Optimizing Electronic-Structure Hamiltonians for Locality

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

© 2025 Louis Wenjun Marquis. All rights reserved.

The author hereby grants to MIT a nonexclusive, worldwide, irrevocable, royalty-free license to exercise any and all rights under copyright, including to reproduce, preserve, distribute and publicly display copies of the thesis, or release the thesis under an open-access license.

Authored by:	Louis Wenjun Marquis Department of Electrical Engineering and Computer Science May 9, 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 Electronic-Structure Hamiltonians for Locality

by

Louis Wenjun Marquis

Submitted to the Department of Electrical Engineering and Computer Science
on May 9, 2025 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

The developments of the “kinetic theory” of gases made within the last ten years have enabled it to account satisfactorily for many of the laws of gases. The mathematical deductions of Clausius, Maxwell and others, based upon the hypothesis of a gas composed of molecules acting upon each other at impact like perfectly elastic spheres, have furnished expressions for the laws of its elasticity, viscosity, conductivity for heat, diffusive power and other properties. For some of these laws we have experimental data of value in testing the validity of these deductions and assumptions. Next to the elasticity, perhaps the phenomena of the viscosity of gases are best adapted to investigation.[1]

Thesis supervisor: Aram Harrow
Title: Professor of Physics

---

[1] Text from Holman (1876): doi:10.2307/25138434.

Acknowledgments

Write your acknowledgments here.

Contents

List of Figures 9

List of Tables 11

1 Introduction 13
- 1.1 Hamiltonian Simulation . . . 13
- 1.2 Trotterization . . . 14
- 1.3 Second Quantized Fock States . . . 15
- 1.4 Jordan-Wigner Transformation . . . 16
- 1.5 Double Factorization of Electronic Structure Hamiltonian . . . 17
  - 1.5.1 Double-Factorized Expansion Method . . . 20
  - 1.5.2 Discussion . . . 22

2 Quantum Arithmetic Circuit Design for Double-Factorized Electronic Structure Hamiltonian Simulation 23
- 2.1 Summation . . . 24
- 2.2 Squaring . . . 25
- 2.3 Phase Rotation . . . 26
- 2.4 Error Analysis . . . 27
- 2.5 Complexity Analysis . . . 29
- 2.6 Numerics . . . 30
- 2.7 Discussion . . . 31

3 Convex Optimization of Decomposition of Electron-Electron Interaction Hamiltonian Term 33
- 3.1 Lower Bounds on Rotation Complexity . . . 33
- 3.2 K-Range Hamiltonian . . . 34
- 3.3 Decomposition of Hamiltonian Term . . . 35
- 3.4 Gradient Descent . . . 36
- 3.5 Gate Count Analysis . . . 36
- 3.6 Numerics . . . 36

4 Conclusion 37
- 4.1 Directions for Further Work . . . 37

A Quantum Circuit Diagrams 39

List of Figures

1.1 Construction of $CR_z(\varphi)$ in (1.36) with two $CNOT$ gates and three R_z gates 21

A.1 Quantum circuit diagram for $U_A^{(r)}$ using arithmetic circuits (uncomputing not shown) 39

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 $[w]_j = 1$. 40

A.3 Quantum circuit diagram for $|w\rangle\,|z\rangle \rightarrow |w\rangle\,|z + [w]_j 2^{M_r-(\frac{m}{2}-1)+j} w\rangle$. 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. 41

List of Tables

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. In particular, 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.

1.1 Hamiltonian Simulation

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

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

Operators and states can be thought of as matrices and vectors, but where a vector maps integer indices to values, a state is a function that maps a continuous variable (such as position) to an amplitude value. An operator maps state functions to state functions how a matrix maps vectors to vectors.

The Hamiltonian $\hat{H}$ 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. $\hat{H}$, and in turn $U = e^{-i\hat{H}t}$ can be very complex, and usually it's not feasible to simulate U exactly. Instead, we try 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.

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

This task is accordingly named Hamiltonian simulation.

One important Hamiltonian is the electronic structure Hamiltonian, describing 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.

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

$\vec{r}_p$ 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 $e^{-i\hat{H}t}$, or the states that are invariant (up to a phase) under $e^{-i\hat{H}t}$. The ability to simulate the behavior of $e^{-i\hat{H}t}$ on an arbitrary state can directly be used to find the eigenvalues using the quantum phase estimation algorithm. The details of this algorithm are not essential in the topic of this paper, the relevant fact is the power unlocked by performing $e^{-i\hat{H}t}$.

1.2 Trotterization

The Hamiltonian H of a system may often be a sum of many terms $\hat{h}_j$, as exemplified by the electronic structure Hamiltonian in (1.3).

$$\hat{H} = \sum_j \hat{h}_j \tag{1.4}$$

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

$$U = e^{-i\hat{H}t} = e^{-i\sum_j \hat{h}_j t} = \lim_{M\to\infty} (\prod_j e^{-i\hat{h}_j \frac{t}{M}})^M \tag{1.5}$$

In other words, for sufficiently large M, U is essentially equivalent to iterating M times through smaller evolutions $e^{-i\hat{h}_j \frac{t}{M}}$. We can denote $\Delta t = \frac{t}{M}$ the step size. This is the first-order Trotter-Suzuki Algorithm. This is very convenient because it means that if we can simulate a Hamiltonian evolution with $\hat{h}_j$, we can also approximately simulate U. This will be the case most of the time in this paper.

However, M must be sufficiently large to make the approximation sufficiently precise. At the same time, the computation cost of the algorithm scales linearly with the number of individual evolutions and therefore with M. It turns out that, to achieve an error threshold of ϵ, a cost of $\frac{L^3\Lambda^2t^2}{\epsilon}$ is necessary, in which L is the number of terms h_j, and $\Lambda = \max_j ||h_j||$.

First-order Trotterization as described is the most basic of such Hamiltonian simulation algorithms, and there is a rich field of literature finding improved methods to decrease error and computation cost for this problem.

One example is qDRIFT, a randomized version of first-order Trotterization. Its computation cost is $O(\frac{\lambda^2t^2}{\epsilon})$, in which $\lambda = \sum_j ||h_j||$.

The power of such Trotter-like techniques (breaking $e^{-i\hat{H}t}$ into many $e^{-i\hat{h}_jt}$) allows us to analyze each term $\hat{h}_j$ independently of the others. If one can simulate $e^{-i\hat{h}_jt}$ for every term, one can simulate U.

1.3 Second Quantized Fock States

We would like to model the electronic structure system described by $\hat{H}$ 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.

$$\psi = \phi_1 \otimes \phi_2 - \phi_2 \otimes \phi_1 \tag{1.6}$$

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 $\vec{f} \in \{0,1\}^n$.

$$|\vec{f}\rangle = |f_0, f_1, ..., f_{n-1}\rangle \tag{1.7}$$

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.

$$\begin{aligned} a_i^\dagger |\vec{f}\rangle &= a_i^\dagger |f_0, ..., f_{n-1}\rangle \\ &= \delta_{f_i,1}(-1)^{\sum_{j=0}^{i-1}} |f_0, ..., f_i \oplus 1, ..., f_{n-1}\rangle \end{aligned} \tag{1.8}$$

$$\begin{aligned} a_i |\vec{f}\rangle &= a_i |f_0, ..., f_{n-1}\rangle \\ &= \delta_{f_i,0}(-1)^{\sum_{j=0}^{i-1}} |f_0, ..., f_i \oplus 1, ..., f_{n-1}\rangle \end{aligned} \tag{1.9}$$

$a_i^\dagger$ 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 $n_i = a_i^\dagger a_i$ [1].

$$n_i |\vec{f}\rangle = f_i |f_0, ..., f_{n-1}\rangle \tag{1.10}$$

The creation and annihilation operators have anti-commutation relations [1] that are analogous to typical commutation relations of creation and annihilation operators.

$$\{a_i^\dagger, a_j\} = \delta_{i,j} \tag{1.11}$$

$$\{a_i, a_j\} = \{a_i^\dagger, a_j^\dagger\} = 0 \tag{1.12}$$

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.

1.4 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.

$$|\psi\rangle = A_0 |0\rangle + A_1 |1\rangle \text{ such that } |A_0|^2 + |A_1|^2 = 1 \tag{1.13}$$

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

$$|\psi\rangle = \sum_{\vec{x}\in\{0,1\}^n} A_{\vec{x}} |\vec{x}\rangle \text{ such that } \sum_{\vec{x}\in\{0,1\}^n} |A_{\vec{x}}|^2 = 1 \tag{1.14}$$

It is then straightforward that basis states (1.7) of an n-orbital many-electron physical system correspond to basis states (1.14) of n qubits in a quantum computer. This is the Jordan-Wigner Encoding.

$$|\vec{f}\rangle \leftrightarrow |\vec{x}\rangle \tag{1.15}$$

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 (1.8) and annihilation (1.9) operators on the physical system also correspond to qubit operators.

$$a_i \leftrightarrow Z_0 Z_1 ... Z_{i-1} |0\rangle \langle 1|_i \tag{1.16}$$

$$a_i^\dagger \leftrightarrow Z_0 Z_1 ... Z_{i-1} |1\rangle \langle 0|_i \tag{1.17}$$

$$n_i = a_i^\dagger a_i \leftrightarrow |1\rangle \langle 1|_i \tag{1.18}$$

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.

1.5 Double Factorization of 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.

$$\hat{H} = \sum_{i,j} h_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_j^\dagger a_k a_l \tag{1.19}$$

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

$$\begin{aligned} h_{ij} &= \langle \phi_i| \left(-\frac{\hbar^2}{2m_e}\nabla^2 - \sum_I \frac{e^2}{4\pi\epsilon_0} \frac{Z_I}{|\vec{r} - \vec{R}_I|} \right) |\phi_j\rangle \\ &= \int \phi_i^*(\vec{r}) \left(-\frac{\hbar^2}{2m_e}\nabla^2 - \sum_I \frac{e^2}{4\pi\epsilon_0} \frac{Z_I}{|\vec{r} - \vec{R}_I|} \right) \phi_j(\vec{r}) d^3\vec{r} \end{aligned} \tag{1.20}$$

$$\begin{aligned} h_{ijkl} &= \langle \phi_i, \phi_j| \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{r}_1 - \vec{r}_2|} |\phi_l, \phi_k\rangle \\ &= \iint \phi_i^*(\vec{r}_1)\phi_j^*(\vec{r}_2) \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{r}_1 - \vec{r}_2|} \phi_l(\vec{r}_1)\phi_k(\vec{r}_2) d^3\vec{r}_1 d^3\vec{r}_2 \end{aligned} \tag{1.21}$$

Observe from (1.20) that h_{ij}, which is real, must be symmetric between i and j. From (1.21), 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.

$$\begin{aligned}
\hat{H} &= \sum_{i,j} h_{ij} a_i^\dagger a_j + \frac{1}{2}\sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_j^\dagger a_k a_l \\
&= \sum_{i,j} h_{ij} a_i^\dagger a_j - \frac{1}{2}\sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_j^\dagger a_l a_k \\
&= \sum_{i,j} h_{ij} a_i^\dagger a_j - \frac{1}{2}\sum_{i,j,k,l} h_{ijkl} a_i^\dagger (\delta_{jl} - a_l a_j^\dagger) a_k \\
&= \sum_{i,j} h_{ij} a_i^\dagger a_j - \frac{1}{2}\left(\sum_{i,j,k} h_{ijkj} a_i^\dagger a_k - \sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_l a_j^\dagger a_k\right) \\
&= \sum_{i,j} (h_{ij} - \frac{1}{2}\sum_k h_{ikjk}) a_i^\dagger a_j + \frac{1}{2}\sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_l a_j^\dagger a_k \\
&= \sum_{i,j} h'_{ij} a_i^\dagger a_j + \frac{1}{2}\sum_{i,j,k,l} h'_{iljk} a_i^\dagger a_l a_j^\dagger a_k \\
&= \hat{H}_{1e} + \hat{H}_{2e}
\end{aligned} \tag{1.22}$$

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

$$\begin{aligned}
h'_{ij} &= h_{ij} - \frac{1}{2}\sum_k h_{ikjk} \\
\hat{H}_{1e} &= \sum_{i,j} h'_{ij} a_i^\dagger a_j \\
h'_{iljk} &= h_{ijkl} \\
\hat{H}_{2e} &= \frac{1}{2}\sum_{i,j,k,l} h'_{iljk} a_i^\dagger a_l a_j^\dagger a_k
\end{aligned} \tag{1.23}$$

As a reminder, we would like to simulate $U = e^{-i\hat{H}t}$ on a quantum computer. Trotterization allows us to study each term in (1.22) independently. That is, we just need to find a way to simulate $U_1 = e^{-i\hat{H}_{1e}\Delta t}$ and $U_2 = e^{-i\hat{H}_{2e}\Delta t}$ for arbitrary Δt. The direct brute force method is to substitute the operator correspondances from (1.16) and (1.17) into (1.22) to obtain the corresponding qubit Hamiltonian. With this method, it turns out that U_1 is a sum of $O(n^2)$ terms and $U_2 = e^{-\hat{H}_{2e}t}$ becomes $O(n^4)$ terms. We can iterate over the terms with a Trotter-like or qDRIFT, but the $O(n^4)$ can make such an iteration very tedious. We'd like to see if there's a more clever method.

Recall that because $h'_{iljk} = h_{ijkl}$ is symmetric between (i,l) and (j,k), h' can 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.

$$h'_{iljk} = \sum_r \lambda_r Q^{(r)}_{i,l} Q^{(r)}_{j,k} \tag{1.24}$$

$$
\begin{aligned}
\hat{H}_{2e} &= \frac{1}{2}\sum_{i,j,k,l} h'_{ijkl} a_i^\dagger a_l a_j^\dagger a_k \\
&= \frac{1}{2}\sum_{i,j,k,l}\sum_r \lambda_r Q_{i,l}^{(r)} Q_{j,k}^{(r)} a_i^\dagger a_l a_j^\dagger a_k \\
&= \frac{1}{2}\sum_r \lambda_r \sum_{i,l} Q_{i,l}^{(r)} a_i^\dagger a_l \sum_{j,k} Q_{j,k}^{(r)} a_j^\dagger a_k \\
&= \frac{1}{2}\sum_r \lambda_r \left(\sum_{i,j} Q_{i,j}^{(r)} a_i^\dagger a_j\right)^2 \\
&= \sum_r \hat{H}_{2e}^{(r)}
\end{aligned}
\tag{1.25}
$$

We have written $\hat{H}_{2e}$ as a sum of terms $\hat{H}_{2e}^{(r)}$. If we could simulate $U_2^{(r)} = e^{-i\hat{H}_{2e}^{(r)}\Delta t}$ for an arbitrary Δt, then we can Trotter or qDRIFT over these terms to implement U_2.

$$
\hat{H}_{2e}^{(r)} = \frac{1}{2}\lambda_r \left(\sum_{i,j} Q_{i,j}^{(r)} a_i^\dagger a_j\right)^2 \tag{1.26}
$$

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 an $n \times n$ matrix to be further eigendecomposed with eigenvalues $\lambda_s'^{(r)}$ and real orthonormal eigenvectors $R_s^{(r)}$ as the columns of $R^{(r)}$, which is orthogonal.

$$
\begin{aligned}
Q^{(r)} &= R^{(r)} \begin{bmatrix} \lambda_0'^{(r)} & & \\ & \ddots & \\ & & \lambda_{n-1}'^{(r)} \end{bmatrix} R^{(r)T} \\
Q_{i,j}^{(r)} &= \sum_s \lambda_s'^{(r)} R_{is}^{(r)} R_{js}^{(r)}
\end{aligned}
\tag{1.27}
$$

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 $R^{(r)}$ 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 basis $\{\tilde{\phi}_s = \sum_s R_{is}^{(r)} \phi_i\}_{s\in[0,n)}$. This rotated basis also has its own number operators $\tilde{n}_s^{(r)}$.

$$
\begin{aligned}
\tilde{a}_s^{(r)\dagger} &= \sum_i R_{is}^{(r)} a_i^\dagger \\
\tilde{a}_s^{(r)} &= \sum_i R_{is}^{(r)} a_i \\
\tilde{n}_s^{(r)} &= \tilde{a}_s^{(r)\dagger} \tilde{a}_s^{(r)}
\end{aligned}
\tag{1.28}
$$

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

$$
\begin{aligned}
H_{2e}^{(r)} &= \frac{1}{2}\lambda_r \left(\sum_{i,j} Q_{i,j}^{(r)} a_i^\dagger a_j \right)^2 \\
&= \frac{1}{2}\lambda_r \left(\sum_{i,j} \sum_s \lambda_s'^{(r)} R_{is}^{(r)} R_{js}^{(r)} a_i^\dagger a_j \right)^2 \\
&= \frac{1}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} \sum_i R_{is}^{(r)} a_i^\dagger \sum_j R_{js}^{(r)} a_j \right)^2 \\
&= \frac{1}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} \tilde{a}_s^{(r)\dagger} \tilde{a}_s^{(r)} \right)^2 \\
&= \frac{1}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} \tilde{n}_s^{(r)} \right)^2
\end{aligned}
\tag{1.29}
$$

We would like to simulate $U_2^{(r)} = e^{-i\hat{H}_{2e}^{(r)}\Delta T}$ 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. [2] uses the Thouless Theorem to show that such a rotation operator $U_R(R^{(r)})$ necessary can be calculated from $R^{(r)}$ and and corresponds to a maximum of $\binom{n}{2} = O(n^2)$ gates on the quantum computer.

$$
U_2^{(r)} = e^{-\frac{i\Delta t}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} \tilde{n}_s^{(r)} \right)^2} = U_R(R^{(r)})^\dagger e^{-\frac{i\Delta t}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} n_s \right)^2} U_R(R^{(r)}) = U_R(R^{(r)})^\dagger U_A^{(r)} U_R(R^{(r)})
\tag{1.30}
$$

$$
U_A^{(r)} = e^{-\frac{i\Delta t}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} n_s \right)^2}
\tag{1.31}
$$

1.5.1 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.

$$
\begin{aligned}
U_A^{(r)} &= e^{-\frac{i\Delta t}{2}\lambda_r \left(\sum_s \lambda_s'^{(r)} n_s \right)^2} \\
&= e^{-\frac{i\Delta t}{2}\lambda_r \sum_{s,s'} \lambda_s'^{(r)} \lambda_{s'}'^{(r)} n_s n_{s'}} \\
&= \prod_{s,s'} e^{-\frac{i\Delta t}{2}\lambda_r \lambda_s'^{(r)} \lambda_{s'}'^{(r)} n_s n_{s'}} \\
&= \prod_s e^{-\frac{i\Delta t}{2}\lambda_s (\lambda_s'^{(r)})^2 n_s} \prod_{s<s'} e^{-i\Delta t \lambda_r \lambda_s'^{(r)} \lambda_{s'}'^{(r)} n_s n_{s'}}
\end{aligned}
\tag{1.32}
$$

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.

$$e^{-\frac{i\Delta t}{2}\lambda_r(\lambda_s'^{(r)})^2 n_s} \leftrightarrow R_z(-\frac{\Delta t}{2}\lambda_r(\lambda_s'^{(r)})^2)_s \tag{1.33}$$

$$R_z(\varphi) = \begin{bmatrix} e^{-i\frac{\varphi}{2}} & 0 \\ 0 & e^{i\frac{\varphi}{2}} \end{bmatrix} = e^{-i\frac{\varphi}{2}} \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} = e^{-i\frac{\varphi}{2}}(|0\rangle\langle 0| + e^{i\varphi}|1\rangle\langle 1|) \tag{1.34}$$

$$e^{-\frac{i\Delta t}{2}\lambda_r\lambda_s'^{(r)}\lambda_{s'}'^{(r)} n_s n_{s'}} \leftrightarrow CR_z(-\frac{\Delta t}{2}\lambda_r\lambda_s'^{(r)}\lambda_{s'}'^{(r)})_{s,s'} \tag{1.35}$$

$$CR_z(\varphi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\varphi} \end{bmatrix} \tag{1.36}$$

There are therefore n R_z operators and $\binom{n}{2}$ CR_z operators in this decomposition. Each CR_z can be decomposed 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 $U_A^{(r)}$.

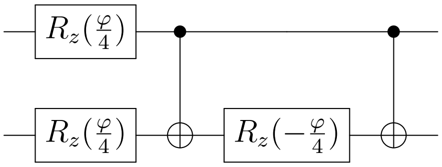

Figure 1.1: Construction of $CR_z(\varphi)$ in (1.36) with two $CNOT$ gates and three R_z gates

[3] 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 $n, \frac{1}{\epsilon}$.

$$\begin{aligned} \tilde{N}_T &= \frac{3n(n-1)}{2}(10 + \log_2(\frac{1}{\epsilon_z})) \\ &= \frac{3n(n-1)}{2}(10 + \log_2(\frac{\frac{3n(n-1)}{2}}{\epsilon})) \\ &= O(n^2\log_2(\frac{n}{\epsilon})) \end{aligned} \tag{1.37}$$

1.5.2 Discussion

Similarly to (1.32), $U_R(R^{(r)})$ in (1.30) can be decomposed into at most $O(n^2)$ non-Clifford gates as described in [2], resulting in a $O(n^2 \log_2(\frac{n}{\epsilon}))$ T-count. Combined with (1.37), 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 $U_R(R^{(r)})$ decomposes using the [2] 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 $U_R(R^{(r)})$ 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 (1.32).

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

We can substitute the number operator correspondance from (1.18) 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.

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

(1.32) 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.

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 (and therefore $\tilde{U}_A^{(r)}$ as a whole) will be an approximation.

$$|\vec{x}\rangle \to |\vec{x}\rangle |0^{\frac{m}{2}}\rangle |0^m\rangle \tag{2.6}$$

$$\to |\vec{x}\rangle |\tilde{y}_r(\vec{x})\rangle |0^m\rangle \tag{2.7}$$

$$\to |\vec{x}\rangle |\tilde{y}_r(\vec{x})\rangle |\tilde{y}_r(\vec{x})^2\rangle \tag{2.8}$$

$$\to e^{-i\frac{\Delta t}{2}\lambda_r \tilde{y}_r(\vec{x})^2} |\vec{x}\rangle |\tilde{y}_r(\vec{x})\rangle |\tilde{y}_r(\vec{x})^2\rangle \tag{2.9}$$

$$\to e^{-i\frac{\Delta t}{2}\lambda_r \tilde{y}_r(\vec{x})^2} |\vec{x}\rangle |0^{\frac{m}{2}}\rangle |0^m\rangle \tag{2.10}$$

$$\to e^{-i\frac{\Delta t}{2}\lambda_r \tilde{y}_r(\vec{x})^2} |\vec{x}\rangle \tag{2.11}$$

$$\to e^{-i\frac{\Delta t}{2}\lambda_r (\sum_s \tilde{\lambda}_s'^{(r)} |1\rangle\langle 1|_s)^2} |\vec{x}\rangle \tag{2.12}$$

$$\approx \tilde{U}_A^{(r)} |\vec{x}\rangle \tag{2.13}$$

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.

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

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

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

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 [3] 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 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 [3]. 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.

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

A Toffoli gate can be constructed from 2 Hadamard gates, 6 CNOT gates, and 7 T-gates. We can then calculate the total T-count of $\tilde{U}_A^{(r)}$.

$$\begin{aligned}\tilde{N}_T &= 7\tilde{N}_{CCNOT} + \tilde{N}_{T;Rz} \\ &= 21mn - 70n + 98m^2 - 28m + 10m + 4m\log_2(\frac{2m}{\epsilon}) \\ &= 21mn - 70n + 98m^2 - 18m + 4m\log_2(\frac{2m}{\epsilon}) \\ &= O(mn + m^2 + m\log_2(\frac{1}{\epsilon}))\end{aligned} \tag{2.40}$$

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 Numerics

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 (1.37), 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. [4] 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 $|\lambda_r|Y_r^2 \approx n\log_2 nE_h$.

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

In practice, $\Delta << t$, so t is not a useful order of magnitude for Δt. [4] also mentions a value $\Delta t = 0.01\frac{1}{E_h}$ which is probably closer to the actual of Δt.

$$m_H^* \approx 2\log_2(n) + 2\log_2\left(\frac{n\log_2 n \times 0.01}{10^{-6}}\right) + 4 \approx 4\log_2(n) + 2\log_2(\log_2(n)) + 18 \tag{2.43}$$

$$\tilde{N}_T = 21m(n)n - 70n + 98m(n)^2 - 18m(n) + 4m(n)\log_2(\frac{2m(n)}{\epsilon}) \tag{2.44}$$

We plot this T-count of the “coherent method” as well as that of the “expansion method” in (1.37) as functions of n. It turns out that the expansion method surpasses the coherent method in T-count at approxmiately $n \approx 122$. At $n = 122$, the coherent method requires about $3.94 * 10^5$ T-gates and has $m = 52$, while the expansion method requires about $4.00 * 10^5$ T-gates.

INSERT PLOT

2.7 Discussion

The asymptotic advantage of the coherent method ($O(n^3 \log_2 n)$) over the expansion method ($O(n^4)$) is only useful in $U_A^{(r)}$ if the number of T gates in $U_R(R^{(r)})$ 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 $U_R(R^{(r)})$ can be decomposed “easily” as such.

Chapter 3

Convex Optimization of Decomposition of Electron-Electron Interaction Hamiltonian Term

3.1 Lower Bounds on Rotation Complexity

only results in an overall asymptotic advantage if $U(R^{(r)})$ can also be implemented in with sub-quadratic gate complexity. [2] decomposed $U(R^{(r)})$ into a series of at most $\binom{n}{2}$ two-orbital basis rotations, each implementable in a constant number of quantum gates. However, for certain types of Hamiltonians, the necessary number of such rotations is often much smaller.

The minimum number of rotation is 0, which occurs when $R^{(r)} = I$; that is, when $Q^{(r)}$ is already diagonal. However, if this is the case for all $r \in [0, n^2)$ in $\hat{H}_{2e}$, then all of the terms in $\hat{H}_{2e}$ can be combined into n^2 terms as shown in (3.1), resulting in a $O(n^2)$ complexity for $\hat{H}_{2e}$, in which case the coherent method would not be useful.

$$\begin{aligned}\hat{H}_{2e} &= \frac{1}{2}\sum_r \lambda_r \left(\sum_{i,j} Q^{(r)}_{i,j} a^\dagger_i a_j\right)^2 \\ &= \frac{1}{2}\sum_r \lambda_r (\sum_s \lambda'^{(r)}_s n_s)^2 \\ &= \frac{1}{2}\sum_{s,s'} n_s n_{s'} \sum_r \lambda_r \lambda'^{(r)}_s \lambda'^{(r)}_{s'}\end{aligned} \tag{3.1}$$

In fact, more generally, if $R^{(r)}$ is invariant across r, all of the different $\tilde{n}^{(r)}_s$ are in the same basis across r. Then the basis rotations $U(R^{(r)})$ and $U(R^{(r)})^\dagger$ can be performed at the beginning and end without the need to switch bases between r. Then $\hat{H}_{2e}$ combines into $O(n^2)$ terms in (3.2), just like in (3.1). Since $U(R^{(r)})$ uses $O(n^2)$ but is only performed twice, the total complexity is $O(n^2)$ for $\hat{H}_{2e}$, and the coherent would not be useful in this case either.

$$\begin{aligned}\hat{H}_{2e} &= \frac{1}{2}\sum_r \lambda_r \left(\sum_s \lambda_s'^{(r)} \tilde{n}_s\right)^2 \\ &= \frac{1}{2}\sum_{s,s'} \tilde{n}_s \tilde{n}_{s'} \sum_r \lambda_r \lambda_s'^{(r)} \lambda_{s'}'^{(r)}\end{aligned} \tag{3.2}$$

Moreover, if all the $Q^{(r)}$ are diagonally dominated (the magnitude of the terms on its diagonal dominates over the non-diagonal terms) or if they have very few ($O(\sqrt{n})$) nonzero off-diagonal elements, then the expansion of $\hat{H}_{2e}$ also contains a reduced number of terms that make the coherent method unnecessary.

For the coherent method to be useful, all of the $U(R^{(r)})$ should decompose to $o(n^2)$ rotations, but $Q^{(r)}$ should have $\omega(\sqrt{n})$ non-zero terms (not a tight bound). This opens an area for future work to address: identifying useful Hamiltonians that satisfy these conditions.

In the next section, we will focus on one family of Hamiltonians that satisfy these conditions.

3.2 K-Range Hamiltonian

Define an electronic structure Hamiltonian from (1.22) as K-range if its values h_{ijkl} satisfy the following condition.

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

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 $\bar{h}_{ijkl}$.

$$\begin{gathered}\hat{H} = \sum_{i,j} h_{ij} a_i^\dagger a_j + \frac{1}{2}\sum_{i,j,k,l} \bar{h}_{ijkl} a_i^\dagger a_j^\dagger a_k a_l \\ \text{such that } \bar{h}_{ijkl} = 0 \text{ if } \max(|i-l|,|j-k|) > K\end{gathered} \tag{3.4}$$

Then, accordingly with (1.22), this Hamiltonian can be rearranged, and accordingly with (1.25), the second term can be factored. The condition does not change.

$$\begin{aligned}\hat{H} &= \sum_{i,j} h'_{ij} a_i^\dagger a_j + \frac{1}{2}\sum_{i,j,k,l} h'_{iljk} a_i^\dagger a_l a_j^\dagger a_k \\ &= \sum_{i,j} h'_{ij} a_i^\dagger a_j + \frac{1}{2}\sum_r \lambda_r \left(\sum_{i,j} Q_{i,j}^{(r)} a_i^\dagger a_j\right)^2 \\ &\text{such that } h'_{iljk} = \bar{h}_{ijkl} = 0 \text{ if } \max(|i-l|,|j-k|) > K\end{aligned} \tag{3.5}$$

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

$$Q^{(r)} = \begin{bmatrix} Q^{(r)}_{0,0} & \cdots & Q^{(r)}_{0,K} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \vdots & Q^{(r)}_{1,K+1} & \ddots & & & & \vdots \\ Q^{(r)}_{K,0} & \cdots & \ddots & \ddots & \ddots & \ddots & & & 0 \\ 0 & Q^{(r)}_{K+1,1} & \ddots & \ddots & \ddots & \ddots & \ddots & & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & Q^{(r)}_{n-K-2,n-2} & 0 \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & \cdots & Q^{(r)}_{n-K-1,n-1} \\ \vdots & & & & \ddots & Q^{(r)}_{n-2,n-K-2} & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & Q^{(r)}_{n-1,n-K-1} & \cdots & Q^{(r)}_{n-1,n-1} \end{bmatrix} \tag{3.6}$$

If $Q^{(r)}$ has bandwidth K, this guarantees $R^{(r)}$ can be decomposed into at most Kn Givens rotations, corresponding to at most $O(Kn)$ two-qubit basis rotations comprising $U(R^{(r)})$. PROOF? Meanwhile, $Q^{(r)}$ has $\Omega(Kn)$ nonzero terms, so the terms cannot combine in the way of (3.1) and (3.2). Therefore, a K-range Hamiltonian can be expressed as a sum of n^2 terms, each simulable with $O(n(\log n + K))$ gates using the coherent method. We see that the coherent method is therefore particularly useful on such low-range Hamiltonians. For example, if $K = O(\log n)$, the total number of gates in a single Trotter iteration over these terms is $O(n^3 \log n)$, beating the $O(n^4)$ complexity of a brute force method.

3.3 Decomposition of Hamiltonian Term

We can expand our usage of this coherent method to electronic structure Hamiltonians that are "low-rangedly dominant". These are Hamiltonians that are not themselves low-range but can be expressed as a sum of a low-range term $\hat{H}_0$ and a relatively small arbitrary-range term $\Delta\hat{H}$.

$$\begin{gathered} \hat{H} = \hat{H}_0 + \Delta\hat{H} \\ ||\Delta\hat{H}|| \ll ||\hat{H}_0|| \end{gathered} \tag{3.7}$$

$e^{-i\hat{H}_0 t}$ is (asymptotically) cheaper, while $e^{-i\Delta\hat{H}t}$ is asymptotically more expensive but it affects the output to a lesser extent (since it's smaller). The idea, then, is to use a randomized Hamiltonian simulation algorithm such as qDRIFT, which samples terms more if they have a higher weight (operator norm). In simulating (3.7), the algorithm would perform $e^{-i\hat{H}_0 t}$ much more frequently (asymptotically) than $e^{-i\Delta\hat{H}t}$, which is good because the former is cheaper. If the operator norm discrepancy $||\Delta\hat{H}|| \ll ||\hat{H}_0||$ is large enough, $\Delta\hat{H}$ will not contribute too much to the overall cost, while still being performed enough times to simulate $\hat{H}$ precisely.

3.4 Gradient Descent

3.5 Gate Count Analysis

3.6 Numerics

Chapter 4

Conclusion

4.1 Directions for Further Work

Appendix A

Quantum Circuit Diagrams

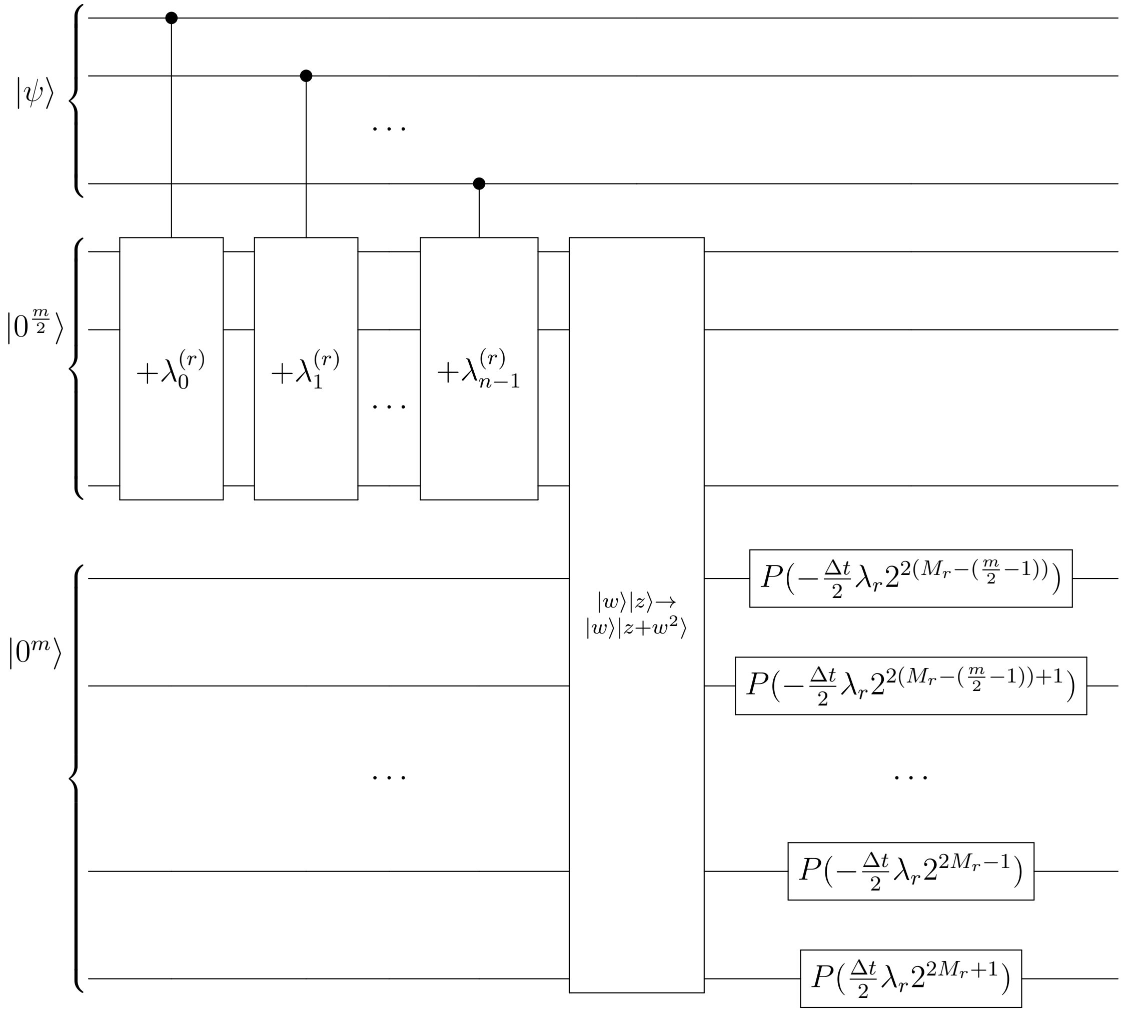
$|\psi\rangle$
$|0^{\frac{m}{2}}\rangle$
$|0^m\rangle$
$+\lambda_0^{(r)}$
$+\lambda_1^{(r)}$
$+\lambda_{n-1}^{(r)}$
$|w\rangle|z\rangle\to$
$|w\rangle|z+w^2\rangle$
$P(-\frac{\Delta t}{2}\lambda_r 2^{2(M_r-(\frac{m}{2}-1))})$
$P(-\frac{\Delta t}{2}\lambda_r 2^{2(M_r-(\frac{m}{2}-1))+1})$
$P(-\frac{\Delta t}{2}\lambda_r 2^{2M_r-1})$
$P(\frac{\Delta t}{2}\lambda_r 2^{2M_r+1})$
$\dots$

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

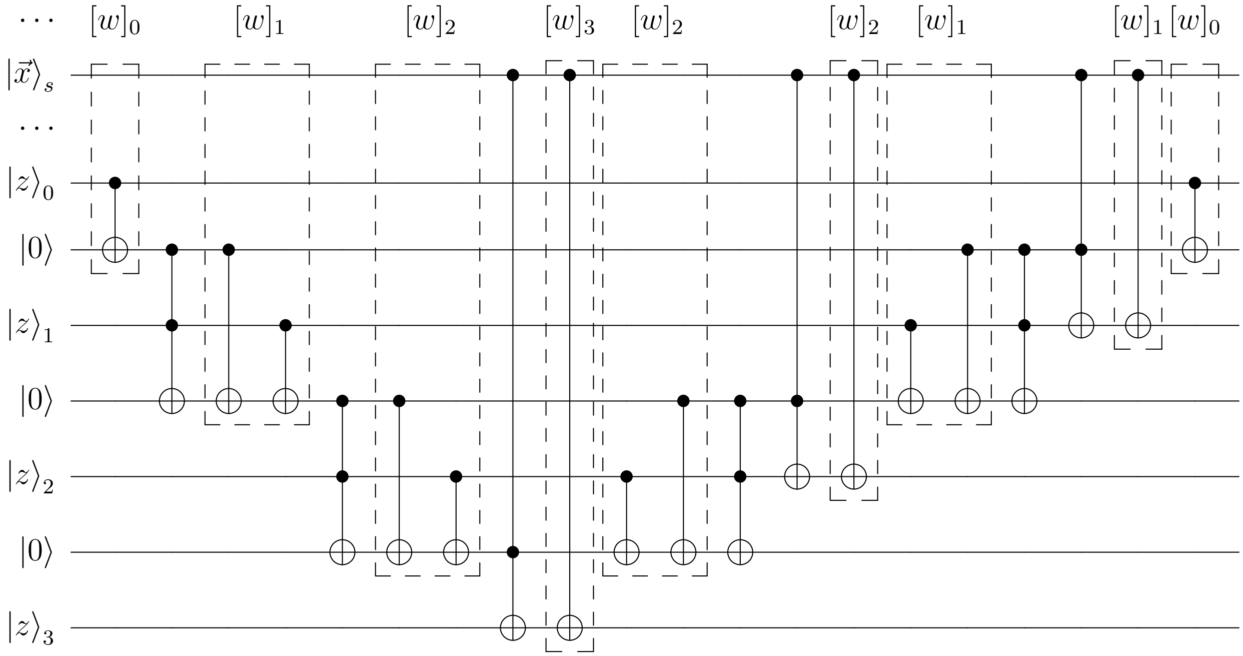
$\cdots$
$[w]_0$
$[w]_1$
$[w]_2$
$[w]_3$
$[w]_2$
$[w]_2$
$[w]_1$
$[w]_1$
$[w]_0$
$|\vec{x}\rangle_s$
$\cdots$
$|z\rangle_0$
$|0\rangle$
$|z\rangle_1$
$|0\rangle$
$|z\rangle_2$
$|0\rangle$
$|z\rangle_3$

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 $[w]_j = 1$.

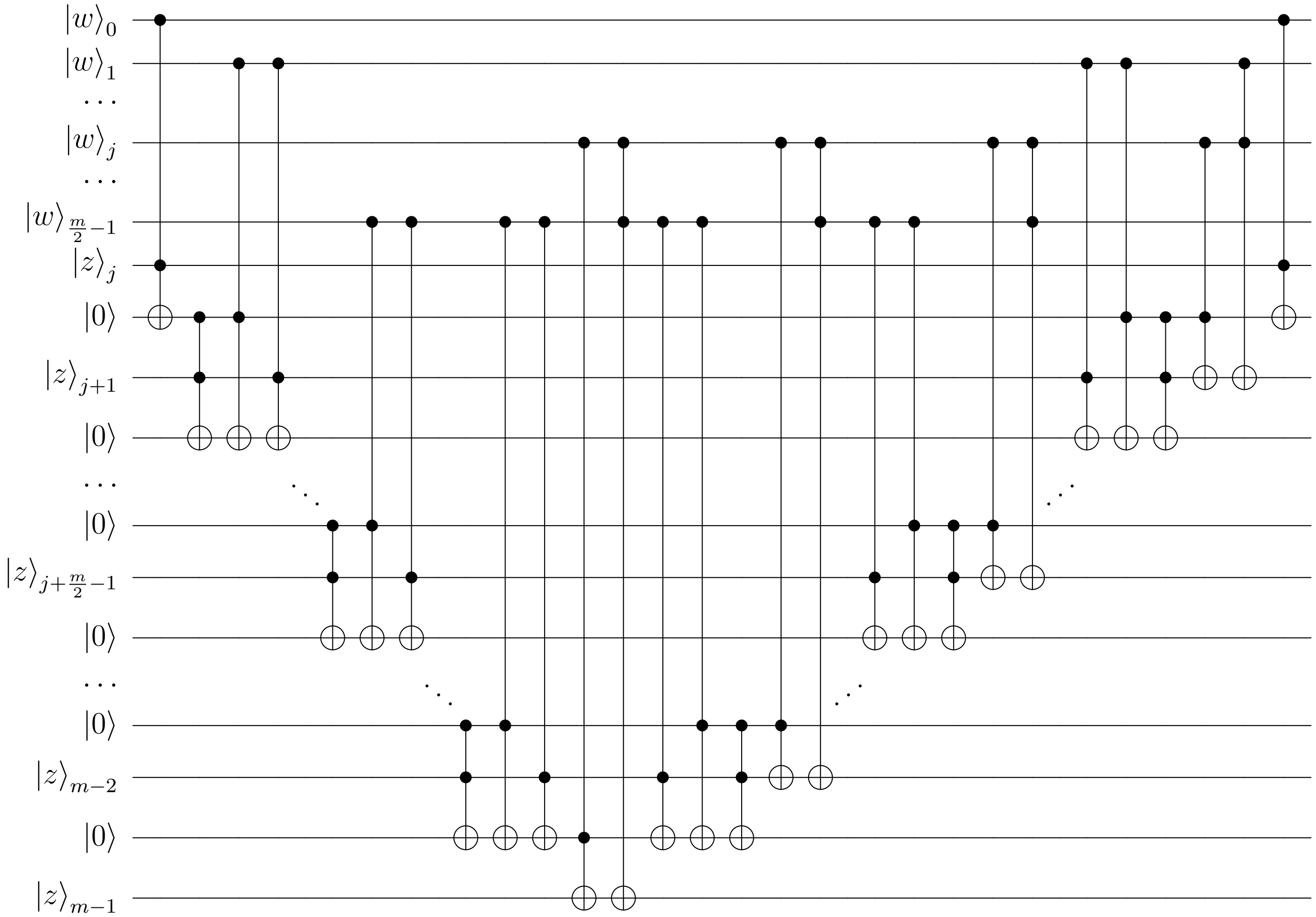
$|w\rangle_0$
$|w\rangle_1$
$\cdots$
$|w\rangle_j$
$\cdots$
$|w\rangle_{\frac{m}{2}-1}$
$|z\rangle_j$
$|0\rangle$
$|z\rangle_{j+1}$
$|0\rangle$
$\cdots$
$|0\rangle$
$|z\rangle_{j+\frac{m}{2}-1}$
$|0\rangle$
$\cdots$
$|0\rangle$
$|z\rangle_{m-2}$
$|0\rangle$
$|z\rangle_{m-1}$

Figure A.3: Quantum circuit diagram for $|w\rangle |z\rangle \rightarrow |w\rangle |z + [w]_j 2^{M_r-(\frac{m}{2}-1)+j} w\rangle$. 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] 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.

[2] 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.

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

[4] 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.