Optimizing Quantum Simulation of Low-Range Electronic Structure Hamiltonians

by

Louis Wenjun Marquis

S.B. Computer Science and Engineering, Mathematics
Massachusetts Institute of Technology, 2025

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2025

Authored by:	Louis Wenjun Marquis Department of Electrical Engineering and Computer Science May 13, 2025
Certified by:	Aram Harrow Professor of Physics, Thesis Supervisor
Accepted by:	Katrina LaCurts Department of Electrical Engineering and Computer Science Chair, Master of Engineering Thesis Committee

Optimizing Quantum Simulation of Low-Range Electronic Structure Hamiltonians

by

Louis Wenjun Marquis

Submitted to the Department of Electrical Engineering and Computer Science
on May 13, 2025 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

To be written.

Thesis supervisor: Aram Harrow
Title: Professor of Physics

Acknowledgments

To be written. Aram, Jakob, of course. Seth Lloyd maybe. Maybe Raji from 8.06? The Copenhagen team half sponsors this right?

Contents

List of Figures

Chapter 1

Introduction

The field of computational chemistry aims to use computers to simulate and better understand the behavior of complex chemical systems in the real world. Drug discovery, materials science, and nanotechnology are only a few areas that would benefit from the ability to computationally predict behavior of molecular systems. However, chemical systems at the particle level are inherently shaped by quantum effects, which are not easily modeled by a classical (non-quantum) computer. For example, the phenomenon of quantum superposition allows a system to be in a linear combination of an exponential number of basis states at the same time. A classical computer must numerically store this exponential amount of data, a prohibitive task.

Quantum computational chemistry aims to perform this task using quantum computers, which can represent such systems more easily because its hardware, by definition, is inherently quantum. A superposition in a chemical system, for example, can represented by physically putting a quantum computer's hardware into superposition.

The first half of this Chapter 1 will summarize well-known techniques in second quantization and Hamiltonian simulation that are relevant to our central problem in this paper, the electronic structure Hamiltonian. The second half will focus on one particular technique called "double factorization" that is useful on electronic structure Hamiltonians. Chapter 2 proposes a quantum circuit that can reduce the asympototic cost of individual terms in the double factorization for certain types of Hamiltonians and numerically analyzes its cost in comparison to competing methods. Chapter 3 explores the usage of this quantum circuit in low-range and related Hamiltonians.

1.1 Hamiltonian Simulation

In any quantum system, the behavior in which its state changes over time is dictated by its Hamiltonian operator H. In particular, if the system starts in state $|\psi(0)\rangle$, then we can determine its state $|\psi(t)\rangle$ at time t.

$$|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle \tag{1.1}$$

A wavefunction maps a continuous physical quantity (such as position) to an amplitude. A state denotes a physical system's wavefunction at a particular moment. An operator linearly

maps wavefunctions to wavefunctions like how a matrix maps vectors to vectors.

The Hamiltonian H depends on many factors, such as the nature of interactions between particles in the system or external forces such as electric or magnetic fields acting in the system. H, and in turn $U = e^{-iHt}$ can be very complex, and usually it's not feasible to simulate U exactly. Instead, we try to find a different operator $\tilde{U}$ that is easy to simulate and sufficiently close to $\tilde{U}$. Usually we choose to define the error as the spectral norm (largest singular value) of the difference between U and $\tilde{U}$. The goal is for the error to be below some variable threshold ϵ.

$$\epsilon \geq ||U - \tilde{U}|| = ||e^{-iHt} - \tilde{U}|| \tag{1.2}$$

At the same time, we want to minimize the cost of simulating $\tilde{U}$ on a quantum computer. To function well in practice, quantum computers require error correction, which generally only work on particular sets of gates. The Clifford+T set is usually considered the most promising set, meaning that we must decompose $\tilde{U}$ as a series of Clifford group gates and T gates. Because T gates over 100 to 10000 times [1] more costly than Clifford gates, the number of T gates, or the "T-count", is often used as the cost metric. Hence, we wish to construct $\tilde{U}$ within error ϵ of e^{-iHt} while minimizing T-count.

This task is accordingly named Hamiltonian simulation.

One important Hamiltonian is the electronic structure Hamiltonian, which describes the behavior of electrons in a molecule consisting of many nuclei and electrons. In modeling this system, we use the Born-Oppenheimer approximation, which assumes that the positions of the nuclei are essentially fixed. This is justified by the fact that the nuclei are far more massive than the electrons, so we can study the movement of electrons separately from the movement of the nuclei. With this assumption, the electronic structure Hamiltonian contains only terms for the electrons' kinetic energies, for the interactions between electrons and nuclei, and for the interactions among electrons.

$$H = -\sum_p \frac{\hbar^2}{2m_e}\nabla_p^2 - \sum_{p,P} \frac{e^2}{4\pi\epsilon_0}\frac{Z_P}{|\vec{r}_p - \vec{R}_P|} + \frac{1}{2}\sum_{p\neq q} \frac{e^2}{4\pi\epsilon_0}\frac{1}{|\vec{r}_p - \vec{r}_q|} \tag{1.3}$$

$\vec{r}_p$ denotes the position of the pth electron, and $\vec{r}_P$ and Z_P denote the position and charge of the Pth nucleus.

The ultimate goal of electronic structure theory is to find the eigenvalues and eigenstates of e^{-iHt}, or the states that are invariant (up to a phase) under e^{-iHt}. The ability to simulate the behavior of e^{-iHt} on an arbitrary state can directly be used to find the eigenvalues using the quantum phase estimation algorithm. The details of this algorithm are explained in [2]; the relevant fact for this paper is the power unlocked by performing e^{-iHt}.

1.2 Second Quantized Fock States

We would like to model the electronic structure system described by H in (1.3). As previously described, the system consists of numerous identical electrons and numerous nuclei at fixed positions. The indistinguishability of the electrons means that the joint wavefunction is only determined by how many electrons are in each basis state, rather than which electron is

in which state. For example, electrons 1 and 2 being in respective states ϕ_1 and ϕ_2 must create the same wavefunction as them being in the states ϕ_2 and ϕ_1. In addition, the joint wavefunction of many identical fermions must be anti-symmetric upon exchange of any pair of fermions. For example, if we want to express a state where one electron is in state ϕ_1 and the other is in state ϕ_2, a simple tensor product $\phi_1 \otimes \phi_2$ would not be a valid joint wavefunction. Instead, it must be anti-symmetrized.

$$\psi = \phi_1 \otimes \phi_2 - \phi_2 \otimes \phi_1 \tag{1.4}$$

This required process can make many-body calculations very cumbersome. Fortunately, second quantization significantly simplifies the representation of many-body systems by introducing the Fock basis, whose states are inherently anti-symmetrized.

Suppose for a single electron we'd like to use a basis $\{\phi_i\}$. Although the basis has infinite size, usually only basis states below a certain energy threshold are relevant, restricting our concern to a finite basis $\{\phi_i\}_{i\in[0,n)}$.

If there are many identical particles, we often want to express a state where f_i particles are in the ϕ_i state. We can label such a state with a vector $\vec{f}$. This is a Fock state $|\vec{f}\rangle$. For fermions (such as electrons), $f_i \in \{0, 1\}$ due to the Pauli-Exclusion Principle, so $\vec{f} \in \{0, 1\}^n$.

$$|\vec{f}\rangle = |f_0, f_1, ..., f_{n-1}\rangle \tag{1.5}$$

With n orbitals (basis states), there are 2^n such Fock states, one for each n-bitstring $\vec{f} \in \{0, 1\}^n$. These 2^n states form a complete basis of the many-body Hilbert space. That is, any state possible for a many-body system within these n orbitals can be expressed as a linear combination of these Fock states.

We can define fermionic creation $a_i^\dagger$ and annihilation a_i operators that map Fock states to Fock states.

$$\begin{aligned} a_i^\dagger |\vec{f}\rangle &= a_i^\dagger |f_0, ..., f_{n-1}\rangle \\ &= \delta_{f_i,1}(-1)^{\sum_{j=0}^{i-1}} |f_0, ..., f_i \oplus 1, ..., f_{n-1}\rangle \end{aligned} \tag{1.6}$$

$$\begin{aligned} a_i |\vec{f}\rangle &= a_i |f_0, ..., f_{n-1}\rangle \\ &= \delta_{f_i,0}(-1)^{\sum_{j=0}^{i-1}} |f_0, ..., f_i \oplus 1, ..., f_{n-1}\rangle \end{aligned} \tag{1.7}$$

$a_i^\dagger$ transforms a Fock state with $f_i = 0$ into a Fock state with $f_i = 1$, thereby "creating" a fermion in state ϕ_i and adding a phase shift to account for anti-symmetry. If orbital i is already occupied, the new state is 0 (it disappears). a_i does the reverse of $a_i^\dagger$ and de-occupies orbital i.

As typical with creation and annihilation operators, their product is a number operator $n_i = a_i^\dagger a_i$ [3].

$$n_i |\vec{f}\rangle = f_i |f_0, ..., f_{n-1}\rangle \tag{1.8}$$

The creation and annihilation operators have anti-commutation relations [3] that are analogous to typical commutation relations of creation and annihilation operators.

$$\{a_i^\dagger, a_j\} = \delta_{i,j} \tag{1.9}$$

$$\{a_i, a_j\} = \{a_i^\dagger, a_j^\dagger\} = 0 \tag{1.10}$$

Second quantization, in short, equips us with a concise representation of many-fermion wavefunction in the form of Fock states and operators $a_i^\dagger, a_i$ to easily manipulate them.

1.3 Jordan-Wigner Transformation

The Fock basis allows us to easily represent physical systems of many electrons. We would like to encode these physical systems on a quantum computer made of qubits so that we can compute on them. Conveniently, it is very straightforward to do so with the Fock basis.

Quantum computers are comprised of qubits. A qubit is simply a system with a two-dimensional Hilbert space. One example is an electron spin (up or down). Given a basis $\{|0\rangle, |1\rangle\}$, the state of a qubit can be any normalized linear combination of these two basis vectors.

$$|\psi\rangle = A_0 |0\rangle + A_1 |1\rangle \text{ such that } |A_0|^2 + |A_1|^2 = 1 \tag{1.11}$$

The joint state of n qubits can be any normalized superposition of all 2^n basis states $\{|\vec{x}\rangle\}_{\vec{x}\in\{0,1\}^n} = \{|x_0, x_1, ..., x_{n-1}\rangle\}_{\vec{x}\in\{0,1\}^n}$.

$$|\psi\rangle = \sum_{\vec{x}\in\{0,1\}^n} A_{\vec{x}} |\vec{x}\rangle \text{ such that } \sum_{\vec{x}\in\{0,1\}^n} |A_{\vec{x}}|^2 = 1 \tag{1.12}$$

It is then straightforward that basis states (1.5) of an n-orbital many-electron physical system correspond to basis states (1.12) of n qubits in a quantum computer. This is the Jordan-Wigner Encoding.

$$|\vec{f}\rangle \leftrightarrow |\vec{x}\rangle \tag{1.13}$$

That is, in the Jordan-Wigner Encoding, the value x_i of the ith qubit represents the occupation f_i of the orbital ϕ_i.

Accordingly, creation (1.6) and annihilation (1.7) operators on the physical system also correspond to qubit operators.

$$a_i \leftrightarrow Z_0 Z_1 ... Z_{i-1} |0\rangle \langle 1|_i \tag{1.14}$$

$$a_i^\dagger \leftrightarrow Z_0 Z_1 ... Z_{i-1} |1\rangle \langle 0|_i \tag{1.15}$$

$$n_i = a_i^\dagger a_i \leftrightarrow |1\rangle \langle 1|_i \tag{1.16}$$

We use the short-hand $U_i = I^{\otimes i} \otimes U \otimes I^{\otimes n-i-1}$ to denote applying a gate U on the ith qubit.

Observe that individual physical and annihilation operators map to a series of i gates (qubit operators). In the average case, this amounts to $O(n)$ gates, which is not ideal. There are other encodings that optimize the efficiency of operator mappings, but they sacrifice the simplicity in basis state mappings exhibited by the Jordan-Wigner Encoding. For the purposes of this project, the Jordan-Wigner Encoding is sufficient.

1.4 Second-Quantized Electronic Structure Hamiltonian

Equipped with Fock states and creation and annihilation operators, we can rewrite the electron structure Hamiltonian from (1.3) in a simpler, second-quantized form.

$$H = \sum_{i,j} h_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_j^\dagger a_k a_l \tag{1.17}$$

The real coefficients h_{ij} and h_{ijkl} are the projections of H onto the chosen single-particle basis $\{\phi_i\}_{i \in [0,n)}$. These can be calculated on a classical computer.

$$\begin{aligned} h_{ij} &= \langle \phi_i | \left(-\frac{\hbar^2}{2m_e} \nabla^2 - \sum_I \frac{e^2}{4\pi\epsilon_0} \frac{Z_I}{|\vec{r} - \vec{R}_I|} \right) |\phi_j\rangle \\ &= \int \phi_i^*(\vec{r}) \left(-\frac{\hbar^2}{2m_e} \nabla^2 - \sum_I \frac{e^2}{4\pi\epsilon_0} \frac{Z_I}{|\vec{r} - \vec{R}_I|} \right) \phi_j(\vec{r}) d^3\vec{r} \end{aligned} \tag{1.18}$$

$$\begin{aligned} h_{ijkl} &= \langle \phi_i, \phi_j | \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{r}_1 - \vec{r}_2|} | \phi_l, \phi_k \rangle \\ &= \iint \phi_i^*(\vec{r}_1) \phi_j^*(\vec{r}_2) \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{r}_1 - \vec{r}_2|} \phi_l(\vec{r}_1) \phi_k(\vec{r}_2) d^3\vec{r}_1 d^3\vec{r}_2 \end{aligned} \tag{1.19}$$

Observe from (1.18) that h_{ij}, which is real, must be symmetric between i and j. From (1.19), h_{ijkl}, which is also real, must also be symmetric between i and l, j and k, and (i,l) and (j,k). Inspired by this symmetry, we can rearrange the operators in the second summation using anti-commutation relations to put i,l and j,k next to each other.

$$\begin{aligned} H &= \sum_{i,j} h_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_j^\dagger a_k a_l \\ &= \sum_{i,j} h_{ij} a_i^\dagger a_j - \frac{1}{2} \sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_j^\dagger a_l a_k \\ &= \sum_{i,j} h_{ij} a_i^\dagger a_j - \frac{1}{2} \sum_{i,j,k,l} h_{ijkl} a_i^\dagger (\delta_{jl} - a_l a_j^\dagger) a_k \\ &= \sum_{i,j} h_{ij} a_i^\dagger a_j - \frac{1}{2} \left(\sum_{i,j,k} h_{ijkj} a_i^\dagger a_k - \sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_l a_j^\dagger a_k \right) \\ &= \sum_{i,j} (h_{ij} - \frac{1}{2} \sum_k h_{ikjk}) a_i^\dagger a_j + \frac{1}{2} \sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_l a_j^\dagger a_k \\ &= \sum_{i,j} h'_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{i,j,k,l} h'_{iljk} a_i^\dagger a_l a_j^\dagger a_k \\ &= H_{1e} + H_{2e} \end{aligned} \tag{1.20}$$

We introduced new coefficients h'_{ij} and h'_{iljk} for this new ordering and label the summations H_{1e} and H_{2e}.

$$\begin{aligned} h'_{ij} &= h_{ij} - \frac{1}{2}\sum_k h_{ikjk} \\ H_{1e} &= \sum_{i,j} h'_{ij} a_i^\dagger a_j \end{aligned} \tag{1.21}$$

$$\begin{aligned} h'_{iljk} &= h_{ijkl} \\ H_{2e} &= \frac{1}{2}\sum_{i,j,k,l} h'_{iljk} a_i^\dagger a_l a_j^\dagger a_k \end{aligned} \tag{1.22}$$

1.5 Trotterization

We would like to simulate $U = e^{-iHt}$ on a quantum computer. Unfortunately, simulating U itself is very difficult: we don't even know what the eigenstates of H are. The main problem is H is sum of many terms, each of which act very differently on a state. We'd like to study each term separately. Fortunately, Trotterization allows us to do so.

Consider a generic Hamiltonian H that is the sum of L terms h_j.

$$H = \sum_{j=0}^{L-1} h_j \tag{1.23}$$

We can use the Lie-Trotter formula to approximate the target operator $U = e^{-iHt}$ as the product of many smaller rotations.

$$U = e^{-iHt} = e^{-i\sum_{j=0}^{L-1} h_j t} = \lim_{M\to\infty} (\prod_{j=0}^{L-1} e^{-ih_j \frac{t}{M}})^M \tag{1.24}$$

In other words, for sufficiently large M, U is essentially equivalent to iterating M times through smaller evolutions $e^{-ih_j \frac{t}{M}}$. We can denote $\Delta t = \frac{t}{M}$ the step size. This is the first-order Trotter-Suzuki Algorithm. One can use the term "Trotterize" to denote the breaking of e^{-iHt} into the product of many $e^{-ih_j \Delta t}$

However, M must be sufficiently large to make the approximation sufficiently precise. In first-order Trotterization, $M = O(\frac{(L\Lambda t)^2}{\epsilon})$ (in which $\lambda = max_j||h_j||$, the largest singular value of a term), scales quadratically with the number of terms L. Each iteration requires L steps, totalling $O(ML = O(\frac{L^3(\Lambda t)^2}{\epsilon}))$ steps.

We can reduce the asymptotic number of iterations necessary with the second-order Trotter-Suzuki Algorithm, which essentially iterates $[h_j]_j$ in ascending order then descending order.

$$U = e^{-iHt} = e^{-i\sum_{j=0}^{L-1} h_j t} = \lim_{M\to\infty} (\prod_{j=0}^{L-1} e^{-ih_j \frac{t}{2M}} \prod_{j=L-1}^{0} e^{-ih_j \frac{t}{2M}})^M \tag{1.25}$$

Second-order Trotterization requires $M = O(\frac{(L\Lambda t)^{1.5}}{\sqrt{\epsilon}})$. In fact, one can continue increasing the order: in general, the $2k$th order Trotterization requires $M = O(\frac{(L\Lambda t)^{1+\frac{1}{2k}}}{\epsilon^{\frac{1}{2k}}})$. However, increasing the order k quickly increases the constant factor, so in practice it's rarely useful to go beyond $k = 3$.

Randomizing the order of the steps $e^{-ih_j\Delta t}$ can significantly improve performance. [4] proposed the qDRIFT algorithm, which samples among terms $[h_j]_j$, assigning h_j a probability proportional to $||h_j||$. It requires at most $\frac{2\lambda^2 t^2}{\epsilon} \leq \frac{2L^2\Lambda^2 t^2}{\epsilon}$ total steps, in which $\lambda = \sum_j ||h_j||$.

The power of such techniques like Trotterization and qDRIFT (breaking e^{-iHt} into many $e^{-ih_j\Delta t}$) allows us to analyze each term h_j independently of the others. If one can simulate $e^{-ih_j\Delta t}$ for arbitrary Δt for every term, one can approximately simulate $U = e^{-iHt}$.

For example, one can Trotterize the electronic structure Hamiltonian (1.20). One can substitute the operator correspondances from (1.14) and (1.15) into (1.20) to obtain the corresponding qubit Hamiltonian, which would be a sum of many tensor products of Paulis. However, with $O(n^4)$ terms in (1.22), there end up being $L = O(n^4)$ tensor products of Paulis in the qubit Hamiltonian. Each such term can contain up to $O(n)$ Paulis, requiring $O(n)$ gates to implement individually, with methods such as [5], [6], [7], [8]. However, most of these gates are actually Clifford gates, which don't contribute to the T-count. Taking the implementation of [8], there are only $O(1)$ non-Clifford gates (namely R_z), which cost $O(\log_2 1\epsilon)$ T gates (where ϵ is the error bound for the step). If we only consider asymptotic scaling in n, then we can consider a such a step to have $O(1)$ cost. This relies on the assumption that the relative cost of T gates eclipses the $O(n)$ quantity of Clifford gates in a step.

If we accordingly assume a constant cost per step, with $L = O(n^4)$, the number of iterations with second-order Trotter-Suzuki becomes $M = O(L^{1.5}) = O(n^6)$, totalling $O(ML) = O(n^{10})$ total steps. In the rest of this paper, we will refer to this direct Trotterization as the "brute force method" We'd like to see if there's a more concise method.

1.6 Double Factorization of Electronic Structure Hamiltonian

Recall that $h'_{iljk} = h_{ijkl}$ is symmetric between (i, l) and (j, k). h' can then be treated as a real symmetric $n^2 \times n^2$ matrix and eigendecomposed with eigenvalues λ_r and eigenvectors $Q^{(r)}$ whose indices are (i, l) ordered pairs. A Cholesky decomposition is also an option but for now we will use an eigendecomposition. Denote $R = \text{rank}(h')$.

$$h'_{iljk} = \sum_{r=0}^{R-1} \lambda_r Q^{(r)}_{i,l} Q^{(r)}_{j,k} \tag{1.26}$$

$$
\begin{aligned}
H_{2e} &= \frac{1}{2}\sum_{i,j,k,l} h'_{ijkl} a_i^\dagger a_l a_j^\dagger a_k \\
&= \frac{1}{2}\sum_{i,j,k,l}\sum_{r=0}^{R-1} \lambda_r Q_{i,l}^{(r)} Q_{j,k}^{(r)} a_i^\dagger a_l a_j^\dagger a_k \\
&= \frac{1}{2}\sum_{r=0}^{R-1} \lambda_r \sum_{i,l} Q_{i,l}^{(r)} a_i^\dagger a_l \sum_{j,k} Q_{j,k}^{(r)} a_j^\dagger a_k \\
&= \frac{1}{2}\sum_{r=0}^{R-1} \lambda_r \left(\sum_{i,j} Q_{i,j}^{(r)} a_i^\dagger a_j\right)^2 \\
&= \sum_{r=0}^{R-1} H_{2e}^{(r)}
\end{aligned}
\tag{1.27}
$$

We have written H_{2e} as a sum of terms $H_{2e}^{(r)}$, defined in (1.28). The number R of terms is the rank of h', and in the worst case $R = n^2$. However, for molecular systems, h' is often low-rank, with R often being $O(n)$, such as in the carbon-hydrogen systems studied in [9]. If we could simulate $U_2^{(r)} = e^{-iH_{2e}^{(r)}\Delta t}$ for an arbitrary Δt, then we could Trotter or qDRIFT over these R terms to implement U_2, far fewer than the $O(n^4)$ terms in direct Trotterization of H_{2e}.

$$
H_{2e}^{(r)} = \frac{1}{2}\lambda_r \left(\sum_{i,j} Q_{i,j}^{(r)} a_i^\dagger a_j\right)^2 \tag{1.28}
$$

Observe that h'_{iljk} is symmetric between i and l and between j and k, implying that $Q_{i,j}^{(r)}$ is symmetric between i and j. Each n^2-length vector $Q^{(r)}$ can then be treated as a symmetric $n \times n$ matrix to be further eigendecomposed with eigenvalues $\lambda_s'^{(r)}$ and real orthonormal eigenvectors $u_s^{(r)}$ as the columns of $u^{(r)}$, which is orthogonal.

$$
\begin{aligned}
Q^{(r)} &= u^{(r)} \begin{bmatrix} \lambda_0'^{(r)} & & \\ & \ddots & \\ & & \lambda_{n-1}'^{(r)} \end{bmatrix} u^{(r)T} \\
Q_{i,j}^{(r)} &= \sum_s \lambda_s'^{(r)} u_{is}^{(r)} u_{js}^{(r)}
\end{aligned}
\tag{1.29}
$$

We introduce the operators $\tilde{a}_s^{(r)\dagger}, \tilde{a}_s^{(r)}$, which are respectively linear combinations of the the creation operators $\{a_i^\dagger\}_{i\in[0,n)}$ and annihilation operators $\{a_i\}_{i\in[0,n)}$. Because $u^{(r)}$ is orthogonal (and therefore unitary), these $\tilde{a}_s^{(r)\dagger}, \tilde{a}_s^{(r)}$ are actually creation and annihilation operators of a different Fock basis; that is, the Fock basis created from the single-particle basis $\{\tilde{\phi}_s = \sum_s u_{is}^{(r)} \phi_i\}_{s\in[0,n)}$. This rotated basis also has its own number operators $\tilde{n}_s^{(r)}$.

$$\begin{aligned}\tilde{a}_s^{(r)\dagger} &= \sum_i u_{is}^{(r)} a_i^\dagger \\ \tilde{a}_s^{(r)} &= \sum_i u_{is}^{(r)} a_i \\ \tilde{n}_s^{(r)} &= \tilde{a}_s^{(r)\dagger} \tilde{a}_s^{(r)}\end{aligned} \tag{1.30}$$

We can rewrite $H_{2e}^{(r)}$ in terms of $\tilde{a}_s^{(r)\dagger}, \tilde{a}_s^{(r)}, \tilde{n}_s^{(r)}$.

$$\begin{aligned}H_{2e}^{(r)} &= \frac{1}{2}\lambda_r \left(\sum_{i,j} Q_{i,j}^{(r)} a_i^\dagger a_j \right)^2 \\ &= \frac{1}{2}\lambda_r \left(\sum_{i,j} \sum_s \lambda_s'^{(r)} u_{is}^{(r)} u_{js}^{(r)} a_i^\dagger a_j \right)^2 \\ &= \frac{1}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} \sum_i u_{is}^{(r)} a_i^\dagger \sum_j u_{js}^{(r)} a_j \right)^2 \\ &= \frac{1}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} \tilde{a}_s^{(r)\dagger} \tilde{a}_s^{(r)} \right)^2 \\ &= \frac{1}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} \tilde{n}_s^{(r)} \right)^2\end{aligned} \tag{1.31}$$

We would like to simulate $U_2^{(r)} = e^{-iH_{2e}^{(r)}\Delta t}$ for an arbitrary Δt. If the operator's eigenbasis were the same as the qubits' computational basis, it would be very simple. However, its eigenbasis is the Fock basis constructed from the rotated basis $\{\tilde{\phi}_s\}$. Meanwhile, the qubit computational basis $\{|\vec{x}\rangle\}$ corresponds to the original physical Fock basis $\{|\vec{f}\rangle\}$ constructed from $\{\phi_i\}$. To deal with this, we have to essentially rotate the operator U_r into the computational basis. [10] uses the Thouless Theorem to show that such a rotation operator $U_R(u^{(r)})$ necessary can be calculated from $u^{(r)}$ and and corresponds to a maximum of $\binom{n}{2} = O(n^2)$ gates on the quantum computer.

$$U_2^{(r)} = e^{-\frac{i\Delta t}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} \tilde{n}_s^{(r)}\right)^2} = U_R(u^{(r)})^\dagger e^{-\frac{i\Delta t}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} n_s\right)^2} U_R(u^{(r)}) = U_R(u^{(r)})^\dagger U_A^{(r)} U_R(u^{(r)}) \tag{1.32}$$

$$U_A^{(r)} = e^{-\frac{i\Delta t}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} n_s\right)^2} \tag{1.33}$$

1.6.1 Double-Factorized Expansion Method

We'd like to find a qubit analog to $U_A^{(r)}$. One way is to expand the square inside $U_A^{(r)}$ and factor it into $\binom{n+1}{2}$ individual operators, since the terms all commute. Let's call this the "double-factorized expansion method", or "expansion method" for short.

$$
\begin{aligned}
U_A^{(r)} &= e^{-\frac{i\Delta t}{2}\lambda_r\left(\sum_s \lambda_s'^{(r)} n_s\right)^2} \\
&= e^{-\frac{i\Delta t}{2}\lambda_r \sum_{s,s'} \lambda_s'^{(r)}\lambda_{s'}'^{(r)} n_s n_{s'}} \\
&= \prod_{s,s'} e^{-\frac{i\Delta t}{2}\lambda_r \lambda_s'^{(r)}\lambda_{s'}'^{(r)} n_s n_{s'}} \\
&= \prod_s e^{-\frac{i\Delta t}{2}\lambda_s(\lambda_s'^{(r)})^2 n_s} \prod_{s<s'} e^{-i\Delta t\lambda_r\lambda_s'^{(r)}\lambda_{s'}'^{(r)} n_s n_{s'}}
\end{aligned} \tag{1.34}
$$

Each operator with a single n_s corresponds (up to a phase) to a R_z operator on the sth qubit in a quantum computer. Each operator with $n_s n_{s'}$ corresponds (up to a phase) to a controlled R_z operator on the sth and s'th qubits in a quantum computer.

$$
e^{-\frac{i\Delta t}{2}\lambda_r(\lambda_s'^{(r)})^2 n_s} \leftrightarrow R_z(-\frac{\Delta t}{2}\lambda_r(\lambda_s'^{(r)})^2)_s \tag{1.35}
$$

$$
R_z(\varphi) = \begin{bmatrix} e^{-i\frac{\varphi}{2}} & 0 \\ 0 & e^{i\frac{\varphi}{2}} \end{bmatrix} = e^{-i\frac{\varphi}{2}} \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} = e^{-i\frac{\varphi}{2}}(|0\rangle\langle 0| + e^{i\varphi}|1\rangle\langle 1|) \tag{1.36}
$$

$$
e^{-\frac{i\Delta t}{2}\lambda_r\lambda_s'^{(r)}\lambda_{s'}'^{(r)} n_s n_{s'}} \leftrightarrow CR_z(-\frac{\Delta t}{2}\lambda_r\lambda_s'^{(r)}\lambda_{s'}'^{(r)})_{s,s'} \tag{1.37}
$$

$$
CR_z(\varphi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\varphi} \end{bmatrix} \tag{1.38}
$$

There are therefore n R_z operators and $\binom{n}{2}$ CR_z operators in this decomposition. Each CR_z can be decomposed (up to a phase) to two $CNOT$s and three R_zs, shown in 1.1. This results in $n + 3\binom{n}{2} = \frac{3n(n-1)}{2}$ R_z gates total in $U_A^{(r)}$.

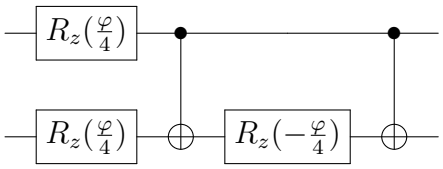

Figure 1.1: Construction of $CR_z(\varphi)$ in (1.38) with two $CNOT$ gates and three R_z gates

[11] achieves a fault-tolerant implementation of R_z using at most $10 + 4\log_2(\frac{1}{\epsilon_z})$ T-gates, in which ϵ_z is the the desired error bound of R_z. Since $U_A^{(r)}$ contains $\frac{3n(n-1)}{2}$ R_z gates and error (which is by operator norm) propagates sub-additively, we need $\epsilon_z \leq \frac{\epsilon}{\frac{3n(n-1)}{2}}$ to guarantee a threshold ϵ for the error of $U_A^{(r)}$.

We can then calculate the upper bound for the T-count $\tilde{N}_T$ of $U_A^{(r)}$ using the expansion method in terms of $n, \frac{1}{\epsilon}$.

$$\begin{aligned}\tilde{N}_T &= \frac{3n(n-1)}{2}(10 + \log_2(\frac{1}{\epsilon_z})) \\ &= \frac{3n(n-1)}{2}(10 + \log_2(\frac{\frac{3n(n-1)}{2}}{\epsilon})) \\ &= O(n^2 \log_2(\frac{n}{\epsilon}))\end{aligned} \tag{1.39}$$

1.7 Discussion

Similarly to (1.34), $U_R(u^{(r)})$ in (1.32) can be decomposed into at most $O(n^2)$ non-Clifford gates as described in [10], resulting in a $O(n^2 \log_2(\frac{n}{\epsilon}))$ T-count. Combined with (1.39), one can achieve an overall T-count of $O(n^2 \log_2(\frac{n}{\epsilon}))$ for $U_2^{(r)}$.

We'd like to find alternative fault-tolerant implementations of $U_2^{(r)}$ with a lower asymptotic T-count. For this to be possible, the chemical system must have a Hamiltonian with "easy" basis rotations; that is, it double factorizes such that all $U_R(u^{(r)})$ decomposes using the [10] method to $o(n^2)$ non-Clifford gates. We also need to both implement $U_A^{(r)}$ more efficiently.

Chapter 2 introduces and explicitly constructs a quantum circuit that simulates $U_R(u^{(r)})$ with $O(n \log_2(\frac{n}{\epsilon}) + \log_2(\frac{1}{\epsilon})^2)$ T-count. We call this quantum circuit the "coherent method", in contrast to the "expansion method". Chapter 3 explores types of Hamiltonians with "easy" basis rotations and how to effectively leverage the "coherent method" for these Hamiltonians.

Chapter 2

Quantum Arithmetic Circuit Design for Double-Factorized Electronic Structure Hamiltonian Simulation

We now propose and analyze a quantum circuit that approximately simulates $U_A^{(r)}$ from (1.34).

$$U_A^{(r)} = e^{-\frac{i\Delta t}{2}\lambda_r\left(\sum_s \lambda_s'^{(r)} n_s\right)^2} \tag{2.1}$$

We can substitute the number operator correspondance from (1.16) to find the equivalent qubit operator.

$$U_A^{(r)} = e^{-\frac{i\Delta t}{2}\lambda_r\left(\sum_s \lambda_s'^{(r)} n_s\right)^2} \leftrightarrow e^{-\frac{i\Delta t}{2}\lambda_r\left(\sum_s \lambda_s'^{(r)} |1\rangle\langle 1|_s\right)^2} \tag{2.2}$$

This qubit operator, when acting on a qubit computational basis state $|\vec{x}\rangle$, it simply applies a phase shift that is a function of $\vec{x}$.

$$e^{-\frac{i\Delta t}{2}\lambda_r\left(\sum_s \lambda_s'^{(r)} |1\rangle\langle 1|_s\right)^2} |\vec{x}\rangle = e^{-\frac{i\Delta t}{2}\lambda_r\left(\sum_s \lambda_s'^{(r)} x_s\right)^2} |\vec{x}\rangle \tag{2.3}$$

We want to find an efficient way to approximately simulate $U_A^{(r)}$. That is, we want to find an operator $\tilde{U}_A^{(r)}$ that applies the specified phase rotation on $|\vec{x}\rangle$ as precisely as possible and with as few gates as possible. As a reminder, the error (and therefore precision) is measured by the operator norm of the difference in the ideal and approximate operators. ϵ is a parameter that limits the maximum error of this approximation and it affects the cost of $\tilde{U}_A^{(r)}$. A tighter bound requires more gates to achieve that bound.

$$\epsilon \geq ||U_A^{(r)} - \tilde{U}_A^{(r)}|| = ||e^{-\frac{i\Delta t}{2}\lambda_r\left(\sum_s \lambda_s'^{(r)} n_s\right)^2} - \tilde{U}_A^{(r)}|| \tag{2.4}$$

(1.34) presented a method that simulated $U_A^{(r)}$ exactly (no error) but it required $O(n^2)$ gates. We now present a $\tilde{U}_A^{(r)}$ that requires $O(n \log_2(\frac{n}{\epsilon}) + \log_2(\frac{1}{\epsilon})^2)$ gates. The idea is to coherently calculate the phase of the phase rotation on a separate qubit register and use

these qubits representing the phase as controls to apply a phase rotation. Hence, we call this the "coherent" method.

More formally, denote the value of the summation in the phase rotation as $y_r(\vec{x})$.

$$\begin{aligned} y_r(\vec{x}) &= \sum_s \lambda_s'^{(r)} x_s \\ U_A^{(r)} |\vec{x}\rangle &= e^{-i\frac{\Delta t}{2}\lambda_r y_r(\vec{x})^2} |\vec{x}\rangle \end{aligned} \tag{2.5}$$

We begin with a register of n qubits in a Fock state $|\vec{x}\rangle$ and introduce an ancilla register of $\frac{m}{2}$ and another of m qubits, both initalized to zeros. We calculate an approximation $\tilde{y}_r(\vec{x})$ onto the smaller ancilla register. We use the value stored in the smaller register to compute an approximation $\tilde{y}_r(\vec{x})^2$ into the larger register. Then we use values of the m ancilla qubits to rotate the phase by $e^{-i\frac{\Delta t}{2}\lambda_r \tilde{y}_r(\vec{x})^2} |\vec{x}\rangle$. Finally, we uncompute the ancilla register and remove them afterwards. These steps comprise the proposed $\tilde{U}_A^{(r)}$. If we wish to construct this circuit fault-tolerantly (i.e. with Clifford gates and T gates), the (2.9) step, which is non-Clifford, will be an approximation.

$$|\vec{x}\rangle \to |\vec{x}\rangle |0^{\frac{m}{2}}\rangle |0^m\rangle \tag{2.6}$$
$$\to |\vec{x}\rangle |\tilde{y}_r(\vec{x})\rangle |0^m\rangle \tag{2.7}$$
$$\to |\vec{x}\rangle |\tilde{y}_r(\vec{x})\rangle |\tilde{y}_r(\vec{x})^2\rangle \tag{2.8}$$
$$\to e^{-i\frac{\Delta t}{2}\lambda_r \tilde{y}_r(\vec{x})^2} |\vec{x}\rangle |\tilde{y}_r(\vec{x})\rangle |\tilde{y}_r(\vec{x})^2\rangle \tag{2.9}$$
$$\to e^{-i\frac{\Delta t}{2}\lambda_r \tilde{y}_r(\vec{x})^2} |\vec{x}\rangle |0^{\frac{m}{2}}\rangle |0^m\rangle \tag{2.10}$$
$$\to e^{-i\frac{\Delta t}{2}\lambda_r \tilde{y}_r(\vec{x})^2} |\vec{x}\rangle \tag{2.11}$$
$$= e^{-i\frac{\Delta t}{2}\lambda_r (\sum_s \tilde{\lambda}_s'^{(r)} |1\rangle\langle 1|_s)^2} |\vec{x}\rangle \tag{2.12}$$
$$\approx \tilde{U}_A^{(r)} |\vec{x}\rangle \tag{2.13}$$

2.1 Summation

Because $\lambda'^{(r)}$ can assume any real (including irrational) values, a finite set of qubits will not be able to represent $y_r(\vec{x}) = \sum_s \lambda_s'^{(r)} x_s$ precisely. Instead, we compute an approximation $\tilde{y}_r(\vec{x})$ that can be expressed in $\frac{m}{2}$ bits. We essentially truncate (or round) each term in the sum and add them together. m can be increased to enhance precision at the cost of ancilla qubits and more gates, and vice versa. In short, we'd like to use the qubits $|\vec{x}\rangle$ to transform $|0^{\frac{m}{2}}\rangle$ to $|\tilde{y}_r(\vec{x})\rangle$.

$$|\vec{x}\rangle |0^{\frac{m}{2}}\rangle \to |\vec{x}\rangle |\tilde{y}_r(\vec{x})\rangle = |\vec{x}\rangle |\sum_s \tilde{\lambda}_s'^{(r)} x_s\rangle \tag{2.14}$$

The encoding between values z and qubit states must be able to handle negative numbers and must be able to hold the all possible values of $\tilde{y}_r(\vec{x})$. We define Y_r as the largest number (by absolute value) that the register must hold, and we assign qubit j to represent a value

$2^{M_r-(\frac{m}{2}-1)+j}$, where M_r is defined below. We use a signed method, so qubit $\frac{m}{2}-1$ represents the sign of the number.

$$Y_r = \max_{\vec{x}\in\{0,1\}^n} |y_r(\vec{x})| = \max\left(\sum_{s:\lambda_s^{(r)}>0} |\lambda_s^{(r)}|, \sum_{s:\lambda_s^{(r)}<0} |\lambda_s^{(r)}|\right) \tag{2.15}$$

$$M_r = \lfloor \log_2 Y_r \rfloor + 2 \tag{2.16}$$

$$|\vec{y}\rangle \leftrightarrow \sum_{j=-\infty}^{\frac{m}{2}-1} (-1)^{\delta_{j,\frac{m}{2}-1}} y_j 2^{M_r-(\frac{m}{2}-1)+j} \tag{2.17}$$

Then this register can hold all values between -2^{M_r} and $2^{M_r}-2^{M_r-(\frac{m}{2}-1)}$ that are multiples of $2^{M_r-(\frac{m}{2}-1)}$. By the definition of M_r, all possible $\tilde{y}_r(\vec{x})$ lie within this range. We can then define the bits $[\lambda_s^{(r)}]_j$ of each $\lambda_s^{(r)}$ in this signed binary form.

$$\lambda_s^{(r)} = \sum_{j=-\infty}^{\frac{m}{2}-1} (-1)^{\delta_{j,\frac{m}{2}-1}} [\lambda_s^{(r)}]_j 2^{M_r-(\frac{m}{2}-1)+j} \tag{2.18}$$

Since we only care about the bits for the values $2^{M_r-(\frac{m}{2}-1)}$ to 2^{M_r}, $\tilde{\lambda}_s^{(r)}$ is the accordingly truncated version of $\lambda_s^{(r)}$.

$$\tilde{\lambda}_s^{(r)} = \sum_{j=0}^{\frac{m}{2}-1} (-1)^{\delta_{j,\frac{m}{2}-1}} [\lambda_s^{(r)}]_j 2^{M_r-(\frac{m}{2}-1)+j} \tag{2.19}$$

As shown in A.1, (2.7) can be achieved by a series of controlled additions. There are many ways to implement such an adder, we chose the most direct way, which simply converts a classical adder into quantum circuits. A.2 shows how to implement a controlled addition of an arbitrary w in the described signed form.

$$w = \sum_{j=0}^{\frac{m}{2}-1} (-1)^{\delta_{j,\frac{m}{2}-1}} [w]_j 2^{M_r-(\frac{m}{2}-1)+j} \tag{2.20}$$

2.2 Squaring

We now have a register of $\frac{m}{2}$ qubits representing the value $\tilde{y}_r(\vec{x})$ and use this to calculate a value of $\tilde{y}_r(\vec{x})^2$ on the register of m qubits.

$$|\vec{x}\rangle\,|\tilde{y}_r(\vec{x})\rangle\,|0^m\rangle \to |\vec{x}\rangle\,|\tilde{y}_r(\vec{x})\rangle\,|\tilde{y}_r(\vec{x})^2\rangle \tag{2.21}$$

We now design a circuit that adds the square of an arbitrary signed $\frac{m}{2}$-qubit $|w\rangle$ to an m-qubit register.

$$|w\rangle\,|z\rangle \to |w\rangle\,|z+w^2\rangle \tag{2.22}$$

Because the smallest value represented by a qubit in the $|w\rangle$ register is $2^{M_r-(\frac{m}{2}-1)}$, the smallest value represented by a qubit in the $|z\rangle$ register will be $2^{2(M_r-(\frac{m}{2}-1))}$. So qubit k in the latter register will have value $(-1)^{\delta_{k,m-1}}2^{2(M_r-(\frac{m}{2}-1))+k}$.

We can expand one of the w to turn w^2 into a sum.

$$w^2 = \sum_{j=0}^{\frac{m}{2}-1} (-1)^{\delta_{j,\frac{m}{2}-1}}[w]_j 2^{M_r-(\frac{m}{2}-1)+j} w \tag{2.23}$$

If one is to add w^2 to z, it's equivalent to adding each of these terms separately. The jth term in (2.23) is only nonzero if $[w]_j = 1$, so it's equivalent to a having a control on the jth qubit in $|w\rangle$. Adding $2^{M_r-(\frac{m}{2}-1)+j}w$ is equivalent to adding w but shifted up j qubits. Note that because $[w]_{\frac{m}{2}-1}$ is a sign bit, it must continue to be added to all bits in $|z\rangle$ until the end, accordingly with classical signed addition. A.3 shows a quantum circuit that adds $[w]_j 2^{M_r-(\frac{m}{2}-1)+j}w$. We can handle the $(-1)^{\delta_{j,\frac{m}{2}-1}}$, which flips the sign for $j = \frac{m}{2}-1$, by performing a controlled subtraction instead of an addition. A controlled subtraction is equivalent to the Hermitian (in this case, a reverse circuit) of the controlled addition.

Performing the circuit described in 2.23 using $w = \tilde{y}_r(\vec{x})$ will successfully accomplish 2.21.

2.3 Phase Rotation

We now have a register of m qubits in which the kth qubit has state $|[\tilde{y}_r(\vec{x})^2]_k\rangle$.

$$\tilde{y}_r(\vec{x})^2 = \sum_k (-1)^{\delta_{k,m-1}}[\tilde{y}_r(\vec{x})^2]_k 2^{2(M_r-(\frac{m}{2}-1))+k} \tag{2.24}$$

As described in (2.9), we'd like to apply a phase of $e^{-i\frac{\Delta t}{2}\lambda_r \tilde{y}_r(\vec{x})^2}$. We can rewrite this phase in the form $e^{i\sum_k [\tilde{y}_r(\vec{x})^2]_k \varphi_k}$.

$$\begin{aligned} -\frac{\Delta t}{2}\lambda_r \tilde{y}_r(\vec{x})^2 &= \frac{\Delta t}{2}\lambda_r \sum_k (-1)^{\delta_{k,m-1}}[\tilde{y}_r(\vec{x})^2]_k 2^{2(M_r-(\frac{m}{2}-1))+k} \\ &= \sum_k [\tilde{y}_r(\vec{x})^2]_k \varphi_k \end{aligned} \tag{2.25}$$

$$\varphi_k = -(-1)^{\delta_{k,m-1}}\frac{\Delta t}{2}\lambda_r 2^{2(M_r-(\frac{m}{2}-1))+k} \tag{2.26}$$

$$\begin{aligned} -\frac{\Delta t}{2}\lambda_r \tilde{y}_r(\vec{x})^2 &= \frac{\Delta t}{2}\lambda_r \sum_k (-1)^{\delta_{k,m-1}}[\tilde{y}_r(\vec{x})^2]_k 2^{2(M_r-(\frac{m}{2}-1))+k} \\ &= \sum_k [\tilde{y}_r(\vec{x})^2]_k \varphi_k \end{aligned} \tag{2.27}$$

We can then decompose the phase operation into several R_z gates. These gates are shown in A.1.

$$
\begin{aligned}
|\vec{x}\rangle\,|\tilde{y}_r(\vec{x})\rangle\,|\tilde{y}_r(\vec{x})^2\rangle &\to e^{i\frac{\Delta t}{2}\lambda_r\tilde{y}_r(\vec{x})^2}\,|\vec{x}\rangle\,|\tilde{y}_r(\vec{x})\rangle\,|\tilde{y}_r(\vec{x})^2\rangle \\
&= |\vec{x}\rangle\,|\tilde{y}_r(\vec{x})\rangle\,e^{i\sum_k[\tilde{y}_r(\vec{x})^2]_k\varphi_k}\,|\tilde{y}_r(\vec{x})^2\rangle \\
&= |\vec{x}\rangle\,|\tilde{y}_r(\vec{x})\rangle\prod_k e^{i[\tilde{y}_r(\vec{x})^2]_k\varphi_k}\,|\tilde{y}_r(\vec{x})^2\rangle \\
&= |\vec{x}\rangle\,|\tilde{y}_r(\vec{x})\rangle\prod_k(|0\rangle\langle 0| + e^{i\varphi_k}\,|1\rangle\langle 1|)_k\,|\tilde{y}_r(\vec{x})^2\rangle \\
&= |\vec{x}\rangle\,|\tilde{y}_r(\vec{x})\rangle\,e^{i\sum_k\varphi_k}\prod_k R_z(\varphi_k)_k\,|\tilde{y}_r(\vec{x})^2\rangle
\end{aligned}
\tag{2.28}
$$

$$
R_z(\varphi) = \begin{bmatrix} e^{-i\frac{\varphi}{2}} & 0 \\ 0 & e^{i\frac{\varphi}{2}} \end{bmatrix} = e^{-i\frac{\varphi}{2}}\begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} = e^{-i\frac{\varphi}{2}}(|0\rangle\langle 0| + e^{i\varphi}\,|1\rangle\langle 1|) \tag{2.29}
$$

We observe that factoring the phase into R_z rotations results in a global phase factor $e^{i\sum_k\varphi_k}$. However, such a phase can be ignored as long as it is independent of $\vec{x}$ (which it is).

Because the R_z gate is not a Clifford gate, it cannot be directly implemented in a fault-tolerant way. Instead, it must be approximated using Clifford and T gates. The number of such gates scales with $O(\log_2(\frac{1}{\epsilon_z}))$, where ϵ_z is the desired precision of the R_z operator. An implementation by [11] achieves a T-count of $10 + 4\log_2(\frac{1}{\epsilon_z})$.

Once the phase has been applied, it is necessary to uncompute the values $\tilde{y}_r(\vec{x}), \tilde{y}_r(\vec{x})^2$ computed in the ancilla registers and reset the registers to zeros. One can simply do this by applying all the previous gates (in the summation and squaring) in reverse order. Then the empty registers can be safely removed and $\tilde{U}_A^{(r)}$ has successfully been applied.

2.4 Error Analysis

The size of the ancilla registers is determined by m and affects the precision of the applied phase. A larger m means more ancilla registers and a more precise phase, but increases the gate cost. If we want to guarantee an error below ϵ accordingly with (2.4), m must be sufficiently large. We now calculate the relation between ϵ and m. This is to say that we upper bound the error $||U_A^{(r)} - \tilde{U}_A^{(r)}||$ in terms of m.

The error originates from both the truncation of $\lambda_s^{(r)}$ into $\tilde{\lambda}_s^{(r)}$ and the inherent error in applying non-Clifford (R_z) gates. The error then propagates into $\tilde{U}_A^{(r)}$. So we begin by bounding the error in $\tilde{\lambda}_s^{(r)}$.

$$
|\lambda_s^{(r)} - \tilde{\lambda}_s^{(r)}| = \sum_{j=-\infty}^{-1}[\lambda_s^{(r)}]_j 2^{M_r-(\frac{m}{2}-1)+j} \le \sum_{j=-\infty}^{-1} 2^{M_r-(\frac{m}{2}-1)+j} \le 2^{M_r-(\frac{m}{2}-1)} \le \frac{4Y_r}{2^{\frac{m}{2}-1}} \tag{2.30}
$$

We propagate this error to $\tilde{y}(\vec{x})$.

$$
|y_r(\vec{x}) - \tilde{y}(\vec{x})| = |\sum_s(\lambda_s^{(r)} - \tilde{\lambda}_s^{(r)})x_s| \le n|\lambda_s^{(r)} - \tilde{\lambda}_s^{(r)}| \le \frac{Y_r}{2^{\frac{m}{2}-3}} \tag{2.31}
$$

We propagate this error to $\tilde{y}(\vec{x})^2$.

$$\begin{aligned} |y_r(\vec{x})^2 - \tilde{y}(\vec{x})^2| &= |2y_r(\vec{x})(y_r(\vec{x}) - \tilde{y}(\vec{x})) - (y_r(\vec{x}) - \tilde{y}(\vec{x}))^2| \\ &\leq \max\left(2|y_r(\vec{x})|n\frac{Y_r}{2^{\frac{m}{2}-3}}, \left(n\frac{Y_r}{2^{\frac{m}{2}-3}}\right)^2\right) \end{aligned} \tag{2.32}$$

Recall that there is some error generated from the m R_z gates in $\tilde{U}_A^{(r)}$. Each gate generates error ϵ_z, which is variable depending on how many Clifford + T gates we use to implement it. Since the operator norm is a norm, the total error is sub-additive.

$$||e^{-\frac{i\Delta t}{2}\lambda_r\left(\sum_s \tilde{\lambda}_s'^{(r)}|1\rangle\langle 1|_s\right)^2} - \tilde{U}_A^{(r)}|| \leq m\epsilon \tag{2.33}$$

We can propagate both errors into the overall error $||U_A^{(r)} - \tilde{U}_A^{(r)}||$.

$$\begin{aligned} ||U_A^{(r)} - \tilde{U}_A^{(r)}|| &\leq ||U_A^{(r)} - e^{-\frac{i\Delta t}{2}\lambda_r\left(\sum_s \tilde{\lambda}_s'^{(r)}|1\rangle\langle 1|_s\right)^2}|| + ||e^{-\frac{i\Delta t}{2}\lambda_r\left(\sum_s \tilde{\lambda}_s'^{(r)}|1\rangle\langle 1|_s\right)^2} - \tilde{U}_A^{(r)}|| \\ &\leq \max_{|\psi\rangle:|||\psi\rangle||=1} ||(U_A^{(r)} - e^{-\frac{i\Delta t}{2}\lambda_r\left(\sum_s \tilde{\lambda}_s'^{(r)}|1\rangle\langle 1|_s\right)^2})\,|\psi\rangle\,|| + m\epsilon_z \\ &= \max_{\vec{x}\in\{0,1\}^n} ||(e^{-i\frac{\Delta t}{2}\lambda_r y_r(\vec{x})^2} - e^{-i\frac{\Delta t}{2}\lambda_r \tilde{y}_r(\vec{x})^2})\,|\vec{x}\rangle\,|| + m\epsilon_z \\ &= \max_{\vec{x}\in\{0,1\}^n} ||(e^{-i\frac{\Delta t}{2}|\lambda_r|(y_r(\vec{x})^2 - \tilde{y}_r(\vec{x})^2)} - 1)\,|\vec{x}\rangle\,|| + m\epsilon_z \\ &\leq \max_{\vec{x}\in\{0,1\}^n} \frac{\Delta t}{2}|\lambda_r||y_r(\vec{x})^2 - \tilde{y}_r(\vec{x})^2| + m\epsilon_z \\ &\leq \frac{\Delta t}{2}|\lambda_r| \max_{\vec{x}\in\{0,1\}^n} \max\left(2|y_r(\vec{x})|n\frac{Y_r}{2^{\frac{m}{2}-3}}, \left(n\frac{Y_r}{2^{\frac{m}{2}-3}}\right)^2\right) + m\epsilon_z \\ &= n\frac{|\lambda_r|Y_r^2\Delta t}{2^{\frac{m}{2}-4}} \max\left(1, \frac{n}{2^{\frac{m}{2}-2}}\right) + m\epsilon_z \end{aligned} \tag{2.34}$$

We want m to be large enough and ϵ_z small enough that this quantity is at most ϵ. We can calculate the thresholds m^* and ϵ_z^* above which this is the case. To make the calculations simpler, we can (arbitrarily) restrict both terms to at most $\frac{\epsilon}{2}$.

$$\begin{aligned} \frac{\epsilon}{2} &\geq n\frac{|\lambda_r|Y_r^2\Delta t}{2^{\frac{m}{2}-4}} \max\left(1, \frac{n}{2^{\frac{m}{2}-2}}\right) \\ \implies m^* &\approx \max(2\log_2\left(n\frac{|\lambda_r|Y_r^2\Delta t}{\epsilon}\right) + 4, \log_2\left(n^2\frac{|\lambda_r|Y_r^2\Delta t}{\epsilon}\right) + 6) \\ &= 2\log_2(n) + 2\log_2\left(\frac{|\lambda_r|Y_r^2\Delta t}{\epsilon}\right) + 4 \end{aligned} \tag{2.35}$$

We choose the first input of the max. Both options are asymptotically the same, so we choose the one with the larger constant factor to establish an upper bound on the threshold.

If it turns out that $\log_2\left(n^2\frac{|\lambda_r|Y_r^2\Delta t}{\epsilon}\right) < 2$, then m^* won't be much larger than (and may be less than) $2\log_2(n)$ anyway.

Clearly m^* is asymptotically logarithmic in n, as long as $|\lambda_r|$ and Y_r are not exponential in n (in reality they are sublinear). Referring back to (2.38), the total asymptotic gate cost of $\tilde{U}_A^{(r)}$ is therefore $O(mn + m^2) = O(n\log n + (\log n)^2) = O(n\log n)$.

m^* is also logarithmic in Δt and in $\frac{1}{\epsilon}$.

Meanwhile, the restriction on $m\epsilon_z$ leads to a threshold ϵ_z^*.

$$\begin{aligned} \frac{\epsilon}{2} &\geq m\epsilon_z \\ \implies \epsilon_z^* &= \frac{\epsilon}{2m} \end{aligned} \tag{2.36}$$

2.5 Complexity Analysis

We would like to establish an upper bound on the cost of $\tilde{U}_A^{(r)}$. In implementing fault-tolerant quantum circuits, the T-count is the usual cost metric, since T-state distillation is usually the bottleneck.

The controlled constant addition in A.2 requires $\frac{m}{2} - 1$ ancilla qubits (in addition to the target register) and its gate count depends on the values of $[w]_j$. At maximum, it requires $1.5m - 5$ Toffolis and $2.5m - 7$ CNOTs.

Next, we wish to find the cost of the squaring circuit in A.3. Observe that for each $|z\rangle_{j+2}, ..., |z\rangle_{m-1}$, there are 8 Toffolis [1]. There are 4 additional Toffolis (2 for the highest bit and 2 at the beginning and end). This is a total of $8(m - j - 2) + 4$ Toffolis in A.3.

Since the squaring protocol performs A.3 for each $j \in [0, \frac{m}{2})$, we can calculate the total number of Toffolis in the squaring protocol.

$$\begin{aligned} \sum_{j=0}^{\frac{m}{2}-1}(8(m-j-2)+4) &= 8m^2 - 8\frac{\frac{m}{2}(\frac{m}{2}-1)}{2} - 8m + 4m \\ &= 7m^2 - 2m \end{aligned} \tag{2.37}$$

In our implementation of $\tilde{U}_A^{(r)}$, there are $2n$ controlled constant additions (n to compute and n to uncompute) and 2 squaring protocols (once to compute and uncompute). The phase rotations only contribute n single-qubit gates, which are negligible. We can calculate the total number of Toffolis in $\tilde{U}_A^{(r)}$.

$$\begin{aligned} \tilde{N}_{CCNOT} &= 2n(1.5m - 5) + 2(7m^2 - 2m) \\ &= 3mn - 10n + 14m^2 - 4m \\ &= O(mn + m^2) \end{aligned} \tag{2.38}$$

[1] When iterating through each of the qubits in $|w\rangle$, inevitably one or more of these qubits will be $|w\rangle_j$. One of the gates in an iteration is a Toffoli controlled by $|w\rangle_j$ and the iterated qubit, which in this case is itself. Then that Toffoli will actually be a CNOT with a lower cost. We ignore this edge case because we only need an upper bound

Given ϵ_z, each R_z gate requires $10 + 4\log_2(\frac{1}{\epsilon_z})$ T-gates using the implementation in [11]. There are m R_z gates in $\tilde{U}_A^{(r)}$. We can use the threshold ϵ_z^* calculated in (2.36) to upper-bound this T-count in terms of m and ϵ.

$$\begin{aligned}\tilde{N}_{T;Rz} &= m(10 + 4\log_2(\frac{1}{\epsilon_z})) \\ &= 10m + 4m\log_2(\frac{2m}{\epsilon})\end{aligned} \tag{2.39}$$

The traditional construction of the Toffoli gate uses 2 Hadamard gates, 6 CNOT gates, and 7 T-gates. However, [12] proposed a construction that uses only 4 T-gates. We can calculate the total T-count of $\tilde{U}_A^{(r)}$ if we use the latter construction.

$$\begin{aligned}\tilde{N}_T &= 4\tilde{N}_{CCNOT} + \tilde{N}_{T;Rz} \\ &= 12mn - 40n + 56m^2 - 16m + 10m + 4m\log_2(\frac{2m}{\epsilon}) \\ &= 12mn - 40n + 56m^2 - 6m + 4m\log_2(\frac{2m}{\epsilon}) \\ &= O(mn + m^2 + m\log_2(\frac{1}{\epsilon}))\end{aligned} \tag{2.40}$$

We can substitute the asymptotic threshold $m^* = O(\log_2(\frac{n}{\epsilon}))$ from (2.35) to obtain an asymptotic $\tilde{N}_T$ in terms of n and $\frac{1}{\epsilon}$.

$$\begin{aligned}\tilde{N}_T &= O(\log_2(\frac{n}{\epsilon})n + (\log_2(\frac{n}{\epsilon}))^2 + \log_2(\frac{n}{\epsilon})\log_2(\frac{1}{\epsilon})) \\ &= O(n\log_2(\frac{n}{\epsilon}) + \log_2(\frac{1}{\epsilon})^2)\end{aligned} \tag{2.41}$$

2.6 Numerical Cost Estimates

The gate cost of this $\tilde{U}_A^{(r)}$ is asympotically lower in n than the $O(n^2\log_2(\frac{n}{\epsilon}))$ cost of the expansion method in (1.39), which is a significant advantage. It's useful to get a (very rough) numerical estimate of the gate counts to understand the scale (n) at which the advantage appears.

The value of Δt depends on the particular Trotter-like algorithm used. An obvious but very loose upper bound for Δt is t itself. [13] establishes $t = 6000\frac{1}{E_h}$ and $\epsilon = 10^{-6}$ as typical values. For estimates of $|\lambda_r|Y_r^2$, we decomposed Hamiltonians of hydrogen chain systems for various n (n is the number of Hs and the number of orbitals) to obtain values of $|\lambda_r|$ and Y_r. It turns out for these systems that $|\lambda_r|Y_r^2 \approx n\log_2 nE_h$.

$$
\begin{aligned}
m_H^* &\approx 2\log_2(n) + 2\log_2\left(\frac{|\lambda_r| Y_r^2 t}{\epsilon}\right) + 4 \\
&\approx 2\log_2(n) + 2\log_2\left(\frac{n\log_2 n \times 6000}{10^{-6}}\right) + 4 \\
&\approx 4\log_2(n) + 2\log_2(\log_2(n)) + 69
\end{aligned} \tag{2.42}
$$

In practice, $\Delta << t$, so t is not a useful order of magnitude for Δt. [13] also mentions a value $\Delta t = 0.01 \frac{1}{E_h}$ which is probably closer to the actual of Δt.

$$
m_H^* \approx 2\log_2(n) + 2\log_2\left(\frac{n\log_2 n \times 0.01}{10^{-6}}\right) + 4 \approx 4\log_2(n) + 2\log_2(\log_2(n)) + 30.57 \tag{2.43}
$$

$$
\tilde{N}_T = 12m(n)n - 40n + 56m(n)^2 - 6m(n) + 4m(n)\log_2(\frac{2m(n)}{\epsilon}) \tag{2.44}
$$

We plot this T-count of the "coherent method" as well as that of the "expansion method" in (1.39) as functions of n. It turns out that the expansion method surpasses the coherent method in T-count at approxmiately $n \approx 64$. At $n = 64$, the coherent method requires about $2.51 * 10^5$ T-gates and has $m = 60$, while the expansion method requires about $2.57 * 10^5$ T-gates.

We reiterate that this is a very rough estimate, intended to show the order of magnitude of the threshold n, rather than a specific practical value. The number of ancillas and the gate count of the coherent method are highly dependent on the scale of values λ_r, Y_r which are highly dependent on the specific Hamiltonian. The values of ϵ and Δt are also vary significantly across use cases.

Additionally, the circuit design describes an uncontrolled implementation of $e^{-i\hat{H}_2^{(r)}\delta t}$, while in practice the main use case of an operation e^{-iHt} is in phase estimation to obtain the ground state of H. Phase estimation will require a controlled version of e^{-iHt}, and therefore a controlled version of $e^{-i\hat{H}_2^{(r)}\delta t}$. This will turn the Toffoli gates into $CCCNOT$ gates, $CNOT$ gates into Toffolis, R_z gates into CR_z gates, and CR_z gates into CCR_z gates, which will increase the gate count of both the expansion and coherent methods and will shift the threshold n. However, such a transformation will only affect the the gate counts by a constant factor (e.g. a $CCCNOT$ can be implemented with 3 Toffolis) and will not affect the asymptotic performance of the coherent and expansion methods.

INSERT PLOT

2.7 Discussion

The asymptotic (in n) advantage of the coherent method ($O(n\log_2 n)$) over the expansion method ($O(n^2)$) is only useful in $U_A^{(r)}$ if the number of T gates in $U_R(u^{(r)})$ can also be reduced

from $O(n^2 \log_2(\frac{n}{\epsilon}))$. Otherwise, the latter complexity dominates and maintains the overall complexity unchanged.

In Chapter 3, we explore chemical systems and Hamiltonians for which it is the case that $U_R(u^{(r)})$ can be decomposed "easily" as such.

Chapter 3

Low Range Hamiltonians

3.1 Sparse Hamiltonians

We would like to identify Hamiltonians for which the double factorization has "easy" rotations; that is, each $U_R(u^{(r)})$ decomposes to a small number of basis rotations.

Consider the minimum case: for all $r \in [0, n^2)$, $U_R(u^{(r)}) = I$, which occurs when $u^{(r)} = I$. That is, $Q^{(r)}$ is already diagonal with $\lambda_s'^{(r)} = Q^{(r)}_{s,s}$ and zero basis rotations are necessary. In this case, the Hamiltonian H_{2e} itself can actually be written as a sum of products of number operators.

$$\begin{aligned} H_{2e} &= \frac{1}{2}\sum_r \lambda_r \left(\sum_{i,j} Q^{(r)}_{i,j} a_i^\dagger a_j\right)^2 \\ &= \frac{1}{2}\sum_r \lambda_r (\sum_s \lambda_s'^{(r)} n_s)^2 \\ &= \frac{1}{2}\sum_{s,s'} n_s n_{s'} \sum_r \lambda_r \lambda_s'^{(r)} \lambda_{s'}'^{(r)} \\ &= \frac{1}{2}\sum_s n_s \sum_r \lambda_r (\lambda_s'^{(r)})^2 + \sum_{s<s'} n_s n_{s'} \sum_r \lambda_r \lambda_s'^{(r)} \lambda_{s'}'^{(r)} \end{aligned} \tag{3.1}$$

With H_{2e} in the form (3.1) analogous to (1.34) in the expansion method, $U_2 = e^{-iH_{2e}t}$ corresponds to a product of n R_z gates and $\binom{n}{2}$ CR_z gates, which, just like in the expansion method, costs $O(n^2 \log_2(\frac{n}{\epsilon}))$ T gates. Then Trotterization over H_{2e} is not necessary because there is only one term. In such a case, it's unlikely that double factorization will be useful at all, since H_{2e} is already so simple.

More generally, if H_{2e} is very sparse, then a direct brute force Trotterization will not involve as many terms as the worst case $O(n^4)$, and its gate count may be low enough to make double factorization (and thereby the coherent method) unnecessary. At the same time, the coherent method requires the number of rotations composing $u^{(r)}$ to be small, and having very few rotations tends to make $Q^{(r)}$ (and thereby H_{2e}) very sparse. Therefore, this opens the problem of finding Hamiltonians H_{2e} that are sufficiently dense but for which the $Q^{(r)}$ are sufficiently easily decomposable.

3.2 K-Range Hamiltonian

Define an electronic structure Hamiltonian from (1.20) as K-range if its values h_{ijkl} satisfy the following condition.

$$\begin{aligned} H &= \sum_{i,j} h_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_j^\dagger a_k a_l \\ \text{such that } h_{ijkl} &= 0 \text{ if } \min(\max(|i-l|, |j-k|), \max(|i-k|, |j-l|)) > K \end{aligned} \tag{3.2}$$

Any K-range Hamiltonian can be rewritten in the following more restrictive form. This is because, for an index (i, j, k, l) such that $\max(|i-k|, |j-l|) > K$ but $\max(|i-l|, |j-k|) \leq K$, we can substitute $h_{ijkl} a_i^\dagger a_j^\dagger a_l a_k = -h_{ijkl} a_i^\dagger a_j^\dagger a_k a_l$. This term satisfies the more restrictive condition and can be combined with the original term indexed (i, j, l, k). From this process of substitutions and combination we get new terms $\bar{h}_{ijkl}$.

$$\begin{aligned} H &= \sum_{i,j} h_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{i,j,k,l} \bar{h}_{ijkl} a_i^\dagger a_j^\dagger a_k a_l \\ \text{such that } \bar{h}_{ijkl} &= 0 \text{ if } \max(|i-l|, |j-k|) > K \end{aligned} \tag{3.3}$$

Then, accordingly with (1.20), this Hamiltonian can be rearranged into H_{1e} and H_{2e}, and accordingly with (1.27), H_{2e} can be double factorized. The condition does not change.

$$\begin{aligned} H_{2e} &= \frac{1}{2} \sum_{i,j,k,l} h'_{iljk} a_i^\dagger a_l a_j^\dagger a_k \\ &= \frac{1}{2} \sum_r \lambda_r \left(\sum_{i,j} Q_{i,j}^{(r)} a_i^\dagger a_j \right)^2 \\ &= \sum_{r=0}^{R-1} H_{2e}^{(r)} \\ \text{such that } h'_{iljk} &= \bar{h}_{ijkl} = 0 \text{ if } \max(|i-l|, |j-k|) > K \end{aligned} \tag{3.4}$$

Observe that fewer than $K^2 n^2$ indices $iljk$ satisfy the range condition, so H_{2e} can be expressed as the sum of $O(K^2 n^2)$ Pauli tensor products in context of the brute force method. As one might expect, decreasing K makes H_{2e} sparser.

Observe also that the range condition implies, for all r, that $Q_{i,j}^{(r)} = 0$ if $|i-j| > K$. This means that all $Q^{(r)}$ are band matrices with bandwidth K.

$$Q^{(r)} = \begin{bmatrix} Q^{(r)}_{0,0} & \cdots & Q^{(r)}_{0,K} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \vdots & Q^{(r)}_{1,K+1} & \ddots & & & & \vdots \\ Q^{(r)}_{K,0} & \cdots & \ddots & \ddots & \ddots & \ddots & & & 0 \\ 0 & Q^{(r)}_{K+1,1} & \ddots & \ddots & \ddots & \ddots & \ddots & & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & Q^{(r)}_{n-K-2,n-2} & 0 \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & \cdots & Q^{(r)}_{n-K-1,n-1} \\ \vdots & & & & \ddots & Q^{(r)}_{n-2,n-K-2} & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & Q^{(r)}_{n-1,n-K-1} & \cdots & Q^{(r)}_{n-1,n-1} \end{bmatrix} \tag{3.5}$$

If $Q^{(r)}$ has bandwidth K, this guarantees $u^{(r)}$ can be decomposed into fewer than Kn Givens rotations, corresponding to at most $O(Kn)$ two-qubit basis rotations comprising $U(u^{(r)})$ [1]

Therefore, a K-range Hamiltonian can be expressed as a sum of R terms $H^{(r)}_{2e}$, each simulable with either the $O(n^2)$ expansion method or the $O(n(\log n + K))$ T-gates (we ignore ϵ factors for simplicity).

3.1 lists bounds on second-order Trotterization T-counts using the brute force, double-factorized expansion, and double-factorized coherent methods. It considers their performance on generic H_{2e} as well as ones that are K-range, linear-rank, or both. We chose to list second-order Trotterization costs rather than first-order to show that the double-factorized coherent method holds advantages over brute-force methods even in higher-order Trotterization, reduces the asymptotic impact of L, which is more present in the latter.

Observe that the coherent method has the lowest asymptotic upper bound among all three methods for low-rank K-range Hamiltonians, as long as $K = o(n)$ (otherwise it's the same as the expansion method). This remains the case ever for higher-order Trotterization. Of course, this is assuming that K is not so low (i.e. $K = 0$) that H_{2e} simplifies into a sum of commuting terms (which comprise one Trotter term) in the way of (3.1). For generic-rank K-range H_{2e}, it performs asymptotically better than brute force if $K = \omega(n^{0.25})$ and better than expansion if $K = o(n)$. This also remains the case for arbitrary kth-order Trotterization but with a generic lower bound $K = \omega(n^{\frac{1}{3+\frac{1}{k}}})$. For example, for fourth-order Trotterization ($k = 2$), we would need $K = \omega(n^{\frac{1}{3.5}})$.

[1] To see this, observe that (as quoted from [10] and appropriated for our purposes), when a Givens rotation matrix $r_{pq}(\theta)$ left multiplies a matrix, it effects a rotation between rows p and q which can be used to zero out a single element in one of the rows. In [10] there were $\binom{n}{2}$ elements below the diagonal, in $Q^{(r)}$ there are $\binom{n}{2} - \binom{n-K}{2} = \frac{2Kn-K^2-2n+K}{2} < Kn$. So $u^{(r)}$ requires less than Kn Givens rotations (on each side) to make $Q^{(r)}$ diagonal.

	L	$\tilde{N}_T$	$M = O(L^{1.5})$	Total $= ML\tilde{N}_T$
Generic H_{2e}				
Brute Force	$O(n^4)$	$O(1)$	$O(n^6)$	$O(n^{10})$
DF Expansion Method	$R = O(n^2)$	$O(n^2)$	$O(n^3)$	$O(n^7)$
DF Coherent Method	$R = O(n^2)$	$O(n^2)$	$O(n^3)$	$O(n^7)$
K-Range				
Brute Force	$O(K^2n^2)$	$O(1)$	$O(K^3n^3)$	$O(K^5n^5)$
DF Expansion Method	$R = O(n^2)$	$O(n^2)$	$O(n^3)$	$O(n^7)$
DF Coherent Method	$R = O(n^2)$	$O(Kn)$	$O(n^3)$	$O(Kn^6)$
Low Rank ($R = O(n)$)				
Brute Force	$O(n^4)$	$O(1)$	$O(n^6)$	$O(n^{10})$
DF Expansion Method	$R = O(n)$	$O(n^2)$	$O(n^{1.5})$	$O(n^{5.5})$
DF Coherent Method	$R = O(n)$	$O(n^2)$	$O(n^{1.5})$	$O(n^{5.5})$
Low Rank and K-Range				
Brute Force	$O(K^2n^2)$	$O(1)$	$O(K^3n^3)$	$O(K^5n^5)$
DF Expansion Method	$R = O(n)$	$O(n^2)$	$O(n^{1.5})$	$O(n^{4.5})$
DF Coherent Method	$R = O(n)$	$O(Kn)$	$O(n^{1.5})$	$O(Kn^{3.5})$

Table 3.1: Asymptotic T-count of the brute force, double-factorized expansion method, and double-factorized coherent method on various types of Hamiltonians. We assume a second-order Trotterization. We ignore $\log_2 n$ terms and $\log_2 \frac{1}{\epsilon}$ terms for simplicity.

3.3 Extraction of Low-Range Terms from Practical Electronic Structure Hamiltonians

We have shown some asymptotic advantages for the coherent method within a deterministic Trotterization assumption. In practice, the large empirical discrepancy between the absolute values of the largest and smallest terms in typical electronic structure Hamiltonians makes qDRIFT much more suitable than deterministic Trotterization. In addition, electronic structure Hamiltonians typically are not strictly low-range, but often elements in high-range indices tend to be very small, and much of the "weight" is concentrated in low-range areas. We can combine these two concepts to potentially leverage the asymptotic advantage of the coherent method towards simulating generic electronic structure Hamiltonians that are "low-range dominant".

We propose a framework for decomposition in which H_{2e} is broken into a sum of a term H_L that has small rank R and low range K and a term H_H that contains "everything else" and has no restrictions. Then we would qDRIFT over the R terms in H_L and the $O(n^4)$ tensor-product Pauli terms in H_H. Setting $R = 0$ would be equivalent to a conventional qDRIFT approach over the entire H_{2e}. The idea is that as R is increased and weight is "moved"

from H_H to H_L, H_L should gradually "capture" most of the weight in the Hamiltonian if it is indeed low-range dominant. Then, H_H would be left with many very small terms, which can be handled effectively with qDRIFT, whose complexity scales only with the total weight, rather than the number of terms. Meanwhile, H_L is able to handle the larger-valued low-range "weight" more efficiently due to its asymptotic advantages.

$$\begin{aligned} H_{2e} &= \frac{1}{2} \sum_{i,j,k,l} h'_{iljk} a_i^\dagger a_l a_j^\dagger a_k \\ &= H_L + H_H \end{aligned} \tag{3.6}$$

Unfortunately, it was not feasible to provide a specific implementation or numerical results in context of this framework, due to both the limited time frame of this project and the fact that T-count advantages of the coherent method only manifest asymptotically above a threshold at which Hamiltonian sizes become difficult for classical computers to manipulate. However, we hope that presenting this potential framework both conveys the applicability of the coherent method in realistic electronic structure Hamiltonians and inspires future work in this direction.

Chapter 4

Conclusion

In this work, we have designed in detail a quantum circuit that reduces the asymptotic gate complexity of simulating certain types of electronic structure Hamiltonians, in particular those that are low-range. With specific calculations of T counts and error bounds, we showed that asymptotic advantages of the proposed "double-factorized coherent method" over the competing "double-factorized expansion method" begin to manifest at a fairly low threshold of n: for example, fewer than a hundred orbitals for hydrogen chain systems.

Furthermore, we proposed a framework for adapting the usage of the double-factorized coherent method from strictly-low-range electronic structure Hamiltonians to those that are "low-range dominant", which tend to be the case for practical systems. We intend to continue formulating more specific decomposition techniques in this framework and numerically analyze the performance of the coherent method in conjunction with qDRIFT and qDRIFT-like methods; unfortunately this was not feasible within the timeframe of this thesis. We strongly encourage this framework as an area of future work for others inspired by the findings in this paper.

Appendix A

Quantum Circuit Diagrams

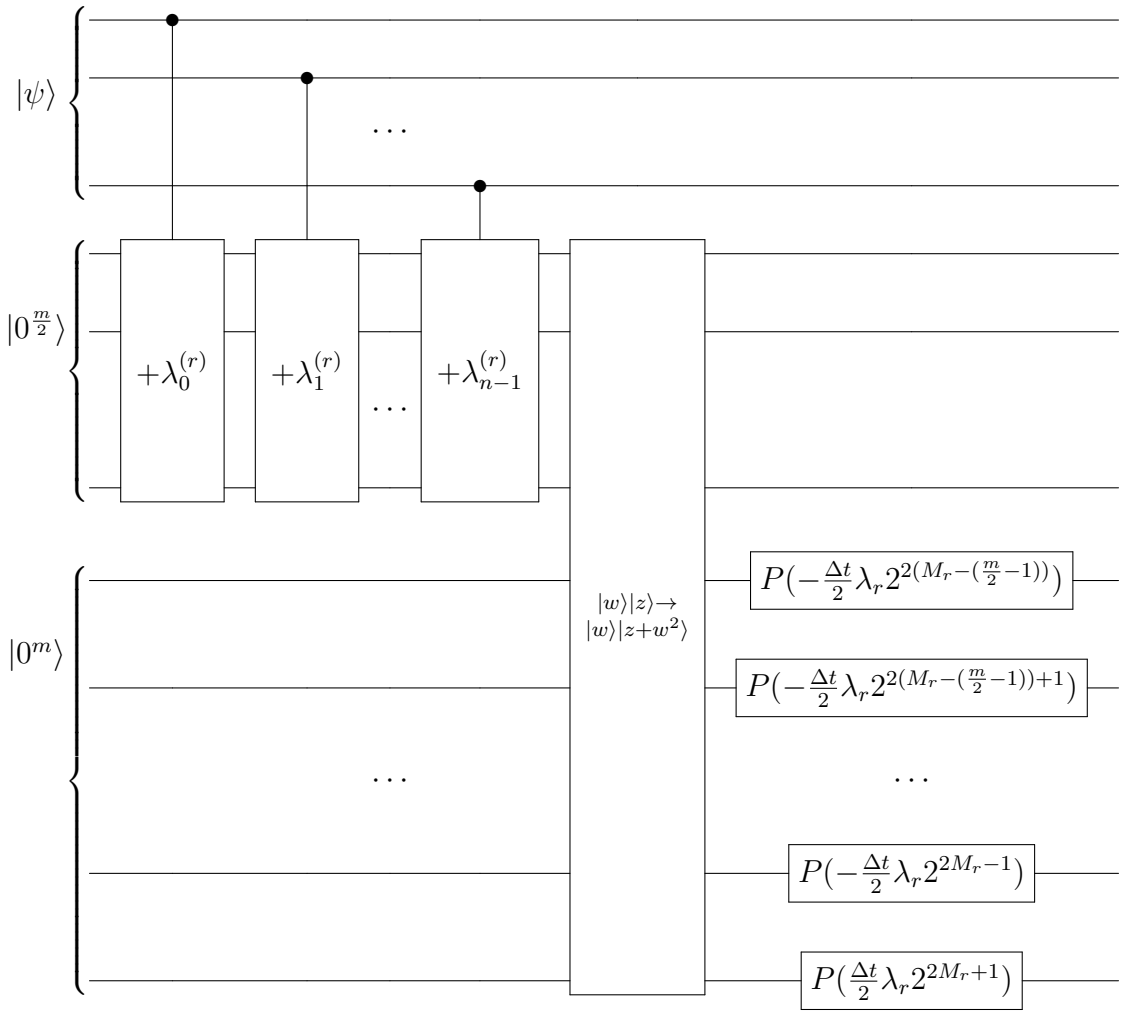

Figure A.1: Quantum circuit diagram for $U_A^{(r)}$ using arithmetic circuits (uncomputing not shown)

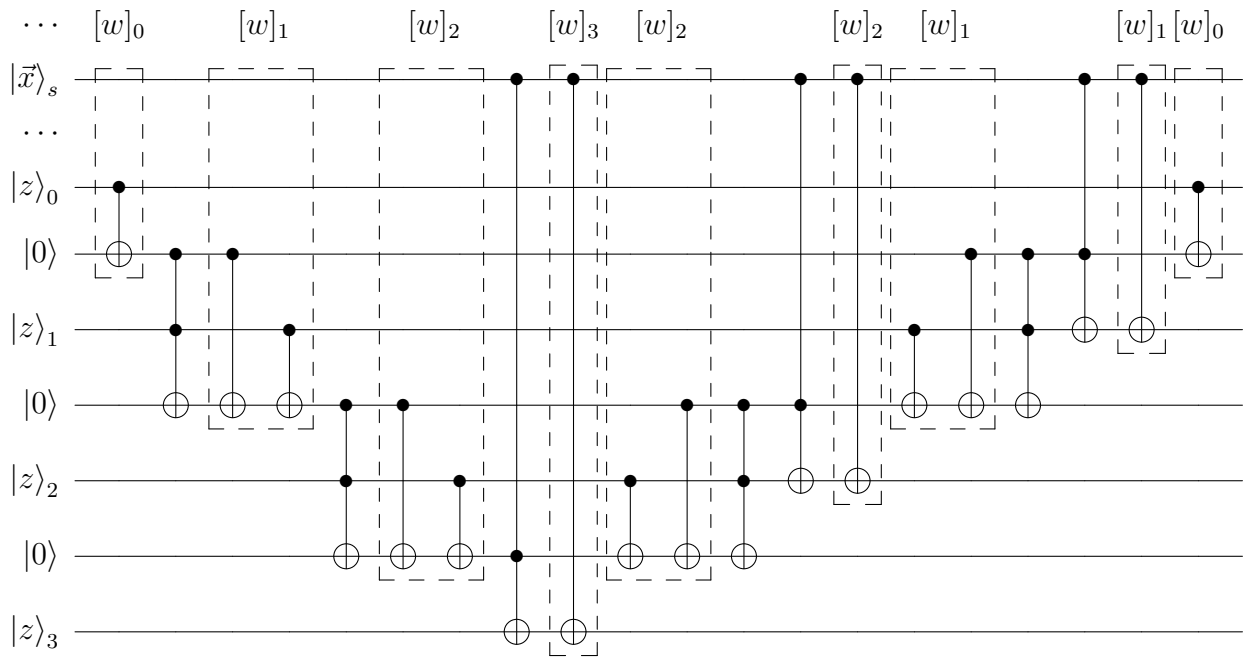

Figure A.2: Example quantum circuit diagram for a controlled addition of w for $\frac{m}{2} = 4$. The label $[w]_j$ indicates that the correspondingly boxed gates are only applied when $[w]_j = 1$.

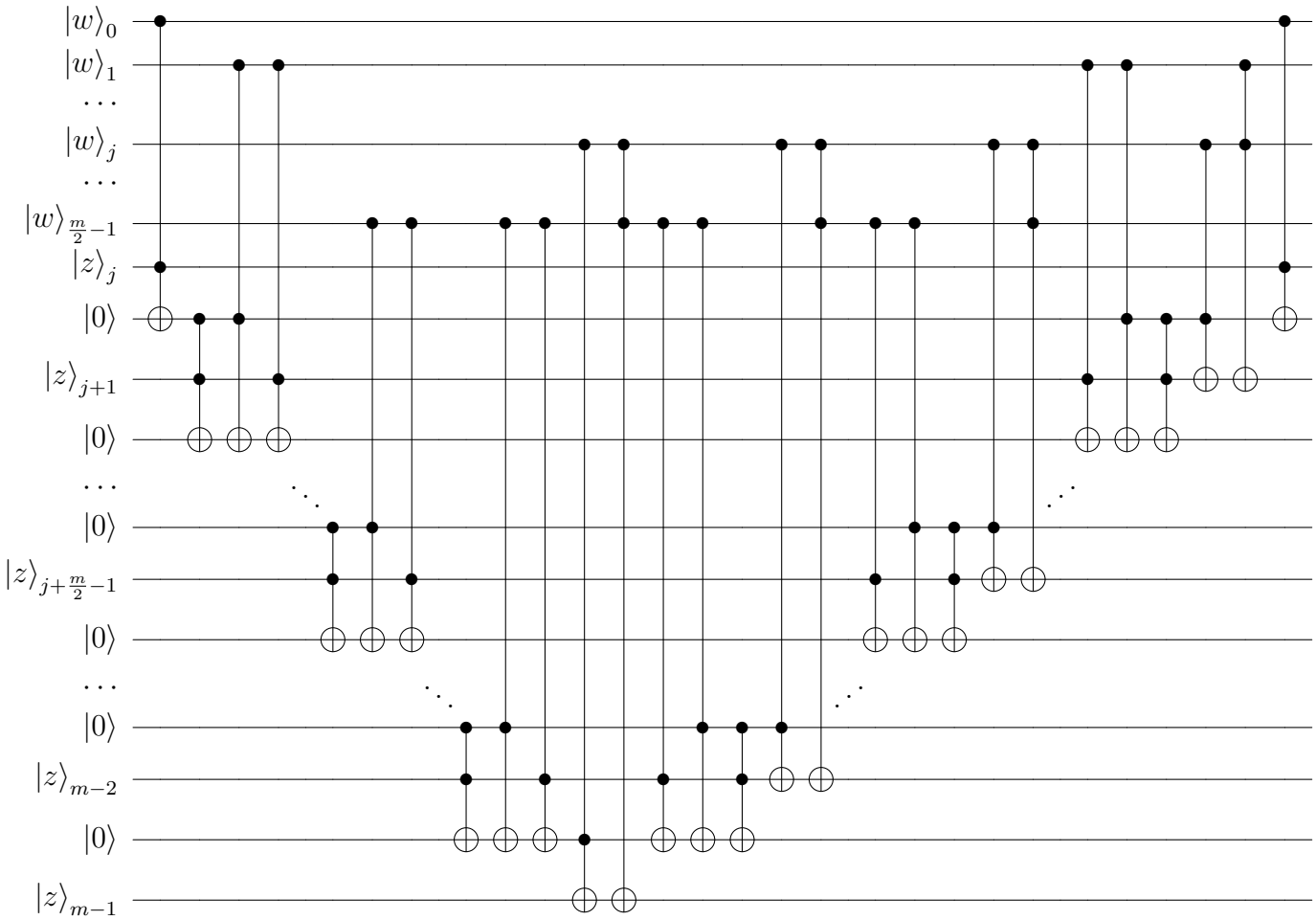

Figure A.3: Quantum circuit diagram for $|w\rangle|z\rangle \to |w\rangle|z + [w]_j 2^{M_r-(\frac{m}{2}-1)+j} w\rangle$. This is essentially the equivalent of A.2 if the added input were quantum, and if the controlling qubit $|\vec{x}\rangle_s$ were instead a qubit in w, and if the . Qubits 0 to $j-1$ of the $|z\rangle$ register are not shown.

References

[1] V. von Burg, G. H. Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer. "Quantum computing enhanced computational catalysis". In: *Physical Review Research* 3.3 (July 2021). ISSN: 2643-1564. DOI: 10.1103/physrevresearch.3.033055. URL: http://dx.doi.org/10.1103/PhysRevResearch.3.033055.

[2] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[3] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan. "Quantum computational chemistry". In: *Reviews of Modern Physics* 92.1 (Mar. 2020). ISSN: 1539-0756. DOI: 10.1103/revmodphys.92.015003. URL: http://dx.doi.org/10.1103/RevModPhys.92.015003.

[4] E. Campbell. "Random Compiler for Fast Hamiltonian Simulation". In: *Physical Review Letters* 123.7 (Aug. 2019). ISSN: 1079-7114. DOI: 10.1103/physrevlett.123.070503. URL: http://dx.doi.org/10.1103/PhysRevLett.123.070503.

[5] N. Cody Jones, J. D. Whitfield, P. L. McMahon, M.-H. Yung, R. V. Meter, A. Aspuru-Guzik, and Y. Yamamoto. "Faster quantum chemistry simulation on fault-tolerant quantum computers". In: *New Journal of Physics* 14.11 (Nov. 2012), p. 115023. DOI: 10.1088/1367-2630/14/11/115023. URL: https://dx.doi.org/10.1088/1367-2630/14/11/115023.

[6] J. T. Seeley, M. J. Richard, and P. J. Love. "The Bravyi-Kitaev transformation for quantum computation of electronic structure". In: *The Journal of Chemical Physics* 137.22 (Dec. 2012), p. 224109. ISSN: 0021-9606. DOI: 10.1063/1.4768229. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.4768229/13999577/224109_1_online.pdf. URL: https://doi.org/10.1063/1.4768229.

[7] S. B. Bravyi and A. Y. Kitaev. "Fermionic Quantum Computation". In: *Annals of Physics* 298.1 (2002), pp. 210–226. ISSN: 0003-4916. DOI: https://doi.org/10.1006/aphy.2002.6254. URL: https://www.sciencedirect.com/science/article/pii/S0003491602962548.

[8] J. D. Whitfield, J. Biamonte, and A. A.-G. and. "Simulation of electronic structure Hamiltonians using quantum computers". In: *Molecular Physics* 109.5 (2011), pp. 735–750. DOI: 10.1080/00268976.2011.552441. eprint: https://doi.org/10.1080/00268976.2011.552441. URL: https://doi.org/10.1080/00268976.2011.552441.

[9] B. Peng and K. Kowalski. "Highly Efficient and Scalable Compound Decomposition of Two-Electron Integral Tensor and Its Application in Coupled Cluster Calculations". In: *Journal of Chemical Theory and Computation* 13.9 (2017). PMID: 28834428, pp. 4179–4192. DOI: 10.1021/acs.jctc.7b00605. eprint: https://doi.org/10.1021/acs.jctc.7b00605. URL: https://doi.org/10.1021/acs.jctc.7b00605.

[10] I. D. Kivlichan, J. McClean, N. Wiebe, C. Gidney, A. Aspuru-Guzik, G. K.-L. Chan, and R. Babbush. "Quantum Simulation of Electronic Structure with Linear Depth and Connectivity". In: *Physical Review Letters* 120.11 (Mar. 2018). ISSN: 1079-7114. DOI: 10.1103/physrevlett.120.110501. URL: http://dx.doi.org/10.1103/PhysRevLett.120.110501.

[11] P. Selinger. *Efficient Clifford+T approximation of single-qubit operators*. 2014. arXiv: 1212.6253 [quant-ph]. URL: https://arxiv.org/abs/1212.6253.

[12] C. Jones. "Low-overhead constructions for the fault-tolerant Toffoli gate". In: *Phys. Rev. A* 87 (2 Feb. 2013), p. 022328. DOI: 10.1103/PhysRevA.87.022328. URL: https://link.aps.org/doi/10.1103/PhysRevA.87.022328.

[13] D. Wecker, B. Bauer, B. K. Clark, M. B. Hastings, and M. Troyer. "Gate-count estimates for performing quantum chemistry on small quantum computers". In: *Phys. Rev. A* 90 (2 Aug. 2014), p. 022305. DOI: 10.1103/PhysRevA.90.022305. URL: https://link.aps.org/doi/10.1103/PhysRevA.90.022305.