can represent a subset of valid transformations, was searched using particle swarm optimization.

Fermionic level labeling – The embedding of electronic sites onto qubits is another choice that can drastically change the resource requirement. Due to factorially large embedding space, a simple greedy approach that tries a transposition at a time was used.

Intra-excitation term ordering – Each term implemented in a quantum circuit was e^{O_k} (see above). O_k, once transformed to qubit operators, becomes a sum of Pauli strings, a tensor

[1]The compressed operations follow the so-called *hard-core bosonic algebra*, hence the name "Bosonic" encoding.

Figure 6.2: Different choices of target qubit implmeneting the same $e^{i\theta ZZ/2}$.

product of Pauli matrices. e^{O_k} was implemented by applying the circuit that implements the matrix exponentiation of each Pauli-string summand, one after another. An exhaustive search was performed to find the best ordering of the summands for each e^{O_k}.

Target qubit choice – Notice both circuits implement $e^{i\theta ZZ/2}$ in Fig. 6.2. The target qubit choice ($\oplus$ of CNOT) in the circuit-level implementation can in fact result in different gate cancellations. Previously, all Pauli strings from the same O_k term shared the same target.

Inter-excitation term ordering – The relative ordering between O_k can also be used to optimize quantum circuits, aiming to expose as much similarity between neighboring O_k's. Previously, a doubly-greedy approach was used, one to group as many O_k's implementable with the same target and the other to find the best O_k-ordering within each group, to find locally-optimized circuits.

For Box 2 of Fig. 6.1, recall in a typical VQE approach, one simply measures the expectation values of a set of operators that correspond to the Hamiltonian. Shown in [1] was that this is nothing but first-order perturbation theory. Therefore, by additionally measuring expectation values of yet another set of operators that correspond to second-order corrections, the energy estimate is improved, and perhaps more importantly, which excitation term would need to be additionally considered to improve ground-state energy estimate is determined. Indeed, the rapid convergence of the energy estimate afforded by it in turn resulted in needing a fewer number of excitation terms to reach a pre-specified convergence goal, such as the oft-discussed chemical

accuracy in the literature [86].

The baseline approach for Box 1 of Fig. 6.1 detailed in this section will be contrasted with our proposed methods in the next section. For Box 2 of Fig. 6.1, our work also employs the second-order perturbation theory based approach (i.e. HMP2 in [1]).

6.3 Proposed methodologies

We drastically improve the quality of optimization while massively streamlining the procedure. In contrast to the baseline, we use:

1. *Hybrid encoding*: a generalization of bosonic encoding to encompass partial quantum information compressibility for arbitrarily many orbitals,

2. *Advanced sorting*: a proper superset of the union of Intra-excitation term ordering, Target qubit choice, and Inter-excitation term ordering, and

3. *Advanced fermion-to-qubit transformation*: a proper superset of the union of fermion-to-qubit transformation matrix and fermionic level labeling.

Further, our methodologies are based on well-studied graph properties and optimization problems, obviating the need for custom heuristics, amenable to the use of powerful commercial solvers.

Fig. 6.3 shows a complete flow diagram for our advanced compilation and optimization methodology for VQE simulations of fermionic systems. Briefly, for a given set of excitation terms, we first classify the excitation terms into bosonic, hybrid, and fermionic classes (see Section 6.3.1), depending on the parity symmetry of each excitation terms. The bosonic terms can be

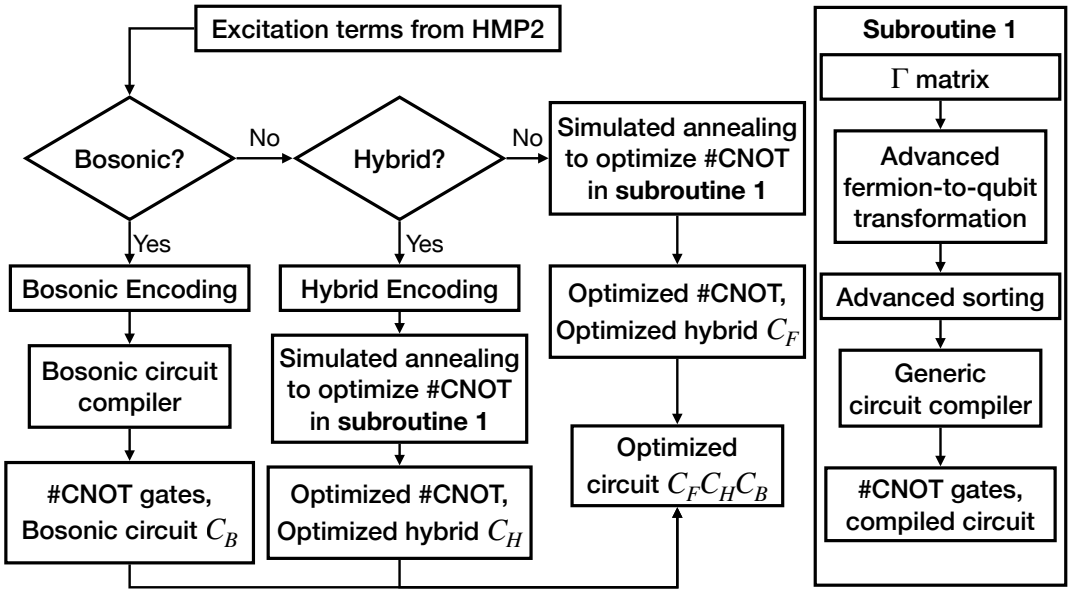


Figure 6.3: Our procedure to compile and optimize a VQE circuit.

simply compiled to quantum circuits using the known procedure described in [4]. For the hybrid and fermionic terms, we use the procedures described in Section 6.3.1 and Section 6.3.2. Recall, in order to compile the unitary operators into a quantum circuit, a choice of fermion-to-qubit transformation must be made. We search for the transformation that admits the best optimization using the procedure described in Section 6.3.3.

6.3.1 Hybrid encoding

Consider a state that is a linear superposition of $|00\rangle$ and $|11\rangle$, or $|01\rangle$ and $|10\rangle$. There is exactly one quantum bit of information encoded, so a single qubit is sufficient. Now consider a system of n qubits where, by the circuit elements to be applied and with the knowledge of the input state, a two-qubit subsystem state can be predicted to be a superposition of $|00\rangle$ and $|11\rangle$, or $|01\rangle$ and $|10\rangle$. Instead of two qubits, using only one qubit suffices. The wave function satisfying

this property have number parity symmetry. Specifically,

$$P_{ij}|\psi\rangle = \pm|\psi\rangle, \tag{6.1}$$

where

$$P_{ij} = (-1)^{c_i^\dagger c_i + c_j^\dagger c_j} \tag{6.2}$$

is the number parity operator for spin orbitals i and j.

Any symmetry-preserving operator over two qubits that commutes with P_{ij} can be compressed to a single-qubit operator. The compressed encoding can be recovered to a two-qubit state through a CNOT gate (up to single qubit gates), i.e.,

$$\mathrm{CNOT}(a\,|0\rangle + b\,|1\rangle)\,|0\rangle \mapsto a\,|00\rangle + b\,|11\rangle \tag{6.3}$$

for an even-parity compression and

$$\mathrm{X}_1\mathrm{CNOT}(a\,|0\rangle + b\,|1\rangle)\,|0\rangle \mapsto a\,|10\rangle + b\,|01\rangle \tag{6.4}$$

for an odd-parity compression.

Applied to two pairs of subsystems that each contain the symmetry, explored in [4] was the so-called bosonic encoding. Consider a double-excitation unitary $\exp(\theta c_p^\dagger c_q^\dagger c_r c_s - h.c.)$ on $|\psi\rangle$, an eigenstate of the number parity operators P_{pq} and P_{rs} with eigenvalues $+1$. Leveraging the

parity symmetry and defining the so-called hard-core bosonic operators

$$d_{pq}^{\dagger} := c_p^{\dagger} c_q^{\dagger} \tag{6.5}$$

$$d_{rs} := c_r c_s, \tag{6.6}$$

$d_{pq}^{\dagger}$ and d_{rs} map to the Pauli raising/lowering operators $\sigma^{\pm}$. The reduced, "bosonic" unitary operator is then $\exp(\theta \sigma_p^+ \sigma_r^- - h.c.)$.

Consider a different $|\psi\rangle$, an eigenstate of the number parity operator P_{pq} with eigenvalue +1, but not an eigenstate of the operator P_{rs}. Following the procedure above, we can reduce the excitation operator to $\sigma_p^+ c_r c_s$, a one-qubit and two-fermion operator. The operator is hereafter referred to as a hybrid double excitation term.

Note a hybrid double excitation term, in order for it to be reduced, requires the input state to contain the symmetry. This raises an important challenge: Ordering of the excitation terms in the ansatz circuit. For example, consider two double excitation terms $h_1 = c_2^{\dagger} c_3^{\dagger} c_5 c_6$ and $h_2 = c_4^{\dagger} c_5^{\dagger} c_7 c_8$ and consider an input state $|\psi\rangle$ with the pair symmetry on (5,6), i.e., $P_{56}|\psi\rangle = |\psi\rangle$. Applying $e^{\theta_1 h_1 - h.c.}$ first, the (5,6) symmetry is preserved, hence compressible. Apply $e^{\theta_2 h_2 - h.c.}$ first, then try to apply $e^{\theta_1 h_1 - h.c.}$: the action of the former breaks the input-state symmetry for the latter, preventing the its compression.

Benefits – The minimum CNOT counts for a double-excitation term was 13 in [4]. For bosonic, the count is reduced to two [4]. If a hybrid-enabled reduction is applicable, the count becomes seven (See Fig. 6.4(a)). Thus, finding an optimal ordering of the excitation terms to be applied in our ansatz circuit that enables maximal compression via symmetry preservation becomes an important task.

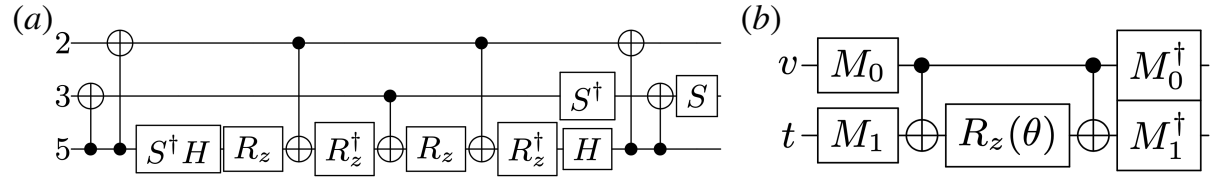


Figure 6.4: (a) Optimized circuit for the term $U = e^{\theta_{2356} c_2^\dagger c_3^\dagger c_5 c_6 - h.c.}$. The R_z gate used in the circuit refers to $R_z(\theta_{2356}/2)$. The qubit indices are labeled on the left of the circuit. (b) Template circuit for the unitary $U = e^{-i\frac{\theta}{2}\sigma_0^v \otimes \sigma_1^t \otimes \cdots}$, where t denotes the target qubit location and v denotes control qubit locations. $\sigma_i \in \{X, Y, Z\}$, and $M_i \in \{H, S^\dagger H, \mathbf{1}\}$. If $\sigma_i^j = X$, $M_i^j = H$. If $\sigma_i^j = Y$, $M_i^j = S^\dagger H$. If $\sigma_i^j = Z$, $M_i^j = \mathbf{1}$.

Solution – In general, bosonic and hybrid encoding can be considered for any pair of spin orbits. Here, we limit the consideration to the spin degree of freedom, i.e. $c_p^\dagger c_{p+1}^\dagger$ for odd p, since most significant physical excitation terms tend to preserve such symmetry. The symmetry-preserving ordering problem is now ready to be mapped to the well-known graph vertex coloring problem (GVCP). We do this in three steps: Graph construction, reduction, and coloring.

Graph construction – Consider a set of hybrid terms that are the rotation associated with $h_i = c_{p_i}^\dagger c_{p_i+1}^\dagger c_{q_i} c_{r_i}$ (or also $h_i = c_{p_i}^\dagger c_{q_i}^\dagger c_{r_i} c_{r_i+1}$ if the index ordering matters as in the UCC ansatz), where $1 \leq i \leq N_h$ and N_h is the number of hybrid excitation terms. The goal is to maximize the number of hybrid terms that admit compressed implementation. To achieve this, we first map the set of the hybrid terms to a directed graph G. In particular, the graph G contains N_h vertices, where each vertex corresponds to one hybrid excitation term h_i. We assign a directed edge from h_j to h_k if h_j rotation breaks the parity symmetry required for h_k rotation. Specifically, we use a *sufficient* condition for h_i breaking the parity symmetry required for h_j, i.e.,

$$B(h_i, h_j) = (q_i = p_j) \vee (r_i = p_j) \vee (q_i = p_j + 1) \vee (r_i = p_j + 1), \tag{6.7}$$

where $\vee$ denotes logical or operation and the directed edge from h_i to h_j exists if the logical

$B(h_i, h_j)$ evaluates to true. Concretely,

$$G = (V, E), \tag{6.8}$$

$$V = \{h_i | 1 \leq i \leq N_h\}, \tag{6.9}$$

$$E = \{(h_i, h_j) | \text{ if } B(h_i, h_j)\}, \tag{6.10}$$

where (x, y) denotes a directed edge from x to y.

Graph reduction – In this step, we consider the source and sink vertices of G. Sink vertices S_{sink} are the vertices that do not have any outgoing edge. Source vertices S_{source} are the vertices that do not have any incoming edge. By design, the sink vertices of the graph G do not break any parity symmetry used for hybrid-excitation compression. Therefore, all the sink vertices are implemented first. For the source vertices, the parity symmetry needed is always preserved regardless of which hybrid excitation terms were implemented beforehand. Therefore, all the source vertices are implemented last. Remove the sink and source vertices from G. Repeat the removal until there is no more sink or source vertices left.

Graph coloring – We start with a reduced G without any source or sink. Note, not all the hybrid excitation terms in the reduced graph may be implemented with compression, since implementing one with compression may break the parity symmetry required for another. In order to thus maximize the number of hybrid excitation terms that can be implemented with compression, we use GVCP.

Start by constructing an undirected graph via removing the direction information in the reduced graph G. Label each vertex with a different color such that no two vertices sharing the same edge have the same color, while minimizing the number of colors used. In other words, for

a given undirected graph $G = (V, E)$, we aim to find its chromatic number k and the coloring of each vertex when only k colors are used. Note every vertex with the same color can be implemented with compression. By minimizing the number of colors used, then finding the largest set of vertices with the same color, we obtain a heuristic solution to the problem of finding the maximal number of vertices that can be implemented with compression.

Summing up, referring the maximal set of vertices that can have the same color as S_{color}, we convert sets S_{sink}, S_{color}, and S_{source} to circuits C_{sink}, C_{color}, and C_{source}, respectively, using the procedure described in the next section. The resulting circuit is of the form $C = C_{source}C_{color}C_{sink}$. The hybrid terms or vertices that lie outside the three sets are folded into the fermionic terms.

6.3.2 Advanced sorting

Instead of Trotterizing at the excitation-operator level, implementing e^{O_k} one at a time, a Pauli-string level Trotterization can be applied: Expand O_k according to $\theta \sum_\kappa \otimes_{i=1}^N \sigma_{\kappa,i}$, $\sigma_{\kappa,i} \in \{I, X, Y, Z\}$, then reorder the Pauli strings across *different* O_k terms. Finding the best ordering of Pauli strings can be viewed as a traveling salesman problem (TSP), with the distance as the gate-count savings between the consecutive Pauli strings, whose matrix exponentiations are to be implemented one after the other. Now, allow for each Pauli string to have its own "target qubit," in contrast to the baseline. Mapping each string to a cluster, and each subvertex of the cluster being a possible target, the problem becomes determining which subvertex to visit per cluster while visiting each cluster once to make a loop. Hence generalized TSP (GTSP).

Note the exact form of the string is determined by the excitation terms considered and the

choice of fermion-to-qubit transformation. Assuming a set of Pauli strings are provided, our task is to determine the implementation order of the strings based on the gate cancellation that occurs in between them. We adapt a well-known meta-heuristic algorithm to find a locally optimal solution for the GTSP.

The input for the problem is a set of excitation terms (E), either the fermionic excitation terms, or a segment of the hybrid excitation terms, i.e., source, color, or sink, and a fermion-to-qubit transformation. Using the transformation, the input excitation terms are converted to a set of Pauli string operators $E \rightarrow \{P_k | 1 \leq k \leq M\}$, where M is the number of the Pauli strings, $P_k = \otimes_{i=1}^{N} \sigma_{k;i}$ and $\sigma_{k;i} \in \{I, X, Y, Z\}$. Individually exponentiated, a conventional circuit synthesis method uses $2(w-1)$ CNOT gates for each string, where w is the weight of the string defined as the number of non-I matrices in a given string. In particular, one chooses a target qubit to be any one of the qubits a non-I Pauli matrix acts on, then uses the circuit template shown in Fig. 6.4 (b).

Consider now implementing two circuit elements that correspond to two different Pauli strings in a row. Depending on the strings and the specific choice of the target qubit, different numbers of CNOT gates may be canceled. Example: Suppose we have $P_1 = XXXY$ and $P_2 = XXYX$. In the first scenario, we consider both strings with the common target of the fourth qubit, i.e., $t_1 = t_2 = 4$. At the interface of two circuits that correspond to $[P_1, t_1]$ and $[P_2, t_2]$, CNOT cancellations occur, resulting in a single CNOT (see Fig. 6.5 (a)). Let us consider the second scenario where $t_1 = t_2 = 1$. Then at the interface, a CNOT cancellation occurs, resulting in two CNOTs (see Fig. 6.5 (b)). Point made: The choice of target matters. Note $t_1 = t_2$ is required for CNOT cancellations. Indeed, when given $[P_1, t_1]$ and $[P_2, t_2 = t_1]$, the number of CNOT cancellations obtained at the interface may be computed as $2(w_1 - 1) + 2(w_2 - 1) - \sum_i \omega_i$,

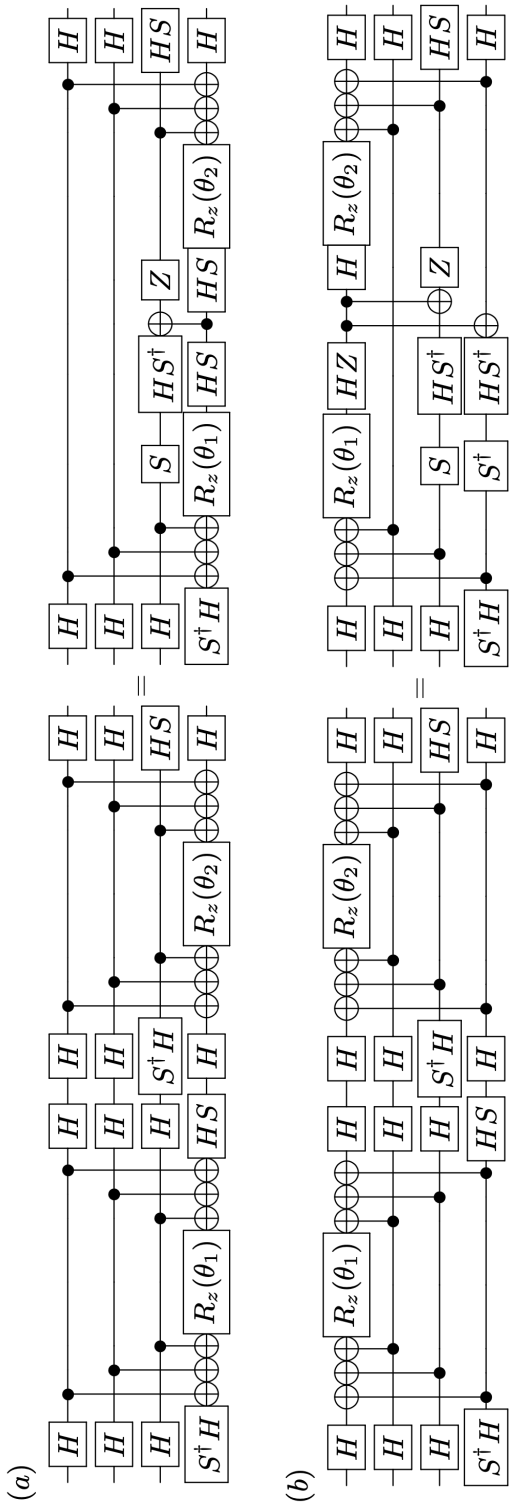


Figure 6.5: Circuits for the unitary $U = e^{-i\frac{\theta_1}{2}XXXY} e^{-i\frac{\theta_2}{2}XXYX}$ with two different target qubit choices. (a) the fourth qubit. (b) the first qubit.

where w_k is the weight of the strings P_k, i traverses every non-target qubit, and $\omega_i \in \{0, 1, 2\}$ is zero if any of $\sigma_{k,i}$ of P_k, $k \in \{1, 2\}$, is identity, two if we have, on the target, any one of (X, Y), (Y, X), (X, X), (Y, Y), or (Z, Z) collisions and, on the control, any one of (X, X), (Y, Y), or (Z, Z) collisions, and one otherwise. See [1, 4, 5] for the CNOT cancellation rules.

In order to maximize the cancellation, we map the sorting problem to GTSP. In GTSP, we are given a partition of vertices into clusters and we aim to find a minimum-length path that includes exactly one vertex from each cluster. Note the two degrees of freedom: the ordering of the strings and the target qubit choice for each string. We assign each vertex of a weighted undirected graph G with each $[P_i, t_i]$. We insert an edge if and only if $P_i \neq P_j$, and the weight of the edge is the number of CNOT reduction expected from placing the circuit blocks for $e^{iP_i\theta_i}$ and $e^{iP_j\theta_j}$ next to each other using target qubits t_i and t_j, respectively. To sum up, the weighted undirected graph G is of the form

$$G = (V, E) \tag{6.11}$$

$$V = \{[P_k, t_k] | 1 \leq k \leq M, 1 \leq t_k \leq N \ni \sigma_{k,t_k} \neq I\} \tag{6.12}$$

$$E = \{(v_{k_1}, v_{k_2}, d_{k_1k_2}) | (v_{k_1}, v_{k_2}) \in V^2 \ni P_{k_1} \neq P_{k_2}\}, \tag{6.13}$$

where (v_a, v_b, d) denotes an edge connecting v_a and v_b with weight d. We now partition all the vertices V in the graph G into M clusters i.e. $V = V_1 \cup V_2 \cup \cdots \cup V_M$. The jth cluster contains all the vertices with the same P_j and all possible target qubit indices. In order to maximize the CNOT reduction, the objective is to find a maximum weight cycle containing exactly one vertex from each cluster. This problem is equivalent to solving the GTSP.

The method we use to solve GTSP is as follows. We first multiply the weight of each

edge by -1 so that the minimum path length in the graph G corresponds to the maximum CNOT reduction. If we find the optimal path X^*, we can convert it to the optimal permutation (the order in which each cluster is visited) and the optimal target qubit choices (the specific vertex visited in each cluster). With the ordering and the target qubits used determined, the ansatz circuit C is determined.

6.3.3 Advanced fermion-to-qubit transformation

Consider $\Gamma \in \mathrm{GL}(N, 2)$, the group of binary $N \times N$ invertible matrices, in contrast to being restricted to an upper-triangular one as in the baseline. For $x \in \mathbb{F}_2^N$, where $\mathbb{F}_2$ is the binary field, Γx is also in $\mathbb{F}_2^N$. Indeed, Γ denotes a linear reversible circuit, a subset of Clifford [87]. Since Clifford maps a Pauli string to another, U_Γ, a unitary implied by Γ, maps a Pauli string P_k to another Pauli string $U_\Gamma P_k U_\Gamma^\dagger$.

One of the commonly used fermion-to-qubit transformation is the Jordan-Wigner (JW) transformation: $c_i = (\bigotimes_{j<i} Z_j)\sigma_i^-$, where Z_j here is the Pauli-Z matrix on qubit j. Applying the transformation to fermionic excitation operators, one obtains Pauli strings. Make no further changes with $\Gamma = 1$: This is the JW transformation. In the advanced fermion-to-qubit transformation, $\Gamma \neq 1$ is used to enable the best optimization in the steps considered previously.

Problem – The choice of Γ depends heavily on the relatively small number of excitation terms considered, since asymptotically Bravyi-Kitaev (BK) transformation and ternary tree transformation [88] are already optimal. Further, the search space for Γ is prohibitively large ($\sim 2^{N^2}$), even for a modest number of qubits N.

Solution – We exploit the topology formed by the excitation terms, i.e., we keep track

of index pairs of neighboring fermion operators in the creation or annihilation part of every fermionic double excitation term. This way, we can construct a graph with its vertices being the indices and its edges being the index pairs. The graph constructed, if disjoint, is divided into disjoint pieces, where each piece contains a vertex set. We use these sets as blocks in a block diagonal Γ.

To illustrate, let us consider an $N \times N$ Γ that is block diagonal in blocks of size $N/2 \times N/2$. Given a Pauli string P_x obtained from using the JW transformation, the block diagonal Γ transforms the first half of P_x separately from the latter half of P_x, since $P_x \mapsto U_\Gamma P_x U_\Gamma^\dagger$. While it may not guarantee a globally optimal solution, the block diagonal Γ provides a way to locally optimize within each block. Indeed, each block can be searched separately, thereby reducing the search problem size dramatically.

For each block, we use simulated annealing (SA), instead of particle swarm optimization that tends to get stuck in local minima, to find an optimal Γ. Briefly, inspired by the heating and cooling processes of a thermodynamic system, SA is a procedure to approximate the global optimum of a function, $f(x)$, over a discrete search space $x \in \mathcal{X}$. In particular, in analogy to the thermodynamic free energy, we use the Metropolis-Hasting sampling algorithm in the $\mathcal{X}$ space such that $\pi(x) \propto \exp(-f(x)/T)$, where $\pi(x)$ is the unnormalized probability distribution and T is a parameter in analogy to the temperature in the thermodynamics. We can then gradually reduce the temperature T to $T = 0$ such that the probability of sampling the global minimum becomes overwhelmingly large compared to other configurations. Therefore, the global minimum can be approximated.

6.4 Examples

6.4.1 Example - hybrid encoding

In this section we demonstrate our procedures to categorize hybrid terms. Suppose we have nine hybrid double excitation fermionic terms $\{h_0, h_1, \cdots, h_8\}$ such that the directed graph, constructed according to Section 6.3.1 is that shown in Fig. 6.6 (a). One example is

$$h_0{=}a_9^\dagger a_{12}^\dagger \boldsymbol{a_3 a_4}, h_1{=}\boldsymbol{a_{11}^\dagger a_{12}^\dagger} a_3 a_6,\ h_2{=}a_{20}^\dagger a_{21}^\dagger \boldsymbol{a_5 a_6},$$

$$h_3{=}a_{19}^\dagger a_{22}^\dagger \boldsymbol{a_5 a_6}, h_4{=}a_{13}^\dagger a_{16}^\dagger \boldsymbol{a_1 a_2},\ h_5{=}a_{11}^\dagger a_{14}^\dagger \boldsymbol{a_5 a_6},$$

$$h_6{=}\boldsymbol{a_{13}^\dagger a_{14}^\dagger} a_5 a_8, h_7{=}a_{13}^\dagger a_{16}^\dagger \boldsymbol{a_7 a_8},\ h_8{=}\boldsymbol{a_{17}^\dagger a_{18}^\dagger} a_3 a_8.$$

Notice the part in bold font can potentially be implemented with compression. Taking 'Sink' and 'Source' terms out iteratively, then dropping the directedness of the edges of the graph, we obtain an undirected graph, shown in Fig. 6.6 (b).

Next, to detail the coloring procedure, we show two random orders in Fig. 6.6 (c). The two random orders are: Order 1 $h_1{-}h_5{-}h_0{-}h_6{-}h_7$ and Order 2 $h_1{-}h_7{-}h_6{-}h_5{-}h_0$. For Order 1, we color the first term h_1 with blue. Since the next term h_5 is connected to h_1, we add a new color red. The third term h_0 is connected to h_1 but not h_5. Since we minimize the number of colors used, we reuse the red color for h_0. h_6 is connected with h_5, but not h_1 nor h_0, so we reuse the blue color. Lastly, h_7 neighbors h_6 only, we can reuse the color red. A similar procedure can be employed for Order 2. Here, for h_5, it is connected to both h_1 and h_6, whose respective colors are blue and red. Thus, a third color cyan is added.

According to Fig. 6.6 (c), Order 1 contains the largest color set which is red. As a result, we have $S_{sink} = \{h_2, h_3\}$, $S_{source} = \{h_4, h_8\}$, $S_{color} = \{h_0, h_5, h_7\}$ implemented with compression while $\{h_1, h_6\}$ are considered as regular fermionic terms.

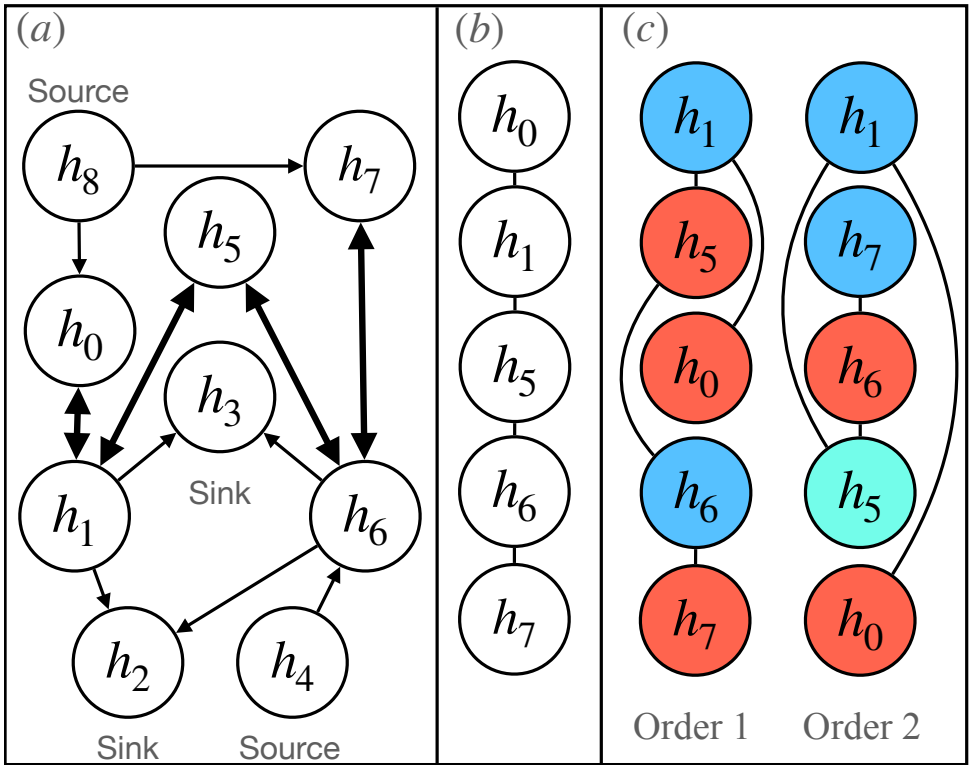


Figure 6.6: Example procedure to categorize hybrid terms using the graph vertex coloring problem. See Section 6.4.1 for detailed description. (a) *Graph construction* – Directional graph for nine hybrid double excitation terms where the direction $h_i \rightarrow h_j$ implies the term h_i breaks the parity symmetry of h_j. (b) *Graph reduction* – After removing the sink and source vertices iteratively, the directedness of the edges of the remaining graph is dropped. (c) *Graph coloring* – The coloring order is from up to down. The lines denote the edges and the pair of vertices adjoined by an edge must be be colored differently.

6.4.2 Example - advanced sorting

Recall in our advanced sorting, we first construct a weighted, undirected graph, given a set of Pauli strings. Consider an example, where P_0=$IIXXYXII$, P_1=$IIXXXYII$, and P_2=$XXIIIIXY$. The valid target qubits are then $\{3, 4, 5, 6\}$, $\{3, 4, 5, 6\}$, and $\{1, 2, 7, 8\}$, re-

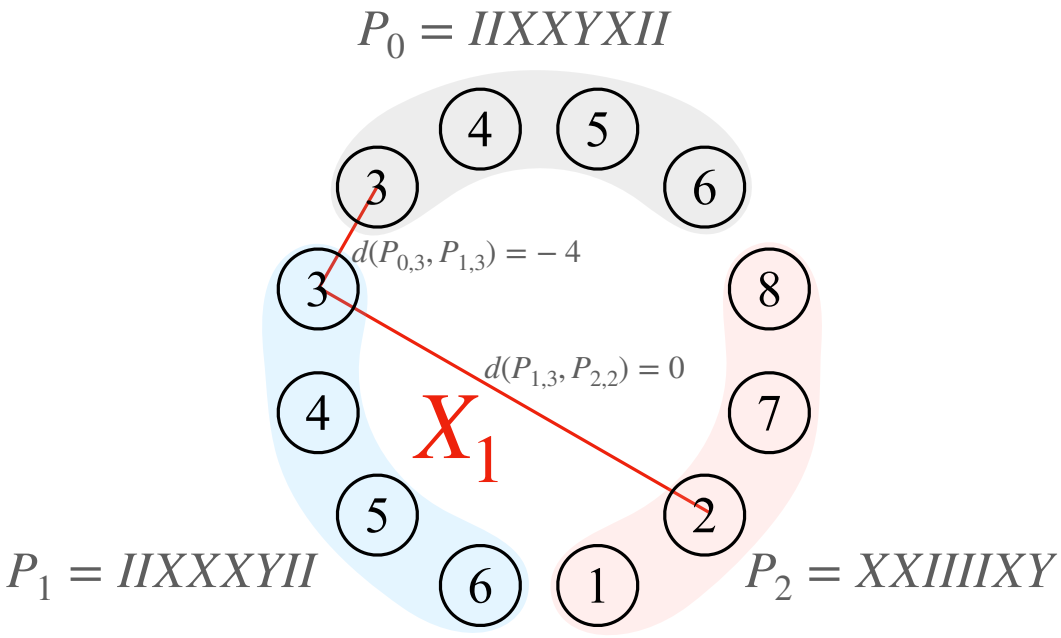


Figure 6.7: GTSP graph constructed for our example advanced sorting problem. Each of the circled numbers inside the same color block (cluster) represents a valid target qubit index of Pauli string P_j. A valid path traverses each cluster once, visiting one circled number per cluster, such as in $X_1 : ([P_0, 3], [P_1, 3], [P_2, 2])$. The path weight of X_1 is the sum of the weights of the edges in the path, i.e., $([P_0, 3], [P_1, 3], -4)$ and $([P_1, 3], [P_2, 2], 0)$. The path weight is thus -4.

spectively. The graph constructed can then be visualized as in Fig. 6.7 (edges not shown for visual clarity). A valid path is a series of connected edges, where each cluster (Pauli string) is visited exactly once. The weight d of an edge $([P_j, t], [P_k, t'], d)$ is computed as the negative of the number of CNOT gates cancelled when implementing $e^{-i\theta P_j/2}$ and $e^{-i\theta' P_k/2}$ one after the other with target qubits t and t', respectively, according to the method detailed in Section 6.3.2. In this example, among all 4^3 possible paths, we aim to find a path that minimizes the path weight, defined as the sum of the weights of the edges in the path. Our sorting problem is thus fully mapped to the GTSP problem.

6.4.3 Example - optimization for advanced fermion-to-qubit transformation

We show in this section an example where (i) we reduce the search space of Γ based on the excitation term topology and (ii) the effect of the choice of a different Γ on the Pauli strings whose matrix exponentiations are to be implemented. To illustrate (i), consider two excitation terms $a_9^\dagger a_8^\dagger a_3 a_1$ and $a_6^\dagger a_5^\dagger a_2 a_1$. The creation part has connected clusters $\{8, 9\}$ and $\{5, 6\}$, while the annihilation part has a connected cluster $\{1, 2, 3\}$. Therefore our Γ matrix candidate may consist of a 3×3 block for indices $\{1, 2, 3\}$ and two 2×2 blocks for indices $\{8, 9\}$ and $\{5, 6\}$ respectively. To illustrate (ii), consider an example Pauli string $P = \mathrm{XXIIXY}$, say, obtained as a part of the JW transformation of some fermionic excitation term. For a block diagonal Γ candidate, consider

$$\Gamma = \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right], \tag{6.14}$$

where we explore non-identity upper left and lower right 2×2 blocks in the hopes to change the input Pauli string to a form more amenable to better circuit optimization by the rest of our circuit optimization procedures. The particular Γ shown above corresponds to applying CNOTs on the

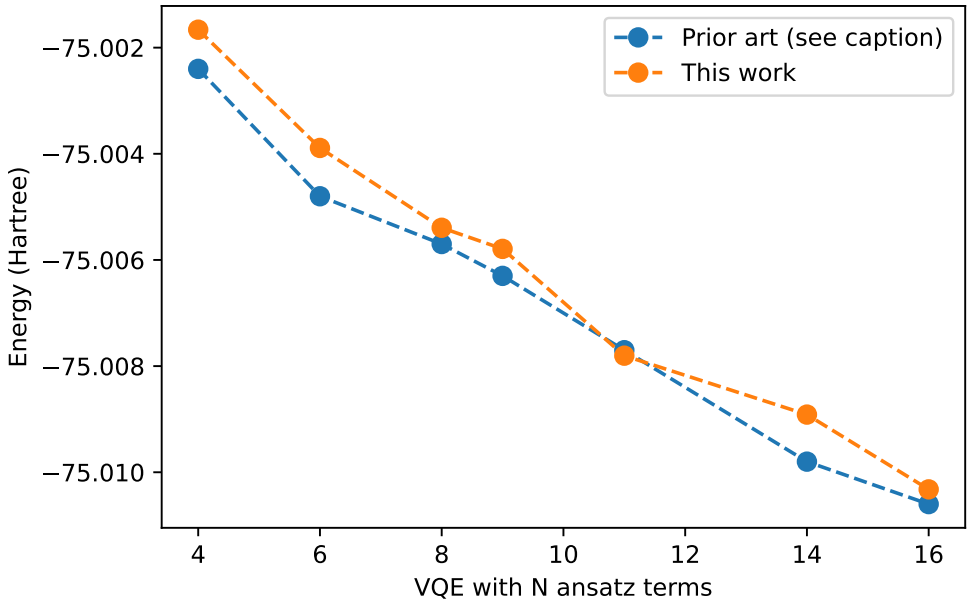


Figure 6.8: Ground-state energy estimate of the water molecule as a function of the number of ansatz terms in STO-3G basis. Blue: Energies reported in [1]. Orange: Energies obtained via the methodology reported in this chapter. Observed is that the energies obtained between the two methods are comparable. The number of ansatz term sufficient to reach the chemical accuracy is 17 (not shown) for both methods.

first two and the last two qubits, which transforms P as

$$(\mathrm{CNOT} \otimes \mathrm{II} \otimes \mathrm{CNOT}) \times (\mathrm{XXIIXY}) \times (\mathrm{CNOT} \otimes \mathrm{II} \otimes \mathrm{CNOT}) = \mathrm{XIIIYZ} \tag{6.15}$$

, which is a different and shorter Pauli string. The search space for the candidate Γ here is indeed a much smaller one to explore than that for the full 6×6 matrix in our current example. Our block-diagonal-based strategy can be employed to an arbitrary-sized generalized transformation we consider, i.e., a linear reversible circuit, which can be implemented using [89].

6.5 Circuit optimization results

In this section, we present the result for the optimized CNOT gate counts for a number of common molecules. We first generate the excitation terms using HMP2 algorithm proposed in [1]. We then use our advanced compilation and optimization methodology to synthesize optimal circuits.

Before we present the optimized results, we first provide the implementation details of the optimization solvers we used in our subroutines. In the hybrid encoding subroutine, one of our tasks is to search for an optimal solution for the graph coloring problem. We use a randomized, greedy coloring algorithm to minimize the number of distinct colors. Specifically, we color the vertices in multiple different orders, where the different orders are generated at random. Each time we assign a color to a vertex, we bias our selection, choosing the same color(s) as much as possible. A new color is added, only if needed. Our randomized greedy coloring algorithm thus explores beyond a local minimum, since we randomly generate several coloring orders and return the best found solution. In the advanced sorting subroutine, a GTSP solver is required. We use the genetic algorithm (GA) to solve the GTSP problem[2] as described in [93].

In Table 6.1, we show the CNOT gate counts of the optimized VQE circuits, simulating a variety of molecules. Compared therein are the CNOT gate counts obtained with different fermion-to-qubit transformations considered previously, including the JW, BK, and the generalized transformation (GT) [1], along with our methodology.

[2]Well-known heuristics such as Lin-Kernighan (LK) [90], used to obtain a good solution to the famous TSP, can indeed be adapted to handle the GTSP [91, 92]. However, the modifications needed to obtain good solutions to the GTSP are often non-trivial; For example, a 2-opt, LK solution of a GTSP, obtained by assuming a pre-determined set of vertices (one per cluster) to visit, can be one of the worst paths to visit each cluster, the moment we change the vertices to visit. We therefore leave the modified Lin-Kernighan approach as a future work and use here instead a simpler-to-implement GA.

The savings obtained by our advanced compilation and optimization vary in the suite of molecules we consider, ranging from 3.56%[3] (NH_3) to 24.00% (HF, LiH). Our methodology is capable of further optimizing the quantum circuits over the previous state of the art obtained by the GT approach [1]. These savings come with no hidden costs or accuracy loss in energy estimates. Indeed, we show in Fig. 6.8 the convergence of the ground-state energy estimates for the water molecule, comparing the results obtained by our advanced approach to those reported in [1]. We confirm the energy convergence hardly changes, evidenced by the fact that both approaches still use 17 excitation terms to reach chemical accuracy.

6.6 Discussion and outlook

While we focus on the compilation and optimization of the UCCSD ansatz, we note that our framework extends to other types of fermionic unitary operations. For example, the advanced sorting routine is immediately applicable to the qubit coupled-cluster (QCC) method. Moreover, one can also extend our optimization framework to the quantum simulation of time evolution of a fermionic system.

It is intriguing to further explore the fermion-to-qubit transformation by studying the Clifford conjugation of the Jordan-Wigner transformation, or encoding one fermionic degree of freedom with multiple qubits [94], instead of GL(N) we explored in this work. This may be of tremendous practical importance to future modular hardware running Hamiltonian simulations, since we may be able to use the Clifford transformation, pay upfront as a one-time cost, to "lo-

[3]The relatively small improvement ratio may be due to the inefficiency of the heuristic solvers used or the absence of a significantly better solution than known. Preliminary results (not shown) suggest the former to blame in part, with frequent local minima trapping. Another possibility may be the problem formulation itself, where the molecular orbital basis may be better chosen to reflect the inherent symmetry of the molecule.

calize" the interactions between "nearby," connected modules only. This way, inter-modular connection resources, expected to be very expensive [95–97], can be saved by a significant amount.

Table 6.1: Number of CNOT gates used for the VQE simulation of different molecules with different fermion to qubit transformations

Molecule	N_e	JW	BK	GT	Adv	Improve(%)
HF	3	30	29	25	19	24.00
LiH	3	30	29	25	19	24.00
BeH_2	9	70	71	60	53	11.67
NH_3	52	485	607	478	461	3.56
$H_2O(4)$	4	42	50	33	27	18.18
$H_2O(5)$	5	44	52	35	29	17.14
$H_2O(6)$	6	46	47	37	31	16.21
$H_2O(8)$	8	68	88	63	50	20.63
$H_2O(9)$	9	71	89	66	53	19.69
$H_2O(11)$	11	93	110	87	67	22.98
$H_2O(12)$	12	95	112	89	70	21.34
$H_2O(14)$	14	114	140	111	88	20.72
$H_2O(16)$	16	135	166	131	105	19.85
$H_2O(17)$	17	137	168	133	107	19.55

Note: N_e is the number of excitation terms considered in the UCCSD ansatz. JW/BK are the number of two-qubit gates with Jorden-Wigner and Bravyi-Kitaev transformations. GT is the number of two-qubit gates reported in [1]. The ansatz terms are determined according to the HMP2 ordering with STO-3G basis set and ground state geometry, detailed in [1]. In the first four rows, we report the results for the cases where the HMP2 method was used to reach chemical accuracy. In the rest of the rows, we report the results for the HMP2 progression for a water molecule. The number in the parentheses next to H_2O indicates the total number of excitation terms N_e considered for the UCCSD ansatz. The last result, $H_2O(17)$ achieves the chemical accuracy.

Chapter 7: Experimental demonstration of better initialization

7.1 Overview

In many optimization scenarios, having a well-informed initial guess closer to the true ground state often leads to faster convergence. The CAFQA is one such method capable of providing an alternative, typically superior initialization compared to the default Hartree-Fock (HF) initial state. The theoretical underpinnings of CAFQA are comprehensively detailed in [98], with a concise introduction provided in Section 7.2.

However, prior to this chapter, no experimental demonstrations had been conducted. In this chapter, we present the first experimental demonstration showcasing the benefits of improved initialization on a trapped-ion quantum computer.

This demonstration was made possible through a collaborative effort between theorists and experimentalists, who collectively selected the appropriate instance for demonstration and facilitated robust VQE runs on the trapped-ion quantum computer. While these efforts may not directly contribute to circuit ansatz optimization, they represent crucial and labor-intensive components of the overall research endeavor. Further details regarding these efforts are provided in Appendix C. The experimental results are presented in 7.3.

7.2 Better initialization with CAFQA

CAFQA stands for a Clifford Ansatze for Quantum Accuracy. The concept behind CAFQA is relatively straightforward: it aims to improve the initialization for VQE using classical computing resources. This is achieved for the following reasons:

1. CAFQA employs Clifford ansatz, which consists of Clifford gates. Clifford gatesets are sets of special gates that can be efficiently simulated classically [99]. In this chapter, the ansatz takes the form depicted in Fig. 7.1.

2. The discrete Clifford search space can be search efficiently via Bayesian optimization technique.

While Clifford ansatz is not universal, further optimization of VQE from the CAFQA result may be necessary to fine-tune the parameters (θ) in the ansatz if higher precision is required. In essence, the CAFQA result can be viewed as a generalized method for initialization, where the Hartree-Fock initialization is merely a special case.

7.3 Experimental demonstration

We selected LiH and BeH_2 as our test cases for this study. To accommodate the limitations of the quantum hardware, we employed a frozen-core approximation with the STO-3G basis, which effectively constrained the number of qubits to four. Additionally, we utilized qubit-wise commuting techniques to further reduce the number of Hamiltonian measurements required.

The VQE ansatz chosen for this study is depicted in Fig. 7.1. Initialization for the CAFQA method was prepared using a budget of 1000 for LiH and 3000 for BeH_2, based on guidelines

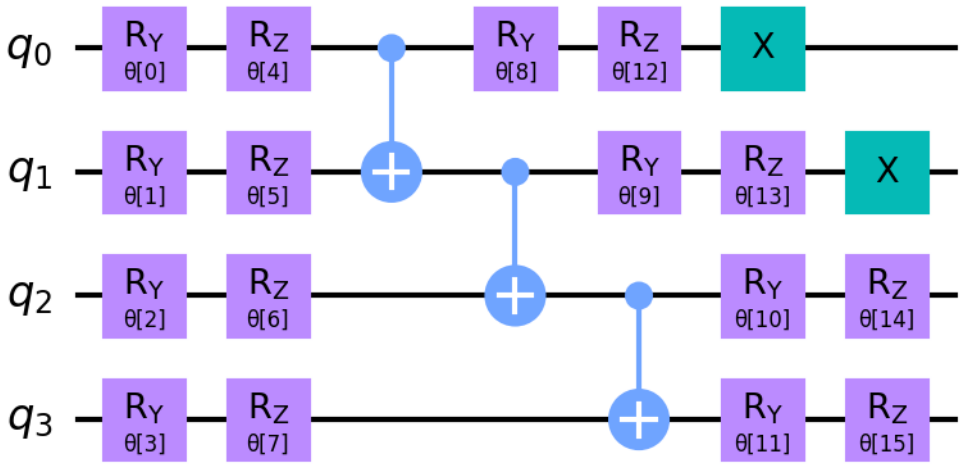


Figure 7.1: The Clifford ansatz used for experimental demonstration. To make it Clifford, $\theta[i] = \{0, \pi/2, \pi, 3\pi/2\}$. For further VQE optimization on a trapped-ion quzntum computer, the circuit was later transpired into trapped-ion native gate set $\{XX, R_x, R_z\}$ using Qiskit with optimization level 3. [7]

outlined in [98].

For classical optimization, we opted for the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm as the optimizer. The learning rates were calculated with a budget of 25.

The rationale behind these choices was to tailor our approach to the specific characteristics of the quantum hardware while ensuring efficient utilization of classical resources for optimization. Through careful selection of test cases, initialization methods, and optimization strategies, we aimed to demonstrate the effectiveness of CAFQA in enhancing the performance of VQE on real quantum hardware.

The results obtained for LiH and BeH_2 are depicted in Fig. 7.2 and Fig. 7.3 respectively. Each circuit was executed with a shot number of 300. Notably, both experiments demonstrate the accelerated convergence of the VQE with CAFQA optimization compared to traditional HF

initialization.

However, it is important to emphasize that these results serve as illustrative examples rather than rigorous proofs. To establish statistically meaningful conclusions, it would be necessary to conduct a more extensive analysis. This would involve generating multiple (potentially degenerate) CAFQA initialization outcomes, denoted as N_C, and subsequently performing VQE optimizations for each initialization a total of N_V times. In parallel, a similar procedure would be carried out using HF initialization.

By analyzing the outcomes, one would anticipate observing that the majority of the time, CAFQA initialization leads to faster optimization compared to HF initialization. Unfortunately, due to constraints on machine time, we were only able to showcase a scenario where $N_C = 1$ and $N_V = 1$ for both LiH and BeH_2. Despite this limitation, the results depicted in Figures 7.2 and 7.3 still provide valid evidence for the advantage of CAFQA initialization.

While a comprehensive demonstration involving larger values of N_C and N_V remains a project for future exploration, the presented findings serve as a promising indication of the efficacy of CAFQA in enhancing VQE performance.

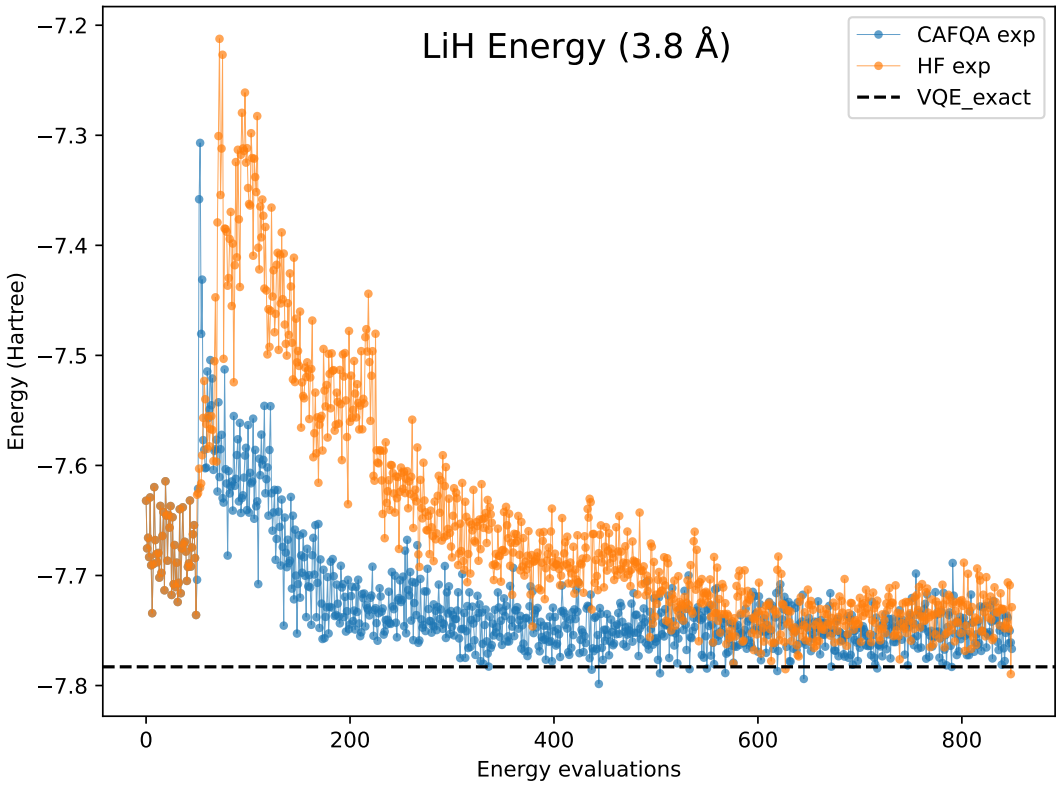


Figure 7.2: Post-CAFQA VQA tuning for LiH with bond length 3.8 Å demonstrated with a trapped-ion quantum computer. CAFQA initialization leads to 2.5x faster convergence compared to HF initialization.

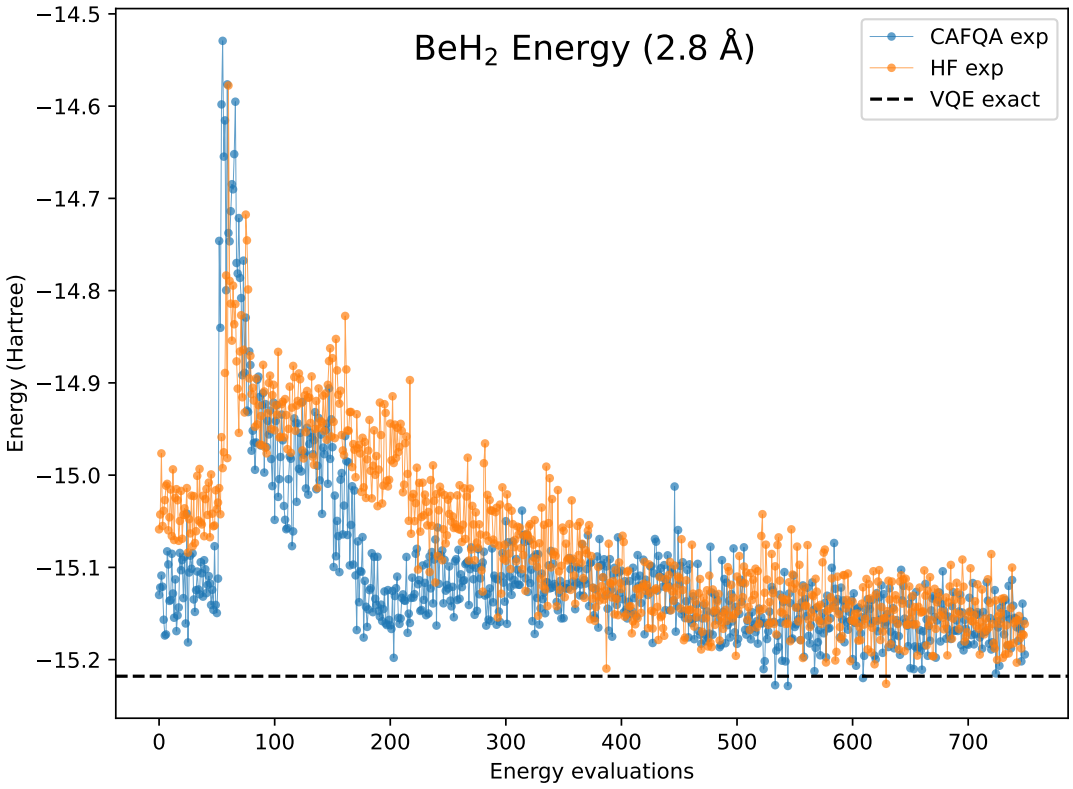


Figure 7.3: Post-CAFQA VQA tuning for BeH_2 with bond length 2.8 Å demonstrated with a trapped-ion quantum computer. CAFQA initialization leads to 2x faster convergence compared to HF initialization.

Chapter A: Qubit space reduction

In this section, we detail the procedures for the qubit space reduction (QSR) technique that optimizes the number of Pauli strings to be measured used in the main text. There are roughly two categories we consider: One is to treat the qubits that are in classically accessible states classically and the other is to take advantage of bosonic states, i.e., we expend only a single qubit for the two spin orbitals in the same spatial orbital with degenerate energy due to a particular choice of the ansatz.

As to the former, consider a Pauli string $S = A_0 A_1 \cdots A_{n-1}$ to be measured, where $A_i \in \{\sigma_x, \sigma_y, \sigma_z, I\}$, where I is a two-by-two identity operator. Conjugating this with a quantum state $|\Psi\rangle = |\psi\rangle|\phi\rangle$, where, without loss of generality, $|\psi\rangle$ denotes a quantum state of the qubits that are entangled due to the ansatz and $|\phi\rangle$ denotes a quantum state of the qubits that are in a classically accessible state, such as a computational basis state, we may write

$$\begin{aligned} \langle\Psi| S |\Psi\rangle &= \langle\psi| \langle\phi| \bigotimes_{i\in P} A_i \bigotimes_{j\in Q} A_j |\psi\rangle |\phi\rangle \\ &= \langle\psi| \bigotimes_{i\in P} A_i |\psi\rangle \langle\phi| \bigotimes_{j\in Q} A_j |\phi\rangle , \end{aligned} \tag{A.1}$$

where we used sets P and Q to denote the set of qubits that are entangled and the set of qubits that are in a classically accessible state, respectively. Clearly, the second tensor product in the

last line of Eq. (A.1) can be classically efficiently computed. Thus, we may evaluate just the first tensor product on a quantum computer and this amounts to a reduction in the number of Pauli strings to measure since there often are multiple second tensor products with the same first tensor product in the full set of Pauli strings to measure.

As to the latter, we start by a simple example to aid the description. Consider a quantum state $|\Psi\rangle$ defined according to

$$|\Psi\rangle = \sum_m |\psi_m\rangle \left(c_{m,0} |00\rangle + c_{m,1} |11\rangle\right). \tag{A.2}$$

Since the separated-out, two-qubit states $|00\rangle$ and $|11\rangle$ do not encode any more than a single qubit of information, this may be compressed to

$$|\Psi_{\text{comp}}\rangle = \sum_m |\psi_m\rangle \left(c_{m,0} |0\rangle + c_{m,1} |1\rangle\right). \tag{A.3}$$

Consider now a Pauli string, living in the full space spanned by $|\Psi\rangle$ in Eq. (A.2), $S = A_0 A_1 \cdots A_{n-1}$ to measure. Without loss of generality we designate $A_0 A_1$ as the Pauli product that lives in the separated-out two-qubit state space in Eq. (A.2). Conjugating S with $|\Psi\rangle$, we obtain

$$\begin{aligned}\langle\Psi| S |\Psi\rangle = \sum_{m,n} \langle\psi_m| \bigotimes_{i>1} A_i |\psi_n\rangle \left(c^*_{m,0} c_{n,0} \langle 00| A_0 A_1 |00\rangle + c^*_{m,0} c_{n,1} \langle 00| A_0 A_1 |11\rangle \right. \\ \left. + c^*_{m,1} c_{n,0} \langle 11| A_0 A_1 |00\rangle + c^*_{m,1} c_{n,1} \langle 11| A_0 A_1 |11\rangle\right).\end{aligned} \tag{A.4}$$

Consider further a compressed Pauli string $S_{\text{comp}} = A_{\text{comp}} A_2 \cdots A_{n-1}$, living in the compressed

space spanned by $|\Psi_{\text{comp}}\rangle$. Conjugating S_{comp} with $|\Psi_{\text{comp}}\rangle$, we obtain

$$\begin{aligned}\langle\Psi| S_{\text{comp}} |\Psi\rangle = \sum_{m,n} \langle\psi_m| \bigotimes_{i>1} A_i |\psi_n\rangle \, (c^*_{m,0}c_{n,0} \langle 0| A_{\text{comp}} |0\rangle + c^*_{m,0}c_{n,1} \langle 0| A_{\text{comp}} |1\rangle \\ + c^*_{m,1}c_{n,0} \langle 1| A_{\text{comp}} |0\rangle + c^*_{m,1}c_{n,1} \langle 1| A_{\text{comp}} |1\rangle).\end{aligned} \tag{A.5}$$

Inspecting Eq. (A.4) and Eq. (A.5), we observe that all that we need to satisfy for the two expressions to agree are

$$\langle 00| A_0 A_1 |00\rangle = \langle 0| A_{\text{comp}} |0\rangle \,, \tag{A.6}$$

$$\langle 00| A_0 A_1 |11\rangle = \langle 0| A_{\text{comp}} |1\rangle \,, \tag{A.7}$$

$$\langle 11| A_0 A_1 |00\rangle = \langle 1| A_{\text{comp}} |0\rangle \,, \tag{A.8}$$

$$\langle 11| A_0 A_1 |11\rangle = \langle 1| A_{\text{comp}} |1\rangle \,. \tag{A.9}$$

For all possible A_0 and A_1, we report in Table A.1 the conversion table for A_{comp}. Numerous null matrices that appear in Table A.1, together with the reduced set of operators to measure that consists of three single-qubit Pauli matrices and an identity matrix, significantly reduces the measurement overhead.

A_0	A_1	A_{comp}
I	I	I
I	σ_x	0
I	σ_y	0
I	σ_z	σ_z
σ_x	I	0
σ_x	σ_x	σ_x
σ_x	σ_y	σ_y
σ_x	σ_z	0
σ_y	I	0
σ_y	σ_x	σ_y
σ_y	σ_y	-σ_x
σ_y	σ_z	0
σ_z	I	σ_z
σ_z	σ_x	0
σ_z	σ_y	0
σ_z	σ_z	I

Table A.1: Conversion table from $A_0 A_1$ to A_{comp} useful for reducing the number of measurements required for VQE simulations. 0 denotes a null matrix.

Chapter B: Number of two-qubit gate distribution using general commuting

In this section we briefly discuss the distribution of extra CNOTs required when using GC with QSR (see Sec. 4.3). Note in GC one creates groups of multiple Pauli strings, drawn from the original set of strings required by a VQE simulation, that can simultaneously be measured. Each group can require a different number of extra CNOTs. It also requires a different number of measurements, typically smaller than the number of Pauli strings inside the group. To illustrate these requirements, in Fig. B.1 we show the distribution of the number of extra CNOTs in the number of required measurements for the GC method for the water molecule example we use in the main text. We observe that, as expected, the number of extra CNOTs and the number of measurements to perform generally increases as a function of a larger ansatz size.

In the main text Fig. 4.3, we opt to use the average number of extra CNOTs as opposed to the maximal number of extra CNOTs for a given ansatz. This is so, since we expect the quantum computational error expected from each application of a two-qubit gate to even out between the circuits that require a smaller than average number of extra CNOTs and the circuits that require a larger than average number of extra CNOTs. Thus we select the average number as a representative number that may best reflect a practical setting.

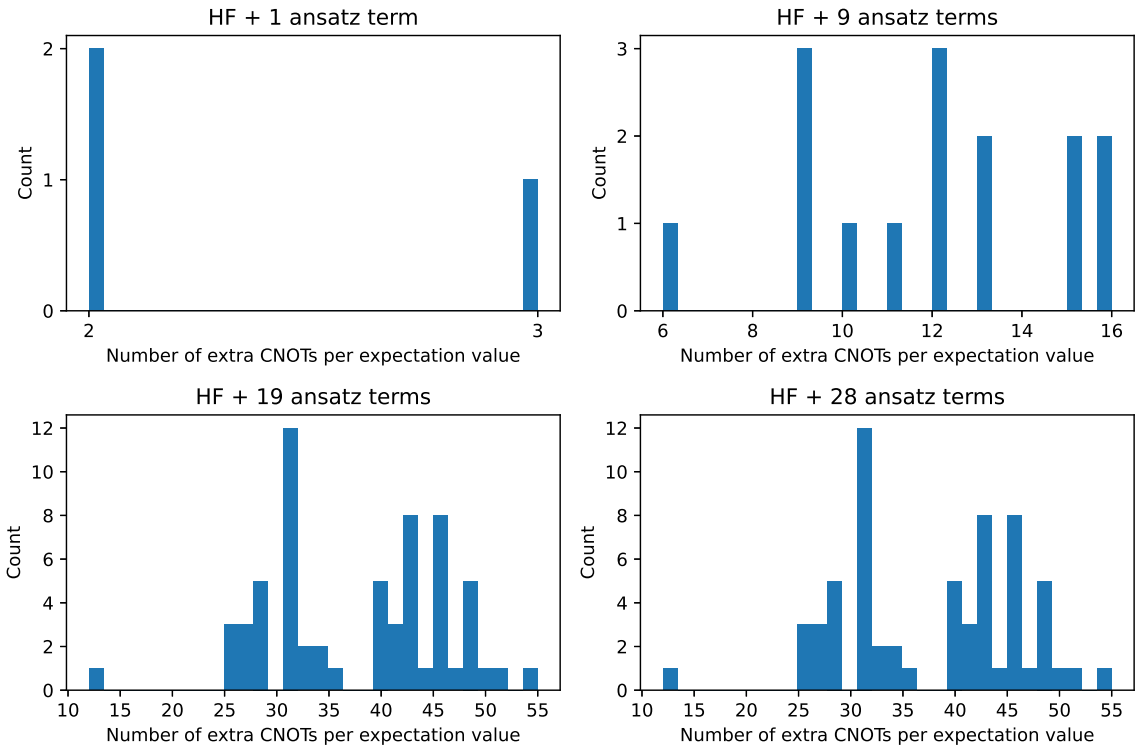


Figure B.1: Histogram of the extra number of CNOTs required for the GC method per measurement value. The Pauli strings to be measured come from the Hamiltonian in HMP2 in (4.12). Shown are the cases for HF+1, HF+9, HF+19 and HF+28 for the water molecule example used in the main text.

Chapter C: Appendix for CAFQA implementation

C.1 VQE assisted by the circuit interface

To better run the VQE, we developed a circuit interface customized for the EURIQA-Blue system trapped-ion machine.

The entire workflow consists of the two parts, the trapped-ion machine and the interface.

Related to this project, the trapped-ion machine by can be considered as an independent system with its own legacy software environment (such as Qiksit Terra version 0.16.1) which can take the OpenQASM2.0 [100] strings as the circuit input and the bright/dark state of the trapped ions as the output.

Despite executing pulse sequences in sequence, there may still be overhead caused by the preparation stage executed on classical computers. Additionally, occasional hardware failures may lead to the generation of garbage outputs.

Based on the above property, we customized a robust API which servers the following features.

1. It can submit OpenQASM2.0 string from the user environment running VQE program (client) to a completely different Nix environment running a trapped-ion machine (host), and retrieve the outcome from the host. Usually the client will send all the circuits that in

principle executed in parallel. This is the most basic and important feature. It can also pass metadata to the host to specify the ions used and the number of shots.

2. It can submit jobs in parallel. This is essential to make the host more efficient without making significant upgrades to it. We found more than 60% of the overall run time can be reduced by submission in parallel.

3. It has a job queuing system that can be paused or restarted manually without disrupting the client. VQE runs can last for days and occasionally we need to pause the job submission to do hardware maintenance or manual calibrations. Circuit jobs are cached locally so the API can be paused or resumed without the communication to the client.

4. It has a job return criteria. Unexpected hardware failures can happen occasionally and result in invalid data. Due to the complexity of the trapped-ion machine, such an error might not be captured actively and remain unnoticed before it is recovered on its own after a short period of time. Although the chance of such events is rare, it can still be encountered since the VQE run can last for days. If the failure is not recognized and dealt with properly before returning the trapped-ion outcome to the client, it can ruin the rest of the VQE iterations. We observed the pattern of several failures from test VQE runs, and concluded such a rare but significant failure can be characterized by the absurdly large deviation of the observed probability histogram from the target probability histogram. As a result, the API will examine the deviation and rerun the jobs through the queuing system mentioned above before returning them to the client.

C.2 Trapped-ion quantum computer

For the CAFQA demonstration, we utilized the EURIQA-Blue trapped-ion quantum computer, which is hosted by the Duke Quantum Center at the Monroe group. It's important to highlight that much of the circuit optimization discussed is generally applicable across different hardware platforms. However, it was assumed that an all-to-all connectivity for two-qubit gates was available. If not, additional two-qubit gates would be necessary when adapting circuits for systems with only local connectivity. The all-to-all connectivity of trapped-ion quantum computers makes them particularly suitable for testing VQE circuits that require a minimal number of two-qubit gates.

The trapped-ion quantum computer we are currently using can accommodate up to 32 addressable ions on a single chip, with this capacity being limited solely by the number of Acousto-Optic Modulator (AOM) channels. In principle, the scalability of trapped-ion QCs can be significantly expanded using Quantum Charge-Coupled Device (QCCD) architecture to connect multiple chips, potentially supporting the number of qubits on the order of 10^2 to 10^3 . [101] Further scaling to host the number of qubits on the order of 10^4 to 10^5 is feasible if photonic links are used to connect these QCCD units. [21]

In a trapped-ion chip, working with a longer ion chain introduces greater engineering challenges. These include difficulties in trapping the ions and reduced motional mode spacing, which can lead to increased difficulty in addressing a particular frequency. Given the constraints of human resources and machine time, it's crucial to strike a balance between the number of qubits utilized and the project's specific requirements.

For the CAFQA project, the primary objective was to demonstrate the effectiveness of the

CAFQA initialization rather than maximizing the number of qubits used. Consequently, circuits with just four qubits were deemed sufficient. This approach differs significantly from previous setups, such as the one used in [39], which involved a 15-ion chain necessary for implementing a Bacon-Shor subsystem code. In our current work, we opted for a shorter seven-ion chain. This configuration offers advantages like stable voltage and greater mode spacing, which collectively lead to fewer calibration procedures and extended run times.

To provide more detail, we use a configuration of seven $^{171}Yb^{+}$ ions arranged in a linear chain, which offers all-to-all connectivity, individually addressed by perpendicular laser through AOM and operates at room temperature. In this setup, the two ions at either end of the chain serve as cap ions. This arrangement is crucial to ensure that the five central ions are evenly spaced. The middle five ions are later functioned as five physical qubits.

The qubits in our trapped-ion setup are defined using the hyperfine levels of the ground state $^2S_{1/2}$ of each ion, which are separated by a frequency of 12.6 GHz. This configuration is specified as follows: [102]

$$|0\rangle \equiv |F = 0, m_F = 0\rangle, \tag{C.1}$$

$$|1\rangle \equiv |F = 1, m_F = 0\rangle. \tag{C.2}$$

Qubit state measurement is achieved via state-dependent fluorescence, utilizing the dipole-allowed $^2S_{1/2} \rightarrow {}^2P_{1/2}$ cycling transition. [103] This method allows for the reliable state readouts by measuring the emitted fluorescence when the ion is excited. Our state preparation and measurement (SPAM) error is 0.27(4)%.

For single-qubit operations, a composite SK1 pulse sequence is employed. [104] This pulse

sequence achieves a gate fidelity of 99.7(2)% (not corrected for state preparation and measurement errors, SPAM) when applied to the five qubits within the chain.

The two-qubit gates are implemented using the Mølmer-Sørensen (MS) protocol. [105] This approach involves the coupling of ions via the Coulomb force while they are aligned linearly, akin to springs oscillating through collective phonon modes that spread across the entire ion chain. A key advantage of the MS gate is that it does not require the cooling of the phonon mode to ground state.

By directing a bichromatic laser at the ions, a state-dependent optical dipole force is created. This force causes a physical displacement of the ions depending on their spin states: a spin-up state results in an upward shift, and a spin-down state in a downward shift, both perpendicular to the ion chain. Consequently, a superimposed qubit state induces a dipole.

When two bichromatic lasers are simultaneously directed at ions at indices i and j, the states $|\uparrow_i\uparrow_j\rangle$ and $|\downarrow_i\downarrow_j\rangle$ maintain the same relative distance, while the states $|\uparrow_i\downarrow_j\rangle$ and $|\downarrow_i\uparrow_j\rangle$ experience a relative shift in ion positions. This interaction leads to an effective native gate described by $XX(\theta) = \exp[-i\theta X_i X_j/2]$ (also see Eq. (3.12)) for ion pair i and j. It is important to highlight that the XX gate, unlike the CNOT gate, features a variable angle, making it particularly useful in certain VQE ansatz. This capability allows for the precise tailoring of interactions, such as the small-angle XX gate application demonstrated for a water molecule, as shown in [4].

Importantly, this method allows full connectivity between any qubit pair because the operation is mediated by the shared phonon mode. However, because this common motional bus also drives ions other than i and j, it is essential to disentangle the motional mode at the end of each operation. Additionally, variations in mode spacing or fluctuations in driving frequency require the design of waveforms that are robust to such changes. We achieve this through an amplitude

modulation (AM) laser pulse shaping scheme, where the waveform is discretized into multiple segments and we do linear interpolation between these segments. [39, 106] Overall, the fidelity of our two-qubit gates varies between 99.3(1)% and 98.9(2)% (not SPAM corrected).

C.3 Noisy simulation

Before running the experiment on hardware, we did preliminary trial runs on noisy simulation for sanity check. The results are similar to what we would have expected shown in [98]. We performed noise simulations based on hardware performance, where the main source of error is the control noise. Since we use composite SK1 pulses that are first-order insensitive to control noise, we assume that the single-qubit gate is perfect and focus on the main errors caused by the two-qubit XX gate. In our experiments, the two dominating errors accomplished with the XX gate are the single-qubit X-flip and Z-flip errors. To simulate the corresponding noise models, we evaluate the X-flip and Z-flip error rates as follows.

An ideal XX gate is parity-preserving such that in-parity $P_{01} + P_{10} = 0$ after applying XX gate on zero state, and non-zero in-parity implies that there is a single-qubit X-flip error. By preparing $N\times$ XX-gates circuit and measuring it, we evaluate the X-flip error rate per qubit per XX-gate as $p_X = (P_{01} + P_{10})/2N$. Our experimental data shows that p_X is very low. To measure the Z-axis error, we need to first rotate the $|00\rangle$ state into x-y plane by applying $R_Y(\pi/2)^{\otimes 2}$ on both qubits, then apply XX$(\pi/4)$-XX$(-\pi/4)$ for N times, vary the phase around the Z-axis by applying $R_Z(\phi)^{\otimes 2}$, and finally rotate it back from the x-y plane by appending $R_Y(\pi/2)^{\otimes 2}$. We can extract the Z-flip error rate per qubit per XX gate using $p_Z = (1 - C)/4N$, where C is the contrast by fitting population P_1 over different phases ϕ.

Bibliography

[1] Qingfeng Wang, Ming Li, Christopher Monroe, and Yunseong Nam. Resource-optimized fermionic local-hamiltonian simulation on a quantum computer for quantum chemistry. *Quantum*, 5:509, 2021.

[2] Qingfeng Wang, Ze-Pei Cian, Ming Li, Igor L Markov, and Yunseong Nam. Ever more optimized simulations of fermionic systems on a quantum computer. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2023.

[3] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, 23(4):550–560, 1997.

[4] Yunseong Nam, Jwo-Sy Chen, Neal C Pisenti, Kenneth Wright, Conor Delaney, Dmitri Maslov, Kenneth R Brown, Stewart Allen, Jason M Amini, Joel Apisdorf, Aleksey Beck, Kristin M. Blinov, Vandiver Chaplin, Mika Chmielewski, Coleman Collins, Shantanu Debnath, Andrew M. Ducore, Kai M. Hudek, Matthew Keesan, Sarah M. Kreikemeier, Jonathan Mizrahi, Phil Solomon, Mike Williams, Jaime David Wong-Campos, Christopher Monroe, and Jungsang Kim. Ground-state energy estimation of the water molecule on a trapped-ion quantum computer. *npj Quantum Information*, 6(1):1–6, 2020.

[5] Yunseong Nam, Neil J. Ross, Yuan Su, Andrew M. Childs, and Dmitri Maslov. Automated optimization of large quantum circuits with continuous parameters. *npj Quantum Information*, 4(1), May 2018.

[6] Robert M. Parrish, Lori A. Burns, Daniel G. A. Smith, Andrew C. Simmonett, A. Eugene DePrince, Edward G. Hohenstein, Uğur Bozkaya, Alexander Yu. Sokolov, Roberto Di Remigio, Ryan M. Richard, Jérôme F. Gonthier, Andrew M. James, Harley R. McAlexander, Ashutosh Kumar, Masaaki Saitow, Xiao Wang, Benjamin P. Pritchard, Prakash Verma, Henry F. Schaefer, Konrad Patkowski, Rollin A. King, Edward F. Valeev, Francesco A. Evangelista, Justin M. Turney, T. Daniel Crawford, and C. David Sherrill. Psi4 1.1: An open-source electronic structure program emphasizing automation, advanced libraries, and interoperability. *Journal of Chemical Theory and Computation*, 13(7):3185–3197, 2017.

[7] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023.

[8] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[9] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1):4213, 2014.

[10] Christopher J Cramer. *Essentials of computational chemistry: theories and models*. John Wiley & Sons, 2013.

[11] Attila Szabo and Neil S Ostlund. *Modern quantum chemistry: introduction to advanced electronic structure theory*. Courier Corporation, 2012.

[12] Trygve Helgaker, Poul Jorgensen, and Jeppe Olsen. *Molecular electronic-structure theory*. John Wiley & Sons, 2013.

[13] Qingfeng Kee Wang and Joel M Bowman. Two-component, ab initio potential energy surface for co2—h2o, extension to the hydrate clathrate, co2@(h2o) 20, and vscf/vci vibrational analyses of both. *The Journal of Chemical Physics*, 147(16), 2017.

[14] John C Slater. Atomic shielding constants. *Physical review*, 36(1):57, 1930.

[15] S Francis Boys. Electronic wave functions-i. a general method of calculation for the stationary states of any molecular system. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 200(1063):542–554, 1950.

[16] Robert F Stewart. Small gaussian expansions of slater-type orbitals. *The Journal of Chemical Physics*, 52(1):431–438, 1970.

[17] RHWJ Ditchfield, Warren J Hehre, and John A Pople. Self-consistent molecular-orbital methods. ix. an extended gaussian-type basis for molecular-orbital studies of organic molecules. *The Journal of Chemical Physics*, 54(2):724–728, 1971.

[18] John C Slater. The theory of complex spectra. *Physical review*, 34(10):1293, 1929.

[19] Richard P Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7), 1982.

[20] David P DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik: Progress of Physics*, 48(9-11):771–783, 2000.

[21] Kenneth R Brown, Jungsang Kim, and Christopher Monroe. Co-designing a scalable quantum computer with trapped atomic ions. *npj Quantum Information*, 2(1):1–10, 2016.

[22] Christopher Monroe and Jungsang Kim. Scaling the ion trap quantum processor. *Science*, 339(6124):1164–1169, 2013.

[23] Rainer Blatt and David Wineland. Entangled states of trapped atomic ions. *Nature*, 453(7198):1008–1015, 2008.

[24] David J Wineland and Dietrich Leibfried. Quantum information processing and metrology with trapped ions. *Laser Physics Letters*, 8(3):175–188, 2011.

[25] Andrew M Steane. The ion trap quantum information processor. *arXiv preprint quant-ph/9608011*, 1996.

[26] Michel H Devoret and John M Martinis. Implementing qubits with superconducting integrated circuits. *Experimental aspects of quantum computing*, pages 163–203, 2005.

[27] Michel H Devoret, Andreas Wallraff, and John M Martinis. Superconducting qubits: A short review. *arXiv preprint cond-mat/0411174*, 2004.

[28] Göran Wendin. Quantum information processing with superconducting circuits: a review. *Reports on Progress in Physics*, 80(10):106001, 2017.

[29] Jay M Gambetta, Jerry M Chow, and Matthias Steffen. Building logical qubits in a superconducting quantum computing system. *npj quantum information*, 3(1):2, 2017.

[30] Michel H Devoret and Robert J Schoelkopf. Superconducting circuits for quantum information: an outlook. *Science*, 339(6124):1169–1174, 2013.

[31] TM Graham, Y Song, J Scott, C Poole, L Phuttitarn, K Jooya, P Eichler, X Jiang, A Marra, B Grinkemeyer, et al. Multi-qubit entanglement and algorithms on a neutral-atom quantum computer. *Nature*, 604(7906):457–462, 2022.

[32] Loïc Henriet, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. Quantum computing with neutral atoms. *Quantum*, 4:327, 2020.

[33] David S Weiss and Mark Saffman. Quantum computing with neutral atoms. *Physics Today*, 70(7):44–50, 2017.

[34] Pieter Kok, William J Munro, Kae Nemoto, Timothy C Ralph, Jonathan P Dowling, and Gerard J Milburn. Linear optical quantum computing with photonic qubits. *Reviews of modern physics*, 79(1):135, 2007.

[35] Hari Krovi. Models of optical quantum computing. *Nanophotonics*, 6(3):531–541, 2017.

[36] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum error correction for beginners. *Reports on Progress in Physics*, 76(7):076001, 2013.

[37] Daniel A Lidar and Todd A Brun. *Quantum error correction*. Cambridge university press, 2013.

[38] Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. *Nature*, 614(7949):676–681, 2023.

[39] Laird Egan, Dripto M Debroy, Crystal Noel, Andrew Risinger, Daiwei Zhu, Debopriyo Biswas, Michael Newman, Muyuan Li, Kenneth R Brown, Marko Cetina, et al. Fault-tolerant control of an error-corrected qubit. *Nature*, 598(7880):281–286, 2021.

[40] Dolev Bluvstein, Simon J Evered, Alexandra A Geim, Sophie H Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, et al. Logical quantum processor based on reconfigurable atom arrays. *Nature*, 626(7997):58–65, 2024.

[41] Nhung H Nguyen, Muyuan Li, Alaina M Green, C Huerta Alderete, Yingyue Zhu, Daiwei Zhu, Kenneth R Brown, and Norbert M Linke. Demonstration of shor encoding on a trapped-ion quantum computer. *Physical Review Applied*, 16(2):024057, 2021.

[42] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[43] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[44] Dmitri Maslov. Basic circuit compilation techniques for an ion-trap quantum machine. *New Journal of Physics*, 19(2):023035, 2017.

[45] P. Jordan and E. Wigner. Über das paulische äquivalenzverbot. *Zeitschrift für Physik*, 47(9):631–651, 1928.

[46] Sergey B. Bravyi and Alexei Yu. Kitaev. Fermionic quantum computation. *Annals of Physics*, 298(1):210 – 226, 2002.

[47] Mark Steudtner and Stephanie Wehner. Fermion-to-qubit mappings with varying resource requirements for quantum simulation. *New Journal of Physics*, 20(6):063010, jun 2018.

[48] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, 2017.

[49] Cornelius Hempel, Christine Maier, Jonathan Romero, Jarrod McClean, Thomas Monz, Heng Shen, Petar Jurcevic, Ben P. Lanyon, Peter Love, Ryan Babbush, Alán Aspuru-Guzik, Rainer Blatt, and Christian F. Roos. Quantum chemistry calculations on a trapped-ion quantum simulator. *Physical Review X*, 8:031022, Jul 2018.

[50] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, and J. M. Martinis. Scalable quantum simulation of molecular energies. *Physical Review X*, 6:031007, Jul 2016.

[51] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski. Cloud quantum computing of an atomic nucleus. *Physical Review Letters*, 120:210501, May 2018.

[52] O. Shehab, K. Landsman, Y. Nam, D. Zhu, N. M. Linke, M. Keesan, R. C. Pooser, and C. Monroe. Toward convergence of effective-field-theory simulations on digital quantum computers. *Physical Review A*, 100:062319, Dec 2019.

[53] D. Maslov, Y. Nam, and J. Kim. An outlook for quantum computing [point of view]. *Proceedings of the IEEE*, 107(1):5–10, Jan 2019.

[54] Emily Grumbling and Mark Horowitz, editors. *Quantum Computing: Progress and Prospects*. The National Academies Press, Washington, DC, 2019.

[55] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.

[56] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018.

[57] James D. Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. Simulation of electronic structure hamiltonians using quantum computers. *Molecular Physics*, 109(5):735–750, 2011.

[58] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[59] Yunseong Nam, Yuan Su, and Dmitri Maslov. Approximate quantum fourier transform with o(n log(n)) t gates. *npj Quantum Information*, 6(1):1–6, Mar 2020.

[60] Mark R. Hoffmann and Jack Simons. A unitary multiconfigurational coupled-cluster method: Theory and applications. *The Journal of Chemical Physics*, 88(2):993–1002, 1988.

[61] Rodney J. Bartlett, Stanislaw A. Kucharski, and Jozef Noga. Alternative coupled-cluster ansätze ii. the unitary coupled-cluster method. *Chemical Physics Letters*, 155(1):133 – 140, 1989.

[62] Dmitry A Fedorov, Bo Peng, Niranjan Govind, and Yuri Alexeev. Vqe method: a short survey and recent developments. *Materials Theory*, 6(1):2, 2022.

[63] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H Booth, et al. The variational quantum eigensolver: a review of methods and best practices. *Physics Reports*, 986:1–128, 2022.

[64] Peter G Szalay, Thomas Muller, Gergely Gidofalvi, Hans Lischka, and Ron Shepard. Multiconfiguration self-consistent field and multireference configuration interaction methods and applications. *Chemical reviews*, 112(1):108–181, 2012.

[65] U. Schollwöck. The density-matrix renormalization group. *Reviews of Modern Physics*, 77(1):259–315, Apr 2005.

[66] Joonho Lee, William J. Huggins, Martin Head-Gordon, and K. Birgitta Whaley. Generalized unitary coupled cluster wave functions for quantum computation. *Journal of Chemical Theory and Computation*, 15(1):311–324, Jan 2019.

[67] Harper R. Grimsley, Sophia E. Economou, Edwin Barnes, and Nicholas J. Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature Communications*, 10(1):3007, Jul 2019.

[68] Masuo Suzuki. General theory of fractal path integrals with applications to many-body theories and statistical physics. *Journal of Mathematical Physics*, 32(2):400–407, 1991.

[69] Jonathan Romero, Ryan Babbush, Jarrod R McClean, Cornelius Hempel, Peter J Love, and Alán Aspuru-Guzik. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. *Quantum Science and Technology*, 4(1):014008, oct 2018.

[70] Chr. Møller and M. S. Plesset. Note on an approximation treatment for many-electron systems. *Phys. Rev.*, 46:618–622, Oct 1934.

[71] Pranav Gokhale, Olivia Angiuli, Yongshan Ding, Kaiwen Gui, Teague Tomesh, Martin Suchara, Margaret Martonosi, and Frederic T Chong. Minimizing state preparations in variational quantum eigensolver by partitioning into commuting families. *arXiv preprint arXiv:1907.13623*, 2019.

[72] Alex Bocharov, Martin Roetteler, and Krysta M. Svore. Efficient synthesis of universal repeat-until-success quantum circuits. *Physical Review Letters*, 114:080502, Feb 2015.

[73] Jacob T. Seeley, Martin J. Richard, and Peter J. Love. The bravyi-kitaev transformation for quantum computation of electronic structure. *The Journal of Chemical Physics*, 137(22):224109, 2012.

[74] David Poulin, M. B. Hastings, Dave Wecker, Nathan Wiebe, Andrew C. Doherty, and Matthias Troyer. The trotter step size required for accurate quantum simulation of quantum chemistry, 2014.

[75] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer. Improving quantum algorithms for quantum chemistry, 2014.

[76] Dave Wecker, Bela Bauer, Bryan K. Clark, Matthew B. Hastings, and Matthias Troyer. Gate-count estimates for performing quantum chemistry on small quantum computers. *Physical Review A*, 90(2), Aug 2014.

[77] Ryan Babbush, Jarrod McClean, Dave Wecker, Alán Aspuru-Guzik, and Nathan Wiebe. Chemical basis of trotter-suzuki errors in quantum chemistry simulation. *Physical Review A*, 91(2), Feb 2015.

[78] Andrew Tranter, Peter J Love, Florian Mintert, Nathan Wiebe, and Peter V Coveney. Ordering of trotterization: Impact on errors in quantum simulation of electronic structure. *Entropy*, 21(12):1218, 2019.

[79] Ketan N. Patel, Igor L. Markov, and John P. Hayes. Optimal synthesis of linear reversible circuits. *Quantum Information and Computation*, 8(3):282–294, March 2008.

[80] James Kennedy and Russell C Eberhart. A discrete binary version of the particle swarm algorithm. In *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, volume 5, pages 4104–4108. IEEE, 1997.

[81] Bochen Tan and Jason Cong. Optimal qubit mapping with simultaneous gate absorption. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–8, 2021.

[82] Kanav Setia, Richard Chen, Julia E. Rice, Antonio Mezzacapo, Marco Pistoia, and James D. Whitfield. Reducing qubit requirements for quantum simulations using molecular point group symmetries. *Journal of Chemical Theory and Computation*, 16(10):6091–6097, 2020. PMID: 32833450.

[83] Yukio Kawashima, Erika Lloyd, Marc P. Coons, Yunseong Nam, Shunji Matsuura, Alejandro J. Garza, Sonika Johri, Lee Huntington, Valentin Senicourt, Andrii O. Maksymov, Jason H. V. Nguyen, Jungsang Kim, Nima Alidoust, Arman Zaribafiyan, and Takeshi Yamazaki. Optimizing electronic structure simulations on a trapped-ion quantum computer using problem decomposition. *Communications Physics*, 4:245, November 2021.

[84] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim. Benchmarking an 11-qubit quantum computer. *Nature Communications*, 10(1), Nov 2019.

[85] Joshua Ramette, Josiah Sinclair, Zachary Vendeiro, Alyssa Rudelis, Marko Cetina, and Vladan Vuletić. Any-to-any connected cavity-mediated architecture for quantum computing with trapped ions or rydberg arrays. *arXiv preprint arXiv:2109.11551*, 2021.

[86] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C Benjamin, and Xiao Yuan. Quantum computational chemistry. *Reviews of Modern Physics*, 92(1):015003, 2020.

[87] Sergey Bravyi, Dmitri Maslov, and Yunseong Nam. Constant-cost implementations of clifford operations and multiply controlled gates using global interactions. *arXiv preprint arXiv:2207.08691*, 2022.

[88] Zhang Jiang, Amir Kalev, Wojciech Mruczkiewicz, and Hartmut Neven. Optimal fermion-to-qubit mapping via ternary trees with applications to reduced quantum states learning. *Quantum*, 4:276, 2020.

[89] Ketan N Patel, Igor L Markov, and John P Hayes. Optimal synthesis of linear reversible circuits. *Quantum Information and Computation*, 8(3):282–294, 2008.

[90] Shen Lin and Brian W Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516, 1973.

[91] Daniel Karapetyan and Gregory Gutin. Lin–kernighan heuristic adaptations for the generalized traveling salesman problem. *European Journal of Operational Research*, 208(3):221–232, 2011.

[92] Keld Helsgaun. Solving the equality generalized traveling salesman problem using the lin–kernighan–helsgaun algorithm. *Mathematical Programming Computation*, 7(3):269–287, 2015.

[93] John Silberholz and Bruce Golden. The generalized traveling salesman problem: A new genetic algorithm approach. In *Extending the horizons: advances in computing, optimization, and decision technologies*, pages 165–181. Springer, 2007.

[94] Yu-An Chen, Anton Kapustin, and Đorđe Radičević. Exact bosonization in two spatial dimensions and a new class of lattice gauge theories. *Annals of Physics*, 393:234–253, 2018.

[95] Naomi H. Nickerson, Joseph F. Fitzsimons, and Simon C. Benjamin. Freely scalable quantum technologies using cells of 5-to-50 qubits with very lossy and noisy photonic links. *Physical Review X*, 4(4), dec 2014.

[96] G. Arnold, M. Wulf, S. Barzanjeh, E. S. Redchenko, A. Rueda, W. J. Hease, F. Hassani, and J. M. Fink. Converting microwave and telecom photons with a silicon photonic nanomechanical interface. *Nature Communications*, 11(1), sep 2020.

[97] N Leung, Y Lu, S Chakram, RK Naik, N Earnest, R Ma, K Jacobs, AN Cleland, and DI Schuster. Deterministic bidirectional communication and remote entanglement generation between superconducting qubits. *npj Quantum Information*, 5(1):1–5, 2019.

[98] Gokul Subramanian Ravi, Pranav Gokhale, Yi Ding, William Kirby, Kaitlin Smith, Jonathan M Baker, Peter J Love, Henry Hoffmann, Kenneth R Brown, and Frederic T Chong. Cafqa: A classical simulation bootstrap for variational quantum algorithms. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, pages 15–29, 2022.

[99] Daniel Gottesman. The heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006*, 1998.

[100] Andrew W Cross, Lev S Bishop, John A Smolin, and Jay M Gambetta. Open quantum assembly language. *arXiv preprint arXiv:1707.03429*, 2017.

[101] David Kielpinski, Chris Monroe, and David J Wineland. Architecture for a large-scale ion-trap quantum computer. *Nature*, 417(6890):709–711, 2002.

[102] Steve Olmschenk, Kelly C Younge, David L Moehring, Dzmitry N Matsukevich, Peter Maunz, and Christopher Monroe. Manipulation and detection of a trapped yb+ hyperfine qubit. *Physical Review A*, 76(5):052314, 2007.

[103] Rachel Noek, Geert Vrijsen, Daniel Gaultney, Emily Mount, Taehyun Kim, Peter Maunz, and Jungsang Kim. High speed, high fidelity detection of an atomic hyperfine qubit. *Optics letters*, 38(22):4735–4738, 2013.

[104] Kenneth R. Brown, Aram W. Harrow, and Isaac L. Chuang. Arbitrarily accurate composite pulse sequences. *Phys. Rev. A*, 70:052318, Nov 2004.

[105] Anders Sørensen and Klaus Mølmer. Quantum computation with ions in thermal motion. *Physical review letters*, 82(9):1971, 1999.

[106] Shantanu Debnath, Norbert M Linke, Caroline Figgatt, Kevin A Landsman, Kevin Wright, and Christopher Monroe. Demonstration of a small programmable quantum computer with atomic qubits. *Nature*, 536(7614):63–66, 2016.