Contextuality and Sparsity in Quantum Algorithms

A dissertation

submitted by

William M. Kirby

In partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Physics

TUFTS UNIVERSITY

February 2023

ADVISOR: Peter J. Love

DOCTORAL COMMITTEE: Kenneth Brown

Timothy Atherton

Cristian Staii

Gary Goldstein

For my parents.

Contextuality and Sparsity in Quantum Algorithms

William M. Kirby

ADVISOR: Peter J. Love

In this thesis we study quantum algorithms motivated by two unifying concepts: contextuality and sparsity. Contextuality is a characteristic feature of quantum mechanics, and identifying contextuality in quantum algorithms provides a means for distinguishing them from their classical counterparts. We first describe how contextuality may be identified in variational quantum eigensolvers (VQEs), which are a leading algorithm for noisy intermediate-scale quantum computers. We then show how to construct a classical phase-space model for any noncontextual Hamiltonian, which provides a classical simulation algorithm for noncontextual VQE and allows us to prove that the NONCONTEXTUAL HAMILTONIAN PROBLEM is only NP-complete, rather than QMA-complete. Finally, we describe an approximation method called contextual subspace VQE that permits us to partition a general Hamiltonian into a noncontextual part and a contextual part, and estimate its ground state energy using a technique that combines classical simulation of the noncontextual part with quantum simulation of the contextual part. By using more quantum resources (in qubits and simulated terms of the Hamiltonian), we can increase the accuracy of the approximation. We present results of simulating contextual subspace VQE on electronic structure Hamiltonians, and find that to reach chemical accuracy in most cases it requires fewer qubits and simulated terms than standard VQE.

A Hamiltonian is sparse in some particular basis when it contains only polynomially-many nonzero entries in each row and column, meaning that each basis state only propagates into polynomially-many others at first order. Time evolution

under sparse Hamiltonians is generally efficiently simulable by quantum algorithms provided efficient unitaries (oracles) exist that identify the locations and values of the nonzero matrix elements. We present two new results on sparse Hamiltonian simulation. The first is an extension of VQE to the sparse Hamiltonian input model, which expands the domain of VQE beyond local Hamiltonians. The second contribution is an explicit construction of the sparse input oracles for a broad class of second-quantized Hamiltonians. This bridges the gap between existing simulation methods that analyze sparse Hamiltonian simulation in terms of oracle queries and the gate-level decompositions that will be required to implement these methods in practice.

Acknowledgments

The work presented in this thesis was previously published in [KL19a, KL20, KTL21, KL21, KHKL21]. This work was supported by the National Science Foundation, Grants No. DGE-1842474 and PHY-1818914. Refs. [KL19a, KL20, KL21, KHKL21] are copyrighted by the American Physical Society, and reproduced per Author's Rights under the transfer of copyright agreement.

I would like to especially thank: Peter, for being an unfailingly supportive, kind, and insightful advisor, and providing a great graduate school experience for me; the rest of my committee, Ken Brown, Tim Atherton, Cristian Staii, and Gary Goldstein, for helping me through this part of the process as well as for their support throughout my time at Tufts even before they were on my committee; my past mentors in physics research, notably Fred Strauch, Bill Wootters, Charlie Doret, and Shelby Kimmel; my wife Anna, who always keeps me grounded and happy, and makes my life make sense; my mom, dad, and brother, for the same reasons (and my mom deserves a special thank you because she homeschooled me for twelve years); the current and former members of the quantum computing group at Tufts, whose friendship and conversations have been extremely valuable to me; and finally the whole department, professors, students, and staff for everything they do.

William M. Kirby

TUFTS UNIVERSITY

February 2023

Contents

Abstract **iii**

Acknowledgments **v**

List of Tables **x**

List of Figures **xii**

Chapter 1 Introduction **2**

1.1 Historical overview 3

1.2 Building blocks of quantum computing 8

1.3 Quantum simulation 14

1.3.1 Representing physical systems on quantum computers 15

1.3.2 Simulating time evolution 24

1.3.3 Simulating ground state energies 31

1.4 Contextuality 39

1.5 Overview and summary of results 44

Chapter 2 Testing for contextuality in Pauli Hamiltonians and variational quantum eigensolvers **50**

2.1 Introduction and prior work 50

2.2 Defining contextuality 52

2.3 Necessary and sufficient conditions for contextuality 59

2.4 Quantifying contextuality 69

2.5 Evaluation of Contextuality in VQE experiments to date 73
2.6 Discussion 76

Chapter 3 Classical simulation of noncontextual Hamiltonians 77

3.1 Introduction 77
3.1.1 Local Hamiltonian Problems 78
3.1.2 Noncontextual Hamiltonians 79
3.2 Quasi-quantized model for a noncontextual Hamiltonian 81
3.2.1 Ontic states 81
3.2.2 Epistemic states 83
3.3 Classical simulation of a noncontextual Hamiltonian 92
3.4 Discussion 94
3.5 Appendix: obtaining an independent set of Pauli measurements from a commuting set 95
3.6 Appendix: More general quasi-quantized models 99

Chapter 4 Contextual subspace variational quantum eigensolver 104

4.1 Introduction 104
4.2 Noncontextual approximation and quantum correction 106
4.2.1 Hamiltonian partitioning and contextual subspace 106
4.2.2 Mapping a contextual subspace to a stabilizer subspace . . . 111
4.2.3 Restricting the Hamiltonian to a contextual subspace 116
4.3 Contextual subspace VQE 118
4.3.1 Method 118
4.3.2 Choosing noncontextual Hamiltonians 120
4.3.3 Applications 122
4.4 Conclusion 125
4.5 Appendix: Moving qubits from the noncontextual approximation to the quantum correction 127
4.6 Appendix: choosing noncontextual sub-Hamiltonians from molecular electronic-structure Hamiltonians 129

4.6.1 Types of Pauli terms in electronic structure Hamiltonians . . 129
4.6.2 Possible choices for the noncontextual sub-Hamiltonian . . . 131

Chapter 5 Variational quantum eigensolvers for sparse Hamiltonians 136

5.1 Introduction . . . 136
5.2 Sparse VQE . . . 139
5.3 Obtaining sparse decompositions . . . 142
5.4 Evolution under one-sparse unitary operators . . . 149
5.5 Conclusion . . . 151
5.6 Appendix: oracle implementations for example applications . . . 152
5.6.1 Oracles for Pauli operators . . . 152
5.6.2 Oracles for Bosonic operators . . . 157

Chapter 6 Oracles for sparse Hamiltonians in compact mapping 164

6.1 Introduction . . . 164
6.2 Main results for plane wave momentum basis . . . 166
6.2.1 Comparison to direct encoding . . . 169
6.2.2 Comparison to prior work on electronic-structure Hamiltonians 171
6.3 Compact encoding and Hamiltonians . . . 172
6.3.1 Compact encoding of Fock states . . . 172
6.3.2 Case study: light-front quantization . . . 175
6.3.3 Hamiltonian . . . 177
6.4 Sparse Hamiltonians . . . 180
6.4.1 Sparse Hamiltonian Simulation for Compact-Encoded Fock States . . . 181
6.4.2 Sparsity of general interactions in the Fock basis . . . 188
6.5 Location oracle . . . 189
6.5.1 Examples . . . 190
6.5.2 General method . . . 198
6.5.3 Analysis . . . 201
6.6 Matrix element (value) oracle . . . 203

6.7 Example applications 207
6.7.1 Free boson and fermion theory 208
6.7.2 $\lambda\phi^4$ theory 210
6.7.3 Massive Yukawa model 213
6.8 Beyond the plane wave momentum basis 218
6.9 Conclusion 220

Chapter 7 Conclusion **221**

List of Tables

2.1 Evaluation of contextuality in VQE experiments. CD_0 is the minimum number of terms we must remove from the Hamiltonian to reach a noncontextual set, as a fraction of the total number of terms ($|\mathcal{S}|$). In [DMH$^+$18], Dumitrescu *et al.* compute the ground-state energies of the deuteron for effective field theories with dimension 2^N for $N = 1, 2, 3$ ($|\mathcal{S}| = 1, 4, 3$), and extrapolate from these to the infinite-dimensional space. Thus, $\mathcal{S}$ is different for each value of N. For $N = 1$, $\mathcal{S} = \{Z\}$. For $N = 2$, $\mathcal{S} = \{ZI, IZ, XX, YY\}$. For $N = 3$, $\mathcal{S} = \{IZ, XX, YY\}$. All of these are noncontextual. 74

2.2 Details of VQE experiments tested for contextuality in Table 2.1. $\underline{\mathcal{S}}$ is a set sufficient to generate the noncontextual part of the Hamiltonian via inference trees (there is not space to represent all of the terms in the noncontextual parts of the Hamiltonians in this table). For brevity, we represent some operators in this table as Pauli operators with subscripts: the subscript indicates which qubit the operator acts on. So for example, X_2X_3 means IXX (in a three-qubit system.) "Error" refers to the error achieved in the estimated ground state resulting from the VQE experiments, which is either represented as a fidelity of the ground state itself or in terms of error in the ground state energy either in Hartree or in percent (the errors are all simply those reported in the various papers). Note that an error of 1.6 mHa is chemical accuracy. 75

4.1 Types of electronic structure Hamiltonian terms, showing their fermionic forms and the resulting one or more Pauli terms. All indices i, j, k, l are assumed to be distinct: otherwise the distinct types can collapse into each other. In the last line, all products are tensor products, and each $Q^{(X,Y)}$ is either X or Y such that an even number are X (so an even number are Y as well). 129

6.1 Glossary of parameter definitions. 169

6.2 Summary of qubit and gate count costs for our algorithm, for a second-quantized Hamiltonian H in the plane wave momentum basis. The parameters are defined in Table 6.1. 169

List of Figures

1.1 This figure shows a parameterized quantum circuit that is expressive enough to generate any two-qubit state with real amplitudes. 34

1.2 Schematic of the VQE algorithm. All of the green nodes correspond to classical components of the algorithm, while the single blue node corresponds to the quantum subroutine. 35

1.3 The Peres-Mermin square, an illustration of contextuality. The observables in the square are two-qubit Pauli operators (by convention the tensor product symbols are suppressed, so for example $IZ \coloneqq I \otimes Z$). The numbers at the ends of the rows and columns are the products of the corresponding rows or columns. In a noncontextual hidden-variable model based on simultaneous assignments of values (± 1) to the observables in the square, those value assignments would have to satisfy the row and column products, since each row and column commutes and is therefore simultaneously measurable. However, in order to satisfy all of the row products, the value assignments would have to contain an even number of -1s, while to satisfy the column products they would have to contain an odd number of -1s, so a noncontextual hidden-variable model necessarily violates one or the other. 41

2.1 Inference trees for $\pm YY$ over $\{XI, IX, ZI, IZ\}$ 54

2.2 Example of Corollary 2.4.1: steps to construct a contextual inference tree for $-XI$, beginning with the Peres-Mermin square trees in Fig. 2.1. Note that $-XI$ is the root of T_2, the final tree, and $\{XI\}$ is the determining set, since every other leaf appears twice. 58
2.3 Compatibility graphs for contextual sets of four Pauli operators. . . 60
2.4 Sequence of trees in Lemma 2.3, demonstrating that since F commutes universally, it may be moved up to become a child of the root. 61
2.5 Sequence of trees in Lemma 2.4. In the first tree, the P_i are parents of pairs of leaves, and τ_1, τ_2 are subtrees that are remnants of the original children of R. In the second tree, the C_i are parents of leaves from within the same commuting clique, with one C_i for each clique. In the third tree, the τ_i have been absorbed into the C_i. 63
2.6 First compatibility graph in Fig. 2.3, together with its contextual inference tree. Note that each leaf in the tree appears twice, so the determining set is empty. 65
2.7 Second compatibility graph in Fig. 2.3, together with its contextual inference tree. Note that each leaf in the tree appears twice, so the determining set is empty. 66
2.8 Third compatibility graph in Fig. 2.3, together with its contextual inference tree. Note that each leaf in the tree appears twice, so the determining set is empty. 67

4.1 CS-VQE approximation errors versus number of qubits used on the quantum computer, for tapered molecular Hamiltonians. All Hamiltonians whose curves overlap in the region below chemical accuracy have the same total numbers of qubits. The solid black lines indicate chemical accuracy. Within each subplot, the ordering of the legend matches the vertical ordering of the leftmost points in the curves. . 122

4.2 Number of terms simulated on the quantum processor required to reach chemical accuracy using CS-VQE versus using full VQE. The dashed line marks equality. All points represent either one, two, or three Hamiltonians. 123

5.1 Hadamard test circuit realizing estimation of the real part of the matrix element $\langle 0|V^\dagger G_j U|0\rangle$. 140

5.2 Circuit to implement the O_L oracle as given by (5.46), for the Pauli $P(a,b)$ and input $|x, y=0\rangle$. x_k is the kth bit in x, y_k is the kth bit in y, and a_k is the kth bit in a. 154

5.3 Circuit to implement the O_V oracle as given by (5.49), for the Pauli $P(a,b)$ and input $|x\rangle|z\rangle$. x_k is the kth bit in x, b_k is the kth bit in b, and $|z\rangle$ is a single qubit. We do not show the register encoding $|x \oplus a\rangle$ as in (5.49) because the action of the oracle does not depend on it. . 154

5.4 Circuit to implement $|w\rangle|w'=0\rangle \mapsto |w\rangle|w'=w\oplus v\rangle$, for a classical binary number v and a binary number w stored in qubits. w represents the occupation of a mode in the input state x; w_{N-1} is the most significant bit and w_0 is the least. w' represents the occupation of the corresponding mode in the output state, which is initially set to $|0\rangle$. The circuit shown is for five bits in w and w'; to generalize to N bits, one should extend the 'adds' column and continue to cascade the 'carry' blocks as in the example above. If there is a carry from the most significant bit, it means that w' has reached or exceeded the occupation cutoff. Therefore, we flip the 'mult' qubit from $|1\rangle$ to $|0\rangle$ if there is a carry from the most significant bit: the matrix element will later be multiplied by the value of the 'mult' qubit, so this sets the matrix element to zero when the occupation reaches the cutoff. . 158

5.5 Circuit to implement $|w\rangle|w' = 0\rangle \mapsto |w\rangle|w' = w \ominus v\rangle$, for a classical binary number v and a binary number w stored in qubits. w represents the occupation of a mode in the input state x; w_{N-1} is the most significant bit and w_0 is the least. w' represents the occupation of the corresponding mode in the output state, which is initially set to $|0\rangle$. The circuit shown is for five bits in w and w'; to generalize to N bits, one should extend the 'adds' column and continue to cascade the 'carry' blocks as in the example above. If there is a carry from the most significant bit, it means that w' has reached or exceeded the occupation cutoff. Therefore, we flip the 'mult' qubit from $|1\rangle$ to $|0\rangle$ if there is a carry from the most significant bit: the matrix element will later be multiplied by the value of the 'mult' qubit, so this sets the matrix element to zero when the occupation reaches the cutoff. . 159

6.1 Diagram of the example 3-point interaction (6.31), which describes annihilation of a pair of particles of type '1' to form a single particle of type '2' (we read time from left to right). 179

6.2 Schematic of the circuit to implement the location oracle. Labeled wires are input and output registers, and unlabeled wires are ancilla registers (each initially in the all 0s state). The inputs and outputs are $|\mathcal{F}\rangle$ (the incoming Fock state), $|i\rangle$ (the sparsity index), and $|\mathcal{F}'_i\rangle$ (the outgoing Fock state), all as in (6.44). The intermediate quantities computed are J_i (the list of modes in $|\mathcal{F}\rangle$ from which the incoming particles are taken, as in (6.77)), $\mathbf{Q}$ (the total incoming and outgoing momentum), and $A(\mathbf{Q}, i)$ (the list of momenta of the outgoing particles, as in (6.78)). The circuit hides some additional ancillas, does not show uncomputation of the ancillas, and shows the action when i is a valid index for an outgoing state. At the points marked by red dashed lines we have to check whether this is true, as described in detail in the text. 199

6.3 Gate counts (in log-local operations) to implement oracles for $\lambda\phi^4$ theory in 1+1D. The equal-time cutoffs are $[-\Lambda, \Lambda]$, for Λ defined in terms of K by (6.112). The exact gate counts for light-front and equal-time quantization are given by the points and crosses, respectively. The solid line K^3 and dashed line $K^4/5$ are included to illustrate that the datapoints are indeed converging to their expected asymptotic scalings of $O(K^3)$ for light-front and $O(K^4)$ for equal-time. Python code to generate the data for this plot is available on GitHub at `https://github.com/wmkirby1/quspice/blob/main/Plots_for_2105.10941.ipynb`. 214

6.4 Gate counts (in log-local operations) to implement oracles for the massive Yukawa model. The equal-time cutoffs are $[-\Lambda, \Lambda]$, for Λ defined in terms of K by (6.112). The exact gate counts for light-front and equal-time quantization are given by the points and crosses, respectively. The line K^3 is included to illustrate that the datapoints are indeed converging to their expected asymptotic scaling of $O(K^3)$. Python code to generate the data for this plot is available on GitHub at `https://github.com/wmkirby1/quspice/blob/main/Plots_for_2105.10941.ipynb`. 215

Contextuality and Sparsity in Quantum Algorithms

Chapter 1

Introduction

This thesis is all about quantum computing, and more specifically, quantum simulation of physical systems. The introduction will review and introduce the material necessary to understand the technical work that appears in later chapters, and also provide context for that work. The main criterion for inclusion as well as organizing principle for the papers and topics I will cover in the introduction will be "what has been useful to me?" or "what is useful in contextualizing the technical work to appear after?"

The introduction will be organized as follows. Section 1.1 will give a high-level historical overview of quantum computing. Section 1.2 will introduce the basic building blocks in the theory of quantum computation. Section 1.3 will summarize the goals of quantum simulation, and provide literature review and discussion of the main techniques. Section 1.4 will introduce the concept of quantum contextuality and discuss its role in quantum computing, broadly, as well as providing literature review on that topic. Finally, Section 1.5 will summarize the structure of the remainder of the thesis and the main results to be presented, as well as briefly reviewing papers I have contributed to that are not included in this thesis.

1.1 Historical overview

Quantum computing and quantum simulation were first suggested by Richard Feynman in his 1981 lecture "Simulating Physics with Computers" [Fey82], which concluded with the following oft-quoted sentence:

> I'm not happy with all the analyses that go with just the classical theory, because nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.

The development of the field so far has made his last clause appear particularly prescient, but both quantum computing hardware and theory have been advancing steadily for the last several decades. At the time of this writing we appear to be close to finally realizing *quantum advantage* for useful tasks, meaning in this case a quantum simulation with dramatically lower resource requirements than the equivalent classical simulation.

After Feynman's lecture, a few key early results laid the groundwork for the field. In 1992, David Deutsch and Richard Jozsa presented a quantum algorithm for determining whether a boolean function on n bits is constant (all values are the same) or balanced (equal numbers of its values are zero and one), given the promise that it is one or the other [DJ92]. The function is given as an oracle, and the Deutsch-Jozsa algorithm succeeds deterministically in a constant number of queries to the oracle, while any deterministic classical algorithm would have to query the oracle for every possible input to the function, i.e., $O(2^n)$ times. This algorithm generalized Deutsch's earlier work [DP85], which performed the same task for boolean functions that take one bit as input, but this cannot offer exponential advantage over classical computing relative to the oracle since there are only two possible inputs.

Next in the lineage of early results came Shor's 1994 algorithm [Sho94] for factoring integers, which requires

$$O(n^2 \log(n) \log\log(n)) \tag{1.1}$$

operations to factor an n-bit integer. The best known classical algorithm for the same task is generalized number-field sieve, which requires

$$O\left(e^{n^{1/3}\log^{2/3}(n)}\right) \tag{1.2}$$

operations [LL93]. While this is subexponential rather than properly exponential, it is still much worse than any polynomial runtime, so it is considered strictly inefficient. Shor's algorithm is usually credited with bringing quantum computing into the mainstream, since unlike the Deutsch-Jozsa algorithm it solves a task of broad interest instead of one contrived for the purpose of demonstrating quantum advantage.

Another important quantum algorithm discovered in the mid-90s was Grover's search algorithm [Gro96], which offers a quadratic speed-up over classical computing in unstructured search, i.e., finding an element satisfying some efficiently-testable property in an unordered list. The provably optimal scaling of any classical algorithm for this task is $\Theta(N)$ where N is the length of the list, i.e., no classical algorithm can scale asymptotically better than the algorithm that simply looks at items in the list one after another until the one satisfying the property is found, which requires looking at $N/2$ elements on average and $N-1$ in the worst case [Gro96]. However, Grover's search requires only $O(\sqrt{N})$ operations to accomplish the same task. Since any NP-complete problem can be solved by an unstructured search over the possible solutions, this means that Grover's search offers a quadratic speed-up for NP-complete problems, although this cost is still exponential (since $O(\sqrt{2^n}) = O(2^{n/2})$).

Also in 1996, Seth Lloyd published "Universal Quantum Simulators" [Llo96], which of the algorithms mentioned so far is the most relevant to this thesis, since it finally realized Feynman's challenge of efficiently simulating quantum physics using a quantum computer. Lloyd demonstrated that time evolution under any Hamiltonian with local interactions can be simulated efficiently on a quantum computer using what we would now refer to as *Trotterization* (see Section 1.3.2). This estab-

lished that quantum computers can be used to simulate quantum dynamics, which continues to be one of the two main goals in the field of quantum simulation.

Shortly before this, Alexei Kitaev had developed the first variant of the phase estimation algorithm, which can be used estimate the energy of an energy eigenstate of some Hamiltonian given the capacity to simulate time-evolution under that Hamiltonian, or can be used to project onto an approximate energy eigenstate if the input state is a superposition of energy eigenstates [Kit95]. In combination with Lloyd's work on quantum simulation of time-evolution in [Llo96], phase estimation makes it possible to simulate energies of Hamiltonians, i.e. statics, which is the other main goal of quantum simulation.

In 2002, Kitaev, Shen, and Vyalyi published their seminal textbook "Classical and Quantum Computation" [KSV02], which in fact presented a number of new results, including refinements of phase estimation and the proof that QMA-complete problems exist, although the latter result had been presented by Kitaev in some earlier lectures. QMA is the quantum analog of NP — it is the class of problems for which a prover with access to a quantum computer can efficiently verify a solution in the form of a quantum state — so the proof of QMA-completeness is the quantum analog of the celebrated Cook-Levin theorem [Coo71, Lev73] in classical complexity theory, which established the existence of NP-complete problems.

The 2000s yielded an explosion of quantum algorithm development, so at this point one ceases to cite all new results, but a few highlights are worth mentioning. In [ATS03], Aharonov *et al.* defined the *sparse Hamiltonian input model* and gave the first efficient quantum algorithm for its simulation. Sparse Hamiltonians are one of the main topics of this thesis and are introduced thoroughly in Section 1.3.1, so we leave the details for later sections, but they are a much broader class of Hamiltonians than the local Hamiltonians that were previously know to be simulable using Lloyd's method [Llo96]. Around the same time, Childs *et al.* showed that exponential quantum advantage could be obtained with respect to an oracle in a similar setting [CCD$^+$03]: this result is significant because it was the first instance of provable quantum advantage in a simulation algorithm, albeit with respect to an

oracle.

In [FGGS00], Farhi *et al.* proposed the first quantum algorithm for solving a classical optimization problem (in this case boolean satisfiability), and for this purpose introduced the idea of adiabatic quantum computing, which has subsequently been applied in many other quantum algorithms. Their idea was to map a SAT problem to a classical, diagonal Hamiltonian H_1, then attempt to simulate its ground state by starting in an initial Hamiltonian H_0 whose ground state is the uniform superposition, and then adiabatically evolving from H_0 to H_1 (by letting the time-dependent Hamiltonian vary as $H(s) = (1-s)H_0 + sH_1$ for rescaled time $s \in [0,1]$) in order to correspondingly evolve the ground state of H_0 into the ground state of the H_1. They showed using the adiabatic theorem that the actual time required for the adiabatic evolution to succeed in mapping the initial ground state to the target final ground state scales as $O(1/\Delta_{\min}^2)$ where $\Delta_{\min}$ is the minimum spectral gap of the interpolating Hamiltonian $H(s)$ along the path s. This established a concrete (though difficult to evaluate) criterion for conditions where it is possible to well-approximate a ground state using an efficient quantum algortihm.

Later, Harrow, Hasidim, and Lloyd developed an algorithm that became known as HHL for solving linear systems of equations, i.e., inverting matrices [HHL09]. For this application, the best classical algorithm has cost $O(N\kappa)$ for an $N \times N$ matrix with condition number κ. The HHL algorithm reduces this to $O(\log(N)\kappa^2)$. Hence, for a matrix with exponential dimension, i.e., one that possesses a classical description of size n but whose dimension is $N = O(\exp(n))$, HHL provides an exponential speed-up provided the condition number does not also scale exponentially with n (in which case neither HHL nor classical algorithms are efficient) and provided the classical description can be translated into an efficient input model.

The final result we will cite from the 2000s is Berry *et al.*'s 2007 paper [BACS07] on simulating sparse Hamiltonians, which contained two major results. The first was a much improved error bound on Hamiltonian simulation by Trotterization, compared to Lloyd's original derivation. The new bound eliminated the assumption that low-order terms in the Trotter expansion dominate the error (which is intuitive but

is now known to fail under certain circumstances [WBC+14, CS19]) and reduced the circuit depth to nearly linear in the length of the time-evolution (the circuit depth using this method is $O\left(t^{1+\frac{1}{2k}}\right)$ for any positive integer k, although later papers reduced this scaling to truly linear — see Section 1.3.2). The second major result was a much improved overall scaling for simulation of sparse Hamiltonians, in particular bringing the dependence on number n of qubits from $O(n^9)$ as in the original paper of Aharonov *et al.* [ATS03] down to $O(\log^*(n))$, which is for all practical purposes constant ($\log^*(n) \leq 5$ for all $n \leq 2^{65536}$). This suggested (correctly, as demonstrated by later work — see Section 1.3.2) that explicit dependence on the number of qubits could be eliminated from the cost of sparse Hamiltonian simulation entirely, leaving as the key parameters sparsity, evolution time, and error.

In the 2010s many of the algorithms we have mentioned so far were further refined, and we will discuss much of the relevant literature in Sections 1.3.2 and 1.3.3. However, for now we will mention one more entirely distinct family of quantum algorithms that will be central to much of this thesis: variational quantum algorithms. These come in two main varieties: the one we will mainly focus on is the *variational quantum eigensolver* or VQE, first proposed and tested in [PMS+14], which is a method for simulating ground state energies of quantum Hamiltonians. This algorithm differs from much of the prior work in two ways: first, it is specifically designed to be well suited for *pre-fault-tolerant* or *NISQ* quantum computers (*NISQ* stands for *noisy, intermediate-scale quantum*) [Pre18]. Second, it is a heuristic, meaning it has no provable guarantee of success (i.e., of convergence to some desired accuracy). It is based on variationally minimizing the expected energy of an *ansatz* (guess) for the ground state, relying on Rayleigh-Ritz variational principle to guarantee that the expected energy will never cross below the ground state energy, so that the accuracy can only improve as the energy decreases. We will discuss VQE in more detail in Section 1.3.3.

The other main type of variational quantum algorithm is the *quantum approximate optimization algorithm* or QAOA, which was introduced by Farhi *et al.* in [FGG14] as a variational alternative to their earlier adiabatic method for solv-

ing classical optimization problems on quantum computers [FGGS00]. This original version of QAOA may be thought of as replacing the adiabatic evolution with a Trotterized time-evolution (see Section 1.3.2), then allowing the Trotter step evolution times to be varied by a classical optimizer. While a number of recent obstructions [CLSS21, BGMZ22, JMSS22, AM22] to the performance of QAOA have raised doubts about whether it can outperform classical heuristics, this remains a rich area of research, including in the search for an ansatz that can yield reliable performance.

1.2 Building blocks of quantum computing

The basic building blocks of a quantum algorithm are states, transformations, and measurements. The Hilbert space of a single quantum bit or *qubit* is spanned by two states called the *logical states* (or *computational basis states*), which are conventionally labeled $|0\rangle$ and $|1\rangle$ (length-1 bitstrings). The logical states of an n-qubit quantum computer are labeled by the 2^n length-n bitstrings, which are equivalent to all possible tensor products of the single-qubit logical states:

$$|00...00\rangle, \quad |00...01\rangle, \quad |00...10\rangle, \quad |00...11\rangle, ... \quad |11...11\rangle. \tag{1.3}$$

The full Hilbert space of n qubits is all possible superpositions of the logical states, which we can write as

$$\alpha_{00...00}|00...00\rangle + \alpha_{00...01}|00...01\rangle + \alpha_{00...10}|00...10\rangle + \cdots + \alpha_{11...11}|11...11\rangle = \sum_b \alpha_b |b\rangle, \tag{1.4}$$

where right-hand side is a compact notation in which the variable b in the sum runs over all length-n bitstrings. The α_b's are complex amplitudes such that $\sum_b |\alpha_b|^2 = 1$ (the property is called *normalization*) [NC01].

Unentangled (product) states of composite systems are the tensor products of states of the subsystems, and we will often omit the tensor product symbol when doing so introduces no ambiguity. For example, if the states of two subsystems are $|\psi_1\rangle$ and $|\psi_2\rangle$, then the state of the composite system may be written in any of the

following forms:

$$|\psi_1, \psi_2\rangle \equiv |\psi_1\rangle|\psi_2\rangle \equiv |\psi_1\rangle \otimes |\psi_2\rangle. \tag{1.5}$$

States may also be represented as row or column vectors with row vectors corresponding to bras and column vectors corresponding to kets. We will always use binary ordering ("physics ordering") for the entries in statevectors, i.e.,

$$\alpha_{00\ldots00}|00\ldots00\rangle + \alpha_{00\ldots01}|00\ldots01\rangle + \alpha_{00\ldots10}|00\ldots10\rangle + \cdots + \alpha_{11\ldots11}|11\ldots11\rangle \leftrightarrow \begin{pmatrix} \alpha_{00\ldots00} \\ \alpha_{00\ldots01} \\ \alpha_{00\ldots10} \\ \alpha_{00\ldots11} \\ \vdots \\ \alpha_{11\ldots11} \end{pmatrix}. \tag{1.6}$$

Measurements are assumed to be performed in the logical basis, i.e., to be projective measurements onto the logical basis states (1.3), with probabilities given by the Born rule. This means that the logical basis is chosen by convention to correspond to whatever physical measurement can be readily performed on the physical qubits. Further details of quantum state and measurement formalism are covered in introductory quantum mechanics courses; see for example [SN20], or for a more quantum computing oriented approach [NC01].

The time evolution of a quantum state can be represented as multiplication on the left by a unitary operator, which when acting on an n-qubit state represented as a vector is a $2^n \times 2^n$ matrix U such that $U^\dagger = U^{-1}$. Equivalently, a unitary operator may be defined by its action on any complete orthonormal basis of the Hilbert space, which will sometimes be more convenient. Unitary operators acting on subsystems may be combined via the tensor product to form unitary operators acting on composite systems [NC01].

In quantum computing, therefore, every transformation of the state of the qubits is described by a unitary operation. Strictly speaking, if the quantum computer is subject to unwanted coupling to an environment, it could be that only

the full evolution of computer plus environment is unitary, leading to nonunitary evolution of the computer subsystem alone. This formalism of obtaining a nonunitary evolution (which is mathematically represented by a completely positive, trace-preserving map) by tensoring in an auxiliary "environment" system, implementing a unitary operation on the combined system, and then tracing out the environment, is called *Stinespring dilation*, or more affectionately, "the church of the larger Hilbert space" [NC01, GL00]. In this thesis, however, we will mostly ignore such considerations and treat quantum computations as based on unitary time-evolutions of the qubits alone. A quantum circuit refers to a complete description of the transformation of the quantum computer from initialization in some easy-to-prepare state, through time evolution, to measurement.

Typically, quantum circuits are expressed as sequences of unitary operations on subsets of the qubits, which are often referred to as *gates* in analog to classical computation. Both the gates and the full time evolution given by their product are unitary operators. Depending on the specific type of quantum hardware used, the gates that can be physically performed can vary; we call these *native gates*. In this thesis, we will focus on algorithms to be performed on *universal quantum computers*, which are quantum computers whose native gatesets are *universal*, meaning that any arbitrary unitary on the whole register of qubits can be approximated by a sufficiently long sequence of the native gates [NC01].

However, the unitaries that can be efficiently implemented on a quantum computer are only those that can be expressed as efficient quantum circuits (i.e., circuits containing only poly(n) native gates for an n-qubit quantum computer), which is a strict subset of the complete set of unitaries. In fact, even the set of unitaries that can be well approximated by efficient quantum circuits is a strict subset of the complete set of unitaries [NC01]. A central task in designing a quantum algorithm is to express (or approximate) the desired unitary transformation of the whole quantum computer as an efficient circuit, written in terms of some gateset. This gateset need not necessarily be the native gateset for some specific hardware, but it should be translatable (or to use the proper term, *transpilable*) into any de-

sired native gateset with at worst poly(n) overhead. This allows us to abstract away from specific hardware, and design algorithms that can be run on any hardware. In this thesis, typical gatesets we will use are composed of one- and two-qubit gates, often parameterized, which can be transpiled onto all standard quantum computing hardware with only constant overhead (this overhead can become linear if the algorithm requires interactions between arbitrary pairs of qubits but the hardware only permits interactions on a specific graph).

Let us now consider some examples of typical gates and quantum circuits. Many of the basic gates are defined in terms of Pauli operators, so we should first define these: a Pauli operator is a tensor product of the usual single-qubit Pauli matrices

$$\begin{aligned} I \coloneqq \sigma_0 \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X \coloneqq \sigma_1 \coloneqq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \\ Y \coloneqq \sigma_2 \coloneqq \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z \coloneqq \sigma_3 \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \end{aligned} \tag{1.7}$$

In writing tensor products of these, to form Pauli operators on more than one qubit, we will usually omit the tensor product symbol, so for example

$$IX \coloneqq I \otimes X, \quad YZ \coloneqq Y \otimes Z, \quad XX \coloneqq X \otimes X \tag{1.8}$$

are all two-qubit Pauli operators.

Some of the basic single-qubit gates we will often see are

$$\begin{aligned} \text{NOT} \coloneqq X \coloneqq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad H \coloneqq \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \\ S \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad T \coloneqq \sqrt{S} \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & \frac{1+i}{\sqrt{2}} \end{pmatrix}, \end{aligned} \tag{1.9}$$

which are called (with varying degrees of creativity) the *NOT* gate, *Hadamard* gate,

phase gate, and T gate, respectively. These are all discrete gates, meaning they have no variable parameters, and they form a universal set for single-qubit unitaries, meaning that any arbitrary single-qubit unitary operation can be approximated arbitrarily well by a sufficiently long sequence of the above gates (the sequence length required to achieve error ϵ scales as $\text{polylog}(1/\epsilon)$ [Kit97, KSV02, NC01, DN05]).

Any universal single-qubit gateset can be promoted to a universal set for an arbitrary number of qubits if it is augmented by an appropriate *entangling gate*, meaning a two-qubit gate that can be performed on any pair of qubits and creates a correlation between their states [NC01]. The most commonly used discrete entangling gate is the controlled-NOT or CNOT gate, defined by

$$\text{CNOT} \coloneqq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1.10}$$

To see why this is called a controlled-NOT, consider the action of CNOT on the logical basis states for two qubits:

$$\begin{aligned} \text{CNOT}|00\rangle &= |00\rangle, \\ \text{CNOT}|01\rangle &= |01\rangle, \\ \text{CNOT}|10\rangle &= |11\rangle, \\ \text{CNOT}|11\rangle &= |10\rangle, \end{aligned} \tag{1.11}$$

i.e., CNOT flips the state of the second qubit (the "target" qubit) conditioned on the state of the first qubit (the "control" qubit). Throughout this thesis we will see many more examples of controlled gates, which follow this same basic principle of applying a gate (in this case NOT) to one or more qubits controlled on the state of some other qubit or qubits.

The last category of gate we will introduce is the *parameterized gate*, which is a gate whose action can be varied continuously by adjusting some parameter.

The most basic examples of these are parameterized gates generated by a Pauli operator, meaning the time evolutions generated by treating a single Pauli operator as a Hamiltonian, e.g.,

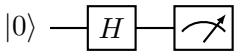
$$e^{i\theta X} = \begin{pmatrix} \cos\theta & i\sin\theta \\ i\sin\theta & \cos\theta \end{pmatrix}, \quad e^{i\theta YZ}, \quad e^{i\theta ZXY}, \tag{1.12}$$

to give one-, two-, and three-qubit examples (the two- and three-qubit examples could also be written out as unitary matrices, but for the sake of space we have not done so). In each case θ is a parameter set by the experimenter. We can also implement controlled versions of parameterized gates, e.g.,

$$\text{C-}e^{i\theta X} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\theta & i\sin\theta \\ 0 & 0 & i\sin\theta & \cos\theta \end{pmatrix}, \tag{1.13}$$

which closely resembles a CNOT for $\theta = \pi/2$ (it is equal to a CNOT up to a phase gate on the control qubit).

Assembling a quantum circuit then simply means writing down the sequence of basic gates to be performed [CCY93, NC01]. Often this is done by writing a *circuit diagram*, in which wires correspond to qubits, gates are written as boxes superimposed on the qubit wires they act on, and time runs from left to right. For example, the single-qubit circuit diagram

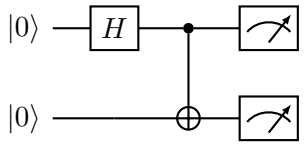


(1.14)

means "prepare the qubit in state $|0\rangle$, apply Hadamard gate, and measure (in the logical basis)." We can check what this circuit does: applying the Hadamard gate defined in (1.9) to $|0\rangle$ produces the equal superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, so by Born's rule, when we then measure we will obtain either outcome with equal probability.

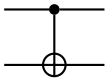
Hence this first example of a quantum circuit uses one qubit to produce a uniformly random bit.

Let us see another example:



. (1.15)

This circuit introduces a symbol we have not yet seen:



(1.16)

means CNOT controlled on the first qubit and targeting the second qubit. Hence the circuit in (1.15) first prepares an equal superposition on the first qubit, i.e., the state both qubits is

$$\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \tag{1.17}$$

immediately after the Hadamard gate, and then the CNOT entangles the two qubits, producing the state

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{1.18}$$

This is a *Bell state* and is one of the simplest examples of an entangled state [NC01], since upon measurement we will obtain either $|0\rangle$ for both qubits or $|1\rangle$ for both qubits, with equal probability, but prior to measurement nothing in the world could determine which of the pairs of outcomes it will be (more on that last claim in Chapter 2). With these examples out of the way, we are now ready to properly introduce one of the main topics of this thesis: quantum simulation of quantum systems.

1.3 Quantum simulation

Most quantum algorithms for simulation of quantum systems fall into one of two categories: simulation of time evolution (dynamics) or approximation of spectra

(statics), usually meaning ground or low-lying energies. Each possesses independent methodologies and bodies of literature. Many algorithms incorporate both, especially in cases of ground state algorithms that use time evolution as a subroutine. Many other algorithms only do one or the other: for example, algorithms for approximating cross-sections by simulating time evolution (e.g., [JLP12]), or variational quantum eigensolvers for approximating ground state energies (e.g., [PMS$^+$14]). This thesis will primarily address the problem of approximating ground state energies, although Chapter 6 is equally relevant to simulating time evolution. In this Section, I will review the literature on both variants of quantum simulation, since they provide the context for the work herein.

1.3.1 Representing physical systems on quantum computers

A prerequisite to simulating a physical system on a quantum computer is encoding the states of the system in qubit states and encoding the operators of the system in qubit operators. Often this involves one or more layers of approximation in addition to one or more encoding steps. There are also often multiple choices of representation on the qubit operator side, which permit different simulation methods. We begin by introducing three main options for this latter choice, which are referred to as *input models* and are characterized by the encoding of the Hamiltonian H: linear combinations of Pauli operators, linear combinations of unitary operators, and sparse matrices.

In each input model, we require that H is defined by a polynomial amount of classical information, since this is required in order to efficiently define the physical system. Note that this does not mean that it is possible to efficiently write down H as a matrix classically, since this will in general have exponential dimension in the system size. Instead, for example, it means that if our input physical Hamiltonian is expressed in second-quantization, it can contain only polynomially-many second-quantized terms, since each of these is classically defined by its coefficient and which modes it acts upon.

The first choice of qubit encoding, linear combinations of Pauli operators

or LCP, is the simplest but most narrowly applicable (subject to the requirement that it be efficient), and in near-term quantum computing simulations it is used almost exclusively, for reasons we will discuss. This input model predates quantum computing since spin Hamiltonians of this form are studied in condensed matter physics, but some representative early references within the quantum computing literature are [Llo96, KSV02, KR03, KKR06, PMS$^+$14]. In this input model, H is expressed as a linear combination of poly(n) Pauli operators (defined in Section 1.2), i.e., the number of Pauli operators with nonzero coefficients is a polynomial in the number of qubits n. Such a Hamiltonian can be written as

$$H = \sum_{i=1}^{N} h_i P_i, \tag{1.19}$$

where $N = \text{poly}(n)$ is the number of terms, P_i are Pauli operators, and h_i are real coefficients. For a given physical model, there can be more than one representation of the physical Hamiltonian as an LCP; we will discuss this further below. In this input model, the classical information used to define H is two lists: a list of coefficients, and a list of Pauli operators, each of which is defined as the tensor product of n Pauli matrices (which could be classically represented as, for example, a string of length n composed of the characters I, X, Y, and Z).

A Hamiltonian expressed in the LCP input model is also called a *local Hamiltonian*, or more explicitly a *k-local Hamiltonian* if each Pauli term has weight (number of nonidentity tensor factors) at most k [Llo96, KSV02, KR03, KKR06]. Note that it is still called a local Hamiltonian even if it is n-local: the "locality" here refers to the fact that each Pauli term alone acts independently on each qubit, since it is a tensor product of single-qubit operators.

The second main input model is linear combinations of unitaries or LCU (introduced in [CW12]), in which H is expressed as

$$H = \sum_{i=1}^{N} h_i U_i, \tag{1.20}$$

where $N = \text{poly}(n)$ is the number of terms, the U_i are unitary operators, and the h_i are coefficients that need not be real in general, but that we can choose to be real (or nonnegative) without loss of generality (since any complex phase can be absorbed into U_i); the whole linear combination must be Hermitian. In order for this representation to be efficient, we require that each unitary U_i is uniquely defined by a polynomial amount of classical information, which typically means it is defined as a quantum circuit containing polynomially-many quantum gates. Note that since the Pauli operators are unitary, LCU is a strict generalization of LCP, but LCPs have sufficiently useful and specific properties of their own that we discuss them separately (for example, the Pauli operators form a Hilbert-Schmidt orthonormal basis for qubit observables, and as unitaries they are implemented as single layers of single-qubit gates [NC01]).

The third main input model is sparse matrices (introduced in [ATS03]). A Hamiltonian matrix in some orthonormal basis is *sparse* if it contains at most $k = \text{poly}(n)$ nonzero entries in each row (and column, since it is Hermitian). Equivalently, it is sparse if it connects each basis state to at most $k = \text{poly}(n)$ other basis states; H is referred to as *k-sparse* and k is referred to as the *sparsity* of H. The sparse matrix input model can be used if H is sparse and two unitary operators called *oracles* can be efficiently implemented. The first oracle, which we will call the *location oracle* or O_L, is defined as

$$O_L|x, i\rangle = |x, y(x, i)\rangle, \tag{1.21}$$

where x is the index of a row in H (i.e., a basis state in the chosen basis), $i \in [k]$ (i.e., i is an integer ranging from 0 to $k-1$), and $y(x, i)$ is the index of the ith nonzero entry in row x of H. Hence, O_L computes the locations of the nonzero entries in H. Note that since k is just the maximum number of nonzero entries in any row in H, for any given row not all values of $i \in [k]$ may be needed to index nonzero entries, and those that are not used can be mapped to any arbitrary values; we will discuss this further in Chapter 6.

The second oracle, which we will call the *value oracle* or O_V, is defined as

$$O_V|x, y, 0\rangle = |x, y, H_{xy}\rangle, \tag{1.22}$$

where again x is the index of a row in H, y is the index of a column in H, and H_{xy} is the value of entry x, y in H. Hence, O_H computes the value of a given entry in H. Note that it stores the value in an extra register of qubits provided the initial state of that register is $|0\rangle$; its action on that extra register if its initial state is not $|0\rangle$ can be arbitrary.

Also, since individual Pauli operators are one-sparse, linear combinations of poly(n) of them are sparse, so LCP is a special case of the sparse matrix input model. Treating LCP as a sparse matrix input requires implementing the location and value oracles for Pauli operators, which we describe how to do as an example of oracle construction in Section 5.6.1. However, this is mostly a proof of principle, since in almost all cases where the LCP input model is available (i.e., efficient), it will be simpler and more efficient to use it rather than the sparse matrix input model. The sparse matrix input model and LCU are strictly inequivalent, since the unitaries in the LCU input model need not be sparse when represented as matrices, and although (as we will see in Chapter 5) sparse matrices can be converted to linear combinations of unitaries, doing so comes with significant overhead.

The above input models are the standard inputs to quantum simulation algorithms in the sense that most quantum simulation algorithms take one of LCP, LCU, or a sparse matrix as the representation of the Hamiltonian that is used in the algorithm itself (many algorithms can accept more than one type). However, translating a physical Hamiltonian into one of these input models requires another encoding step. This encoding step depends on the physical model to be simulated, so here we will be content with reviewing a few important examples.

The first example we will consider is spin models. A spin model is defined by a collection of distinguishable spin-1/2 particles under spin-spin interactions that are typically defined on a lattice. For example, the transverse-field Ising model has

the Hamiltonian

$$H = -J \left(\sum_{\langle i,j \rangle} Z_i Z_j + g \sum_i X_i \right), \tag{1.23}$$

where in the first sum the notation $\langle i, j \rangle$ indicates that i and j run over indices of spins that are neighbors on the lattice. J has units of energy, and g controls the strength of the interaction (first sum) relative to the transverse field (second sum). Since the spins are distinguishable and spin models are typically defined by Hamiltonians expressed as linear combinations of Pauli operators, they may be mapped to qubit systems by simply associating each spin to a qubit and using the same Hamiltonian as an LCP. Operators on spins are represented by the identical operators on the qubits. Since LCP is a special case of both LCU and sparse matrices, as noted above, any of the three standard input models can be used to encode a spin model, but also as noted above, since LCP is always available it should probably be used.

For our second example we consider bosonic systems. Bosons are indistinguishable particles that occupy some collection of modes and satisfy Bose-Einstein statistics. This implies that each mode can in principle contain an infinite number of particles, and in many quantum field theories the set of modes is also infinite, so the Hilbert space of the physical bosons is infinite dimensional. Hence, a prerequisite to mapping this Hilbert space to qubits is to truncate both the allowed occupancies of the modes and the number of modes to be finite. Often this is accomplished by directly imposing cutoffs on both [JLP12, JLP14b, MHM$^+$17, LLY20], although in light-front quantization it is only necessary to impose a cutoff on occupancy (implicitly, by fixing the harmonic resolution) [KKG$^+$22].

Bosonic Hamiltonians are defined in terms of creation and annihilation operators for each mode, which respectively add one or remove one particle from that mode. Such operators are one-sparse in the *Fock basis* (spanned by *Fock* or *occupation number states* that are defined as lists of the occupation numbers of the modes), so they are naturally well-adapted to the sparse matrix input model. Exactly how the oracles are implemented depends on how the occupation numbers are

stored in qubits, but a natural choice is to store them as binary numbers, since this yields logarithmic qubit cost in the occupation number cutoff. An implementation of the sparse matrix oracles for bosonic creation and annihilation operators in this binary occupation number encoding or *direct encoding* is given in Section 5.6.2 (see also [KL21]). An alternative implementation in a *compact encoding*, in which the occupation numbers and identities of only the occupied modes are encoded in qubits, is the subject of Chapter 6 (see also [KKG+22, KHKL21]).

Assuming the occupation numbers are stored as binary numbers, the LCP input model is also efficient provided the occupation number cutoffs are not exponential in the system size [Som05], although it is less natural than the sparse matrix input model. To see this, suppose the occupation number cutoff is Λ, and consider the matrix representation of a bosonic creation operator $a^\dagger$ in the Fock basis: for $\Lambda = 7$, this matrix is

$$a^\dagger \mapsto \begin{pmatrix} 0 & & & & & & & \\ \sqrt{1} & 0 & & & & & & \\ & \sqrt{2} & 0 & & & & & \\ & & \sqrt{3} & 0 & & & & \\ & & & \sqrt{4} & 0 & & & \\ & & & & \sqrt{5} & 0 & & \\ & & & & & \sqrt{6} & 0 & \\ & & & & & & \sqrt{7} & 0 \end{pmatrix} \tag{1.24}$$

(all matrix elements not shown are zero). This 8×8 matrix is a 3-qubit operator, and in general for cutoff Λ the operator corresponding to a creation or annihilation operator will require $\lceil \log_2(\Lambda + 1) \rceil$ qubits (e.g., in the above example, if we were to increase Λ to 8, we would need one more row, but since qubit operators have power-of-two dimensions, this would require a 16×16, i.e., 4-qubit operator).

A bosonic Hamiltonian can always be expressed as a real linear combination of Hermitian terms, which will themselves in general be linear combinations of prod-

ucts of creation and annihilation operators acting on different modes: for example, a Hermitian term could be

$$a_1^\dagger a_2^\dagger a_3 a_4 + a_4^\dagger a_3^\dagger a_2 a_1 \tag{1.25}$$

for some four modes labeled 1, 2, 3, and 4. Provided the products in such a Hermitian term contain at most a constant number C of creation and annihilation operators, the whole linear combination will be at most a $C\lceil \log_2(\Lambda+1) \rceil$-qubit operator [Som05, MMS+19].

In general, an n-qubit Hermitian operator can be represented as a linear combination of at most 4^n Pauli operators. Therefore, each Hermitian term in a bosonic Hamiltonian can be represented as a linear combination of at most

$$4^{C\lceil \log_2(\Lambda+1) \rceil} = \text{poly}(\Lambda) \tag{1.26}$$

Pauli operators. In fact, one can check that the number of Pauli operators appearing in the decomposition of a bosonic creation or annihilation operator is

$$2\Lambda^* - 2 < 4\Lambda - 2, \tag{1.27}$$

where Λ^* is Λ rounded up to the nearest power of two, so at worst we obtain one power of Λ per creation or annihilation operator in the term. Hence the LCP input model for a bosonic Hamiltonian is efficient in the occupation number cutoff Λ [Som05]. Note that the occupations of modes can also be represented as unary rather than binary numbers, and doing this yields significantly cheaper operations at the expense of requiring $O(\Lambda)$ qubits instead of $O(\log \Lambda)$ [Som05]; this is an example of a so-called "space versus time tradeoff," meaning in this case that more space (qubits) is used in exchange for shorter runtime (circuit depth).

As our final example of mapping a physical system to qubits, we consider fermions. Fermions are also indistinguishable particles that occupy some collection of modes, but they satisfy Fermi-Dirac statistics. This implies that each mode can only contain zero or one particles, although the number of modes is typically still

infinite in the physical system. It also implies that fermionic creation and annihilation operators acting on different modes anticommute rather than commute. This second property is what strongly distinguishes a fermionic system from a spin model: if the operators commuted instead, then we could simply let each qubit represent the occupation of a particular mode (after the number of modes has been truncated to a finite number), and single-mode fermionic operators would be represented by single-qubit operators. However, single-qubit operators acting on different qubits commute, so the above plan must be modified to apply to fermions.

There are several options for this modification that yield Hamiltonians in the LCP input model. The first is the Jordan-Wigner encoding [JW28], in which each qubit does represent the occupation of a particular mode, but single-mode fermionic operators are represented as the appropriate transformation operator acting on the qubit encoding the occupation of the mode, tensored into a string of Z operators acting on each preceding qubit (in some predetermined ordering, which is arbitrary provided it is used consistently for all modes). For example, a single fermionic creation operator $b^\dagger$ acting on an isolated fermionic mode is represented in the Fock basis as

$$b^\dagger \mapsto \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \frac{1}{2}(X - iY), \tag{1.28}$$

so in the Jordan-Wigner encoding on n modes, the creation operator acting on the jth mode would be represented as

$$b_j^\dagger \mapsto \frac{1}{2}(X_j - iY_j)Z_{j-1}Z_{j-2}\cdots Z_1 Z_0. \tag{1.29}$$

The string of Z operators provides a ± 1 phase equal to the parity of the modes preceding mode j in the chosen ordering, and one can check that any two such operators acting on different modes anticommute.

However, the string of Z operators has length $n/2$ on average over the creation and annihilation operators on n modes, meaning that creation and annihilation operators in the Jordan-Wigner encoding are $\Theta(n)$-local on average. This is

undesirable because, as we will see, implementing unitaries generated by terms in a Hamiltonian is a very typical subroutine in quantum simulation algorithms, and this becomes much more costly when the Hamiltonian terms are nonlocal. Partly motivated by this, other fermion-to-qubit mappings have been developed, such as the parity and Bravyi-Kitaev mappings [BK02, SRL12]. In these mappings, single qubits no longer represent the occupancies of single modes (although n qubits are still used to represent n modes), so the resulting operators have different forms. In the parity mapping, the qubit encodings of creation and annihilation operators are still $\Theta(n)$-local, but in the Bravyi-Kitaev mapping they are $O(\log n)$-local (in particular, each creation or annihilation operator acts on at most $\lceil \log_2(n+1) \rceil$ qubits), making this the best mapping of the three in terms of operator locality. Various other mappings exist, including Bravyi-Kitaev Superfast [LWG$^+$10, SW18a], which exploits the interaction graph of the Hamiltonian, and a number of other mappings that only work for specific interaction graphs but are advantageous in those cases [VC05, DKBC21, DK21, CK22].

As an aside, since typical fermionic Hamiltonians conserve particle number and total spin, and often possess other symmetries, and typical problems of interest specify their values (e.g., a molecule with a fixed spin and number of electrons), we are often only interested in simulating specific symmetry sectors for fermionic problems. Since this reduces the size of the Hilbert space to be simulated, in some cases it is possible to reduce the number of qubits, although fully exploiting these symmetries usually requires going beyond the LCP input model. A number of works have studied this [BGMT17, BBS$^+$17, SW18b, Ste19]: of these, only the $Z2$-symmetry method in [BGMT17], which has become known as *qubit tapering*, yields an LCP input model (and typically saves a constant number of number qubits, at least two for electronic structure Hamiltonians). I (and collaborators) have also made a contribution to the study of fermion-to-qubit mappings [KFHM22]. Since this work was completed during an internship outside Tufts, it is not included in this thesis, but may be briefly summarized as follows.

For a fermion-number conserving system containing η fermions in n modes,

the number of configurations of the fermions is $\binom{n}{\eta}$, i.e., the number of choices of η out of the n modes. This is the dimension of the Hilbert space to be simulated, so the minimum number of qubits we could possibly hope to use is

$$Q^* = \log_2 \binom{n}{\eta} \xrightarrow{n \gg \eta} \eta \log_2 n, \tag{1.30}$$

where the right-hand side is in the limit of many modes, which is typically the high-accuracy limit since the physical system has infinitely many modes. It is possible to construct an encoding that uses approximately this many qubits [BBS$^+$17], but the resulting encoded fermionic operations have polynomial cost in n. In our work, we constructed the first fermion-to-qubit mapping that in the limit of many modes requires only $O(\text{poly}(\eta)\,\text{polylog}(n))$ qubits and has $O(\text{poly}(\eta)\,\text{polylog}(n))$ operation cost. The importance of this is that both costs are polylogarithmic in the number of modes, which makes it possible to scale to very large numbers of modes and thus approach the physical (continuum) limit more efficiently. However, the degrees of the polynomials in the above scalings place this method well into the fault-tolerant regime.

Fermionic creation and annihilation operators are also one-sparse, just like bosonic creation and annihilation operators, which means that fermionic systems and also quantum field theories containing both fermions and bosons are naturally well-adapted to the sparse matrix input model. Sparse input oracles with minimal qubit requirements for general second-quantized quantum field theories containing both fermionic and bosonic degrees of freedom are the subject of Chapter 6.

1.3.2 Simulating time evolution

In quantum mechanics, the Schrödinger equation describes how a quantum state evolves in time:

$$i\frac{\partial}{\partial t}|\psi\rangle = H|\psi\rangle, \tag{1.31}$$

where $|\psi\rangle$ is the time-dependent quantum state and H is the Hamiltonian of the system (we will work in units $\hbar = 1$). When H is time-independent, this equation can be solved for $|\psi\rangle$, yielding

$$|\psi\rangle = e^{-iHt}|\psi_0\rangle, \tag{1.32}$$

where $|\psi_0\rangle$ is the state at time $t = 0$. Hence, the operator

$$U(H, t) = e^{-iHt} \tag{1.33}$$

is the time evolution operator that propagates states forward in time by t [SN20].

A ubiquitous task in quantum simulation is constructing a quantum circuit that approximates the time evolution operator of some physical system encoded in qubits using one of the input models described in Section 1.3.1. Time evolution is the direct goal of dynamics simulations, and it is also a component of many other algorithms including estimating spectra (see Section 1.3.3), computing transition amplitudes and correlation functions [TD00, INE+22], and approximating scattering cross-sections [JLP12, JLP14a, JLP14b, MHM+17, JKLP18, KKG+22, NPdJB21]. Therefore, much effort has gone into developing efficient quantum algorithms for simulating time evolution, and optimal or nearly-optimal algorithms are now known for the most general cases, as well as many algorithms that have certain desirable properties or are well-adapted to more specific cases. In this section, we review highlights of these existing results, some of which are relevant to the technical results in later chapters.

Some of the conceptually simplest time evolution simulation algorithms are based on Trotter or Trotter-like decompositions of the time evolution unitary, which were originally developed in [Suz85, HR90, Suz91], first used in quantum computing in [Llo96], and subsequently developed and refined in [BACS07, Tha08, CMGB09, WBAG11, KGE14, BMW+15, WHW+15, Som16, HP18, COS19, CMN+18, CST+21].

Suppose the input Hamiltonian has the form

$$H = \sum_{j=1}^{N} H_j, \tag{1.34}$$

such that the time evolutions $e^{iH_j t}$ generated by the terms can be simulated directly on the quantum computer. For example, the LCP input model always satisfies this criterion: in this case each $H_j = h_j P_j$ for some coefficient h_j and Pauli operator P_j (see (1.19)), so the corresponding time evolution is $e^{ih_j P_j t}$, i.e., a rotation generated by P_j, which can be implemented by mapping P_j to a single-qubit Pauli operator Q via a Clifford circuit, then implementing $e^{ih_j Q t}$, and then inverting the Clifford circuit [NC01, WBAG11].

For a Hamiltonian of the form (1.34), Trotter decompositions approximate the time-evolution by first breaking it up into a product of short time-evolutions ("Trotter steps"), then approximating each of these by a product of time-evolutions under the terms H_j. These single-term time-evolutions can be implemented directly, providing an approximation of the full time-evolution on the quantum processor. The simulation error scales with the number r of Trotter steps as $O(1/r^p)$ for some positive integer p called the Trotter order. The Trotter order is set by the precise form of the product of short time-evolutions that forms a Trotter step, and higher Trotter orders require more complex and costly Trotter steps [Suz85, Suz91, CST+21]. Hence, the simulation error can be reduced either by increasing the number of Trotter steps (at the expense of greater circuit depth), or by increasing the Trotter order (also at the expense of greater circuit depth). This leads to complicated trade-offs between the costs of the individual Trotter steps and the numbers of Trotter steps required to reach the desired error, which generally must be evaluated on a problem-by-problem basis, and the error bounds can also often be improved by taking commutation relations among the Hamiltonian terms into account [Tha08, CMGB09, WBC+14, KGE14, WHW+15, BMW+15, Som16, CMN+18, CS19, CST+21]. Simulation of time evolution using Trotter decompositions is called *Trotterization*. An overview of the state-of-the-art can be found

in [CST+21].

Trotterization is important enough as the baseline method for quantum simulation of time evolution that subsequent algorithms are sometimes referred to as "post-Trotter" methods. Next, we will provide a review of these, focusing on the succession of algorithms that advanced the state-of-the-art in simulating general Hamiltonians in the asymptotic regime, before also covering a few algorithms that are useful in specific situations even though they are not optimal asymptotically. The asymptotic algorithms have generally been presented in the sparse matrix input model (which also captures the LCP input model, as noted above), although many of them can also take the LCU input model, and we will note this when applicable.

Aharonov and Ta-Shma presented the first quantum algorithm for simulating sparse Hamiltonians in 2003 [ATS03], while the same year Childs *et al.* demonstrated theoretically exponential quantum advantage with respect to an oracle in a similar setting [CCD+03]. Subsequent works have extended and improved these results [BACS07, Chi10, CK10, CK11, BC12, CW12, BCC+14, BCK15, BCC+15, LC17, LW18, LC19, BCS+20]. We now review these works.

The early results were either based on product formulas (which are similar to Trotterization) [ATS03, BACS07, BCC+14, BCC+15] or quantum walks [Chi10, BC12]. The costs of these algorithms are given in number of queries to the sparse Hamiltonian oracles (1.21) and (1.22). The first [ATS03], by Aharonov and Ta-Shma, established that simulation of sparse Hamiltonians was efficient, but with relatively high polynomial costs in most parameters, including $O(n^9)$ in number of qubits n: the scaling with error was linear in the inverse error, but the optimal scaling is logarithmic in the inverse error. Berry *et al.*'s 2007 paper [BACS07] mainly improved upon this by reducing the scaling with n to nearly constant ($O(\log^*(n))$), and also obtained quadratic scaling in the sparsity and nearly linear scaling in both simulation time and inverse error. The improvements were mainly accomplished via a better method for decomposing the Hamiltonian using a type of graph coloring. This paper also established what has become known as the *no fast-forwarding the-*

orem, which states that evolution under a general Hamiltonian for a time t cannot be simulated using a number of operations that is sublinear in t [BACS07, Theorem 3] (this can be violated for some special types of Hamiltonians [AA17]).

The later product formula based methods achieved optimal dependence on the error ϵ of the simulation [BCC$^+$14, BCC$^+$15], obtaining a number of oracle queries scaling as $\widetilde{O}(\log(1/\epsilon))$, where the $\widetilde{O}$ indicates that dependence of order $O(\log\log(1/\epsilon))$ has been omitted. This may be interpreted as (nearly) linear scaling in the number of digits of accuracy (nearly because of the $\widetilde{O}$), and its optimality is shown in [BCC$^+$14, Theorem 2.2]. The two papers [BCC$^+$14, BCC$^+$15] have all authors in common, and differ only in the details of the method, not in the scaling ultimately achieved. By the authors' own claim, the later paper [BCC$^+$15] is a more natural approach: the time-evolution is broken up into Trotter-like steps, but these are then each approximated by implementing a truncation of the Taylor series of the complex exponential as a linear combination of unitaries (see below and [CW12]). *Oblivious amplitude amplification*, which is a technique introduced in [BCC$^+$14] for boosting the amplitude of part of a superposition in which only the states of some tensor factors corresponding to ancillary qubits are known, is used in the special conditions of [BCC$^+$15] to boost the probability of success to arbitrarily close to one.

A quantum walk based method, on the other hand, achieved linear dependence on the sparsity in [BC12], compared to the quadratic dependence in common to [BACS07, BCC$^+$14, BCC$^+$15]. An efficient correspondence between continuous- and discrete-time quantum walks had previous been established in [Chi10], and [BC12] refined this method as applied to simulation of time-evolution through a combination of modifications including using an ancillary qubit and amplitude amplification to prepare the states required for the quantum walk.

In 2015 Berry *et al.* used a quantum walk structure with techniques borrowed from the product formula approaches to obtain near optimal dependence on all parameters [BCK15]: this work may be viewed mainly as an optimal synthesis of prior methods. In 2020 many of the same authors extended these results for time-

dependent Hamiltonians [BCS+20], also obtaining near optimal dependence on all parameters. In specific cases, related methods that depend only on the interaction-picture or off-diagonal norms of the Hamiltonian may be advantageous, but may also require extra work to cast the Hamiltonian from the sparse oracle input model to the required forms [LW18, KH21, CKH21].

Finally, Low and Chuang developed a technique called *qubitization*, based on quantum signal processing, that achieved fully optimal scaling with simulation time, Hamiltonian norm, and error for the time-independent case [LC17, LC19]. Qubitization is a mostly independent technique from the methods we have already discussed: polynomials of a Hamiltonian are obtained by interleaving applications of a block encoding of the Hamiltonian (see (1.40)) with phases that determine the polynomial implemented. As applied to simulation of time evolution for time t with error ϵ under a time-independent Hamiltonian H with sparsity k, qubitization uses these polynomials to approximate the time-evolution unitary, with a cost of

$$\text{number of oracle queries} = O\left(k\|H\|_{\max}t + \log(1/\epsilon)\right), \tag{1.35}$$

where $\|H\|_{\max}$ (the max-norm of H) is defined to be the maximum magnitude of any entry in H. The precise scaling is actually slightly better; see [GSLW19, Section 5.1], which corrects a minor error in the original qubitization paper [LC19]. As noted above, these are optimal scalings: the linear scaling with t is set by the no fast-forwarding theorem [BACS07], the linear scaling with Hamiltonian norm is set by noting that changing the Hamiltonian norm may be viewed as rescaling time, and the scaling with number of digits of accuracy was established as optimal (up to doubly-logarithmic factors) in [BCC+14].

It is worth mentioning that although all of the above algorithms that are optimal or nearly optimal in all parameters require sparse matrix input oracles of the forms given above in (1.21) and (1.22), the early product formula based methods [ATS03, BACS07, BCC+14, BCC+15] require only a weaker version of the

location oracle:

$$O'_L|x,i,0\rangle = |x,i,y(x,i)\rangle, \tag{1.36}$$

i.e., the index i is saved rather than being uncomputed on the way to computing $y(x,i)$, as in (1.21). This distinction between uncomputing or saving the index i appears harmless, but in fact this version of the oracle is typically much simpler to implement. We will see in Chapter 6 that for the standard version (1.21), some care is needed to properly employ O_L in order to obtain the optimal scaling offered by qubitization [LC19] or the near-optimal algorithm for time-dependent Hamiltonians [BCS$^+$20].

For Hamiltonians in the LCU input model, only some of the above algorithms are applicable. The LCU input model was introduced in [CW12], which gave an efficient algorithm for its simulation, but this has been superseded in most contexts by later algorithms with preferable scaling. Qubitization [LC19] can accept the LCU input model, and provides optimal scaling for time-independent LCU Hamiltonians of the form (1.20). Using qubitization, time evolution for a time t can be simulated with error ϵ using

$$\text{number of oracle queries} = O(\|\vec{h}\|t + \log(1/\epsilon)) \tag{1.37}$$

queries to the so-called "SELECT" oracle

$$\textsc{Select} \coloneqq \sum_{i=1}^{N} |i\rangle\langle i| \otimes U_i, \tag{1.38}$$

which can be implemented efficiently provided the U_i can be (assumed for the LCU input model), and the efficient circuit $\hat{G}$ that prepares the state

$$\hat{G}|0\rangle = |G\rangle \coloneqq \sum_{i=1}^{N} \sqrt{\frac{h_i}{\|\vec{h}\|_1}}|i\rangle. \tag{1.39}$$

In order to use the above construction, we assume that the coefficients $h_i \geq 0$, but we can always obtain this by absorbing their phases into the unitaries in (1.20).

Note that given this, the Hamiltonian can be recovered up to scaling as

$$(\langle G| \otimes \mathbb{1}) \cdot \textsc{Select} \cdot (|G\rangle \otimes \mathbb{1}) = \sum_{i=1}^{N} \frac{h_i}{\|\vec{h}\|_1} U_i = \frac{H}{\|\vec{h}\|_1}. \tag{1.40}$$

For readers familiar with this construction, (1.40) is a *block encoding* of the LCU Hamiltonian, which is the input required for qubitization.

To summarize this section, many quantum algorithms have been developed for simulating time evolution with varying costs, but the most advanced methods for simulating sparse Hamiltonians have now achieved nearly optimal scaling with all parameters [LC19, BCS+20]. These methods require as inputs the two oracles Eq. (1.21) and Eq. (1.22) that encode a sparse Hamiltonian, and their complexities are given in terms of numbers of queries to these oracles. Therefore, in order to use these methods in practice, it is necessary to give gate-level constructions of the oracles. This is the subject of Chapter 6, in which we present such constructions for a broad family of second-quantized Hamiltonians.

1.3.3 Simulating ground state energies

Approximating ground states and ground state energies of quantum systems is an important computational problem across many areas of physics and chemistry. In general cases, this problem is believed to be exponentially challenging for a classical computer [AGDLHG05, PMS+14, OBK+16, KMT+17, MEAG+20, SBW+21]. More precisely, this means that as the size of the system grows according to some parameter (typically number of qubits for applications encoded in qubits, e.g., using the methods in Section 1.3.1), the classical cost in memory, runtime, or both to approximate the ground state energy grows exponentially with the system size provided the desired approximation error is sufficiently small.

Can a quantum computer do better, which would require solving ground state problems in polynomial quantum memory and runtime for at least some useful problems that are classically hard? To begin to answer this question, we must first note that ground state problems are classically hard for two mostly independent

reasons: first, because general quantum states require exponential classical memory to store and exponential classical runtime to manipulate, and second, because general quantum ground state problems are QMA-complete (the quantum generalization of NP-complete) [KSV02, WB03, KR03, KKR06] (ground state problems without superposition are NP-complete [Bar82]). The former difficulty (that of representing states) is eliminated by using a quantum computer, since we can efficiently encode other quantum systems in qubits, as discussed in Section 1.3.1. The latter difficulty (QMA-completeness), however, implies that we should expect worst-case ground state approximation problems to be intractable for quantum computers as well, i.e., even though representing quantum states is easy on a quantum computer, there exist Hamiltonians for which finding the ground state is hard.

Therefore, quantum algorithms for ground state energies must either require additional assumptions that restrict the problem instances away from worst-case QMA problems, or be heuristics, i.e., not have explicit performance guarantees. These two possibilities are not necessarily distinct, and both are well-motivated by the parallel with classical computing of NP-complete problems: although worst-case NP-complete problems are believed to be intractable, specific problems from NP-complete classes are solved routinely using heuristics. In the quantum ground state problem, some typical assumptions or heuristics are finding a "good" initial guess state (one that has nontrivial, i.e., $\Omega(1/\text{poly}(n))$ overlap with the sufficiently low-energy subspace for n qubits) [Kit95, LT20, LT22, DLT22, KMM22], finding a gapped adiabatic path from an efficiently solvable Hamiltonian to the Hamiltonian of interest [FGGS00], assuming bounded correlation length in the ground state [MST$^+$20], or optimizing a variational quantum circuit [PMS$^+$14].

For example, the first of these assumptions could either be regarded as a restriction on the problem class (performance guarantees given a problem with a provably good initial guess state) or as a heuristic (when the initial guess state is well-motivated but may or may not actually be good). This is the most typical assumption in quantum algorithms aimed at high-accuracy simulation of ground state energies in the asymptotic regime: the most well-known of these is *quantum*

phase estimation [Kit95], and recent results have achieved optimal scaling for this class of algorithms [LT20, LT22, DLT22].

In this thesis, however, we will mainly focus on *variational quantum algorithms*, which are algorithms based on the final heuristic listed above, optimizing a parameterized quantum circuit. When applied to the ground state problem for quantum systems, these methods are typically called *variational quantum eigensolvers* or VQEs [PMS+14], which have been experimentally implemented for a number of small molecules and effective field theories [DXP+10, LWG+10, PMS+14, WDB+15, OBK+16, SZZ+17, PGS+17, KMT+17, HMR+18, SWG+18, CRD+18, DMH+18, KMvB+19, KTC+19, GEB+19, SBMS+19, MPJ+19, SM19, NCP+20, AAB+20, KJK+21a, KJK+21b, LYTT21, KGL+22].

A variational quantum eigensolver requires that the Hamiltonian is decomposed into some linear combination of observables that can be measured on the quantum computer. Typically, these are chosen to be the Pauli operators, so the standard input model for VQE is linear combinations of Pauli operators (see Section 1.3.1), although Chapter 5 presents a method for applying VQE to the sparse matrix input model. For simplicity, in this introduction we will focus on linear combinations of Pauli operators: the corresponding Hamiltonian has the form (1.19).

The other main component of a variational quantum eigensolver is the *ansatz*: a quantum circuit that can be varied according to the values of some control parameters. A two-qubit example is given in Fig. 1.1. This example is special because for any two-qubit state with real amplitudes, there is some setting of the parameters in the circuit in Fig. 1.1 that generates that state, and with a slightly more complicated circuit we could similarly parameterize all two-qubit states (with complex amplitudes). In fact in a variational quantum eigensolver we would in general not want our ansatz to be able to generate arbitrary quantum states, since that would require exponentially-many parameters [NC01], but for two qubits this presents no difficulty.

Given a Hamiltonian decomposed into Pauli operators, and an ansatz circuit, the variational quantum eigensolver routine is summarized in Fig. 1.2, and the details

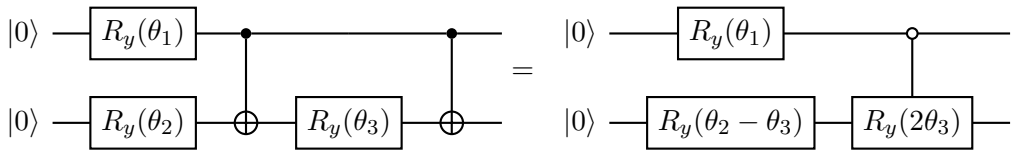


Figure 1.1: This figure shows a parameterized quantum circuit that is expressive enough to generate any two-qubit state with real amplitudes.

are as follows [PMS$^+$14]. The goal is to classically optimize the ansatz parameters in order to minimize the expectation value of the Hamiltonian in the corresponding state. We conventionally denote this state

$$|\psi(\vec{\theta})\rangle = U(\vec{\theta})|\psi_0\rangle, \tag{1.41}$$

where $\vec{\theta}$ is the vector of parameters, $U(\vec{\theta})$ is the unitary implemented by the ansatz circuit, and $|\psi_0\rangle$ is some initial state. By the Rayleigh-Ritz variational principle,

$$E_0 \leq E(\vec{\theta}) \coloneqq \langle\psi(\vec{\theta})|H|\psi(\vec{\theta})\rangle \tag{1.42}$$

for any $\vec{\theta}$, with equality only if $|\psi(\vec{\theta})\rangle$ is an exact ground state, so by minimizing the objective function $E(\vec{\theta})$ we obtain a variational estimate of the ground state energy E_0.

The role of the quantum computer in this algorithm is just to provide an estimate of $E(\vec{\theta})$, the expectation value of the Hamiltonian in the state $|\psi(\vec{\theta})\rangle$ for any given values of $\vec{\theta}$ set by the classical optimizer. To do this, we use the fact that Pauli operators are observables that can be measured directly on the quantum computer, as we will shortly describe. Hence, by repeated measurements we can build up an estimate of the expectation value of each Pauli term in the Hamiltonian. Then we can use the fact that expectation values are linear to obtain an expectation value estimate of the full Hamiltonian as the linear combination of the expectation values of the terms. Symbolically, for a Pauli Hamiltonian of the form (1.19),

$$E(\vec{\theta}) = \langle\psi(\vec{\theta})|H|\psi(\vec{\theta})\rangle = \sum_{i=1}^{N} h_i \langle\psi(\vec{\theta})|P_i|\psi(\vec{\theta})\rangle. \tag{1.43}$$

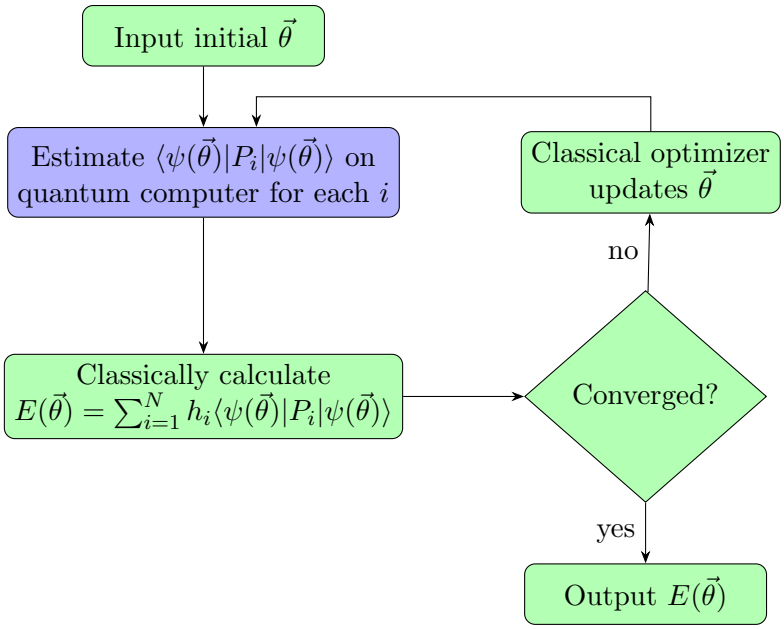


Figure 1.2: Schematic of the VQE algorithm. All of the green nodes correspond to classical components of the algorithm, while the single blue node corresponds to the quantum subroutine.

This value estimated using the quantum computer forms the objective function for the classical optimization.

To perform the measurements, we conventionally define the single-qubit Pauli Z measurement (i.e., measurement in the computational basis $\{|0\rangle, |1\rangle\}$ with the associated measurement results $\{+1, -1\}$) on each qubit to be the measurement that is physically implementable on the quantum processor. Therefore, if we just repeatedly measure in this basis on every qubit, we can use the measurement results to estimate the expectation values of all Pauli terms in the Hamiltonian that are tensor products of single-qubit identity I and Z. For example, if we want to estimate the expectation value of $I \otimes Z \otimes I \otimes Z$, then we just multiply the ± 1 measurement results on the second and fourth qubits for each measurement and take the average of these.

To measure some qubit in a different Pauli basis, we can use the facts that

$$HZH = X, \qquad SHZHS^\dagger = Y \tag{1.44}$$

(where H is the Hadamard gate and S is the phase gate — see Section 1.2). Hence, if we apply the gate H on a qubit immediately before measuring it in the Z basis, this is equivalent to measuring it in the X basis. If we apply $HS^\dagger$ immediately before measuring it in the Z basis, this is equivalent to measuring it in the Y basis. Therefore, using appropriate combinations of these we can perform measurements in bases that yield samples of any Pauli in the Hamiltonian, so by repeating measurements in all of these bases we build up expectation value estimates of all terms in the Hamiltonian (1.19).

Note that the worst-case scenario for this scheme is when no Pauli terms in the Hamiltonian are *locally compatible* (i.e., none of them share a common set of measurement bases for each qubit): in this case we would need a separate set of measurement bases for each term in the Hamiltonian. In cases of interest, however, there are typically many Pauli terms that are locally compatible, meaning that we can simultaneously build up expectation value estimates of many Pauli terms at once. Optimally exploiting this property is the subject of ongoing research, with the main divide being between methods that use versions of graph coloring or other deterministic strategies to partition the Pauli terms into commuting subsets [VYI20, YVI20, GAD$^+$20, YGI22], and "classical shadows" methods that randomly choose measurement bases that permit simultaneous measurement of the locally compatible terms [HKP20, WSHY21, HBRM22, SJM$^+$21, Had21, HHR$^+$21, EFH$^+$22]. The derandomization technique in [HKP21] is a hybrid of the two, using classical precomputation to deterministically choose a sequence of measurement bases that is guaranteed to perform at least as well as a random sequence chosen by the classical shadows method in [HKP20].

The cost of a single iteration step of the classical loop in VQE, i.e., from one parameter setting to the next, is relatively easy to analyze. For the quantum part of the algorithm, each term in the Hamiltonian must be estimated by repeated preparations and measurements, so since each is being estimated essentially as a binomial random variable we should expect the number of measurements required to reach some desired precision ϵ to scale as $O(1/\epsilon^2)$. A naive implementation of

VQE in which separate measurements are used for each term therefore leads to a total measurement scaling of

$$M = O\left(\frac{N}{\epsilon^2}\right) \tag{1.45}$$

where, as in (1.43), N is the number of terms in the Hamiltonian.

The schemes for combining measurements that we described above [VYI20, YVI20, GAD$^+$20, YGI22, HKP20, HBRM22, HKP21, SJM$^+$21, Had21, HHR$^+$21, WSHY21, YGI22, EFH$^+$22] can improve the numerator in (1.45), but the scaling with error remains the same as in (1.45) in all of these methods. *Unitary partitioning* is a different technique for combining measurements, where instead of using the fact that commuting Pauli terms can be measured simultaneously we use the fact that a linear combination of anticommuting Pauli terms can be efficiently unitarily mapped to a single Pauli operator. Thus by appending these unitaries to the end of the state preparations, we can combine a whole anticommuting set of Pauli terms into one measurement, in exchange for increased circuit depth. This method was developed independently in [IYLV20, ZTK$^+$20], the latter of which I contributed to, but which is not included in this thesis.

The costs of the classical components of the iteration step do vary somewhat depending on which classical optimizer is used, but they can generally be expected to be negligible compared to the cost of estimating the terms in the Hamiltonian. The more difficult to analyze cost of VQE is the number of iteration steps required to reach convergence of the approximate ground state energy. As mentioned at the beginning of this section, for general Hamiltonians the problem of accurately estimating the ground state energy is QMA-complete [KSV02, WB03, KR03, KKR06], which means that as in the analogous classical case of NP-complete problems, we should expect worst-case ground state energy problems to be exponentially difficult. In VQE, this would manifest itself by preventing the optimization from converging in polynomial time. However, just as specific cases of NP-complete problems are routinely solved using heuristics for many applications across industry, commerce, and scientific research, specific ground state energy problems corresponding to phys-

ical systems of interest may also be tractable. VQE is a heuristic, so the proof will ultimately be in the pudding, as quantum computers continue to scale and larger and larger problem instances come within reach. The optimization problem in VQE is not the main focus of this thesis, so I will not go into further detail. For a more in-depth review of VQE including discussion of classical optimizers and the challenges of the optimization problem itself, [TCC+22] is an excellent resource.

Finally, although VQE has been for many years the leading algorithm for simulating ground state energies on pre-fault-tolerant quantum computers, some recent techniques that may be loosely referred to as *quantum subspace diagonalization* (QSD) offer an alternative that also has the potential to be feasible on NISQ devices (there is presently no apparent consensus on what term should be used for this family of algorithms, but QSD is suggested in [ELN21] and I support this choice). The basic idea, first suggested in [MKSCdJ17], is to use the quantum processor to evaluate matrix elements of the projection of the Hamiltonian into a subspace spanned by states that can be efficiently obtained via quantum circuits. The resulting circuits are relatively low-depth compared to quantum simulation algorithms for fault-tolerant quantum computers, such as phase estimation. Versions of this include subspaces spanned by unitary operators (applied to some initial state) that approximate real [PM19, SHE20, CMP21, CG22, KMZC+22] or imaginary [MST+20] time evolutions, powers of the Hamiltonian as approximated by linear combinations of time evolutions [SY21], and other heuristically-motivated states not obtained from functions of the Hamiltonian [MKSCdJ17, CRD+18, TRJ+20, HLB+20, YHM+22, BHS+22, TZC+22].

I have also contributed a paper to this area: in [KMM22], we developed an algorithm for constructing a Krylov space (a subspace spanned by powers of the Hamiltonian applied to an initial state) on a quantum computer. Compared to the above prior algorithms, our method does not require approximating real or imaginary time evolution and the resulting Krylov space is exact, so the only source of error relative to the Lanczos method in infinite-precision arithmetic [Lan50, Kan66, Pai71, Saa80] is finite sample noise (as well as noise on the quantum processor). We also provide

explicit error bounds on the ground state energy estimate in terms of the Krylov space dimension and noise rate. To my knowledge our method is the first quantum algorithm to directly use a block encoding of the Hamiltonian without requiring quantum signal processing [LC17, LC19].

To summarize this section, there are many quantum algorithms for approximating ground state energies that will become possible on fault-tolerant quantum computers. However, on NISQ devices the options are much more limited, and VQE is the leading algorithm in this area, although QSD offers a promising alternative. In this thesis, Chapters 2 to 5 will focus primarily on VQE.

1.4 Contextuality

Contextuality is a feature of quantum mechanics that distinguishes it from classical mechanics. It generalizes nonlocality, and although the first proof of existence of contextuality in quantum mechanics appeared around the same time as Bell's theorem and indeed was partly conceived by John Bell, it is less well-known than nonlocality. This first proof of contextuality in quantum mechanics is called the Bell-Kochen-Specker theorem [Bel64, Bel66, KS67], and states that there exist quantum systems for which it is impossible to reproduce the outcome probabilities of every possible measurement as marginals of single joint probability distribution. This impossibility defines contextuality.

Subsequent work has led to a variety of approaches to characterizing and quantifying contextuality. The early works that set the stage for this area of research, by identifying most of the general categories of contextuality (e.g., strong versus weak contextuality, which we discuss further below, and general contextuality versus special cases like nonlocality), are [Bel64, Bel66, KS67, CHSH69, Fin82, Per91, Mer90, Mer93]. A number of more recent works provided rigorous mathematical frameworks for contextuality in settings that are in principle even more general than quantum mechanics, including the sheaf-theoretic approach [AB11], the cohomological approach [IB98, OTR18, Rau19, OR20], and generalized probabilistic

theories [SSW+21]. A seminal paper by Robert Spekkens established a framework for analyzing contextuality in quantum circuits by showing that contextuality can be present in state preparations, transformations, or measurements [Spe05]. Other works have provided witnesses, tests, or quantifications of contextuality in various scenarios [RSKanK12, CSW14, RH14, GHH+14, CKB15, ABM17, dS17, AC17, SSW18, DA18, KL19a, XC19, HZS+22] (the work [KL19a] discussed in Chapter 2 is among these), or experimentally tested such scenarios [KZG+09, LMZ+18]. Contextuality has been connected to resources and models for quantum computing, including negativity of quasiprobability representations [Spe08, FE08, FE09, Fer11, VFGE12, RBD+17, KL17, RBVT+20], magic states [HWVE14], antidistinguishability [LD20], measurement-based quantum computing [Rau13, FRB18], and semiclassical expansions [KL18, KL19b]. It has also been used to generate quantum advantage over classical computing in certain restricted scenarios, including quantum state discrimination [SS18], memory requirements of classical simulation of the stabilizer subtheory [KWB18], computation of nonlinear functions [MK18], postselected metrology [ASYHL+20], computing state overlaps [GaB20], and shallow circuits [BGK18, BGKT20], but there are also known limitations on using contextuality to generate quantum advantage in broader settings (this is discussed in some of the citations in this sentence, and also in [LWE19]). Finally, our works to be discussed in Chapters 3 and 4 of this thesis showed that in a noncontextual variational quantum eigensolver the quantum objective function can be replaced by a classical objective function with no loss in expressivity [KL20], and that a contextual variational quantum eigensolver can be partitioned into a noncontextual part that is modeled using a classical objective function and a contextual correction that is computed on the quantum processor using fewer resources than would be required to run the full VQE [KTL21].

If we restrict to some particular set of measurements corresponding to a set of observables $\mathcal{S}$, properties of the set determine whether a joint probability distribution may exist for the outcomes of those measurements. Measurement contextuality refers to various types of contradictions that can appear in attempts to do this. In

IZ	ZI	ZZ	1
XI	IX	XX	1
XZ	ZX	YY	1
1	1	-1	

Figure 1.3: The Peres-Mermin square, an illustration of contextuality. The observables in the square are two-qubit Pauli operators (by convention the tensor product symbols are suppressed, so for example $IZ \coloneqq I \otimes Z$). The numbers at the ends of the rows and columns are the products of the corresponding rows or columns. In a noncontextual hidden-variable model based on simultaneous assignments of values (± 1) to the observables in the square, those value assignments would have to satisfy the row and column products, since each row and column commutes and is therefore simultaneously measurable. However, in order to satisfy all of the row products, the value assignments would have to contain an even number of -1s, while to satisfy the column products they would have to contain an odd number of -1s, so a noncontextual hidden-variable model necessarily violates one or the other.

this thesis we will focus on "strong contextuality,"[1] which is contextuality in the same vein as the Peres-Mermin square (see Fig. 1.3 and [Per91, Mer90, Mer93], in particular Mermin's outline of a "plausible" hidden-variable theory in [Mer93, §II].) Informally, a set of measurements is strongly contextual if it is impossible to consistently assign values to every measurement in the set. In "weak" versions of contextuality such as Bell inequality violations, joint values may be consistently assignable, but statistical predictions based on the existence of joint probability distributions are violated. It is typical in the literature to use "contextuality" when really "strong contextuality" is meant, so for simplicity we adopt this convention.

An archetypal example of contextuality is the Peres-Mermin square [Per91, Mer90, Mer93], shown in Fig. 1.3. Suppose we want to construct a *noncontextual hidden-variable model* for the set of Pauli operators in the Peres-Mermin square, i.e., a semiclassical model in which the underlying objects are a set of joint value assignments (assignments of values ± 1 to each Pauli operator in the square) and a probability distribution over those joint value assignments. A noncontextual hidden-variable model is very similar to a classical phase-space model, where the phase space points are the joint value assignments, it just may have additional constraints that impose uncertainty relations on the allowed probability distributions, which we

[1] so called in [AB11] and studied in [AB11, RSKanK12, ABM17, AC17, dS17, KWB18, XC19, Rau19, SSW18, DA18, RBVT$^+$20, HZS$^+$22]

will discuss further in Chapter 3 (see also [Spe07, Spe16]). Such a model may be interpreted as describing a set of observables that have definite values at all times, with the probability distribution simply describing the observer's potentially limited knowledge about those values.

For now, suppose we want to construct such a model, starting with the joint value assignments. Any such joint value assignment is a list of nine ± 1s specifying the values assigned to the Pauli operators in the square. The operators in each row and column in the square commute, and the product of the Pauli operators in the row or column is equal to the ± 1 at the end of the row or column. Since each row and column commutes, any one of them corresponds to a set of measurements that an observer could in principle perform jointly. This means that if, for example, the observer measures the first row, IZ, ZI, and ZZ, obtaining the assigned values ± 1, these measurement outcomes must satisfy the row product, i.e., they must multiply to 1. The same is true for any other row or column. We are trying to construct a *noncontextual* hidden-variable model for these observables, which means that the joint value assignment must be independent of the measurement (or *context*) that the observer is about to perform. Hence, it must satisfy all of the row and column products since the observer could perform any of them.

This, however, introduces the problem, because in order to satisfy all of the row products the value assignments would have to contain an even number of -1s (since each row would have to contain an even number), while to satisfy the column products they would have to contain an odd number of -1s (since the first two columns would have to contain an even number and the last column would have to contain an odd number). Therefore, there is no assignment of ± 1s to the Peres-Mermin square that forms a valid noncontextual joint value assignment by satisfying all of the row and column products, and thus the set of observables in the Peres-Mermin square is contextual.

To emphasize what this means, remember that the type of model we are trying to construct (or rule out, in the contextual case) for our set of observables is one in which all of the observables have actual, definite values at all times. In ad-

dition, it is required to be a noncontextual model, which means that if an observer measures some commuting subset of the observables, the measurement outcomes they obtain must be the actual definite values immediately prior to measurement, and those actual definite values cannot be affected by what measurements the observer performs. This last requirement is the noncontextuality condition, since for example if the value assigned to IZ in Fig. 1.3 is allowed to varying depending on whether we measure IZ in the *context* of its row or its column, then it is easy to come up with such conditional value assignments that do satisfy all of the row and column products. The remainder of the noncontextual model would be a set of joint probability distributions over the definite joint value assignments to the observables, which would describe the observer's knowledge about the joint value assignments (this will be discussed in detail in Chapter 3). However, the Peres-Mermin square rules out such models without even having to consider the joint probability distributions, by showing that the joint value assignments cannot satisfy the row and column conditions predicted by quantum mechanics.

The fact that the Peres-Mermin square rules out such models for any set of Pauli observables containing those in the square means that quantum mechanics as a whole does not admit description by a noncontextual hidden-variable model. This is usually taken to imply that quantum mechanics is contextual, and we will also use this language, but it is worth noting that the formalism of quantum mechanics is itself noncontextual in the sense that outcome probabilities for a given measurement do not depend on which other commuting measurements are jointly performed. However, the formalism of quantum mechanics is not a hidden-variable model, so when we say that quantum mechanics is contextual we really mean that any hidden-variable models describing it must be contextual.

Since contextuality (in the sense just described) appears in quantum mechanics but not in classical mechanics, it is natural to study its implications for quantum computing. Chapters 2 to 4 will be primarily occupied with this. Since we will spend significant time getting into it in detail in those chapters, in this introduction we will not say much more, but we should at least explain why we have focused on strong

measurement contextuality: we will be applying analysis of contextuality to VQEs, in which the quantum part of the algorithm consists of a set of Pauli measurements. Hence measurement contextuality is the appropriate type of contextuality to apply. We will see that there is in fact no difference between strong and weak contextuality in this setting, and strong contextuality will be convenient to analyze. We conclude this section by mentioning [BCG$^+$21], which is a thorough review of contextuality, covering most of its known variants and applications, as well as its foundational implications.

1.5 Overview and summary of results

The main results covered in this thesis are as follows. Chapter 2 presents an analytic, classically efficient test for contextuality of VQEs. As discussed in Section 1.3.3, the quantum component of the VQE algorithm consists of preparing a variational ansatz state and measuring the Pauli terms in a Hamiltonian. The contextuality (or lack thereof) of a VQE can thus be characterized by the set of Pauli measurements that are performed. In Chapter 2, we present a necessary and sufficient condition for contextuality of any arbitrary set of Pauli operators, allowing us to check whether any given VQE is contextual. This chapter is based on [KL19a].

If a VQE turns out to be noncontextual, then as discussed in Section 1.4, we expect that it should be possible to construct a noncontextual hidden-variable model for it that replaces the quantum part of the algorithm. In Chapter 3, we explicitly show how to do this. The resulting noncontextual models provide a classical parameterization of the Hamiltonian's spectral range, so minimizing with respect to these parameters yields an approximate ground state energy, just as in VQE but without requiring the quantum computer to evaluate energies. One implication of this is that the noncontextual Hamiltonian problem (the decision problem of verifying whether a noncontextual Hamiltonian's ground state energy lies below some promise gap) is only NP-complete rather than QMA-complete. This chapter is based on [KL20].

Chapter 4 presents a hybrid quantum-classical algorithm that we call *contex-*

tual subspace VQE or CS-VQE, in which a Hamiltonian whose ground state energy we want to simulate is partitioned into a noncontextual part that we simulate classically and a contextual correction that we calculate using the quantum computer. By including as much of the Hamiltonian as possible in the noncontextual part, we do most of the work classically. The resulting quantum correction requires fewer resources (qubits and measurements) in general than doing full VQE on the same Hamiltonian, and at the experimenter's discretion, more quantum resources can be used to obtain higher accuracy in the resulting ground state energy estimate. This chapter is based on [KTL21].

In Chapter 5, we shift our focus from contextual and noncontextual Hamiltonians to sparse Hamiltonians (see Section 1.3.1). Simulation of time evolution under sparse Hamiltonians has been heavily studied, and by extension various algorithms for approximating ground state energies of sparse Hamiltonians that are based on time evolution, like phase estimation, are well-understood. These algorithms require long coherent evolutions, and most likely will only become possible once fault-tolerance is reached. However, the study of VQE has focused almost exclusively on Hamiltonians in the linear combination of Pauli operators input model. In Chapter 5, we present a method for implementing VQE with sparse Hamiltonians. This significantly extends the domain of applicability of VQE, since all Pauli Hamiltonians are sparse Hamiltonians, but sparse Hamiltonians are a much broader category. While this method is less appropriate for very near-term quantum computers than standard VQE, since even the oracles used to encode sparse Hamiltonians are typically too costly to implement on existing devices, it only requires a constant number of queries to those oracles in each coherent evolution. Therefore, it represents an intermediate step on the way to algorithms based on time-evolution under sparse Hamiltonians (in which the number of queries always scales with error and evolution time — see Section 1.3.2), so it may become a useful alternative to standard VQE as well as algorithms like phase estimation on early fault-tolerant quantum computers, and potentially even pre-fault-tolerant quantum computers if they become significantly less noisy than those existing today. This chapter is based

on [KL21].

Finally, in Chapter 6, we present a method for efficiently implementing the sparse Hamiltonian oracles for a broad class of second-quantized Hamiltonians, including most bosonic, fermionic, or mixed Hamiltonians that might appear in quantum field theories. This method is particularly well suited to quantum field theory in the light-front formulation [KKG$^+$22, KJK$^+$21b, KJK$^+$21a], as we will discuss in detail in Chapter 6. Although the resulting oracle implementations likely require fault-tolerance, they demonstrate explicitly that in the long term, quantum computers will be able to simulate generic quantum field theories. This chapter is based on [KHKL21].

Other papers to which I contributed that are not covered in this thesis are [KFHM22, KMM22, ZTK$^+$20, RKR$^+$21, KKG$^+$22, KJK$^+$21a, KJK$^+$21b, WRK$^+$22, RWT$^+$22, RGD$^+$22, AHH$^+$22], which are not included either because they were completed during summer internships not at Tufts or because I am not the first author. The two in which I am the first author are [KFHM22] and [KMM22]. As discussed in Section 1.3.1, [KFHM22] presents the first known method for mapping fermionic systems to qubits in which both the number of qubits and the cost of the encoded fermionic operators are polylogarithmic in the number of fermionic modes (at fixed particle number). As discussed in Section 1.3.3, [KMM22] presents a quantum subspace diagonalization method based on block encoding that avoids many of the sources of error that prior techniques are subject to, and admits rigorous error analysis.

On the remainder of the papers cited above, I was not the first author, but they may be summarized as follows. In [ZTK$^+$20], we developed a technique that has become known as *unitary partitioning* for combining groups of anticommuting Pauli measurements into single measurements in a VQE procedure (unitary partitioning was independently developed in [IYLV20]). For electronic structure Hamiltonians, we proved that this leads to a reduction in the number of distinct measurements by a factor of n, where n is the number of spin-orbitals.

In [RKR$^+$21], we developed extensions of *robust phase estimation* [HBB$^+$09,

KLY15] that add error detection to the procedure via a collection of consistency tests. Robust phase estimation is similar to Kitaev's phase estimation method [KSV02] in that it learns one bit of the phase at a time, but as the name suggests, it is robust to errors up to a fixed threshold. In [RKR$^+$21], we corrected some minor errors in the original analysis of robust phase estimation [KLY15], showing that its main results still hold, and also presented techniques for testing whether the error threshold required for robust phase estimation to succeed has been violated in a given experiment.

In the three closely related papers [KKG$^+$22, KJK$^+$21a, KJK$^+$21b], we developed techniques for simulating quantum field theory in the light-front formulation [PB85, HV87, BPP98, VHL$^+$10]. The light-front formulation of quantum field theory is appealing for quantum simulation because the Hamiltonian block-diagonalizes into blocks with fixed *light-front momentum* (a linear combination of energy and spatial momentum in equal-time coordinates). Therefore, when simulating a closed system on the light-front, the Hamiltonian naturally takes a similar form to quantum mechanical Hamiltonians, allowing us to apply well-developed techniques for quantum simulation of quantum mechanical systems.

The first [KKG$^+$22] of the three papers developed the general theory of quantum simulation of light-front quantum field theory by presenting methods for mapping the physical states and operators to qubit states and operators, and discussing in particular the use of adiabatic state preparation [FGGS00, ATS03] to prepare approximate ground states from which various quantities of interest can then be estimated, including parton distribution functions, form factors, and decay constants. The other two papers, which present similar results in letter form [KJK$^+$21a] and in more detail [KJK$^+$21b], focus on a family of effective light-front field theories known as *basis light-front quantization* [VHL$^+$10], which can be used to generated relatively small effective Hamiltonians. We simulated two- and four-qubit instances of these Hamiltonians using VQE on IBM's freely available quantum processors, using them to calculate mass radius, decay constant, elastic form factor, and charge radius for a pion. These experiments may be consid-

ered similar to simulations of small molecules in minimal bases, which were the earliest demonstrations of quantum simulation for the electronic structure problem [PMS+14, OBK+16, KMT+17, HMR+18, NCP+20].

In [RGD+22], we presented an algorithm for classically simulating Hamiltonians using a method similar to VQE, but with the ansatz circuits restricted to be Clifford circuits, which are efficiently classically simulable [Got98]. For ansatz circuits whose parameterized gates are generated by Pauli operators, this may be accomplished by restricting the gate angles to $\{0, \pi/2, \pi, 3\pi/2\}$. Since such an ansatz is classically simulable, it may be used to construct a classical objective function of the (discrete) parameter settings, which can be passed to the classical optimizer to obtain a completely classical variational simulation method for approximating the ground state. The output of this method may then be used as an approximation in its own right, or as the initialization point for an actual quantum VQE procedure that uses the same ansatz but permits the parameters to vary continuously. We tested the method for electronic structure Hamiltonians and found that it matches the performance of Hartree-Fock in regimes where Hartree-Fock is expected to perform well (specifically, near equilibrium for weakly-correlated molecules), and outperforms Hartree-Fock in some systems where Hartree-Fock performs poorly (specifically, far from equilibrium). This may be explained by noting that Hartree-Fock in the best case finds the lowest energy computational basis state, while the method of [RGD+22] attempts to find the lowest energy among stabilizer states [Got98, NC01], which is a strict superset of the computational basis states.

In [AHH+22], we presented an improvement to the method for quantum computation of the boundary operator given in [UAS+21], which is a component of quantum topological data analysis [LGZ16, UAS+21]. The improvement is based on noticing that in an appropriate representation, the boundary operator takes the form of a sum of anticommuting Pauli operators, which can be mapped to a single Pauli operator using unitary partitioning [IYLV20, ZTK+20]. This allows it to be applied exactly via a single Pauli operation sandwiched between two unitary partitioning circuits. Compared to prior implementations of the boundary operator, this reduces

the scaling with number of qubits from $O(n^2)$ to $O(n)$, and eliminates errors that came from using approximated time-evolution to implement the boundary operator as in [UAS+21].

Finally, in [WRK+22, RWT+22] we developed extensions and improvements to the *contextual subspace variational quantum eigensolver* (CS-VQE) method that will be the topic of Chapter 4 in this thesis. In [WRK+22], we filled in the details of a method for constructing ansatze for CS-VQE by projecting a full VQE ansatz onto the contextual subspace, which was suggested but not fully developed in [KTL21]. In [RWT+22], we studied how unitary partitioning [IYLV20, ZTK+20] can be applied in the context of CS-VQE to further reduce the number of measurements performed on the quantum processor. Both of these papers represent important steps on the way to practical, experimental implementations of CS-VQE, which are now in progress.

Chapter 2

Testing for contextuality in Pauli Hamiltonians and variational quantum eigensolvers

2.1 Introduction and prior work

The variational quantum eigensolver or VQE, introduced in Section 1.3.3, is an important near-term application of NISQ machines. Hence it is natural to consider how the contextuality of VQE procedures is related to any quantum advantage that they may obtain. In this chapter, we present a method to analyze the contextuality of VQE procedures. In particular, we study the contextuality of the target Hamiltonian and observables it is decomposed into. We will focus in this chapter through Chapter 4 on Hamiltonians decomposed into linear combinations of Pauli operators that define the measurements actually performed in order to evaluate the objective function of the VQE instance. Thus our criterion applies to the VQE instance as problem statement and is independent of the variational ansatz, which changes throughout the algorithm.

The set of Hamiltonians that are noncontextual by our definition will include diagonal Hamiltonians that encode a classical objective function. Such problems are addressed by the Quantum Approximate Optimization Algorithm (QAOA), which is closely related to VQE [FGG14]. As we shall see, the set of noncontextual Hamiltonians contains the set of commuting Pauli Hamiltonians, and therefore represents a broader definition of classicality.

One concept upon which we rely is a *closed subtheory*: a set of observables in which all observables whose values are determined with certainty by the values of others in the set are themselves members of the set. We introduce this concept here because it provides a distinction between this work and the criteria for strong contextuality studied in [RSKanK12], which are based on sets of observables that are not necessarily closed subtheories. In [KWB18] it is shown that the efficiency of classical simulation of certain closed subtheories of quantum mechanics is limited by contextuality, so these results of [KWB18] apply to our setting.

In [CSW14] the authors obtain criteria for contextuality based on compatibility graphs, as do we. However, [CSW14] focuses on weak contextuality, that is, violation of noncontextual inequalities, whereas our interest is in strong contextuality. We further discuss the distinction between our condition for contextuality and previously studied criteria in Section 2.6 and Section 2.4.

In Section 2.2, we develop the notion of contextuality we will study. In Section 2.3 prove our main result, an efficiently-testable necessary and sufficient condition for contextuality of any set of Pauli operators (Theorem 2.2 and Theorem 2.5 give two equivalent forms for the condition). In Section 2.4 we discuss several options for quantifying the amount of contextuality possessed by a Hamiltonian, based on the criterion in Section 2.3. In Section 2.5 we evaluate the contextuality of several VQE experiments. We conclude in Section 2.6 with a discussion of our results, and directions for future work. This chapter is based on [KL19a].[1]

[1] ©2019 American Physical Society, reproduced per Author's Rights under the transfer of copyright agreement (`https://journals.aps.org/authors/transfer-of-copyright-agreement`).

2.2 Defining contextuality

We focus on the analysis of strong contextuality for sets of Pauli operators. Let $\mathcal{S}$ be the set of observables that are measured in a VQE procedure: in our case these will be Pauli operators. We will define the (non)contextuality of a VQE procedure by (non)contextuality of $\mathcal{S}$.

A *joint value assignment* is an assignment of one value (± 1) to each observable in $\mathcal{S}$. In an ontological hidden-variable model, joint value assignments correspond to *ontic states* ("real states") of a system, since they may be interpreted as definite ontological values for the observables $\mathcal{S}$. In such a model, a measurement is seen as revealing information about the ontic state, which exists independently whether it is measured or not.

A *context* on a finite dimensional Hilbert space is a set of pairwise-commuting observables whose eigenvalues uniquely specify the (shared) basis states. If $\mathcal{S}$ is a context, it is always possible to consistently assign joint values to the observables in $\mathcal{S}$, since the common eigenstates of $\mathcal{S}$ physically realize such joint value assignments. However, if $\mathcal{S}$ is not a context and contains a pair of observables that do not commute, it may be impossible to assign joint values independent of the context to be measured. Therefore, if we do assign values in this case, they must be context-dependent: hence the term "contextual."

Given any set of observables $\mathcal{S}$, let $\overline{\mathcal{S}}$ be the *closure under inference* of $\mathcal{S}$, i.e., the set of observables whose values can be predicted with certainty from an assignment of values to $\mathcal{S}$. In the language of [KWB18], $\overline{\mathcal{S}}$ is the smallest closed subtheory containing $\mathcal{S}$. The values of $\overline{\mathcal{S}}$ induced by an assignment of values to $\mathcal{S}$ may contain contradictions even if the values of $\mathcal{S}$ alone do not.

An inference occurs when for some observable A' there exists a commuting subset $\mathcal{S}' \subseteq \mathcal{S}$ such that A' is equal to the product of the operators $\mathcal{S}'$. Then since the operators $\mathcal{S}'$ may all be measured simultaneously, in any joint value assignment to $\mathcal{S} \cup \{A'\}$ the value assigned to A' must be the product of the values assigned to $\mathcal{S}'$: we therefore say that the value of A' is directly inferred from $\mathcal{S}$ [Per91]. A' may

now contribute to inferring the values of some other operators that are not directly inferred from $\mathcal{S}$. Thus in general an observable A is inferred from $\mathcal{S}$ if there is an "inference tree" that leads from $\mathcal{S}$ to A:

Definition 2.1. An *inference tree* for a Pauli operator A over a set of Pauli operators $\mathcal{S}$ is a tree whose nodes are Pauli operators and whose leaves are operators in $\mathcal{S}$, such that...

1. The root is A.
2. All children of any particular parent pairwise commute (as operators).
3. Every parent node is the operator product of its children (and thus commutes with them, by property 2).

Remark 2.1. In [KL19a] (the paper upon which this chapter is based), we referred to an observable being inferred from some other set of observables as being "determined by" that set. Correspondingly, we referred to what we are calling "inference trees" herein as "inference trees." The language has been changed in this thesis so that it is more aligned with the conventional language now used in the field.

Fig. 2.1 shows inference trees for the observables $\pm YY$ over

$$\mathcal{S} = \{XI, IX, ZI, IZ\}. \tag{2.1}$$

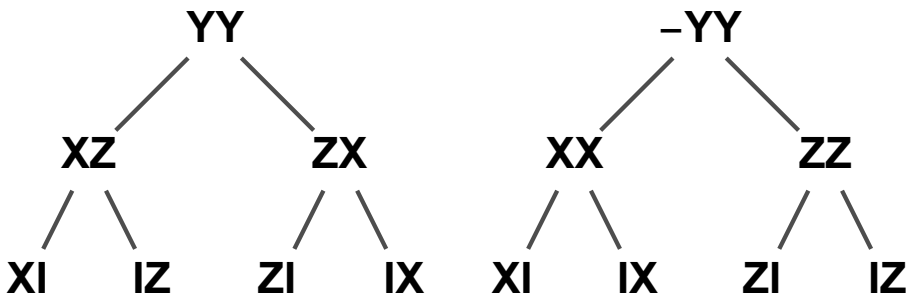
It is easy to check that these trees satisfy the properties of Definition 2.1. This example is a recasting of the classic Peres-Mermin square (see Fig. 1.3 and [Per91, Mer90, Mer93]).

The Jordan product $\frac{1}{2}\{\cdot,\cdot\}$ provides an alternative way to characterize inference. Since a pair of Pauli operators A, B either commute or anticommute,

$$\frac{1}{2}\{A, B\} = \begin{cases} AB \text{ if } [A, B] = 0, \\ 0 \text{ otherwise.} \end{cases} \tag{2.2}$$

Thus, $\overline{\mathcal{S}}$ is the closure of $\mathcal{S}$ under the Jordan product.

Figure 2.1: Inference trees for $\pm YY$ over $\{XI, IX, ZI, IZ\}$



Given an inference tree τ for a Pauli A over a set of Pauli operators $\mathcal{S}$, and a joint value assignment to $\mathcal{S}$, we may now find the inferred value for A. Let $\mathcal{L}$ be the leaves of τ; $\mathcal{L}$ may contain multiple copies of the same operator. Property 3 of an inference tree (see Definition 2.1) implies that if all of the children of some parent node in the tree have values, then the value assigned to the parent must be the product of the values assigned to its children. Hence given an assignment of values $\rho_L = \pm 1$ to each leaf $L \in \mathcal{L}$, we may propagate these values up the tree by taking products of them, so we infer that the value assigned to the root A is the product of the values assigned to the leaves:

$$\rho_A = \prod_{L \in \mathcal{L}} (\rho_L)^{m_L} = \prod_{L \in \mathcal{D}} \rho_L, \tag{2.3}$$

where the exponent m_L is the multiplicity of the operator L in $\mathcal{L}$. Therefore, we say that A is *inferred* from $\mathcal{S}$ if and only if there exists an inference tree for A over $\mathcal{S}$. This also provides a formal definition for $\overline{\mathcal{S}}$:

Definition 2.2. Given a set of Pauli operators $\mathcal{S}$, its *closure under inference* $\overline{\mathcal{S}}$ is the set of Pauli operators for which there exist inference trees over $\mathcal{S}$.

The set $\mathcal{D}$ in (2.3) is a subset of the leaves that we call the *determining set* of τ, defined as follows:

Definition 2.3. For an inference tree τ, the *determining set* is defined to be the set containing one copy of each operator with odd multiplicity as a leaf in τ. If for

some inference tree with root A, the determining set is empty, then every m_L in the first product in (2.3) must be even, so the value assigned to A is 1.

As illustrated by (2.3), the determining set is the subset of the operators included among the leaves of the tree whose values actually determine the value of the root A. Before moving on to our main results, we prove a lemma regarding the forms of inference trees:

Lemma 2.1. *Given any k-ary inference tree, there exists an equivalent binary inference tree, i.e., one that has the same leaves and root.*

Proof. A k-ary inference tree is one in which each parent has at most k children. Given any k-ary inference tree, consider any parent A with m children $\{B_1, ..., B_m\}$, for $2 < m \leq k$: then

$$A = \prod_{i=1}^{m} B_i. \tag{2.4}$$

Since the B_i all commute, B_m commutes with

$$C \coloneqq \prod_{i=1}^{m-1} B_i. \tag{2.5}$$

Therefore, we may replace $\{B_1, ..., B_{m-1}\}$ as children of A by C, which itself has $\{B_1, ..., B_{m-1}\}$ as children. A now has only 2 children, and the new node C has $m - 1$ children. We iterate this operation to obtain a cascade of parents with exactly 2 children, terminating with the parent B_1B_2, which has children B_1 and B_2. Applying this process to every parent in the original k-ary tree results in an equivalent binary inference tree, in the sense that the root and leaves are identical and it is still a valid inference tree. □

We may now state our condition for contextuality:

Definition 2.4. A set $\mathcal{S}$ of Pauli operators is *contextual* if for some Pauli A there exists an inference tree τ for A over $\mathcal{S}$ and an inference tree τ' for $-A$ over $\mathcal{S}$ such that the determining sets of τ and τ' are identical.

By (2.3), the existence of such trees implies that in any joint value assignment, the value assigned to A is both $+1$ and -1, which is a contradiction.

How does this apply to the Peres-Mermin square? Fig. 2.1 gives inference trees for $\pm YY$ over $\mathcal{S} = \{XI, IX, ZI, IZ\}$. In each tree, the set of leaves is $\mathcal{S}$ and each leaf has multiplicity 1, so the determining set for each tree is $\mathcal{S}$. Thus $\mathcal{S}$ satisfies the criteria in Definition 2.4, and is contextual. In plain language, whatever the joint value assignment to $\mathcal{S}$ is, one of the trees in Fig. 2.1 will imply that YY must be assigned the value $+1$ and the other will imply that YY must be assigned the value -1.

The criterion for contextuality in Definition 2.4 depends on a measurement operator ($A \in \overline{\mathcal{S}}$) that may or may not be an element of $\mathcal{S}$. However, for any $\mathcal{S}$ that is contextual according to Definition 2.4, we may obtain a contradiction in the assignment(s) to an operator contained in $\mathcal{S}$. This is demonstrated by the following corollary:

Corollary 2.4.1. *A set $\mathcal{S}$ of Pauli operators is contextual if and only if for some $B \in \mathcal{S}$ there exists an inference tree for $-B$ over $\mathcal{S}$, whose determining set is $\{B\}$.*

Remark 2.2. The plain language statement of the contradiction in this case is: "the value (± 1) assigned to $-B$ must be the same as the value assigned to B."

Proof. "If" follows because B can be taken to be an inference tree for itself (it is both the root and the single leaf). If the premise holds, i.e., there is also an inference tree for $-B$ whose determining set is $\{B\}$, then these two trees satisfy the criteria of Definition 2.4.

"Only if": given inference trees for $\pm A$ with the same determining set $\mathcal{D}$, we may select any operator B in $\mathcal{D}$ and construct an inference tree for $-B$ whose determining set is exactly $\{B\}$. Let the two original inference trees be τ and τ', as in Definition 2.4. Assume that τ is in binary tree form (see Lemma 2.1). Let $p = \{C_0, C_1, C_2, ..., C_d\}$ denote the path from A to B in τ, where d is the depth of τ (so $C_0 = A$ and $C_d = B$), and let C_i' be the sibling of C_i for each $i > 0$. Further, let $\{\tau_1, \tau_2, ..., \tau_d\}$ be the subtrees of τ with $\{C_1', C_2', ..., C_d'\}$ as their roots. Thus, every

leaf of τ except B itself is a leaf of exactly one of the τ_i.

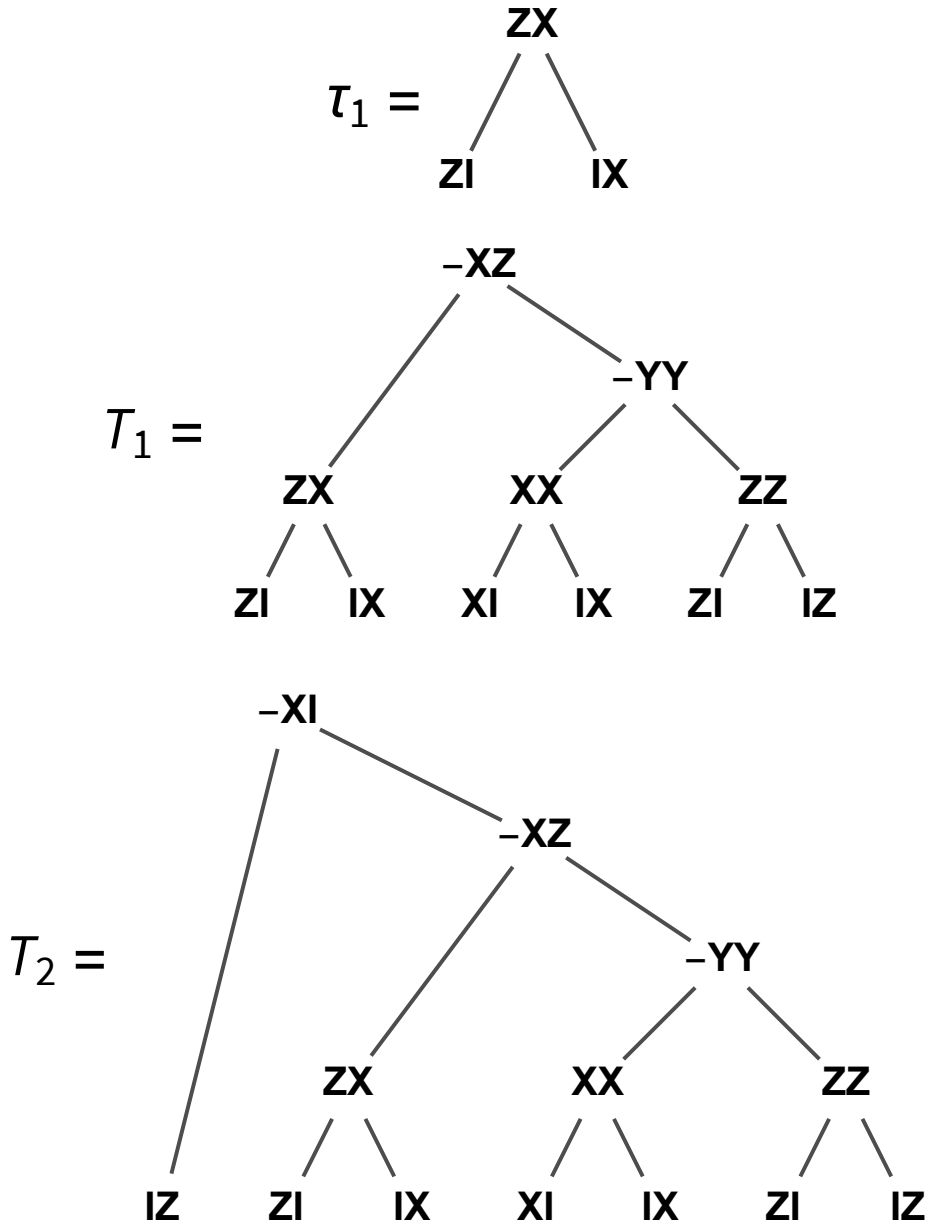
First, construct a new inference tree T_1 for $-C_1$ by letting its children be $-A$ (with τ' attached) and C_1' (with τ_1 attached). T_1 is an inference tree for $-C_1$ because $C_1 C_1' = A$ (since this is the topmost parent-children group in τ), and thus $-AC_1' = -C_1$, since C_1' is self-inverse and commutes with C_1 and A. The leaves in T_1 are the leaves of τ' as well as the leaves of τ_1. Next, construct a new inference tree T_2 for $-C_2$ by letting its children be $-C_1$ (with T_1 attached) and C_2' (with τ_2 attached). As in the previous step, $C_2 C_2' = C_1 \;\Rightarrow\; -C_1 C_2' = -C_2$, so this is a valid inference tree, and its leaves are the leaves of τ', τ_1, and τ_2. We iterate this step until we obtain an inference tree T_d for $-C_d = -B$. The leaves of T_d will be the leaves of $\tau', \tau_1, \tau_2, ..., \tau_d$. As we noted above, the set of leaves of $\tau_1, \tau_2, ..., \tau_d$ is exactly the set of leaves of τ, except for B itself, so the set of leaves of T_d is the set of leaves of τ plus the set of leaves of τ', minus exactly one copy of B. By assumption, τ and τ' have the same determining set $\mathcal{D}$. Thus, every element of $\mathcal{D}$ appears as a leaf in T_d an even number of times (as does every other leaf in τ and τ'), except for B, which appears an odd number of times. Therefore, the determining set for T_d is exactly $\{B\}$. □

We illustrate this construction using the Peres-Mermin square example. Let τ and τ' be the left and right inference trees in Fig. 2.1, and $A = YY$. Let us choose B to be XI. Then the construction requires two steps. In the first step, $C_1 = XZ$, $C_1' = ZX$: τ_1 is given by the first tree in Fig. 2.2. The resulting T_1 is an inference tree for $-C_1 = -XZ$, given by the second tree in Fig. 2.2. In the second step, $C_2 = B = XI$, $C_2' = IZ$, and τ_2 is simply the node IZ, since IZ has no children in τ. Thus T_2 is an inference tree for $-C_2 = -XI$, given by the final tree in Fig. 2.2. Notice that each operator appears twice as a leaf except for XI itself, so we have a contradiction, as expected.

A third equivalent definition of contextuality is also useful:

Corollary 2.4.2. *A set $\mathcal{S}$ of Pauli operators is contextual if and only if there exists an inference tree for $-\mathbb{1}$ (the identity observable) over $\mathcal{S}$, whose determining set is*

Figure 2.2: Example of Corollary 2.4.1: steps to construct a contextual inference tree for $-XI$, beginning with the Peres-Mermin square trees in Fig. 2.1. Note that $-XI$ is the root of T_2, the final tree, and $\{XI\}$ is the determining set, since every other leaf appears twice.



empty.

Remark 2.3. As noted in Definition 2.3, if a determining set for an operator is it means that the value assigned to that operator is 1. Hence the plain language statement of the contradiction in Corollary 2.4.2 is: "the value assigned to $-\mathbb{1}$

(whose eigenvalues are all -1) is $+1$."

Proof. "Only if": Let $\mathcal{S}$ be contextual. Then by Definition 2.4, there exists some Pauli A and inference trees τ and τ' for A and $-A$ (respectively) over $\mathcal{S}$, such that the determining sets of τ and τ' are identical. We construct an inference tree τ'' whose root is $-\mathbb{1}$ with children A and $-A$, and τ and τ' attached to these (respectively). Since the determining sets of τ and τ' are identical, each element in them appears an even number of times as a leaf in τ'', and is thus not in its determining set. Leaves of τ and τ' that are not in their determining sets are also not in the determining set of τ'', so the determining set of τ'' must be empty.

"If": Suppose there exists an inference tree τ for $-\mathbb{1}$ over $\mathcal{S}$ whose determining set is empty. By Lemma 2.1, we may take τ to be a binary tree. Then the children of $-\mathbb{1}$ in τ must be A and $-A$ for some $A \in \mathcal{S}$. Every leaf in τ is therefore either a leaf in the subtree τ' whose root is A or a leaf in the subtree τ'' whose root is $-A$. Since the determining set of τ is empty, each leaf L in τ appears an even number of times. Therefore, L either appears an even number of times in both τ' and τ'' (in which case it is in neither determining set), or an odd number of times in both (in which case it is in both determining sets). Since this argument applies to every leaf in τ, the determining sets of τ' and τ'' must be identical, and thus $\mathcal{S}$ is contextual, by Definition 2.4. □

Definition 2.4, Corollary 2.4.1, and Corollary 2.4.2 formalize the notion of contradiction in induced joint values for $\overline{\mathcal{S}}$. Since $\overline{\mathcal{S}}$ is the smallest closed subtheory containing $\mathcal{S}$, such a contradiction constitutes strong contextuality of $\mathcal{S}$.

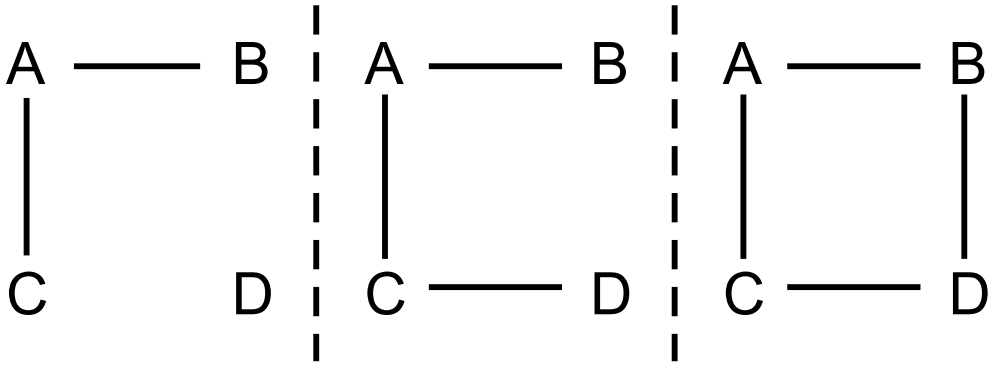
2.3 Necessary and sufficient conditions for contextuality

We now present three theorems that give necessary and sufficient conditions for measurement contextuality as defined in Definition 2.4. We will make use of the following concept:

Definition 2.5. For a set $\mathcal{S}$ of observables, the *compatibility graph* of $\mathcal{S}$ is an undirected graph whose nodes are the observables, and in which a pair of nodes is adjacent if and only if the corresponding observables commute.

Theorem 2.2. *A set of n Pauli operators is contextual if and only if it contains a subset consisting of four operators A, B, C, D whose compatibility graph has one of the forms given in Fig. 2.3.*

Figure 2.3: Compatibility graphs for contextual sets of four Pauli operators.



Remark 2.4. Theorem 2.2 implies that any set of four Pauli operators is contextual if and only if its compatibility graph is identical to one of the graphs in Fig. 2.3, and any set of three or fewer Pauli operators is noncontextual.

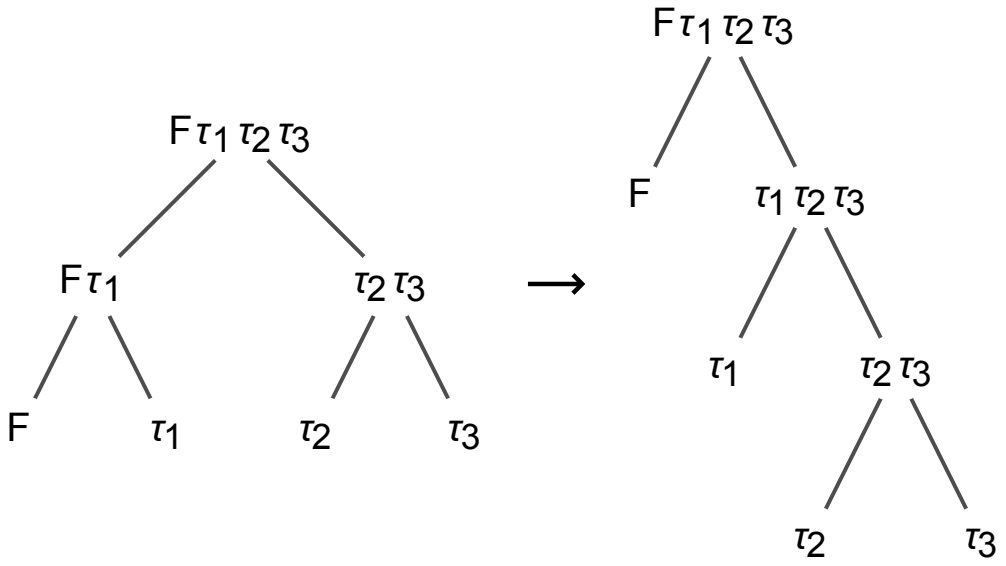
We first prove two lemmas, before proceeding to the main proof of Theorem 2.2.

Lemma 2.3. *Suppose $\mathcal{T}$ is a set of Pauli operators and τ is an inference tree over $\mathcal{T}$ with root R. If any Pauli operator F (which need not be in $\mathcal{T}$) commutes with every operator in $\mathcal{T}$, and F is a node in τ, we may construct a new inference tree τ' over $\mathcal{T}$ with root R, in which F only appears as a child of R.*

Proof. Suppose F is a node in τ that is not a child of R. Then there is a subtree of τ with depth 3 (contains 3 layers), containing F as a leaf (assume τ has been written in binary form). This subtree is the left tree in Fig. 2.4: F, τ_1, τ_2, and τ_3 may themselves be the roots of subtrees. Since F commutes with all operators in $\mathcal{T}$, it commutes with any product of them, so it commutes with $\tau_2\tau_3$. Thus since

$F\tau_1$ commutes with $\tau_2\tau_3$ (they are children of the same parent), τ_1 must commute with $\tau_2\tau_3$ as well. So, we can replace the whole subtree by the right tree in Fig. 2.4. Note that F is now a child of the root in this subtree. By repeating transformations of this form, we may move all instances of F up until they are children of R, the root of τ. □

Figure 2.4: Sequence of trees in Lemma 2.3, demonstrating that since F commutes universally, it may be moved up to become a child of the root.



Lemma 2.4. *If the compatibility graph G for a set of Pauli operators $\mathcal{T}$ is a disjoint union of cliques (complete subgraphs), then $\mathcal{T}$ is noncontextual.*

Proof. Suppose G is a disjoint union of cliques. Consider any inference tree over $\mathcal{T}$ with empty determining set and root R. Since we may take the tree to be binary, and only leaves within the same clique in G commute, if two leaves have the same parent then they must be in the same clique. For any particular commuting clique $\mathcal{C}$, the parent (product) of two operators in $\mathcal{C}$ commutes with every operator in $\mathcal{T}$, since operators not in $\mathcal{C}$ anticommute with each operator in the product. Therefore, we may use Lemma 2.3 to move all such parents up until they are children of R. The resulting tree is the first tree in Fig. 2.5, where $P_1, P_2, ..., P_k$ are parents of pairs of leaves in same clique. Since these all commute, we may group them by clique

and merge the products within cliques, obtaining the second tree in Fig. 2.5, where $C_1, C_2, ..., C_m$ are parents of sets of leaves in the same clique, with one parent per clique (let there be m cliques.) The remaining subtrees τ_1 and τ_2 are the remnants of the subtrees whose roots were the original children of R. Since all parents of pairs of leaves in the same clique have been removed from τ_1 and τ_2, no parent of leaves in τ_1 or τ_2 can be a parent of more than one leaf. But a parent of a single leaf is just that leaf, so we lose no generality by removing such leaves. Therefore, τ_1 and τ_2 cannot contain any parents of leaves, and therefore must be leaves themselves. But since they must commute, they must either be in the same clique, or one or both must be the identity (i.e., not actually present in the tree.)

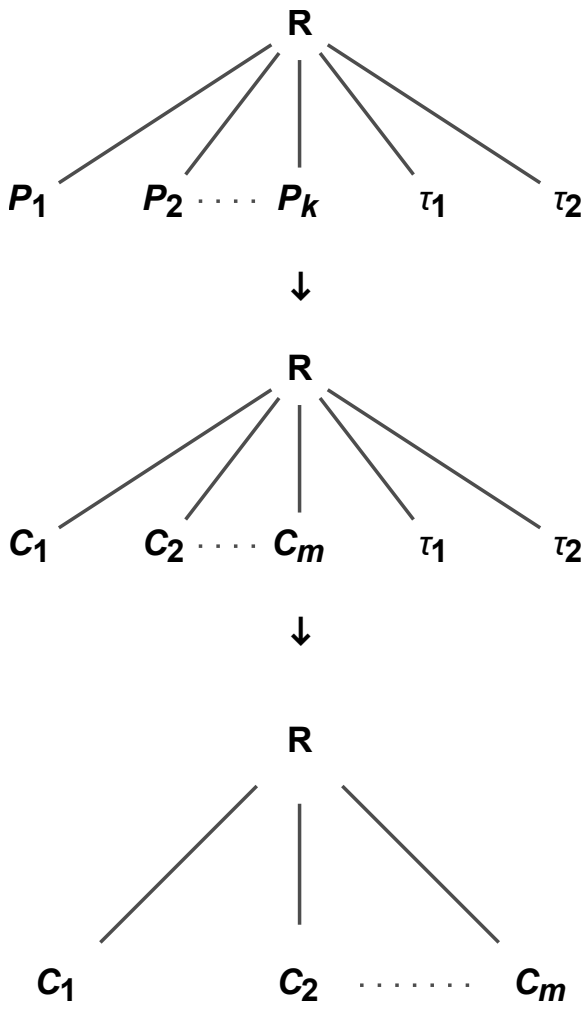
In either case, we can merge them with the product of leaves from their (shared) clique, resulting in the final tree in Fig. 2.5, where $C_1, C_2, ..., C_m$ are parents of sets of leaves in the same clique, with one parent per clique. If τ_1 and τ_2 are not present, then the tree already has this form. Since each C_i is the product of all the leaves from the same clique, and each leaf must appear an even number of times (since the determining set is empty), every C_i is $\mathbb{1}$, and thus the root of the tree is $\mathbb{1}$ as well. Therefore, since this holds for any inference tree over $\mathcal{T}$ with empty determining set, $\mathcal{T}$ is noncontextual, by Corollary 2.4.2. □

Remark 2.5. Fully-commuting or fully-anticommuting sets of Pauli operators are trivial examples of sets whose compatibility graphs are disjoint unions of cliques. A less trivial example is the set $\{XX, ZZ, XZ, ZX\}$, in whose compatibility graph $\{XX, ZZ\}$ and $\{XZ, ZX\}$ are disjoint cliques (the reader may recognize these cliques as the intermediate layers in the two inference trees corresponding to the Peres-Mermin square in Fig. 2.1.) Thus Lemma 2.4 indicates that this set on its own is noncontextual.

We now finally complete the proof of Theorem 2.2:

Proof. "If": If a set of Pauli operators contains a subset that is contextual, then the full set is also contextual, since if values cannot be assigned to the subset then they

Figure 2.5: Sequence of trees in Lemma 2.4. In the first tree, the P_i are parents of pairs of leaves, and τ_1, τ_2 are subtrees that are remnants of the original children of R. In the second tree, the C_i are parents of leaves from within the same commuting clique, with one C_i for each clique. In the third tree, the τ_i have been absorbed into the C_i.



cannot be consistently assigned to the full set. Therefore, in order to show that any set of Pauli operators containing a subset whose compatibility graph has one of the forms Fig. 2.3 is contextual, it is enough to show that the subset itself is contextual. To do this, for each compatibility graph in Fig. 2.3 we construct an inference tree for $-\mathbb{1}$ with empty determining set over a set of Pauli operators with that compatibility graph. The three compatibility graphs and their corresponding trees are shown in

Fig. 2.6 to 2.8. Using the compatibility graphs, one can check that the inference trees are valid (i.e., satisfy the three conditions in Definition 2.1), and thus each of the corresponding sets of Pauli operators is contextual, by Corollary 2.4.2.

"Only if": Let $\mathcal{S}$ be a contextual set of Pauli operators. Then there exists an inference tree τ for $-\mathbb{1}$ over $\mathcal{S}$, with empty determining set, by Corollary 2.4.2. Let $\mathcal{T} \subseteq \mathcal{S}$ be the set of leaves of τ.

By Lemma 2.3, we may move all instances of any leaf $C \in \mathcal{T}$ that commutes with every other operator in $\mathcal{T}$ up until they are children of the root. Since C is self-inverse and must have even multiplicity (since the determining set of τ is empty), all of its instances now cancel, so we may remove them from the tree. Repeating this process for every leaf that commutes with every operator in $\mathcal{T}$ results in a new tree in which every leaf anticommutes with at least one other, so we assume that this is now the case.

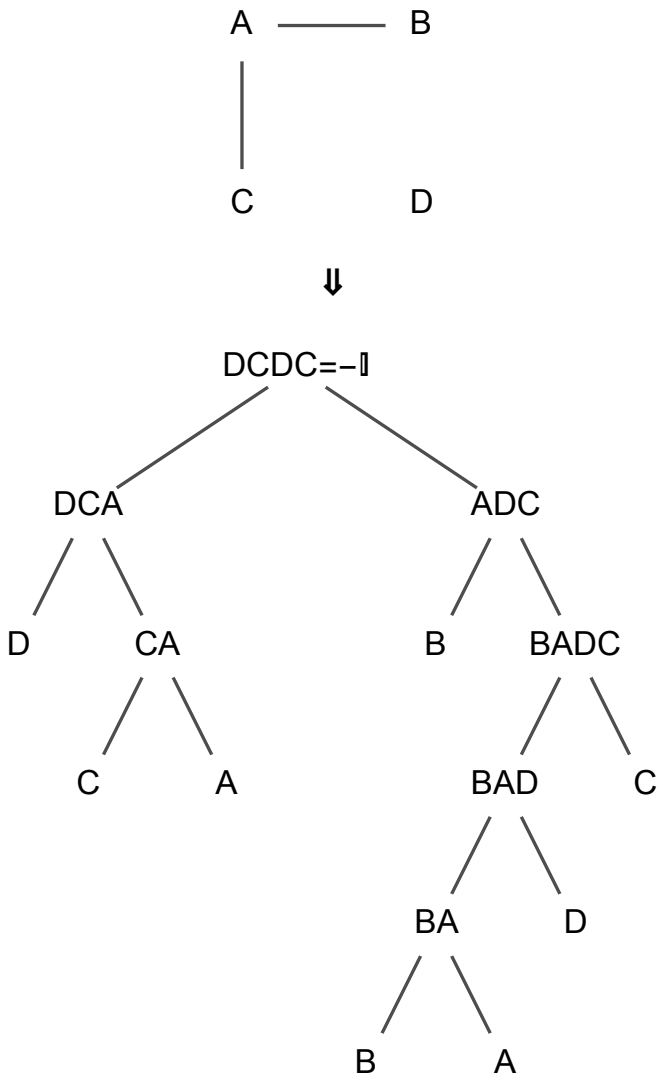
Since $\mathcal{T}$ is contextual, by Lemma 2.4 the compatibility graph of $\mathcal{T}$ cannot be a disjoint union of cliques. Therefore, commutation is not an equivalence relation over $\mathcal{T}$, so there must exist $A, B, C \in \mathcal{T}$ such that A commutes with B and C, but B and C anticommute. We argued above we may take τ to be such that every operator in $\mathcal{T}$ anticommutes with at least one other. Thus, must exist $D \in \mathcal{T}$ such that A anticommutes with D. Therefore, the compatibility graph of $\{A, B, C, D\} \subseteq \mathcal{S}$ must have one of the contextual forms given in Fig. 2.3 (up to swapping B and C in the second form). □

Although Theorem 2.2 relies on the presence or absence of sets of four Pauli operators with the compatibility graphs in Fig. 2.3, it turns out that we can test for contextuality by searching over subsets of size at most three. This is based on our final main theorem, which follows from Theorem 2.2:

Theorem 2.5. *For a set $\mathcal{S}$ of Pauli operators, let $\mathcal{T} \subseteq \mathcal{S}$ be the set obtained by removing any operator that commutes with all others in $\mathcal{S}$. Then $\mathcal{S}$ is noncontextual if and only if commutation is an equivalence relation on $\mathcal{T}$.*

Remark 2.6. That commutation is not transitive in general is a non-classical prop-

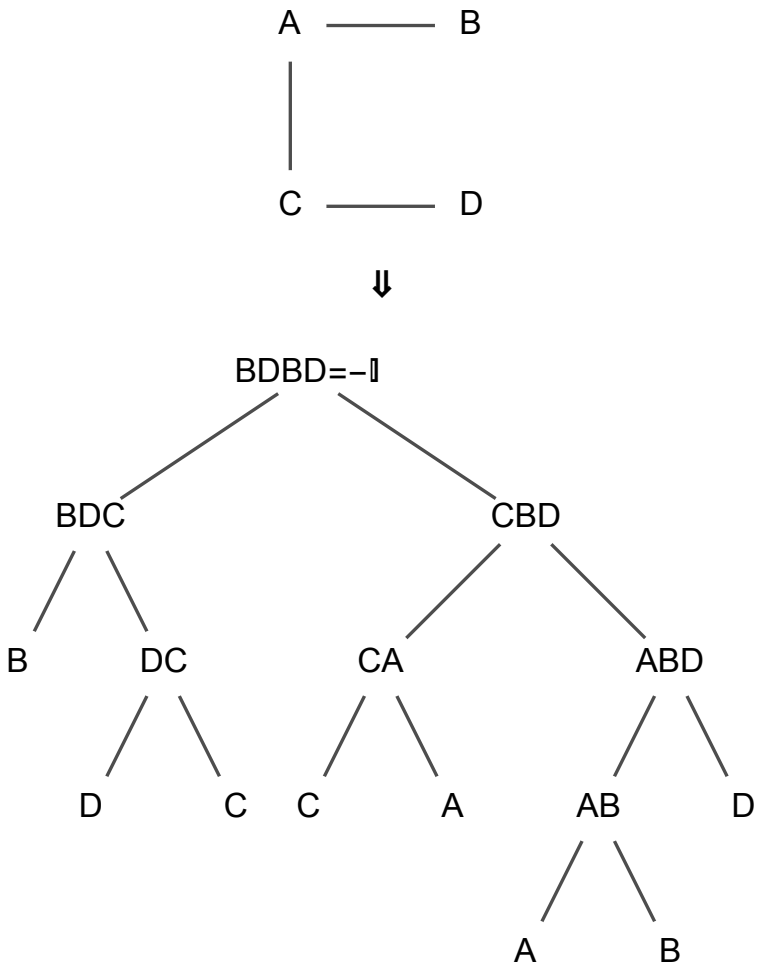
Figure 2.6: First compatibility graph in Fig. 2.3, together with its contextual inference tree. Note that each leaf in the tree appears twice, so the determining set is empty.



erty. Operators that commute with all others in the set cannot contribute to contextuality (see Lemma 2.3); after removing these, nontransitivity of commutation is equivalent to contextuality.

Proof. "If": if there exists any inference tree τ over $\mathcal{S}$ for $-\mathbb{1}$ with empty determining set, then as noted in the "Only if" portion of the proof of Theorem 2.2, by Lemma 2.3 we may obtain an equivalent inference tree over $\mathcal{T}$ (i.e., an inference tree for $-\mathbb{1}$ whose determining set is also empty, and whose leaves are in $\mathcal{T}$.) Thus, the

Figure 2.7: Second compatibility graph in Fig. 2.3, together with its contextual inference tree. Note that each leaf in the tree appears twice, so the determining set is empty.

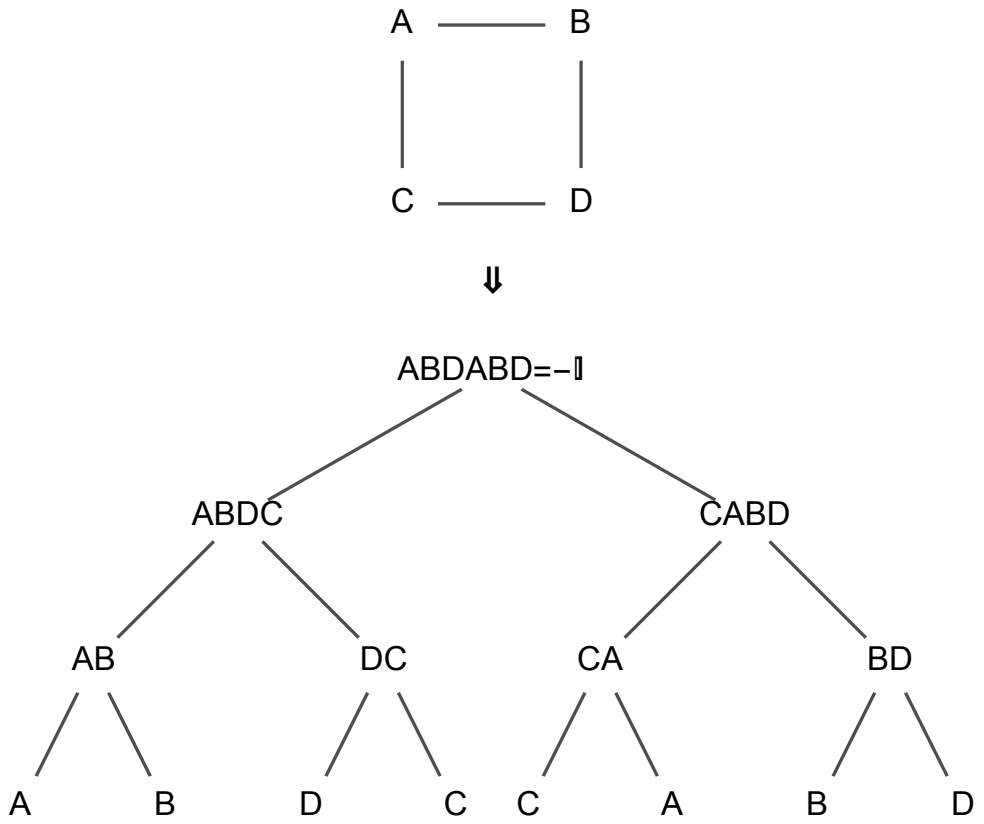


(non)contextuality of $\mathcal{S}$ is identical to the (non)contextuality of $\mathcal{T}$. By Lemma 2.4, if $\mathcal{T}$ is a disjoint union of cliques, then it is noncontextual, and thus by the above argument, $\mathcal{S}$ is as well.

"Only if": by the argument that concludes the proof of Theorem 2.2, if commutation is not an equivalence relation on $\mathcal{T}$, then there is a subset of $\mathcal{T}$ whose compatibility graph has one of the forms Fig. 2.3. By Theorem 2.2, any such subset is contextual, and thus $\mathcal{T}$ is as well. □

Let us see a few examples of noncontextual sets. Denote by $\mathcal{Z}$ the subset of

Figure 2.8: Third compatibility graph in Fig. 2.3, together with its contextual inference tree. Note that each leaf in the tree appears twice, so the determining set is empty.



$\mathcal{S}$ containing all operators in $\mathcal{S}$ that commute with all other operators in $\mathcal{S}$, so that

$$\mathcal{S} = \mathcal{Z} \cup \mathcal{T} \tag{2.6}$$

for $\mathcal{T}$ as defined in Theorem 2.5. Then by Theorem 2.5, $\mathcal{S}$ is noncontextual if and only if

$$\mathcal{T} = C_1 \cup C_2 \cup \cdots \cup C_N, \tag{2.7}$$

where the C_i's are equivalence classes of commutation, i.e., any pair of operators in the same C_i commute but any operator in C_i does not commute with any operator in C_j for $j \neq i$. In terms of this, three examples of noncontextual sets are...

1. Any commuting set of Pauli operators is noncontextual (this is the special case

where $\mathcal{T} = \emptyset$).

2. Any anticommuting set of Pauli operators is noncontextual (this is the special case where $\mathcal{Z} = \emptyset$ and each C_i contains one operator).

3. Generalizing example 2, any set of Pauli operators on which commutation is transitive is noncontextual (this is the special case where $\mathcal{Z} = \emptyset$).

As claimed above, Theorem 2.5 provides an algorithm for testing contextuality of a set $\mathcal{S}$ of Pauli operators in $O(|\mathcal{S}|^3)$ time. First, remove any operators from $\mathcal{S}$ that commute with all others (searching for these takes $O(|\mathcal{S}|^2)$ time) and let $\mathcal{T}$ denote the remaining set (as in Theorem 2.5). Then, search in $\mathcal{T}$ for a set of three operators A, B, C such that A commutes with B and C, but B and C anticommute. Such a set exists if and only if commutation is nontransitive on $\mathcal{T}$, which by Theorem 2.5 implies that such a set exists if and only if $\mathcal{S}$ is contextual. There are $O(|\mathcal{S}|^3)$ subsets of size three in $\mathcal{S}$, so this is the runtime for the search. In many VQE procedures some structure on the set $\mathcal{S}$ is known, which may improve the efficiency of determining whether it is contextual.

It is also worth noting that our definitions and criteria for contextuality only depend on the product and commutation relations of the Pauli operators, so they will apply for any sets of operators whose product and commutation relations are isometric to those of the Pauli operators. In particular, the Majorana operators

$$b_j^\dagger + b_j, \quad i(b_j^\dagger - b_j), \tag{2.8}$$

constructed from the fermionic creation and annihilation operators $b_j^\dagger$ and b_j on a set of fermionic modes $j = 0, 1, 2, ..., n-1$, as well as any products of Majorana operators, are isomorphic to Pauli operators under any of the standard fermion-to-qubit mappings (see Section 1.3.1). This means that our criteria can be applied directly to Majorana operators, providing a method for evaluating contextuality of fermionic Hamiltonians that does not require mapping to qubits and Pauli operators (although it will always have the same output as would first mapping to qubits and

then evaluating contextuality).

2.4 Quantifying contextuality

Can we extend our evaluation procedure to a quantifier of the amount of contextuality present in a contextual Hamiltonian H? One natural quantifier of the contextuality of H is obtained by evaluating the distance from H to any noncontextual Hermitian operator, as suggested in [DA18]. Any choice of operator norm will induce such a quantifier:

$$\mathrm{CSep}_{\mathcal{N}}(H) \coloneqq \min_{H'} \left(\frac{\|H - H'\|_{\mathcal{N}}}{\|H\|_{\mathcal{N}}} \right), \tag{2.9}$$

where H' is any noncontextual Hermitian operator, and $\|\cdot\|_{\mathcal{N}}$ is some operator norm. We call this quantifier the *contextual separation* or $\mathrm{CSep}_{\mathcal{N}}$, where $\mathcal{N}$ identifies the norm that the quantifier is defined with respect to.

Let a *decontextualizing set* $\mathcal{S}'$ be any subset of $\mathcal{S}$ such that $\mathcal{S} \setminus \mathcal{S}'$ is noncontextual. Then we may define another quantifier of contextuality as the minimum of $\sum_{j \in \mathcal{S}'} |h_j|$ over all decontextualizing sets $\mathcal{S}'$ (the h_j are the coefficients of the Pauli terms in the Hamiltonian, as in (1.19)). This quantifier provides an upper bound on the error in the energy estimate induced by "decontextualizing" the Hamiltonian.

We can show that this cost of decontextualization is in fact the special case of contextual separation where the norm is chosen to be the Hilbert-Schmidt norm. The Pauli operators are a Hilbert-Schmidt orthogonal basis for the Hermitian operators. Thus, any noncontextual set of Pauli operators defines a subspace of the Hermitian operators. Any noncontextual Hamiltonian H' is an element of one of these subspaces, so the minimum in (2.9) is achieved by setting H' to be the maximal projection of H onto any noncontextual subspace. Since H in the form (1.19),

$$H = \sum_i h_i \mathcal{P}_i, \tag{2.10}$$

is written as a vector with components h_i in the coordinate system defined by the Pauli operators $\mathcal{P}_i$, the projection of H onto the subspace spanned by a set $\mathcal{T}$ of

Pauli operators is

$$H' = \sum_{\mathcal{P}_i \in \mathcal{T}} h_i \mathcal{P}_i. \tag{2.11}$$

Therefore, if we take the norm in (2.9) to be the Hilbert-Schmidt norm, then (2.9) may be written

$$\mathrm{CSep}_{\mathrm{HS}}(H) = \min_{\mathcal{T} \subseteq \mathcal{S}} \left(\frac{\| \sum_{\mathcal{P}_i \notin \mathcal{T}} h_i \mathcal{P}_i \|}{\| \sum_i h_i \mathcal{P}_i \|} \right), \tag{2.12}$$

where $\mathcal{S} = \{\mathcal{P}_i\}$ (as above) and $\mathcal{T}$ is any noncontextual subset of $\mathcal{S}$. More conveniently,

$$\mathrm{CSep}_{\mathrm{HS}}(H) = \min_{\mathcal{S}' \subseteq \mathcal{S}} \left(\frac{\| \sum_{\mathcal{P}_i \in \mathcal{S}'} h_i \mathcal{P}_i \|}{\| \sum_i h_i \mathcal{P}_i \|} \right) = \min_{\mathcal{S}' \subseteq \mathcal{S}} \sqrt{\frac{\sum_{\mathcal{P}_i \in \mathcal{S}'} h_i^2}{\sum_i h_i^2}}, \tag{2.13}$$

where $\mathcal{S}'$ is any decontextualizing set, as defined in the main text.

Let us define $\vec{h} \coloneqq (h_1, ..., h_m)$, and let $\vec{h}'$ be the projection of $\vec{h}$ onto the span of any subset of the standard basis such that the support of $\vec{h} - \vec{h}'$ corresponds to a noncontextual subset of $\mathcal{S}$. (In other words, the set of observables corresponding to the support of $\vec{h}'$ is a decontextualizing set.) Then (2.13) assumes the useful form

$$\mathrm{CSep}_{\mathrm{HS}}(H) = \min_{\vec{h}'} \sqrt{\frac{\vec{h}'^2}{\vec{h}^2}} = \min_{\vec{h}'} \left(\frac{\| \vec{h}' \|_2}{\| \vec{h} \|_2} \right). \tag{2.14}$$

This suggests another name for contextual separation with Hilbert-Schmidt norm: the *contextual 2-distance*, since (2.14) is the (scaled) 2-distance between the vector $\vec{h}$ and its maximal noncontextual projection. We may then generalize to the *contextual p-distance*:

$$\mathrm{CD}_p(H) = \min_{\vec{h}'} \left(\frac{\| \vec{h}' \|_p}{\| \vec{h} \|_p} \right), \tag{2.15}$$

of which contextual separation is the special case for $p = 2$.

We thus have a family of measures of contextuality for any Hermitian operator. The contextual 1-distance $\mathrm{CD}_1(H)$ is the minimum absolute fractional weight of any decontextualizing set. Thus (as noted in the main text) it has a physical interpretation as an upper bound on the fractional error induced in the energy estimate

by "decontextualizing" the procedure. As noted above, $\mathrm{CD}_2(H)$ is the minimum Hilbert-Schmidt distance of the Hamiltonian from a noncontextual Hamiltonian. For the contextual p-distances for $p > 2$ we do not have such simple physical interpretations, although $\mathrm{CD}_\infty(H)$ is the minimum over all decontextualizing sets $\mathcal{T}$ of the maximum h_i associated to $\mathcal{T}$, as a fraction of the maximum h_i over the entire Hamiltonian.

Prior measures of contextuality include the contextual fraction (CF) (in [AB11, ABM17, MK18, DA18]), relative entropy of contextuality (REC), mutual information of contextuality (MIC), and contextual cost (CC) (all in [GHH$^+$14]), and rank of contextuality (RC) [HZS$^+$22]. In a strict sense our contextual p-distance is a complementary measure to CF and CC, both of which measure the fraction of an empirical model that must be strongly contextual. In particular, Proposition 6.3 in [AB11] states that CF = 1 if and only if the model is strongly contextual, which means that $\mathrm{CD}_p > 0$ if and only if CF = 1 (and correspondingly, CC = 1). Thus CD_p and CF/CC vary over disjoint regions in the space of empirical models. MIC and REC are shown to be equal in [GHH$^+$14], and are related to contextuality as a resource for communication.

Rank of contextuality is the measure most closely related to our CD_p, being the minimum number of noncontextual empirical models ("boxes" in the terminology of [HZS$^+$22]) required to simulate the system of interest. However, rank of contextuality and CD_p are not even necesarily monotonically related, since an adversary could construct a Hamiltonian for which many noncontextual boxes are required for an exact description, but for which the weights of all but one of these boxes are arbitrarily small, thus giving a low CD_p. It is possible that one could define a weighted version of the rank of contextuality that would avoid this problem, or that rank of contextuality might have some other operational meaning in the variational quantum eigensolver, but we do not pursue this herein.

Calculating CD_p via compatibility graphs involves an optimization problem over subgraphs that are disjoint unions of cliques, by Theorem 2.5. Thus evaluating CD_p by strictly graph-theoretic methods is a variant of the clique problem, and

is therefore likely to be NP-complete, so any efficient method for evaluating the contextual p-distance will have to take advantage of the structure of commutation relations that goes beyond compatibility graphs. Finding such a method is an open question.

In [AB11, RSKanK12], the authors do not require that the set of measurements forms a closed subtheory. As a result, our criterion in Theorem 2.2 is different from Proposition 1 in [RSKanK12] (originally proven in a non-quantum setting in [Vor63, Vor67]), which states that a set of measurements admits a joint probability distribution if its compatibility graph is chordal [2]. Note that the first two graphs in Fig. 2.3 are chordal, and would thus be classified as noncontextual by Proposition 1 in [RSKanK12].

The criterion of [RSKanK12] is valuable in capturing strong contextuality as it may exist strictly internally in a set of measurements. In [KWB18] it is demonstrated that for a quantum procedure the efficiency of classical simulation is limited by the presence of contextuality, as noted above: efficient simulation by sampling from the discrete Wigner function is only possible in the absence of contextuality. In showing this, as noted in the introduction the authors assume that their sets of measurements are closed subtheories [KWB18, pp. 1-2], which means exactly that all elements of $\overline{\mathcal{S}}$ must be included. Thus the condition for contextuality we have developed is that upon which their argument is based.

In addition to providing a connection to the simulability results in [KWB18], requiring that the set of measurements be a closed subtheory is important when interpreting a noncontextual joint probability distribution as an ontological hidden-variable theory. If a joint value assignment to operators including a commuting pair A, B does not imply the corresponding assignment to AB, it is difficult to interpret the original assignment as an ontic state of the system. Indeed, such a state is manifestly contextual in the sense that the ontic values can only apply if certain measurements are disallowed. Impossibility of a local-realistic hidden-variable theory for a set of measurements is commonly regarded as equivalent to

[2] A chordal graph contains no induced cycles with length greater than 3.

contextuality of that set (see the introductory discussions in [RSKanK12, HWVE14, GHH$^+$14, CKB15, XC19, Rau19], for example), but apparently we must be careful in associating the two.

The converse of Proposition 1 of [RSKanK12] is proven in [XC19]: this implies that since the third compatibility graph in Fig. 2.3 (the 4-cycle) is nonchordal, unlike the other two it is contextual both by our condition and by that of [XC19]. However, [XC19] uses the fact that a 4-cycle compatibility graph is equivalent to the CHSH scenario [CHSH69, Fin82, AQB$^+$13], and so the contradiction for this case is derived from violation of an inequality rather than directly from value assignability.

2.5 Evaluation of Contextuality in VQE experiments to date

Small scale VQE experiments have already been performed in numerous systems [DXP$^+$10, LWG$^+$10, PMS$^+$14, WDB$^+$15, OBK$^+$16, SZZ$^+$17, PGS$^+$17, KMT$^+$17, HMR$^+$18, SWG$^+$18, CRD$^+$18, DMH$^+$18, KMvB$^+$19, KTC$^+$19, GEB$^+$19, SBMS$^+$19, MPJ$^+$19, SM19, NCP$^+$20, AAB$^+$20, KJK$^+$21a, KJK$^+$21b, LYTT21, KGL$^+$22]. We use the methods in Section 2.3 to assess contextuality of the Hamiltonians in a subset of these. Some other details of these experiments are given in Table 2.2. The results are summarized in Table 2.1, in which we also give CD_0 (see Section 2.4), the minimum size of any decontextualizing set as a fraction of the total number of terms. Note that each simulation of H_2 in the STO-3G minimal basis is noncontextual. This is not surprising if one considers these simulations as encoding a two-dimensional Hilbert space spanned by a bonding and antibonding state, i.e., a single qubit, for which Bell gave a noncontextual hidden-variable theory [Bel66]. We choose to present CD_0 because, unlike the CD_p for $p \geq 1$, it is independent of the Hamiltonian coefficients, and many of the experiments given in Table 2.1 use ranges of values for the coefficients.

As noted in Section 2.4, calculating CD_p for any p (including $p = 0$) is in general hard, since it involves an optimization over subsets of the terms in

Citation:	System:	Contextual?	CD_0	$\lvert\mathcal{S}\rvert$
Dumitrescu *et al.* [DMH$^+$18]	Deuteron	No	0	—
Kandala *et al.* [KMT$^+$17]	H_2	No	0	4
O'Malley *et al.* [OBK$^+$16]	H_2	No	0	5
Hempel *et al.* [HMR$^+$18]	H_2 (BK)	No	0	5
Hempel *et al.* [HMR$^+$18]	H_2 (JW)	No	0	14
Colless *et al.* [CRD$^+$18]	H_2	No	0	5
Kokail *et al.* [KMvB$^+$19]	Schwinger Model	Yes	$\sim$0.14	231
Nam *et al.* [NCP$^+$20]	H_2O	Yes	0.27	22
Hempel *et al.* [HMR$^+$18]	LiH	Yes	0.33	13
Peruzzo *et al.* [PMS$^+$14]	HeH^+	Yes	0.38	8
Kandala *et al.* [KMT$^+$17]	BeH	Yes	$\sim$0.74	164
Kandala *et al.* [KMT$^+$17, KTC$^+$19]	LiH	Yes	$\sim$0.77	99

Table 2.1: Evaluation of contextuality in VQE experiments. CD_0 is the minimum number of terms we must remove from the Hamiltonian to reach a noncontextual set, as a fraction of the total number of terms ($|\mathcal{S}|$). In [DMH$^+$18], Dumitrescu *et al.* compute the ground-state energies of the deuteron for effective field theories with dimension 2^N for $N = 1, 2, 3$ ($|\mathcal{S}| = 1, 4, 3$), and extrapolate from these to the infinite-dimensional space. Thus, $\mathcal{S}$ is different for each value of N. For $N = 1$, $\mathcal{S} = \{Z\}$. For $N = 2$, $\mathcal{S} = \{ZI, IZ, XX, YY\}$. For $N = 3$, $\mathcal{S} = \{IZ, XX, YY\}$. All of these are noncontextual.

the Hamiltonian. For the contextual experiments we consider here, however, we may find CD_0 by brute force search for those with fewer terms (namely, [PMS$^+$14, HMR$^+$18, NCP$^+$20]), and those with larger numbers of terms (namely, [KMvB$^+$19, KMT$^+$17, KTC$^+$19]), the compatibility graphs are sufficiently structured to enable a greedy approximation of CD_0. In particular, all of these sets of terms contain commuting subsets that comprise substantial fractions of the full sets. Therefore, we approximate the largest noncontextual subset by first including the largest commuting subset, then including the largest possible second clique (i.e., a commuting subset that anticommutes with some subset of the first commuting set), then including the largest possible third clique, and so forth. This greatly restricts the number of noncontextual subsets we have to consider, and renders the optimization tractable. We expect this heuristic to give a good approximation to the largest noncontextual subset when the approximate largest noncontextual subsets thus found are of comparable size to the full set of terms: this is the case for the Hamiltonians in [KMvB$^+$19, KMT$^+$17, KTC$^+$19]. We also find that for the Hamiltonians

in [PMS+14, HMR+18, NCP+20], for which we obtained the exact largest noncontextual subsets, our heuristic approach also finds the exact solutions.

Our heuristic approach may still be inefficient if it is hard to find the largest commuting subset of the set of terms, but for the Hamiltonians in [KMvB+19, KMT+17, KTC+19] this task turns out to be simple. In [KMvB+19] for large N nearly all of the terms are diagonal ($\frac{1}{2}N^2 - \frac{1}{2}N + 3$ terms are diagonal, while $2(N-1)$ are not). In the LiH Hamiltonian in [KMT+17, KTC+19] the diagonal terms form a maximal commuting set (for n qubits there can be no more than $2^n - 1$ non-identity commuting Pauli operators). Finally, in the BeH_2 Hamiltonian in [KMT+17] the diagonal terms form a maximal commuting set minus one element (and no other commuting subset is larger).

Citation:	System	$\underline{\mathcal{S}}$	Error
[DMH+18]	Deuteron	various	$< 3\%$
[KMT+17]	H_2	$\{ZI, IZ, XX\}$	< 1.6 mHa
[OBK+16]	H_2	$\{ZI, IZ, XX\}$	< 1.3 mHa
[HMR+18]	H_2 (BK)	$\{ZI, IZ, XX, YY\}$	$F > 0.99$
[HMR+18]	H_2 (JW)	$\{Z_i \vert i = 1, ..., 4\} \cup$ $\{XXYY, YYXX, YXXY, XYYX\}$	$F > 0.97$
[CRD+18]	H_2	$\{ZI, IZ, XX, YY\}$	various
[KMvB+19]	L.S.M.	$\{X_iX_{i+1}, Y_iY_{i+1}, Z_i \vert i =$ $1, 2, ..., 19\} \cup \{Z_{20}\}$	$F > 0.8$
[NCP+20]	H_2O	$\{Z_i \vert i = 1, ..., 5\} \cup$ $\{X_1X_2, Y_1Y_2, X_3X_4, X_3X_5,$ $X_4X_5, Y_3Y_4, Y_3Y_5, Y_4Y_5\}$	< 1.6mHa
[HMR+18]	LiH	$\{X_i, Z_i \vert i = 1, 2, 3\}$	various
[PMS+14]	HeH^+	$\{X_i, Z_i \vert i = 1, 2\}$	3.05mHa
[KMT+17]	BeH	$\{X_i, Z_i \vert i = 1, ..., 6\}$	< 1.6mHa
[KMT+17]	LiH	$\{X_i, Z_i \vert i = 1, ..., 4\}$	< 1.6mHa

Table 2.2: Details of VQE experiments tested for contextuality in Table 2.1. $\underline{\mathcal{S}}$ is a set sufficient to generate the noncontextual part of the Hamiltonian via inference trees (there is not space to represent all of the terms in the noncontextual parts of the Hamiltonians in this table). For brevity, we represent some operators in this table as Pauli operators with subscripts: the subscript indicates which qubit the operator acts on. So for example, X_2X_3 means IXX (in a three-qubit system.) "Error" refers to the error achieved in the estimated ground state resulting from the VQE experiments, which is either represented as a fidelity of the ground state itself or in terms of error in the ground state energy either in Hartree or in percent (the errors are all simply those reported in the various papers). Note that an error of 1.6 mHa is chemical accuracy.

2.6 Discussion

All VQE procedures that have been implemented to date, whether noncontextual or contextual, have been small enough to simulate classically. The purpose of such experiments is not to demonstrate quantum advantage, but to apply current hardware to small examples of real-world applications. Such efforts have been instrumental in developing both experimental and theoretical capabilities; indeed, VQE itself was developed in this context [PMS$^+$14].

For these reasons, we should be clear that our classification of these experiments as contextual or noncontextual is not a judgement of the value of the experiments, but rather a constructive categorization whose purpose is to inform future experiments and theoretical work. As noted above, we may regard a noncontextual Hamiltonian as an instance of an essentially classical problem, akin to quantum algorithms for explicitly classical problems as in QAOA [FGG14] (note that QAOA's diagonal Hamiltonians are always noncontextual.) In spite of this last point, a noncontextual Hamiltonian may be hard to simulate classically, since classical problems can be classically hard. However, contextuality in a VQE procedure provides a strict separation between it and any classical algorithm, by ruling out the existence of a description of the problem in terms of joint probability distributions over a classical phase space, and thus precluding any classical approach either explicitly or implicitly based on such distributions. We suggest therefore that future VQE implementations, even at small scales, should focus on contextual Hamiltonians, according to the criteria we have developed.

Chapter 3

Classical simulation of noncontextual Hamiltonians

3.1 Introduction

In Chapter 2, we presented an efficiently-testable, necessary and sufficient condition for contextuality of an arbitrary set of Pauli operators. We defined a Hamiltonian to be noncontextual if its Pauli decomposition is noncontextual according to the condition for sets of Pauli operators. Given that contextuality is defined by ruling out classical, noncontextual hidden-variable models, we should expect that when a Hamiltonian is noncontextual, such a model can be constructed and a classical simulation method for the Hamiltonian can be implemented based on the model. In this chapter, we show how to construct such models for arbitrary noncontextual Hamiltonians and discuss the implications for their classical simulability. This chapter is based on [KL20].[1]

In particular, we show that when a Hamiltonian is noncontextual according to the criterion of Chapter 2 [KL19a], we may construct a *quasi-quantized* classical model of the associated VQE instance. Quasi-quantized models are descriptions of classical subtheories of quantum mechanics, such as the stabilizer subtheory for odd-

[1] ©2020 American Physical Society, reproduced per Author's Rights under the transfer of copyright agreement (`https://journals.aps.org/authors/transfer-of-copyright-agreement`).

dimensional qudits, and Gaussian subtheory for continuous systems [Spe16, Spe07, vE07, Spe08, BRS12].

Our quasi-quantized model is a hidden-variable theory for prepare-and-measure scenarios where arbitrary states are allowed but the measurements must correspond to a noncontextual set of Pauli observables. This contrasts with [RBVT$^+$20], in which it is shown that for noncontextual states (in the sense of being positively representable over a noncontextual phase space), any set of Pauli measurements permits classical description.

The problem of approximating the ground state energy of a classical or quantum Hamiltonian can be NP-complete or QMA-complete, respectively [Bar82, KSV02, WB03, KR03, BV05, KKR06, OT08, BL08, AE11, Sch11, YB12, AE15]. Our quasi-quantized model allows us to show that the NONCONTEXTUAL HAMILTONIAN PROBLEM (see Section 3.1.1) is only NP-complete, rather than QMA-complete [KSV02]. In other words, the fact that we can describe the VQE procedure in terms of nonnegative joint probabilities places noncontextual VQE unambiguously in the realm of the classical, and this result extends to the computational complexity of the problem.

In Section 3.2, we show how to construct a quasi-quantized model for any noncontextual Hamiltonian. The states of the quasi-quantized model are probability distributions that correspond to quantum states. We prove that these probability distributions reproduce expectation values for the Hamiltonian terms, including the expectation values corresponding to Hamiltonian eigenstates. In Section 3.3, we use our model to construct a classical simulation algorithm for VQE. We also show that the NONCONTEXTUAL HAMILTONIAN PROBLEM is NP-complete. In Section 3.4, we summarize and discuss our results.

3.1.1 Local Hamiltonian Problems

The k-LOCAL HAMILTONIAN PROBLEM is the decision problem of whether the ground state energy of a k-local Hamiltonian lies below some specified energy gap (with size at least $1/\text{poly}(n)$, for n qubits), given the promise that the ground state does not lie

within the gap [KSV02]. The k-LOCAL HAMILTONIAN PROBLEM is QMA-complete, and has also been studied for various subsets of Hamiltonians (such as k-local commuting Hamiltonians, and Hamiltonians with specific interaction graphs) [Bar82, KSV02, WB03, KR03, BV05, KKR06, OT08, BL08, AE11, Sch11, YB12, AE15]. The complexity of the k-LOCAL COMMUTING HAMILTONIAN PROBLEM is a long-standing open problem [BV05, AE11, Sch11, YB12, AE15].

In this paper we focus on Pauli Hamiltonians. The complexity of the corresponding commuting problem is known: the k-LOCAL COMMUTING PAULI HAMILTONIAN PROBLEM is in NP, since it is a special case of the k-LOCAL COMMUTING, FACTORIZABLE QUBIT HAMILTONIAN PROBLEM [BV05] [2]. Since all commuting Pauli Hamiltonians are noncontextual, we will generalize this by proving that the NONCONTEXTUAL HAMILTONIAN PROBLEM is NP-complete.

3.1.2 Noncontextual Hamiltonians

For $\mathcal{S}$ the set of Pauli operators in the Hamiltonian, as in Chapter 2 let $\overline{\mathcal{S}}$ denote the *closure under inference* of $\mathcal{S}$ (Definition 2.2). This means that $\overline{\mathcal{S}}$ is the smallest set containing $\mathcal{S}$ as a subset, such that for every commuting pair $A, B \in \overline{\mathcal{S}}$, $AB \in \overline{\mathcal{S}}$ as well. Since we could in principle measure A and B simultaneously together with their product AB, from any value assignment to A, B we may *infer* the assignment to AB: hence closure under inference. We say that a value assignment is *consistent* if it respects all such inference relations. A value assignment to the observables in $\mathcal{S}$ extends by inference to an assignment to $\overline{\mathcal{S}}$, and noncontextuality is defined by situations in which consistent assignments to $\overline{\mathcal{S}}$ exist (see Chapter 2 and [KL19a]).

As proven in Chapter 2, we can determine whether $\mathcal{S}$ is contextual by the following criterion. Let $\mathcal{Z} \subseteq \mathcal{S}$ be the subset of operators in $\mathcal{S}$ that commute with all operators in $\mathcal{S}$ (we say that elements of $\mathcal{Z}$ commute universally in $\mathcal{S}$). Let $\mathcal{T} \coloneqq \mathcal{S} \backslash \mathcal{Z}$; then $\mathcal{S}$ *is noncontextual if and only if commutation is an equivalence relation on* $\mathcal{T}$ (Theorem 2.5). In other words, if $\mathcal{S}$ is noncontextual, then $\mathcal{T}$ is a union of N disjoint

[2] Note that this problem is distinct from that shown to be in P in [YB12], in which the Hamiltonian is a linear combination of commuting Pauli operators *with coefficients* ± 1

"cliques" $C_1, C_2, ..., C_N$, such that operators in different cliques anticommute, while operators in the same clique commute

Let C_{ij} be the operators in C_i, i.e.,

$$C_i \coloneqq \{C_{ij} \mid j = 1, 2, ..., |C_i|\}. \tag{3.1}$$

In other words,

$$\begin{aligned} \mathcal{S} &= \mathcal{Z} \cup \mathcal{T} \\ &= \mathcal{Z} \cup \bigcup_{i=1}^{N} C_i \\ &= \mathcal{Z} \cup \bigcup_{i=1}^{N} \{C_{ij} \mid j = 1, 2, ..., |C_i|\}. \end{aligned} \tag{3.2}$$

Hence, noncontextual Hamiltonian then has the form

$$H = \underbrace{\sum_{i=1}^{N} \left(\sum_{j=1}^{|C_i|} h_{ij} C_{ij} \right)}_{\text{terms in } \mathcal{T}} + \sum_{B \in \mathcal{Z}} h_B B, \tag{3.3}$$

where h_{ij} and h_B are real coefficients.

We may rewrite this Hamiltonian in a useful way by defining

$$A_{ij} \coloneqq C_{ij} C_{i1} \quad \Rightarrow \quad C_{ij} = A_{ij} C_{i1}, \tag{3.4}$$

where the implication follows because Pauli operators are self-inverse. A_{ij} is itself a Pauli operator (up to a sign) and commutes universally in $\mathcal{S}$, since any operator in C_i or in $\mathcal{Z}$ commutes with both C_{ij} and C_{i1}, and any operator in one of the other cliques anticommutes with both C_{ij} and C_{i1}. Thus, we may rewrite (3.3) as

$$H = \sum_{i=1}^{N} \left(\sum_{j=1}^{|C_i|} h_{ij} A_{ij} \right) C_{i1} + \sum_{B \in \mathcal{Z}} h_B B. \tag{3.5}$$

Note that in (3.5), the only operators appearing on the right-hand side that do not

commute universally in $\mathcal{S}$ are the C_{i1}: C_{i1} and $C_{i'1}$ anticommute for $i \neq i'$.

3.2 Quasi-quantized model for a noncontextual Hamiltonian

In this section we describe our main tool, a *quasi-quantized* (or *epistricted*) model for noncontextual Hamiltonians. A quasi-quantized model is composed of a set of phase space points, called the *ontic states* of the system, and a set of probability distributions over the ontic states, called the *epistemic states* [Spe07, Spe16].

3.2.1 Ontic states

If the set $\mathcal{S}$ is noncontextual, then consistent joint valuations for $\mathcal{S}$ exist. However, in general not *every* assignment of values to $\mathcal{S}$ is consistent. In order to conveniently index the allowed value assignments to $\mathcal{S}$, we will construct a new set $\mathcal{R}$ such that a value assignment to $\mathcal{R}$ induces a value assignment to $\mathcal{S}$ by inference, but such that every value assignment $\{+1, -1\}^{|\mathcal{R}|}$ to $\mathcal{R}$ is allowed. The latter is possible when $\mathcal{R}$ is *independent* in the following sense:

Definition 3.1. A set $\mathcal{R}$ of Pauli operators is *independent* if no value of any operator A in $\mathcal{R}$ can be inferred from any value assignment to a subset of $\mathcal{R}$ not containing A, according to the notion of inference defined in Section 2.2 Equivalently, $\mathcal{R}$ is independent if it contains no operator that can be written as a product of other commuting operators in $\mathcal{R}$.

Requiring $\mathcal{R}$ to be independent means that not only are *some* value assignments to $\mathcal{R}$ allowed (since it is noncontextual), but in fact *all* value assignments to $\mathcal{R}$ are allowed, since there are no inferences within $\mathcal{R}$ that could restrict the value assignments.

Note that in a commuting set of Pauli operators, this definition of independence reduces to the usual definition as applied to sets of generators of an Abelian group, where it means that removing any of the generators reduces the size of the

generated group. For the general noncontextual case, we can also relate Definition 3.1 to this definition of independence by using the Jordan product (2.2), equal to half the anticommutator of its arguments. As discussed in Section 2.2, the Jordan product yields inference relations by returning the matrix product of a pair of Pauli operators if they commute, and zero if they anticommute. Since the Jordan product is commutative, Definition 3.1 nearly reproduces the usual notion of independence for an Abelian group taking the group operation to be the Jordan product, but the Jordan product is not associative. This distinction aside, $\overline{\mathcal{S}}$ is the closure of $\mathcal{S}$ under the Jordan product, and correspondingly, a subset of $\overline{\mathcal{S}}$ that is independent in our sense is also independent as a generating set under the Jordan product.

To obtain $\mathcal{R}$, we first construct

$$G' \coloneqq \mathcal{Z} \cup \left(\bigcup_{i=1}^{N} \{A_{ij} \mid j = 2, 3, ..., |C_i|\} \right). \tag{3.6}$$

All operators in G' commute universally in $\mathcal{S}$. G' will in general be dependent, but we may efficiently obtain an independent set $G \coloneqq \{G_i \mid i = 1, 2, ..., |G|\}$ from G' using the method described in [NC01, §10.5.7] (summarized in Section 3.5). Operators in G commute universally in $\mathcal{S}$ because $G \subseteq \overline{G'}$.

The independent set $\mathcal{R}$ is then given by

$$\mathcal{R} \coloneqq \{C_{i1} \mid i = 1, 2, ..., N\} \cup G. \tag{3.7}$$

In words, this is a set containing exactly one element of each clique C_i, together with an independent generating set for the universally commuting part of the Hamiltonian. The C_{i1} are independent because no pair of them commutes, and since any product of operators in G commutes with all operators in $\mathcal{S}$, the C_{i1} are also independent of G. Therefore, $\mathcal{R}$ is an independent set.

Each operator in $\mathcal{Z}$ and each A_{ij} may be written as a product of operators

in G, by construction. Therefore, since by (3.2) and (3.4),

$$\mathcal{T} = \bigcup_{i=1}^{N} \{A_{ij} C_{i1} \mid j = 1, 2, ..., N\}, \tag{3.8}$$

each operator in $\mathcal{T}$ is a product of one of the C_{i1} and some set of operators in G. Thus, each operator in $\mathcal{S}$ is a product of commuting operators in $\mathcal{R}$, so $\overline{\mathcal{R}} = \overline{\mathcal{S}}$.

The set G has size at most n, and size at most $n-1$ if $\mathcal{T}$ is nonempty, because a set of n independent, commuting Pauli operators forms a complete, commuting set of observables for n qubits. Therefore, if G had size n (or more) with nonempty $\mathcal{T}$, then for any i the set $G \cup \{C_{i1}\}$ would be a commuting set of size $n+1$ (or more), which could not be independent. Thus $\mathcal{R}$ has size at most $\max(N + n - 1, n)$. As pointed out in [RBVT$^+$20, §IV.A], $N \leq 2m + 1$ for $m \coloneqq n - |G|$, so

$$|\mathcal{R}| \leq 2n + 1 \tag{3.9}$$

with equality only when G (and thus $\mathcal{Z}$) is empty. This will be important in Section 3.3.

Note that although $\mathcal{R}$ is not in general a subset of $\mathcal{S}$, it is a subset of $\overline{\mathcal{S}}$. Furthermore, since there is a bijection from the set of ontic states for $\mathcal{R}$ to the set of ontic states for $\overline{\mathcal{S}}$, $|\mathcal{R}|$ is unique for each $\overline{\mathcal{S}}$. Thus closure under inference permits construction of the independent generating set $\mathcal{R}$.

3.2.2 Epistemic states

Epistemic states are joint probability distributions over the ontic states, which complete our quasi-quantized model. We write these joint probabilities as

$$P(c_1, ..., c_N, g_1, g_2, ...), \tag{3.10}$$

where each c_i, g_i is ± 1 and denotes the value assigned to C_{i1} or G_i, respectively.

First consider a commuting Hamiltonian. In this case, $\mathcal{R} = G$, and $\overline{\mathcal{S}} = \overline{\mathcal{R}}$ is the Abelian group generated by $\mathcal{R}$. The observables may be simultaneously mea-

sured, so there is a one-to-one mapping between the ontic states and the simultaneous eigenstates of $\mathcal{S}$. Thus the only constraint on the joint probabilities in this case is normalization.

Next consider the case where all observables pairwise anticommute:

Lemma 3.1. *Let $\vec{A} = (A_1, A_2, ..., A_N)$ be an anticommuting set of Pauli operators. For any unit vector $\vec{a} \in \mathbb{R}^N$, the operator*

$$\vec{a} \cdot \vec{A} \equiv \sum_{i=1}^{N} a_i A_i \tag{3.11}$$

has eigenvalues ± 1. From this it follows that for any state, $\sum_{i=1}^{N} \langle A_i \rangle^2 \leq 1$.

Proof. This lemma consists of two claims:

For the first claim (that the operator $\vec{a} \cdot \vec{A}$ has eigenvalues ± 1), note that

$$\begin{aligned}(\vec{a} \cdot \vec{A})^2 &= \sum_i a_i^2 A_i^2 + \sum_{i>j} a_i a_j \{A_i, A_j\} \\ &= \sum_i a_i^2 \mathbb{1} = |\vec{a}|^2 \mathbb{1} = \mathbb{1},\end{aligned} \tag{3.12}$$

where the second equality follows because the Pauli operators A_i are self-inverse, and by assumption $\{A_i, A_j\} = 0$ for $i > j$. Thus, $\vec{a} \cdot \vec{A}$ is self-inverse, so its eigenvalues are ± 1. This completes the proof of the first claim.

For the second claim (that $\sum_{i=1}^{N} \langle A_i \rangle^2 \leq 1$), note that for any state represented as a density operator ρ, we may write

$$\rho = \frac{1}{2^n} \left(\mathbb{1} + \vec{b} \cdot \vec{A} + \cdots \right), \tag{3.13}$$

for some $\vec{b} \in \mathbb{R}^N$, where the ellipsis indicates the presence in general of additional Pauli terms. The expectation value of any A_i is then

$$\langle A_i \rangle = \mathrm{Tr}(A_i \rho) = b_i, \tag{3.14}$$

since the Pauli operators are Hilbert-Schmidt orthogonal and self-inverse.

Assume that $|\vec{b}| > 0$ (if $|\vec{b}| = 0$ then the theorem holds, by (3.14).) Let $\vec{a} \coloneqq \frac{\vec{b}}{|\vec{b}|}$, so that $|\vec{a}| = 1$. By the first claim in this lemma, the observable $\vec{a} \cdot \vec{A}$ has eigenvalues ± 1. Therefore, $|\langle \vec{a} \cdot \vec{A} \rangle| \leq 1$, so we have

$$\begin{aligned} 1 \geq & |\langle \vec{a} \cdot \vec{A} \rangle| = |\mathrm{Tr}(\vec{a} \cdot \vec{A} \rho)| \\ & = \left| \frac{1}{2^n} \mathrm{Tr}\left(\vec{a} \cdot \vec{A} (\vec{b} \cdot \vec{A}) \right) \right| = |\vec{a} \cdot \vec{b}|, \end{aligned} \tag{3.15}$$

where (as above) the steps in the second line follow because the Pauli operators are Hilbert-Schmidt orthogonal and self-inverse. Therefore,

$$1 \geq |\vec{a} \cdot \vec{b}|^2 = \frac{|\vec{b}|^4}{|\vec{b}|^2} = |\vec{b}|^2 = \sum_{i=1}^{N} b_i^2 = \sum_{i=1}^{N} \langle A_i \rangle^2 \tag{3.16}$$

(since by assumption $|\vec{b}| > 0$.) This completes the proof of the second claim. □

For a general noncontextual set $\mathcal{S}$, construct $\mathcal{R}$ as described in Section 3.2.1. We define the set of allowed joint probabilities, i.e., epistemic states, in our quasi-quantized model as follows:

$$P_{(\vec{q},\vec{r})}(c_1, ..., c_N, g_1, g_2, ...) = \left(\prod_{j=1}^{|G|} \delta_{g_j, q_j} \right) \prod_{i=1}^{N} \frac{1}{2} |c_i + r_i|, \tag{3.17}$$

where the parameters $(\vec{q}, \vec{r})$ that define the distribution can take any values in the set then define the set

$$E \coloneqq E(\mathcal{R}) \coloneqq \left\{ (\vec{q}, \vec{r}) \in \{\pm 1\}^{|G|} \times \mathbb{R}^N \mid |\vec{r}| = 1 \right\}. \tag{3.18}$$

We will abuse terminology by referring to both the joint probabilities and the parameters $(\vec{q}, \vec{r})$ that define them as epistemic states: they contain equivalent information.

In what follows, it will be more convenient for us to define the epistemic states as joint sets of expectation values for the observables in $\mathcal{R}$ than to use the joint probability distributions (3.17) directly. We can do this because of the following theorem:

Theorem 3.2. *For any epistemic state* $(\vec{q},\vec{r})$ *as defined in* (3.18)*, the corresponding joint probability distribution* (3.17) *is equivalent to the set of expectation values*

$$\begin{aligned}\langle G_j\rangle_{(\vec{q},\vec{r})} &= q_j,\\ \langle C_{i1}\rangle_{(\vec{q},\vec{r})} &= r_i.\end{aligned} \tag{3.19}$$

Proof.

1. First assume that (3.19) holds. Since $\langle G_j\rangle = q_j = \pm 1$ for each j (as in (3.18)), the G_j have definite values. The values of the G_j in the ontic state $(c_1,...,c_N,g_1,g_2,...)$ are the g_j, so $P(c_1,...,c_N,g_1,g_2,...)$ can be nonzero only when $q_j = g_j$ for each j. If this holds for each j, then $P(c_1,...,c_N,g_1,g_2,...)$ is just the product over $i = 1,2,...,N$ of the probabilities of obtaining outcome $C_{i1} \mapsto c_i$ given the expectation value $\langle C_{i1}\rangle = r_i$: these probabilities are given by

   $$\frac{1}{2}|c_i + r_i| \tag{3.20}$$

   for each i. Taking the joint probability to be the product of these works only because the C_{i1} do not commute, and thus cannot be correlated. Hence, the condition due to each G_j contributes a factor of

   $$\prod_{j=1}^{|G|}\delta_{g_j,q_j}, \tag{3.21}$$

   and the condition due to each C_{i1} contributes a factor of

   $$\prod_{i=1}^{N}\frac{1}{2}|c_i + r_i|; \tag{3.22}$$

   (3.17) is simply the product of these.

2. Now assume that (3.17) holds. The probabilities for the outcomes of each G_j and each C_{i1} should be obtained as marginals of (3.17). If p_{G_j} denotes the

probability of obtaining the outcome $G_j \mapsto +1$, then

$$\begin{aligned} p_{G_j} &= \sum_{\substack{c_i, g_k = \pm 1, \\ \forall\ k \neq j}} P(c_1, ..., c_N, g_1, g_2, ...)\Big|_{g_j=1} \\ &= \sum_{\substack{c_i, g_k = \pm 1, \\ \forall\ k \neq j}} \left(\prod_{l=1}^{|G|} \delta_{g_l, q_l} \right) \Big|_{g_j=1} \left(\prod_{m=1}^{N} \frac{1}{2} |c_m + r_m| \right) \\ &= \delta_{q_j,1} \sum_{c_i = \pm 1} \left(\prod_{m=1}^{N} \frac{1}{2} |c_m + r_m| \right) \\ &= \frac{1}{2}(q_j + 1) \left(\prod_{m=1}^{N} \frac{1}{2} (|r_m + 1| + |r_m - 1|) \right) \\ &= \frac{1}{2}(q_j + 1), \end{aligned} \tag{3.23}$$

where the fourth equality follows because for $r_m \in [-1, 1]$,

$$|r_m + 1| + |r_m - 1| = 2. \tag{3.24}$$

Similarly,

$$\begin{aligned} p_{C_{i1}} &= \sum_{\substack{c_k, g_j = \pm 1, \\ \forall\ k \neq i}} P(c_1, ..., c_N, g_1, g_2, ...)\Big|_{c_i=1} \\ &= \sum_{\substack{c_k, g_j = \pm 1, \\ \forall\ k \neq i}} \left(\prod_{l=1}^{|G|} \delta_{g_l, q_l} \right) \left(\prod_{m=1}^{N} \frac{1}{2} |c_m + r_m| \right) \Big|_{c_i=1} \\ &= \sum_{\substack{c_k = \pm 1, \\ \forall\ k \neq i}} \left(\prod_{m=1}^{N} \frac{1}{2} |c_m + r_m| \right) \Big|_{c_i=1} \\ &= \frac{1}{2}|r_i + 1| = \frac{1}{2}(r_i + 1), \end{aligned} \tag{3.25}$$

again using (3.24). From (3.23) and (3.25) we can obtain the expectation values for G_j and C_{i1}:

$$\begin{aligned} \langle G_j \rangle_{(\vec{q}, \vec{r})} &= 2p_{G_j} - 1 = q_j, \\ \langle C_{i1} \rangle_{(\vec{q}, \vec{r})} &= 2p_{C_{i1}} - 1 = r_i, \end{aligned} \tag{3.26}$$

which is (3.19) as desired.

□

The model (3.18) is epistricted in the following sense: as in [Spe16], a state is represented by joint knowledge of a set of commuting observables. For a given $(\vec{q}, \vec{r})$, this set is G together with the observable

$$\mathcal{A}(\vec{r}) \coloneqq \sum_{i=1}^{N} r_i C_{i1} \tag{3.27}$$

(which has eigenvalues ± 1, by Lemma 3.1). Note that since the C_{i1} have expectation values r_i as in (3.19), $\mathcal{A}(\vec{r})$ has expectation value 1, since $\vec{r}$ is a unit vector. Hence, we can think of an epistemic state as a stabilizer state (see e.g. [NC01, Sec. 10.3]) in which one of the stabilizers, $\mathcal{A}$, has been rotated by a unitary that preserves the other stabilizers, which we may take to be $q_j G_j$ so that their assigned eigenvalues are $+1$. No probability distributions are allowed that represent more knowledge of the state than simultaneous values of G and $\mathcal{A}(\vec{r})$. Note that our model describes only pure states (as do the models in [Spe16]).

From the expectation values (3.19) for $\mathcal{R}$, we can obtain expectation values for $\mathcal{S}$ as follows. For any $B \in \overline{\mathcal{Z}}$, let $\mathcal{J}_B$ be the set of indices such that $B = \prod_{j \in \mathcal{J}_B} G_j$; then

$$\langle B \rangle_{(\vec{q},\vec{r})} = \left\langle \prod_{j \in \mathcal{J}_B} G_j \right\rangle = \prod_{j \in \mathcal{J}_B} q_j, \tag{3.28}$$

where the second equality follows because $G_j \mapsto q_j = \pm 1$ for all j (in other words, the state is a common eigenstate of the G_j and of B). Similarly, for $C_{i1} B \in \mathcal{T}$,

$$\langle C_{i1} B \rangle_{(\vec{q},\vec{r})} = r_i \prod_{j \in \mathcal{J}_B} q_j. \tag{3.29}$$

Note that this expression indicates that for $C_{i1}B$, the expectation value of the product is the product of the expectation values; this is not true in general, but true in this case because the expectation value of B is ± 1.

For any $(\vec{q}, \vec{r})$, the expectation values (3.19) are produced by a simultaneous

eigenstate of $G \cup \{\mathcal{A}(\vec{r})\}$, so they correspond to valid quantum states. Moreover, we will shortly prove in Theorem 3.4 that there exists an eigenbasis for the Hamiltonian composed of common eigenstates of G and $\mathcal{A}(\vec{r})$ for some $\vec{r}$, which means that all eigenstates of the Hamiltonian can be expressed in our quasi-quantized model. First, we prove the following lemma, which will be used in the proof of Theorem 3.4 to explain why the Hamiltonian has an eigenbasis in common with $\mathcal{A}(\vec{r})$:

Lemma 3.3. *For $|\psi\rangle$ an eigenstate of the full Hamiltonian* (3.5)*, the expectation values of the C_{i1} satisfy $\sum_{i=1}^{N} \langle C_{i1} \rangle^2 = 1$.*

Remark 3.1. In other words, for any energy eigenstate, the $\langle C_{i1} \rangle$ saturate the bound given in Lemma 3.1. This means that every energy eigenvalue can be reproduced from the expectation values (3.28) and (3.29) for some setting of $(\vec{q}, \vec{r})$.

Proof. Since the G_j commute, we may simultaneously diagonalize them. The resulting form for the full Hamiltonian H will be block-diagonal, with each block corresponding to a common eigenspace of the G_j, since the C_{i1} commute with the G_j and therefore do not mix their common eigenspaces. Thus, the eigenstates of the full Hamiltonian are common eigenstates of G.

Given this, if $|\psi\rangle$ is an eigenstate of the full Hamiltonian, then $|\psi\rangle$ is also an eigenstate of any $B \in \overline{G}$, a product of some subset of G, so

$$BC_{i1}|\psi\rangle = C_{i1}B|\psi\rangle = \lambda_B C_{i1}|\psi\rangle, \tag{3.30}$$

where $\lambda_B = \pm 1$ is the eigenvalue of B in the state $|\psi\rangle$. Thus, we may rewrite the full Hamiltonian (3.5), acting on $|\psi\rangle$, as

$$H|\psi\rangle = \left(h'_0 + \sum_{i=1}^{N} h'_i C_{i1} \right) |\psi\rangle, \tag{3.31}$$

for coefficients h'_i defined by

$$\begin{aligned} h'_0 &\coloneqq \sum_{B \in \mathcal{Z}} h_B \lambda_B, \\ h'_i &\coloneqq \sum_{B \in \overline{G}} h_{B,i} \lambda_B \quad \text{for } i > 0, \end{aligned} \tag{3.32}$$

where $h_{B,i}$ is the coefficient of BC_{i1} in the Hamiltonian. Since $|\psi\rangle$ is an eigenstate of H, by (3.31) it must also be an eigenstate of

$$\sum_{i=1}^{N} h'_i C_{i1} = h \sum_{i=1}^{N} \tilde{h}_i C_{i1}, \tag{3.33}$$

the non-identity terms in (3.31), for $h \coloneqq \sqrt{\sum_{i=1}^{N} (h'_i)^2}$, so that the $\tilde{h}_i \coloneqq h'_i/h$ satisfy

$$\sum_{i=1}^{N} \tilde{h}_i^2 = 1. \tag{3.34}$$

By Lemma 3.1, the operator $\sum_{i=1}^{N} \tilde{h}_i C_{i1}$ has eigenvalues ± 1. Since $|\psi\rangle$ is an eigenstate of the operator given in (3.33), it is an eigenstate of $\sum_{i=1}^{N} \tilde{h}_i C_{i1}$:

$$\left(\sum_{i=1}^{N} \tilde{h}_i C_{i1} \right) |\psi\rangle = \pm |\psi\rangle. \tag{3.35}$$

Therefore, if $\langle \cdot \rangle$ denotes expectation value with respect to $|\psi\rangle$,

$$\langle \psi | \left(\sum_{i=1}^{N} \tilde{h}_i C_{i1} \right) |\psi\rangle = \sum_{i=1}^{N} \tilde{h}_i \langle C_{i1} \rangle = \pm 1. \tag{3.36}$$

We know by Lemma 3.1 that

$$\sum_{i=1}^{N} \langle C_{i1} \rangle^2 \leq 1, \tag{3.37}$$

and by construction $\sum_{i=1}^{N} \tilde{h}_i^2 = 1$. Thus the only way (3.36) can be satisfied is if

$$\langle C_{i1} \rangle = s \tilde{h}_i \tag{3.38}$$

for all $i = 1, 2, ..., N$, for fixed $s = \pm 1$. Therefore,

$$\sum_{i=1}^{N} \langle C_{i1} \rangle^2 = \sum_{i=1}^{N} \tilde{h}_i^2 = 1. \tag{3.39}$$

Since this holds for any $|\psi\rangle$ in the eigenbasis of H, it holds for any eigenstate of H. □

Theorem 3.4. *The epistemic states* (3.18) *give sets of expectation values that correspond to valid quantum states, and the set of quantum states described by the epistemic states includes an eigenbasis of any Hamiltonian whose Pauli terms are* $\mathcal{S}$.

Proof. The theorem consists of two claims. For the first, note that for any epistemic state $(\vec{q}, \vec{r})$, there exists a simultaneous eigenstate of the operators G_j, with eigenvalue $q_j = \pm 1$ for each G_j, and of $\mathcal{A}(\vec{r}) \coloneqq \sum_{i=1}^{N} r_i C_{i1}$, with eigenvalue 1. For this state, the expectation value of each G_j is q_j, and the expectation value of each C_{i1} is r_i, as noted above. Thus every epistemic state $(\vec{q}, \vec{r})$ corresponds to a valid quantum state, proving the first claim.

For the second claim, consider first the universally-commuting operators: $\mathcal{Z}$, which are generated by G. Since these may in principle be simultaneously diagonalized, the common eigenstates of G (which are the common eigenstates for all of $\mathcal{Z}$) are a complete set of eigenstates for any Hamiltonian whose terms are a linear combination of $\mathcal{Z}$.

Now suppose we add to such a Hamiltonian a linear combination of the terms in $\mathcal{T}$, each of which is a product of operators in G with one of the C_{i1}. As in the previous paragraph, we may simultaneously diagonalize G: the resulting form for the full Hamiltonian will be block-diagonal, with each block corresponding to a common eigenspace of the G_j. Within each block, the Hamiltonian will take the form of some linear combination of the C_{i1}, obtained by replacing the G_j by their eigenvalues for the current block, as in (3.31) and (3.32). Since the C_{i1} commute with G, we may diagonalize the linear combination of the C_{i1} within each block (i.e., the C_{i1} do not

mix the common eigenspaces of G). Thus, it is still the case that we may take the eigenstates of the full Hamiltonian to be common eigenstates of G. This justifies the condition on $\vec{q}$ (the first set of values in our epistemic states) given in (3.18), namely, $q_j = \pm 1$, since these correspond to the expectation values of the $G_j \in G$.

Regarding the condition on $\vec{r}$ (the second set of values in our epistemic state) in (3.18), we draw on Lemma 3.3. In (3.18), the condition on $\vec{r}$ is that it be a unit vector, so since its components r_i give the expectation values for the C_{i1} as in (3.19), Lemma 3.3 proves that the set of expectation values thus described includes all eigenvalues of the Hamiltonian. □

We show in Section 3.6 how for *any* quantum state we may construct a joint probability distribution that reproduces the expectation values for $\overline{\mathcal{S}}$; however, to simulate noncontextual VQE it is only necessary to reproduce probabilities corresponding to eigenstates.

3.3 Classical simulation of a noncontextual Hamiltonian

Given the model described in Section 3.2, we now define a classical variational algorithm to simulate a noncontextual Hamiltonian. In (3.5), each A_{ij} and each B is a product of operators in G, i.e., is an element of $\overline{G}$. Therefore, we may replace A_{ij} by B and sum over all of $\overline{G}$, obtaining:

$$H = \sum_{B \in \overline{G}} \left(h_B B + \sum_{i=1}^{N} h_{B,i} B C_{i1} \right), \tag{3.40}$$

where the $h_{B,i}$ and h_B are the same real coefficients appearing in Eq. (3.5), just relabeled, and in general now also including some coefficients that are zero (there are $2^{|G|}$ elements of $\overline{G}$, which are guaranteed include all of the terms in $\mathcal{Z}$ but also

many terms not in the Hamiltonian). Thus we can use (3.28) and (3.29) to write:

$$\langle H \rangle_{(\vec{q},\vec{r})} = \sum_{B \in \overline{G}} \left(h_B + \sum_{i=1}^{N} h_{B,i} r_i \right) \prod_{j \in \mathcal{J}_B} q_j. \tag{3.41}$$

We may now treat (3.41) as a classical objective function.

This classical optimization problem will in general be hard. Although a convex special case of (3.41) is obtained when we fix some set of values for the q_j (or when G is empty), in general the function is non-convex. It is also in general frustrated, even if all terms commute, since in that case (3.41) becomes a linear combination of products of the q_j. Thus, as discussed further in the next few paragraphs, we should not expect worst-cases of this optimization to be tractable, but they remain at worst classically hard.

The classical objective function (3.41) representing energies of a noncontextual Hamiltonian also has implications for the complexity of noncontextual Hamiltonians. The statement of the NONCONTEXTUAL HAMILTONIAN PROBLEM is as follows: The inputs are a Hamiltonian H of the form (3.40) with poly(n) terms, together with a "promise" that the lowest eigenvalue of H is either greater than b or less than a for some $a < b \in \mathbb{R}$ such that $b - a > 1/\text{poly}(n)$. The goal is to return YES if the lowest eigenvalue is less than a.

A problem is in NP if for every YES instance, there exists a proof (or *witness*) that is classically verifiable in polynomial time. In our case, the witness for a YES instance is an epistemic state $(\vec{q},\vec{r}) \in E$ (a vector with dimension at most $2n+1$, as discussed in Section 3.2.1) satisfying $\langle H \rangle_{(\vec{q},\vec{r})} < a$. By showing that such a witness can be efficiently verified, we prove the following:

Theorem 3.5. *The* NONCONTEXTUAL HAMILTONIAN PROBLEM *is NP-complete.*

Proof. We demonstrated above that by varying over all epistemic states, the expected energy given in (3.41) varies over all eigenvalues of the Hamiltonian. Therefore, if the Hamiltonian possesses an eigenvalue $\lambda < a$, there is some epistemic state $(\vec{q},\vec{r})$ such that $\langle H \rangle_{(\vec{q},\vec{r})} = \lambda$. This epistemic state serves as a witness of the YES

instance, as long as given the Hamiltonian as in (3.40) we can classically evaluate the objective function in (3.41) in polynomial time.

To show this, we use the fact that by assumption $|\mathcal{S}|$ (the number of terms in the Hamiltonian) is polynomial in n. Thus, the total number of nonzero coefficients h_B and $h_{B,i}$ is poly(n); these are given in the statement of the problem instance. Which coefficients are nonzero also determines the terms in the sum in (3.41) (and upper bounds their number). The only remaining components of (3.41) to be evaluated are the sets of indices $\mathcal{J}_B$: we have one such set for each term, defined to satisfy $B = \prod_{G \in \mathcal{J}_B} G$. The $\mathcal{J}_B$ are obtained directly from the standard method used to construct G (described in Section 3.5), the entirety of which requires poly(n) classical operations given that $|\mathcal{S}| = \text{poly}(n)$. Thus, given a Hamiltonian of the form (3.40) and a witness in the form of an epistemic state $(\vec{q}, \vec{r})$, we can use (3.41) to verify that $\langle H \rangle_{(\vec{q},\vec{r})} < a$, in poly$(n)$ classical operations.

The DIAGONAL LOCAL HAMILTONIAN PROBLEM is NP-complete, as follows from [Bar82]; see also [YB12]. This remains true even for 2-local diagonal Hamiltonians [WB03], so since any 2-local diagonal Hamiltonian is noncontextual with poly(n) Pauli terms, the NONCONTEXTUAL HAMILTONIAN PROBLEM is NP-complete as well. □

3.4 Discussion

In the quantum approximate optimization algorithm (QAOA) [FGG14], the Hamiltonian is diagonal (and thus noncontextual), because it encodes a classical problem, so our method simply recovers the diagonal entries. We have shown that the NONCONTEXTUAL HAMILTONIAN PROBLEM is in NP. Thus, the potential for quantum advantage in noncontextual VQE reduces to the same question that motivates QAOA: can we use cleverly-chosen and/or physically-motivated ansatze to generate otherwise hard-to-reach joint probability distributions, and thus efficiently converge to solutions of classically hard problems?

3.5 Appendix: obtaining an independent set of Pauli measurements from a commuting set

Given a set G' of commuting Pauli operators on n qubits, we wish to obtain an independent commuting set G such that every operator in G' is a product of operators in G. Since G commutes, G is independent if and only if no operator in G is a product of other operators in G (see the discussion following Definition 3.1.) Finding G given G' is a standard procedure, and a method for performing it is given by Nielsen and Chuang in [NC01, §10.5.7]. We summarize the method here in terms of the language used in this work.

The method to calculate G from G' is a multiplicative variant of Gaussian elimination. Let $G' = \{g'_1, g'_2, ..., g'_m\}$, and write each g'_i as

$$g'_i = h_{i1} \otimes h_{i2} \otimes \cdots \otimes h_{in}, \tag{3.42}$$

where each h_{ij} is a single-qubit Pauli operator (including the identity). We may then express G' in an array as:

$$G' = \begin{matrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1n}, \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2n}, \\ h_{31} & h_{32} & h_{33} & \cdots & h_{3n}, \\ & & & \vdots & \\ h_{m1} & h_{m2} & h_{m3} & \cdots & h_{mn}, \end{matrix} \tag{3.43}$$

where we have suppressed the tensor product symbols.

Let h denote the matrix whose entries are h_{ij}, the single-qubit Pauli operators appearing in (3.43), augmented by a vector $\vec{s}$ whose entries s_i are the signs associated to each row in h_i (initially these are all $+1$). We first describe a procedure that, given such a matrix h, transforms it to a matrix in which at most two entries in the first column are non-identity:

1. $h_{11}, h_{21}, ..., h_{m1}$ are the entries in the first column of h. If

$$h_{11} = h_{21} = \cdots = h_{m1} = I, \tag{3.44}$$

then we are already done.

2. If for some k, $h_{k1} = X$, then for each $i \neq k$, if $h_{i1} = X$, multiply row i by row k. This corresponds to multiplying g'_i by g'_k, or the following mapping on the entries in h:

$$\begin{aligned} h_{i1} &\mapsto h_{i1}h_{k1} = X^2 = I, \\ h_{ij} &\mapsto h_{ij}h_{kj}, \ \forall j > 1. \end{aligned} \tag{3.45}$$

Thus when we have completed this step for each $i \neq k$, there will be no Xs in the first column except in row k.

3. If for some l, $h_{l1} = Z$, then for each $i \neq l$, if $h_{i1} = Z$, multiply row i by row l. As in step 2, this corresponds to multiplying g'_i by g'_l, so when we have completed this step for each $i \neq l$, there will be no Zs in the first column except in row l.

4. If there is both an X and a Z in the first column (in rows k and l, respectively), then for each $i \neq k, l$, if $h_{i1} = Y$, multiply row i by row k and row l. This corresponds to multiplying g'_i by $g'_k g'_l$, or the following mapping on the entries in h:

$$\begin{aligned} h_{i1} &\mapsto h_{i1}h_{k1}h_{l1} = YXZ = -iI, \\ h_{ij} &\mapsto h_{ij}h_{kj}h_{lj}, \ \forall j > 1. \end{aligned} \tag{3.46}$$

Thus when we have completed this step for each $i \neq l$, there will be no Ys in the first column at all (if both X and Z are present in the first column).

5. If X and Z are not both present in the first column, and if for some m, $h_{m1} = Y$, then for each $i \neq m$, if $h_{i1} = Y$, multiply row i by row m. As in

step 2, this corresponds to multiplying g'_i by g'_m, so when we have completed this step for each $i \neq m$, there will be no Ys in the first column except in row m.

Note that since we know that the full Pauli operators corresponding to the rows commute, the total phase obtained in multiplying any row by any other row must be ± 1. We record this by multiplying s_i by this sign, when the multiplication takes place in the ith row.

Call this procedure ReduceFirstCol(h). When we have completed the mapping, the transformed matrix h will contain at most two rows in which the first entry is non-identity (and the first entries in those two rows will be different).

Our full procedure to obtain G is then as follows:

1. Let h be the full matrix (3.43). h' will be a submatrix of h that is updated at each iteration; let h' initially be equal to h.

2. Perform the mapping

$$h' \mapsto \text{ReduceFirstCol}(h'). \tag{3.47}$$

3. For each of the (up to) two rows in h' in which the first entry is non-identity, put the corresponding Pauli operators in G. Then let the new h' be the submatrix obtained from the current h' by removing these rows and the first column.

4. If any non-identity entries remain in h', return to step 2. If not, then G is complete.

When this procedure is complete, at most two rows in h will have a non-identity first entry, at most two others will have a non-identity second entry, at most two others will have a non-identity third entry, and so forth. In other words, under some

reordering of the rows, h will have the form:

$$\begin{array}{cccccccc}
 & P_1 & h'_{12} & h'_{13} & h'_{14} & h'_{15} & \cdots & h'_{1n}, \\
[& P_2 & h'_{22} & h'_{23} & h'_{24} & h'_{25} & \cdots & h'_{2n}, \ ] \\
 & I & P_3 & h'_{33} & h'_{34} & h'_{35} & \cdots & h'_{3n}, \\
[& I & P_4 & h'_{43} & h'_{44} & h'_{45} & \cdots & h'_{4n}, \ ] \\
 & I & I & P_5 & h'_{54} & h'_{55} & \cdots & h'_{5n}, \\
[& I & I & P_6 & h'_{64} & h'_{65} & \cdots & h'_{6n}, \ ] \\
 & & & & \vdots & & & \\
[& I & I & I & I & I & \cdots & I, \ ]
\end{array} \tag{3.48}$$

where the bracketed rows may or may not appear, and in cases where they do appear $P_2 \neq P_1$, $P_4 \neq P_3$, $P_6 \neq P_5$ and so forth. Note that in general there may be a collection of rows at the bottom of the matrix that have been reduced entirely to the identity. G will be given by the non-identity rows in (3.48).

To see that G is independent, consider some particular non-identity row i (representing an operator $g_i \in G$). Row i cannot be written as a product of rows below it, since if h_{ij} is row i's first non-identity entry, $h_{i'j} = I$ for all $i' = i+2, i+3, ..., m$. The entry $h_{(i+1)j}$ (immediately below h_{ij}) might not be the identity, but it cannot be equal to h_{ij}, so no product of the entries below h_{ij} can be equal to h_{ij}. Thus g_i cannot be written as the product of any subset of the g_j for $j > i$.

But this means that g_i cannot be written as a product of any subset of the other g_j for $j \neq i$, as we can prove by contradiction: suppose g_i could be written as a product

$$g_i = \prod_{j \in \mathcal{J}} g_j, \tag{3.49}$$

where $\mathcal{J}$ is some set of indices not including i. If $i < j$ for all $j \in \mathcal{J}$, then this violates the condition in the previous paragraph directly. Otherwise, choose the least $j \in \mathcal{J}$: call it j'. Then since the Pauli operators are self-inverse, and all of the

g_k commute, we can rearrange (3.49) as

$$g_{j'} = g_i \prod_{j \in \mathcal{J} \setminus \{j'\}} g_j, \tag{3.50}$$

which violates the condition in the previous paragraph since $j' < i, j$ for all $j \in \mathcal{J} \setminus \{j'\}$. Thus, all of the g_i are independent.

We wish to be able to recover the original set of operators G' from the new independent set G. To accomplish this, we simply need to keep track of the row multiplications carried out in each step above. Since the Pauli operators are self-inverse, we can reconstruct each row i in G' as the product of the corresponding row in G and the rows it is multiplied by in the procedure (including in the multiplication the corresponding signs in $\vec{s}$). Thus we can write each operator in G' as a product of operators in G.

3.6 Appendix: More general quasi-quantized models

In Section 3.2.1, we showed how to construct the ontic states (phase space points) for a noncontextual set of observables. The key step is obtaining the set $\mathcal{R} = \{C_{i1} \mid i = 1, 2, ..., N\} \cup G$, where $\{C_{i1} \mid i = 1, 2, ..., N\}$ is a completely anticommuting set, $G = \{G_i \mid i = 1, 2, ..., |G|\}$ is an independent commuting set whose elements also commute with each C_{i1}, and every operator in $\mathcal{S}$ may be written as a product of commuting operators in $\mathcal{R}$. Specifically, each universally commuting operator in $\mathcal{S}$ (these compose the set $\mathcal{Z}$) may be written as a product of operators in G, and every operator in one of the cliques $C_i \subset \mathcal{T}$ may be written as a product of operators in G with the single operator C_{i1}.

The joint outcome assignments for $\mathcal{R}$ label the phase space points (each of which defines a joint outcome assignment for all of $\mathcal{S}$), so they also label the joint probabilities associated to each phase space point. As in Section 3.2.2, we write

these joint probabilities as

$$P(c_1, ..., c_N, g_1, g_2, ...), \tag{3.51}$$

where each c_i, g_i is ± 1 and denotes the outcome assigned to C_{i1} or G_i, respectively.

In Section 3.2.2, we presented a quasi-quantized model that gives a set of joint probabilities P (by way of a set of expectation values) that is sufficiently general to reproduce the expectation values associated to any eigenstate of the Hamiltonian. However, the set of ontic states in principle admits broader sets of probability distributions, which we discuss in this section.

The probabilities for outcome $+1$ for each operator in $\mathcal{S}$ should be obtained as marginals of the joint probability distribution (3.51). Consider an operator in $\mathcal{S}$ that is written as

$$B \text{ or } C_{i1}B, \tag{3.52}$$

for B a product of operators in G (which, it is understood, may be the identity — the product of no operators in G). The corresponding marginal is the sum of the joint probabilities for outcomes such that the product outcome from (3.52) is $+1$. In other words, if p_O denotes the probability of obtaining outcome $+1$ upon measurement of the operator O, then

$$p_B = \sum_{\substack{c_i, g_j = \pm 1, \\ \text{s.t. } B \mapsto 1}} P(c_1, ..., c_N, g_1, g_2, ...), \tag{3.53}$$

$$p_{C_{i1}B} = \sum_{\substack{c_i, g_j = \pm 1, \\ \text{s.t. } C_{i1}B \mapsto 1}} P(c_1, ..., c_N, g_1, g_2, ...). \tag{3.54}$$

Exactly half of the joint probabilities will appear in each sum (3.53) and (3.54). This is apparent when we note that for any point $(c_1, ..., c_N, g_1, g_2, ...)$ satisfying, say, the condition $B = 1$ for B a product of some subset of the g_j, flipping the sign of any of the coordinates g_j appearing in the product will cause it to violate the condition, and vice versa.

Alternatively, we may visualize the sets of phase space points appearing in the sums (3.53) and (3.54) by first reassigning the outcome labels as

$$c_i \to c_i' = \frac{1}{2}(1 - c_i), \quad g_i \to g_i' = \frac{1}{2}(1 - g_i), \tag{3.55}$$

i.e., the outcome assignments are mapped as

$$1 \to 0, \quad -1 \to 1, \tag{3.56}$$

and products of outcomes become binary sums of outcomes. In other words, we have mapped our outcome space for each observable to $\mathbb{Z}_2$ (also denoted $\mathbb{F}_2$, the field of two elements), as is common practice in the literature (see [GHW04], for example). The marginalizations (3.53) and (3.54) therefore become sums over all the joint probabilities for phase space points satisfying conditions of the form

$$\vec{A} \cdot \vec{c}' + \vec{B} \cdot \vec{g}' = 1 \tag{3.57}$$

where $\vec{c}' \coloneqq (c_1', ..., c_N')$ and $\vec{g}' \coloneqq (g_1', g_2', ...)$, and $\vec{A}, \vec{B}$ are vectors in $(\mathbb{Z}_2)^N, (\mathbb{Z}_2)^{|G|}$, respectively. We can now see that (3.57) is the equation for a hyperplane in the phase space $(\mathbb{Z}_2)^{N+|G|}$, so we may think of the marginal probabilities for outcomes of individual measurements as sums of the joint probabilities over such hyperplanes [GHW04].

We wish to demonstrate that for any state there exists a joint probability distribution P that reproduces as marginals the correct probabilities for the outcomes of any observable in $\mathcal{S}$. Since each ontic state carries with it an outcome assignment to each observable in $\mathcal{S}$, it is enough to show that for any state, P can reproduce as marginals the correct probabilities for the outcomes of any observable in $\mathcal{R}$. The largest subsets of $\mathcal{R}$ that may be measured simultaneously are $\{C_{i1}\} \cup G$ for any $i = 1, 2, ..., N$: let us refer to the probabilities for the joint outcomes for these as

$$P^{(\{C_{i1}\} \cup G)}(c_i, g_1, g_2, ...), \tag{3.58}$$

where the $c_i, g_1, g_2, ... = \pm 1$ label the joint outcomes. Each C_{i1} commutes with all operators in G, so for a given state we can directly determine (via Born's rule, using the appropriate projectors, or by actual measurements if the state is physical) the probabilities (3.58) (separately for each i). P will thus correctly reproduce as marginals the probabilities for the observables in $\mathcal{R}$ if and only if it correctly reproduces as marginals the joint outcome probabilities (3.58), since these correspond to the largest simultaneously measurable subsets of $\mathcal{R}$.

Thus, we wish P to satisfy

$$\begin{aligned} & P^{(\{C_{i1}\}\cup G)}(c_i, g_1, g_2, ...) \\ & = \sum_{\substack{c_j=\pm 1, \\ \forall j \neq i}} P(c_1, ..., c_i, ..., c_N, g_1, g_2, ...) \end{aligned} \tag{3.59}$$

for each $i = 1, 2, ..., N$. Since the C_{i1} pairwise anticommute, the probabilities $P^{(\{C_{i1}\}\cup G)}$ and $P^{(\{C_{j1}\}\cup G)}$ for any $i \neq j$ cannot be correlated. More specifically, the product $C_{i1}B_iC_{j1}B_j$ of the operators $C_{i1}B_i$ and $C_{j1}B_j$ (for $i \neq j$ and B_i, B_j any products of operators in G) is not jointly measurable with $C_{i1}B_i$ and $C_{j1}B_j$, since none of the three commute with each other, and in general the product need not even be in $\overline{\mathcal{S}}$. In any case, it has no bearing on the joint probabilities for $C_{i1}B_i$ and $C_{j1}B_j$; all that is required of the joint probability distribution P is that it correctly reproduce the expressions (3.59), which the assignment (3.60) does. Therefore, we may take the probability for the joint outcome $P(c_1, ..., c_N, g_1, g_2, ...)$ to be proportional to the product of the probabilities $P^{(\{C_{i1}\}\cup G)}(c_i, g_1, g_2, ...)$ over all $i = 1, 2, ..., N$. Including the appropriate normalization gives an expression for the joint probabilities:

$$\frac{P(c_1, ..., c_N, g_1, g_2, ...)}{P^{(G)}(g_1, g_2, ...)} = \prod_{i=1}^{N} \frac{P^{(\{C_{i1}\}\cup G)}(c_i, g_1, g_2, ...)}{P^{(G)}(g_1, g_2, ...)}, \tag{3.60}$$

where $P^{(G)}(g_1, g_2, ...)$ denotes the probability of obtaining the joint outcome $(g_1, g_2, ...)$ for the operators G (which we can also obtain via Born's rule).

As we noted above, given any quantum state we can directly evaluate each of

the probabilities on the right-hand side, since each is the probability for a joint outcome of one of the commuting sets of observables $\{C_{i1}\} \cup G$ (or just G). Therefore, for any quantum state, (3.60) gives a joint probability distribution $P(c_1, ..., c_N, g_1, g_2, ...)$ for the phase space points $(c_1, ..., c_N, g_1, g_2, ...)$, that reproduces as marginals all probabilities associated to the state, for observables in $\mathcal{S}$.

This discussion does not provide specific methods for constructing quasi-quantized models alternative to that presented in the main text. What we have shown, however, is that epistemic states over the ontic states described in Section 3.2.1 may in principle describe the measurable properties of $\mathcal{S}$ for any state. Thus, the door is open for broader sets of epistemic states than those allowed in our model as described in Section 3.2.2. For our purposes, however, the set of epistemic states allowed in our model is sufficient.

Chapter 4

Contextual subspace variational quantum eigensolver

4.1 Introduction

As discussed in Section 1.3.3, the *variational quantum eigensolver* (VQE) is a leading algorithm for quantum simulation on noisy intermediate-scale quantum (NISQ) computers, due to the limited resources it requires in both qubit count and coherence time [DXP+10, LWG+10, PMS+14, WDB+15, OBK+16, SZZ+17, PGS+17, KMT+17, HMR+18, SWG+18, CRD+18, DMH+18, KMvB+19, KTC+19, GEB+19, SBMS+19, MPJ+19, SM19, NCP+20, AAB+20, KJK+21a, KJK+21b, LYTT21, KGL+22]. VQE is a hybrid quantum-classical algorithm in which the expectation value of the Hamiltonian, or other observable, is computed for an ansatz state generated by a parameterized quantum circuit. Optimization of the ansatz parameters is performed iteratively using an optimization algorithm running on a classical computer. VQE algorithms require a large number of measurements to be performed, and give approximate results due to limitations of the ansatz that can be prepared and noise on the quantum device. As a result, the largest experiments to date either do not reach chemical accuracy [KMT+17], do not include all Hamiltonian terms [NCP+20], or simulate restricted models such as Hartree-Fock [AAB+20].

We address these limitations of VQE by providing an approximate simulation method for the full Hamiltonian that can be adjusted to use any amount of available quantum resources. We show that in many cases our method reduces the resources required to reach chemical accuracy. The method is based on the concept of a noncontextual Hamiltonian as defined in Chapter 2 (based on [KL19a]), in which definite values can be assigned to all Pauli terms simultaneously without contradiction. In Chapter 3 (based on [KL20]), we gave a classical algorithm for computing the ground state energies of noncontextual Hamiltonians based on a quasi-quantized model [Spe07, Spe16]. This raises the question of whether a truly hybrid algorithm can be developed for simulating general Hamiltonians, in which the noncontextual parts of the Hamiltonians are computed classically and contextual quantum corrections are computed using VQE. The method described in this chapter is such an algorithm; this chapter is based on [KTL21].

The contextual quantum correction is obtained by performing VQE restricted to quantum states that are consistent with the noncontextual ground state. We refer to the space of such quantum states as the *contextual subspace*. The contextual subspace represents the degrees of freedom that remain after the noncontextual degrees of freedom have been fixed. The resulting quantum computations only involve Hamiltonian terms in the complement of the noncontextual Hamiltonian. We also show how to adjust the noncontextual part of the Hamiltonian, in order to move more of the computation onto the quantum computer, while preserving the structure of the quasi-quantized model. The technique for accomplishing this is related to "subspace-search VQE" [NMF19], in which excited energies are found by restricting the search space to be orthogonal to the (previously approximated) ground state. In our case, we are not looking for excited states, but we are implementing VQE in a restricted search space, and part of the technique for achieving this is similar to that in [NMF19] (see Section 4.2 for details of the technique).

As an example, suppose we want to apply VQE to a Hamiltonian on n qubits, but the available quantum processor has only q qubits. CS-VQE permits us to adjust the noncontextual approximation method so that the associated quan-

tum correction uses exactly q qubits. Increasing the number of qubits used on the quantum processor monotonically improves the quality of the overall approximation, interpolating between the noncontextual approximation with no quantum correction and full VQE. Thus, we can tune the quantum part to fit the available quantum resources, with the classical method making up the difference.

In Section 4.2, we describe the quantum correction procedure. In Section 4.3, we describe how to implement CS-VQE, and show the results of simulating CS-VQE on Hamiltonians for small molecules. Finally, in Section 4.4, we summarize our results and discuss their implications for NISQ computing. This chapter is based on [KTL21].

4.2 Noncontextual approximation and quantum correction

4.2.1 Hamiltonian partitioning and contextual subspace

Let $\mathcal{S}$ be the set of Pauli terms in a general local Hamiltonian H. Suppose we have a partition of $\mathcal{S}$ into a noncontextual subset $\mathcal{S}_{\mathrm{nc}}$ and its complement $\mathcal{S}_{\mathrm{c}}$ (how to obtain such a decomposition will be discussed in Section 4.3.3). This induces a decomposition of H into a noncontextual part H_{nc} whose Pauli terms are $\mathcal{S}_{\mathrm{nc}}$, and H_{c} whose Pauli terms are $\mathcal{S}_{\mathrm{c}}$:

$$H = H_{\mathrm{nc}} + H_{\mathrm{c}}. \tag{4.1}$$

We will require that $\mathcal{S}_{\mathrm{nc}}$ be *closed under inference within* $\mathcal{S}$:

Definition 4.1. $\mathcal{S}_{\mathrm{nc}}$ is *closed under inference within* $\mathcal{S}$ if any operators in $\mathcal{S}$ whose values can be inferred from the values of operators in $\mathcal{S}_{\mathrm{nc}}$ must be included in $\mathcal{S}_{\mathrm{nc}}$. Formally, the equivalent definition is

$$\mathcal{S}_{\mathrm{nc}} = \overline{\mathcal{S}_{\mathrm{nc}}} \cap \mathcal{S} \tag{4.2}$$

(see Section 2.1).

Closure under inference itself was defined in Definition 2.2. The decomposition (4.1) is the basis of the CS-VQE method. We will obtain an efficient classical description of the eigenspaces of H_{nc}, and use this and H_{c} to quantum compute a correction to the noncontextual approximation to the ground-state energy.

Because all terms in H_{nc} can simultaneously be assigned definite values without contradiction we can introduce a phase space description of its eigenspaces as described in Chapter 3. The phase space points are the possible joint value assignments to a set of observables derived from $\mathcal{S}_{\text{nc}}$, which we review below. The eigenstates of H_{nc} are probability distributions over this phase space. As discussed in Chapter 3, this is a quasi-quantized model: a classical phase-space model with an uncertainty relation imposed upon the allowed probability distributions (sometimes called *epistemic states*) on the phase space [Spe07, Spe16].

Let $(\vec{q}, \vec{r})$ be the noncontextual ground state, i.e., the assignment of parameters (as in Eq. (3.18)) to the quasi-quantized model that produces the ground state energy of H_{nc}. If we take the resulting energy of H_{nc} as a classical estimate of the ground state energy of the full Hamiltonian H, we can obtain a quantum correction by minimizing the energy of the remaining terms in the Hamiltonian over the quantum states that are consistent with the noncontextual ground state. Recall that the epistemic state $(\vec{q}, \vec{r})$ is equivalent to the common eigenspace of the generators G_j (whose assigned eigenvalues are $q_j = \pm 1$) and

$$\mathcal{A} \coloneqq \sum_{i=1}^{N} r_i C_{i1} \tag{4.3}$$

(see Section 3.2.2), whose assigned eigenvalue is $+1$. As discussed in Section 3.2.2, this common eigenspace is a stabilizer subspace up to a rotation on one of the stabilizers (which yields $\mathcal{A}$). We refer to this subspace as the *contextual subspace*.

Before we discuss how to find quantum corrections, we establish when such corrections can appear:

Theorem 4.1. *Let $\mathcal{S}$ be a set of Pauli operators, and let $\mathcal{S}_{nc}$ be a noncontextual subset that is closed under inference within $\mathcal{S}$ (see Definition 4.1). Then for any*

noncontextual state $(\vec{q},\vec{r})$ *as in* (3.18) *describing* $\mathcal{S}_{nc}$*, there exists a quantum state consistent with* $(\vec{q},\vec{r})$ *(i.e., that gives the same expectation values for* $\mathcal{S}_{nc}$ *as* $(\vec{q},\vec{r})$*) for which the expectation value of every operator in* $\mathcal{S}_c \coloneqq \mathcal{S} \setminus \mathcal{S}_{nc}$ *is zero.*

Proof. Let $G \cup \{\mathcal{A}\}$ be the independent, commuting set of observables associated to the noncontextual state $(\vec{q},\vec{r})$ describing $\mathcal{S}_{\rm nc}$: the values assigned to $G \cup \{\mathcal{A}\}$ in the noncontextual state $(\vec{q},\vec{r})$ are

$$G_j \;\mapsto\; q_j = \pm 1 \tag{4.4}$$

for each $G_j \in G$, and

$$\mathcal{A} \coloneqq \sum_{i=1}^{N} r_i C_{i1} \;\mapsto\; +1 \tag{4.5}$$

(see Section 3.2.2). Let P be any Pauli operator in $\mathcal{S} \setminus \mathcal{S}_{\rm nc}$.

Case 1. If P anticommutes with any operator in G, then $\langle P \rangle = 0$, since any quantum state consistent with $(\vec{q},\vec{r})$ is a simultaneous eigenstate of G.

Case 2. If P commutes with the operators in G and also with the C_{i1}, then:

1. if P is a product of operators in G, then P can be inferred from G, so P must in fact be included in $\mathcal{S}_{\rm nc}$, since by assumption $\mathcal{S}_{\rm nc}$ is closed under inference within $\mathcal{S}$. This follows immediately from Definition 4.1.

2. if P is not a product of operators in G, then P is unconstrained by the noncontextual state, and may take any expectation value, including zero.

Case 3. Finally, suppose P commutes with the operators in G, but anticommutes with at least one of the C_{i1}. In this case we want to prove that there exists a $+1$-eigenstate of $\mathcal{A}$ for which $\langle P \rangle = 0$, as follows:

Let I_P be the set of indices such that

$$\begin{aligned} i \in I_P \quad &\Rightarrow \quad [P, C_{i1}] = 0, \\ i \notin I_P \quad &\Rightarrow \quad \{P, C_{i1}\} = 0. \end{aligned} \tag{4.6}$$

If I_P is empty, then P anticommutes with all of the C_{i1}: thus since

$$\sum_{i=1}^{N} \langle C_{i1} \rangle^2 = \sum_{i=1}^{N} r_i^2 = 1 \tag{4.7}$$

(see Section 3.2.2), and

$$\langle P \rangle^2 + \sum_{i=1}^{N} \langle C_{i1} \rangle^2 \leq 1 \tag{4.8}$$

(by Lemma 3.1), it follows that $\langle P \rangle = 0$.

The remaining case is when I_P is nonempty; there also exist $i \notin I_P$ by assumption. Let

$$K \coloneqq \sum_{i \in I_P} r_i C_{i1}, \quad L \coloneqq \sum_{i \notin I_P} r_i C_{i1}; \tag{4.9}$$

thus

$$\mathcal{A} = K + L \tag{4.10}$$

and

$$[K, P] = 0, \quad \{L, P\} = 0, \quad \{K, L\} = 0. \tag{4.11}$$

Since K and L are linear combinations of anticommuting Pauli operators, their eigenvalues are $\pm k$ and $\pm l$, respectively, where

$$k \coloneqq \sqrt{\sum_{i \in I_P} r_i^2}, \quad l \coloneqq \sqrt{\sum_{i \notin I_P} r_i^2}. \tag{4.12}$$

Therefore,

$$K^2 = k^2 \mathbb{1}, \quad L^2 = l^2 \mathbb{1}, \quad k^2 + l^2 = 1. \tag{4.13}$$

Since P commutes with K and is a Pauli operator, PK is also an observable with eigenvalues $\pm k$, which commutes with L (since both P and K anticommute with L). Thus, PK commutes with $\mathcal{A}$, so within the $+1$-eigenspace of $\mathcal{A}$ there exist eigenstates $|\pm\rangle$ of PK with eigenvalues $\pm k$, i.e.,

$$\mathcal{A}|\pm\rangle = |\pm\rangle, \quad PK|\pm\rangle = \pm k|\pm\rangle. \tag{4.14}$$

Note that since $P \notin \mathcal{S}_{\text{nc}}$, P cannot be written as a product of operators in G with any of the C_{i1}, so both of the states $|\pm\rangle$ are consistent with the noncontextual state $(\vec{q}, \vec{r})$.

The noncontextual state gives us the expectation values of the C_{i1}:

$$\langle \pm | C_{i1} | \pm \rangle = r_i \tag{4.15}$$

(see Section 3.2.2). This means that in addition to (4.14), we have

$$\langle \pm | K | \pm \rangle = \sum_{i \in I_P} r_i \langle \pm | C_{i1} | \pm \rangle = \sum_{i \in I_P} r_i^2 = k^2. \tag{4.16}$$

Define $|\psi\rangle \coloneqq \frac{1}{\sqrt{2}}\Big(|+\rangle + |-\rangle\Big)$; then

$$\begin{aligned}
\langle \psi | P | \psi \rangle &= \frac{1}{k^2} \langle \psi | P K^2 | \psi \rangle \quad \text{by (4.13)} \\
&= \frac{1}{2k^2} \Big(\langle + | + \langle - | \Big) K \Big(PK|+\rangle + PK|-\rangle \Big) \\
&= \frac{1}{2k} \Big(\langle + | + \langle - | \Big) K \Big(|+\rangle - |-\rangle \Big) \quad \text{by (4.14)} \\
&= \frac{1}{2k} \Big(\langle + | K | + \rangle - \langle + | K | - \rangle + \langle - | K | + \rangle - \langle - | K | - \rangle \Big) \\
&= -\frac{1}{2k} \Big(\langle + | K | - \rangle - \langle - | K | + \rangle \Big) \quad \text{by (4.16).}
\end{aligned} \tag{4.17}$$

But since K and P commute, we similarly have

$$\begin{aligned}
\langle \psi | P | \psi \rangle &= \frac{1}{k^2} \langle \psi | P K^2 | \psi \rangle \quad \text{by (4.13)} \\
&= \frac{1}{2k^2} \Big(\langle + | KP + \langle - | KP \Big) K \Big(|+\rangle + |-\rangle \Big) \\
&= \frac{1}{2k} \Big(\langle + | - \langle - | \Big) K \Big(|+\rangle + |-\rangle \Big) \quad \text{by (4.14)} \\
&= \frac{1}{2k} \Big(\langle + | K | + \rangle + \langle + | K | - \rangle - \langle - | K | + \rangle - \langle - | K | - \rangle \Big) \\
&= \frac{1}{2k} \Big(\langle + | K | - \rangle - \langle - | K | + \rangle \Big) \quad \text{by (4.16).}
\end{aligned} \tag{4.18}$$

Together, (4.17) and (4.18) imply that

$$\langle \psi | P | \psi \rangle = -\langle \psi | P | \psi \rangle = 0, \tag{4.19}$$

so since $|\psi\rangle$ is a state in the +1-eigenspace of $\mathcal{A}$, we are done. □

Theorem 4.1 has two useful corollaries:

Corollary 4.2. *If the noncontextual states of H_{nc} uniquely identify quantum states, then for any noncontextual state the expectation value of every term in H_c is zero, i.e., no quantum correction is possible.*

Proof. By Theorem 4.1, there exists a quantum state $|\psi\rangle$ consistent with the noncontextual state for which the expectation value of every term not in the noncontextual part is zero. Thus if the noncontextual state uniquely identifies a quantum state, it must be $|\psi\rangle$. □

Corollary 4.3. *As an approximation method for a general Hamiltonian, a quasi-quantized model of the noncontextual part is variational, i.e., the resulting approximate ground state energies are lower-bounded by the true ground state energy. It remains variational when the quantum correction is included.*

Proof. By Theorem 4.1, there exists a ground state of the noncontextual part for which the expectation value of every other term in the Hamiltonian is zero. Hence, the ground state energy of the noncontextual part is a possible expectation value for the energy of the full Hamiltonian, so it is lower-bounded by the ground state energy of the full Hamiltonian. To compute the quantum correction we minimize the expectation value of the full Hamiltonian over quantum states consistent with a given noncontextual state. This produces a variational estimate of the energy with the contribution from H_{nc} given by the noncontextual state. □

4.2.2 Mapping a contextual subspace to a stabilizer subspace

We now show how to map the contextual subspace corresponding to the noncontextual ground state to a subspace stabilized by single-qubit Z operators. To achieve this goal we rotate the G_j and subsequently the single operator $\mathcal{A}$ to single-qubit Z operators.

The number of G_j is some $M < n$, as discussed in Section 3.2.2. Therefore, the G_j can be mapped to single-qubit Z operators by a sequence of at most $2M$ $\frac{\pi}{2}$-rotations. These rotations are Clifford operators, so they map the remaining Pauli operators in the Hamiltonian back to single Pauli operators while preserving their commutation relations. This is a well-known result (see for example the solution to problem 3 in [YB12]), but for completeness, we provide a constructive proof:

Lemma 4.4. *For any set of M independent, commuting Pauli operators, there exists an efficiently calculable unitary rotation U, given by a sequence of at most $2M$ $\frac{\pi}{2}$-rotations generated by Pauli operators, that maps the set to a set of distinct single-qubit Z operators.*

Proof. Let $\{B_i \mid i = 1, 2, ..., M\}$ be a commuting set of Pauli operators. We may write B_i as

$$B_i = \bigotimes_{k=1}^{N} \sigma_k^{(B_i)}, \tag{4.20}$$

where each $\sigma_k^{(B_i)} \in \{I, X, Y, Z\}$. We consider two cases:

Case 1. Suppose B_i is not diagonal, meaning that there is some k such that $\sigma_k^{(B_i)} \in \{X, Y\}$. Consider the Pauli operator

$$J_i = \bigotimes_{k=1}^{N} \sigma_k^{(J_i)}, \tag{4.21}$$

defined as follows: for each k...

$$\sigma_k^{(B_i)} = I \;\Rightarrow\; \sigma_k^{(J_i)} = I, \tag{4.22}$$

$$\sigma_k^{(B_i)} = Z \;\Rightarrow\; \sigma_k^{(J_i)} = Z, \tag{4.23}$$

$$\sigma_k^{(B_i)} = X \text{ or } Y \;\Rightarrow\; \sigma_k^{(J_i)} = X \text{ or } Y, \tag{4.24}$$

where the values $\sigma_k^{(J_i)} = X$ or Y in (4.24) are chosen so that $\sigma_k^{(B_i)}$ and $\sigma_k^{(J_i)}$ differ for exactly one k (as we noted above, at least one of the $\sigma_k^{(B_i)}$ is X or Y if B_i is not diagonal.) This guarantees that J_i anticommutes with B_i.

Consider a rotation by $\pi/2$ generated by J_i, i.e.,

$$\exp\left(i\frac{\pi}{4}J_i\right) = \frac{1}{\sqrt{2}}(1 + iJ_i). \tag{4.25}$$

Upon conjugating B_i by this operator, we obtain

$$\begin{aligned}
&\frac{1}{2}(1 + iJ_i)B_i(1 - iJ_i) \\
&= \frac{1}{2}(B_i - iB_iJ_i + iJ_iB_i + J_iB_iJ_i) \\
&= \frac{1}{2}(B_i + 2iJ_iB_i - J_iJ_iB_i) \\
&= iJ_iB_i \\
&= i\bigotimes_{k=1}^{N}\sigma_k^{(J_i)}\sigma_k^{(B_i)},
\end{aligned} \tag{4.26}$$

where the third line follows because J_i anticommutes with B_i, and the fourth line follows because J_i is self-inverse. By the conditions on the $\sigma_k^{(J_i)}$, we see that $\sigma_k^{(J_i)}\sigma_k^{(B_i)} = I$ or $\pm iZ$ for each k, and $\pm iZ$ appears exactly once, so (4.26) becomes

$$\frac{1}{2}(1 + iJ_i)B_i(1 - iJ_i) = \pm\bigotimes_{k=1}^{N}\sigma_k^{(D_i)}, \tag{4.27}$$

where all $\sigma_k^{(D_i)} = I$ except one, which is Z. In other words, the rotation about J_i has mapped B_i to a single-qubit Z operator, as desired.

In each step, we apply the rotation $\exp\left(i\frac{\pi}{4}J_i\right)$ to all operators in the set. Thus we might worry that, having already mapped some subset of the $B_{i'}$ to single-qubit Z operators $D_{i'}$, applying some later rotation $\exp\left(i\frac{\pi}{4}J_i\right)$ to map B_i to a single-qubit Z operator could change the previously obtained $D_{i'}$. This turns out not to be the case, as we now show:

Consider some particular one of the $D_{i'}$, whose expansion as a tensor product is

$$D_{i'} = \bigotimes_{k=1}^{N}\sigma_k^{(D_{i'})} \tag{4.28}$$

where one of the $\sigma_k^{(D_{i'})}$ is Z and the others are I. Let m be the index such that

$\sigma_m^{(D_{i'})} = Z$. $D_{i'}$ commutes with B_i, since the previously applied rotations preserve commutation relations, so since $\sigma_m^{(D_{i'})} = Z$, $\sigma_m^{(B_i)}$ (as defined in (4.20)) must be I or Z. But this implies that J_i also commutes with $D_{i'}$, since we know that $\sigma_k^{(J_i)}$ is I or Z exactly when $\sigma_k^{(B_i)}$ is I or Z (by (4.26)), and thus $\sigma_m^{(J_i)}$ is I or Z. Therefore, the rotation that maps B_i to a single-qubit Z preserves the previously obtained $D_{i'}$.

Case 2. Suppose B_i is diagonal, so $\sigma_k^{(B_i)} \in \{I, Z\}$ for all k. Since any previously-obtained $D_{i'}$ are single-qubit Z operators and we assumed that the entire set is independent, B_i cannot be the product of any subset of the previously-obtained $D_{i'}$. Therefore, there must exist some $m \in \{1, 2, ..., n\}$ such that

$$\sigma_m^{(B_i)} = Z \tag{4.29}$$

and

$$\sigma_m^{(D_{i'})} = I \tag{4.30}$$

for all of the previously-obtained $D_{i'}$.

Apply the rotation $\exp\left(i\frac{\pi}{4}K_i\right)$, for K_i defined by

$$K_i = \bigotimes_{k=1}^{N} \sigma_k^{(K_i)}, \tag{4.31}$$

where

$$\sigma_m^{(K_i)} = Y \tag{4.32}$$

and

$$\sigma_k^{(K_i)} = I \tag{4.33}$$

for all $k \neq m$. Thus $\exp\left(i\frac{\pi}{4}K_i\right)$ commutes with and therefore does not change any previously-obtained $D_{i'}$.

As in Case 1, applying $\exp\left(i\frac{\pi}{4}K_i\right)$ to B_i obtains

$$\exp\left(i\frac{\pi}{4}K_i\right) B_i \exp\left(-i\frac{\pi}{4}K_i\right) = i\bigotimes_{k=1}^{N} \sigma_k^{(K_i)}\sigma_k^{(B_i)}, \tag{4.34}$$

where by construction,

$$\sigma_m^{(K_i)}\sigma_m^{(B_i)} = YZ = iX \tag{4.35}$$

and

$$\sigma_k^{(K_i)}\sigma_k^{(B_i)} = \sigma_k^{(B_i)} \tag{4.36}$$

for all $k \neq m$. In other words, the rotation has changed the Z at the mth spot in B_i into an X, and left B_i otherwise unchanged. We also apply this rotation to all other operators in the set, which does not change those that have already been mapped to single-qubit Z operators, as we noted above. Now B_i is no longer diagonal, so we proceed as described in Step 1 above, applying a second Pauli $\frac{\pi}{2}$-rotation to map B_i to a single-qubit Z operator. □

Let D denote the composition of the Clifford rotations that map the G_j to single-qubit Z operators, as in Lemma 4.4, and let $H' \coloneqq DHD^\dagger$ be the rotated Hamiltonian. After applying D, the noncontextual generators G_j have been mapped to single-qubit Z operators G'_j. Without loss of generality, since we have already found the noncontextual ground state, at this point we can choose the signs of the G'_j such that they all have eigenvalue $+1$ in the noncontextual ground state, and thus stabilize it.

Once D has been applied, we map $\mathcal{A}' = D\mathcal{A}D^\dagger$ to a single-qubit Z operator as well. $\mathcal{A}'$ is a normalized linear combination of the anticommuting Pauli operators $C'_{i1} = DC_{i1}D^\dagger$, as in (4.3), so we use unitary partitioning [IYLV20, ZTK$^+$20, RWT$^+$22], in particular the LCU implementation of it, for reasons we discuss below. Let R denote the resulting rotation; the result of applying it to $\mathcal{A}'$ is

$$R^\dagger \mathcal{A}' R = C'_{11}, \tag{4.37}$$

a single Pauli operator. Also, R is generated by products $C'_{i1}C'_{11}$ of the Pauli operators in $\mathcal{A}'$ [RWT$^+$22], so it commutes with the already diagonalized operators in G'.

C'_{11} commutes with and is independent of the operators in G', so since it is a

single Pauli operator, we can use Lemma 4.4 to map it to a single-qubit Z operator as well, without disturbing the operators in G'. Let $D_{\mathcal{A}'}$ denote the full rotation that maps $\mathcal{A}'$ to a single-qubit Z operator A''. We refer to the resulting final basis as the "rotated basis."

4.2.3 Restricting the Hamiltonian to a contextual subspace

In the rotated basis, we will restrict the Hamiltonian to the subspace stabilized by the noncontextual generators A'' and $\{G'_j \mid j = 1, 2, ..., n_1 - 1\}$ (thus defining n_1). Let us define Ω to be the set of these single-qubit Z operators:

$$\Omega \coloneqq \{A''\} \cup \{G'_j \mid j = 1, 2, ..., n_1 - 1\}. \tag{4.38}$$

Let $\mathcal{H}_1$ denote the Hilbert space of the n_1 qubits acted upon by operators in Ω, and let $\mathcal{H}_2$ denote the Hilbert space of the remaining n_2 qubits. Thus the full Hilbert space is $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$ and the total number of qubits is $n = n_1 + n_2$. Let H'_{c} be the contextual part of the Hamiltonian in the rotated basis:

$$H'_{\mathrm{c}} = \sum_{P \in \mathcal{S}'_{\mathrm{c}}} h_P P. \tag{4.39}$$

The set of Pauli terms in H'_{c} is $\mathcal{S}'_{\mathrm{c}}$ and terms in $\mathcal{S}'_{\mathrm{c}}$ in general act on both of the subspaces $\mathcal{H}_1$ and $\mathcal{H}_2$.

Note that since the rotation that maps $\mathcal{A}$ to A'' is non-Clifford, not only are the Pauli terms H'_{c} different from the terms in the original contextual part of the Hamiltonian, there can be a different number of them. In the original contextual subspace VQE paper [KTL21], this was not true, because we only explicitly diagonalized G, and instead enforced the eigenvalue of $\mathcal{A}$ at the level of the ansatz. However, subsequently in [RWT$^+$22], it was shown that by using the linear combination of unitaries implementation of unitary partitioning to map $\mathcal{A}'$ to the single-qubit Z operator A'', the increase in the number of terms in H'_{c} compared to H_{c} is at worst $O(N^2)$, where N is the number of cliques in the noncontextual part of the Hamil-

tonian (see Section 3.1.2). Furthermore, these terms can be efficiently classically calculated.

We can write the terms P in (4.39) as

$$P = P_1 \otimes P_2, \tag{4.40}$$

where P_1 is a Pauli operator acting on $\mathcal{H}_1$ and P_2 is a Pauli operator acting on $\mathcal{H}_2$. P commutes with an element of Ω if and only if $P_1 \otimes \mathbb{1}_{\mathcal{H}_2}$ does (where $\mathbb{1}_{\mathcal{H}_2}$ denotes the identity operator acting on $\mathcal{H}_2$), since the operators in Ω act only on $\mathcal{H}_1$. Since the noncontextual state corresponds to a subspace stabilized by Ω, if P anticommutes with any element of Ω, then its expectation value in the noncontextual state is zero. Hence, any P that admits a quantum correction must commute with all elements of Ω, and thus $P_1 \otimes \mathbb{1}_{\mathcal{H}_2}$ must as well. The elements of Ω comprise all single-qubit Z operators acting in $\mathcal{H}_1$, so P_1 must be a product of such operators. Hence the expectation value of P_1 is some $p_1 = \pm 1$, determined by the noncontextual state.

Let $|\psi_{(\vec{q},\vec{r})}\rangle$ be any quantum state consistent with the noncontextual state $(\vec{q},\vec{r})$. The action of any term P that admits a quantum correction on $|\psi_{(\vec{q},\vec{r})}\rangle$ has the form

$$\begin{aligned} P|\psi_{(\vec{q},\vec{r})}\rangle &= \big(P_1 \otimes P_2\big)|\psi_{(\vec{q},\vec{r})}\rangle \\ &= p_1\big(\mathbb{1}_{\mathcal{H}_1} \otimes P_2\big)|\psi_{(\vec{q},\vec{r})}\rangle. \end{aligned} \tag{4.41}$$

Therefore, if we denote by $H'_{\mathrm{c}}|_{(\vec{q},\vec{r})}$ the restriction of H'_{c} to its action on the noncontextual ground state $(\vec{q},\vec{r})$,

$$\begin{aligned} H'_{\mathrm{c}}|_{(\vec{q},\vec{r})} &= \sum_{\substack{P \in \mathcal{S}'_{\mathrm{c}} \\ \text{commuting} \\ \text{with } \Omega}} p_1 h_P \mathbb{1}_{\mathcal{H}_1} \otimes P_2 \\ &= \mathbb{1}_{\mathcal{H}_1} \otimes H'_{\mathrm{c}}|_{\mathcal{H}_2} \end{aligned} \tag{4.42}$$

for

$$H'_{\mathrm{c}}|_{\mathcal{H}_2} \coloneqq \sum_{\substack{P \in \mathcal{S}'_{\mathrm{c}} \\ \text{commuting} \\ \text{with } \Omega}} p_1 h_P P_2, \tag{4.43}$$

where p_1 and P_2 are functions of P defined by (4.41). This is a Hamiltonian on n_2 qubits, for n_2 given by

$$n_2 = n - n_1 = n - |G| - 1, \tag{4.44}$$

where $|G|$ is the number of noncontextual generators G_j. Furthermore, if terms $P \in \mathcal{S}'_{\mathrm{c}}$ are distinct only on their tensor factors P_1, the remaining operators P_2 in $H'_{\mathrm{c}}|_{\mathcal{H}_2}$ will be identical. Also, any terms that anticommute with any of the noncontextual generators Ω are dropped entirely (since their expectation values are zero). Thus, the restricted Hamiltonian $H'_{\mathrm{c}}|_{\mathcal{H}_2}$ may contain fewer than $|\mathcal{S}'_{\mathrm{c}}|$ terms. This is illustrated in Fig. 4.2, in Section 4.3.3. To obtain the quantum correction, we perform n_2-qubit VQE on the restricted Hamiltonian $H'_{\mathrm{c}}|_{\mathcal{H}_2}$ within the contextual subspace $\mathcal{H}_2$.

4.3 Contextual subspace VQE

The quantum correction to noncontextual approximations discussed in Section 4.2 allows us to use limited quantum resources to improve a classical simulation result. In this section we explain how we can systematically step back from the original noncontextual approximation in order to enlarge the contextual subspace, thus improving the overall accuracy of the approximation by using more quantum resources. This provides a parameter that can be specified based on the quantum resources available, taking us from the optimal noncontextual approximation at one extreme to full VQE at the other. We call this method *contextual subspace VQE*.

4.3.1 Method

We begin with a Hamiltonian H whose noncontextual approximation is H_{nc}. As discussed above, the noncontextual ground state corresponds to a joint eigenspace of

the noncontextual generators $G \cup \{\mathcal{A}\}$. We can trade accuracy of the noncontextual approximation for an improved quantum correction by decreasing the size of $G \cup \{\mathcal{A}\}$, which increases the dimension of the contextual subspaces. We accomplish this by decreasing the size of G, the set of generators for the commuting part of H_{nc}. Since the number of qubits used in the quantum correction procedure is the total number of qubits minus one minus the number of generators in G (see (4.44)), reducing the size of G increases the dimension of the search space for the quantum processor.

We work in the rotated basis, as in Section 4.2. In this basis, we select some subset of the noncontextual generators G'_j, and remove all terms generated by them from the noncontextual part. Since the G'_j are single-qubit Z operators, this means that for each generator to be dropped we remove from the noncontextual part all terms containing the corresponding single-qubit Z operator as a tensor factor. All the terms thus removed should be added to the quantum correction Hamiltonian H'_{c} (as in Section 4.2.3). We now implement the quantum correction on this expanded H'_{c}, keeping the same noncontextual ground state that we began with, but only applying its value assignments to the generators that remain in the noncontextual part.

The new noncontextual approximation on its own will in general be worse than the original noncontextual approximation. However, after including the new quantum correction the overall approximation cannot be worse, and will in general be better. This is because the values assigned in the original noncontextual approximation and quantum correction are still consistent with the noncontextual ground state, so quantum states that obtain those values are included in the new quantum search space. Thus, provided the state with lowest energy in the search space can be found, in the worst case the new approximation will only recover the original approximation. If the new quantum correction is nonzero for any additional terms, the new approximation will be strictly better than the original approximation. In the limit where we remove all terms from the noncontextual part and simulate them on the quantum computer, there will be no noncontextual approximation left, and we will have recovered full VQE.

The additional terms that can have nonzero quantum corrections after the removal procedure are those that anticommute with any of the generators G'_j that were removed, but commute with the remaining generators. These terms were previously restricted to null expectation values only because the noncontextual state was required to be a joint eigenstate of the removed generators, so when that is no longer enforced their expectation values can vary. Therefore, we can choose which subset of the generators to remove based on which will permit the optimal quantum correction.

Note that classically simulating the noncontextual part of the Hamiltonian is NP-complete, so in worst cases the classical simulation part of CS-VQE will not perform well [KL20]. However, worst case Hamiltonians for standard VQE are QMA-complete, meaning that a similar argument applies to VQE in general. Hence, in both cases we are interested in heuristic performance for specific Hamiltonians of interest, rather than worst cases. Framed in this way, what CS-VQE does is take standard VQE, which is a heuristic for an optimization problem over a set of parameters for a quantum circuit, and transform it into two heuristics (one classical and one quantum) for two smaller optimization problems. In practice, we have found that a combination of Monte-Carlo and gradient descent methods works well for the classical part of the algorithm, but continuing to optimize this is a topic for future work. Also, depending on what the noncontextual Hamiltonian is, it may be possible to determine what its ground state parameters are from physical arguments, as discussed further in the next subsection.

4.3.2 Choosing noncontextual Hamiltonians

In order to implement CS-VQE, we need a method for choosing the partition of a general Hamiltonian into noncontextual and contextual parts as in (4.1). Ideally, since the optimization of the noncontextual part is conducted first, we would like to choose the noncontextual part such that its ground state is a good approximation of the ground state of the full Hamiltonian, so that the corresponding expectation value assignments to the terms in the noncontextual part are as close as possible to

their values in the true ground state. As a heuristic for this goal, we can simply try to choose the noncontextual part to be as large as possible under some norm on the Pauli coefficients.

However, even this optimization will be classically hard in worst cases, since choosing a noncontextual subset of the Pauli terms is a generalization of the disjoint cliques problem [JSW93], which is NP-complete. Therefore, in general cases as must use a heuristic to solve it. One such heuristic is a pure greedy algorithm, in which we successively add terms to the noncontextual part in order of decreasing coefficient magnitude until no more terms can be added without making it contextual. This algorithm runs in $O(N^3)$ time for an N-term Hamiltonian. This is because, although testing a particular Hamiltonian for noncontextuality takes $O(N^3)$ time [KL19a], when that Hamiltonian is the result of adding a single new term to a Hamiltonian that we already know is noncontextual, all we need to do to check that the new Hamiltonian is noncontextual is make sure that the new term either commutes with all existing terms (checking this takes $O(N)$ time) or is not a member of any triple in which commutation is not transitive (checking this takes $O(N^2)$ time since we have to iterate over all possible other pairs of terms that could complete the triple). A greedy algorithm that adds optimal terms one at a time requires $O(N)$ steps, which multiplied by the $O(N^2)$ cost of each step yields the claimed overall $O(N^3)$ runtime. Note that this is a tighter analysis than was originally given for this greedy algorithm in [KL20]. This method is not optimal, but is efficient.

When applied to molecular electronic structure Hamiltonians, in all examples we tested (see Section 4.3.3, below), the above algorithm yields a noncontextual Hamiltonian composed of all diagonal Pauli terms plus one clique of off-diagonal terms. The corresponding fermionic terms are all number and Coulomb/exchange terms plus one single-excitation and the corresponding number-excitations. Although we originally observed this when using the greedy algorithm described above, we can in fact motivate why this is likely to always be the best choice of terms for molecular electronic structure Hamiltonians near equilibrium, since the resulting Pauli Hamiltonians are not in fact worst cases for the hard classical problem of

choosing the optimal noncontextual subset. Since it is tangential to the main thrust of this chapter, we leave this discussion to Appendix 4.6.

4.3.3 Applications

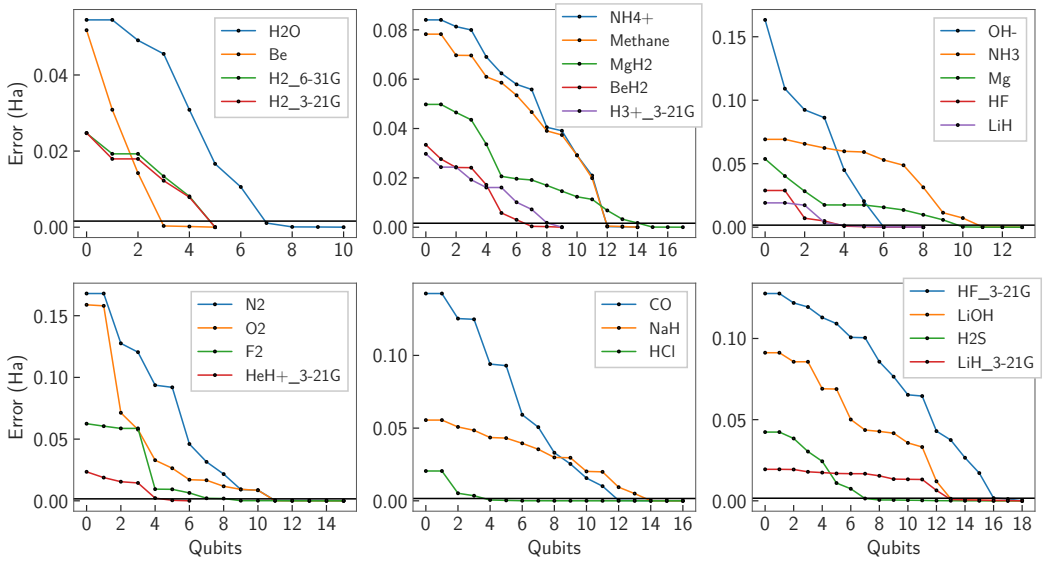


Figure 4.1: CS-VQE approximation errors versus number of qubits used on the quantum computer, for tapered molecular Hamiltonians. All Hamiltonians whose curves overlap in the region below chemical accuracy have the same total numbers of qubits. The solid black lines indicate chemical accuracy. Within each subplot, the ordering of the legend matches the vertical ordering of the leftmost points in the curves.

We tested CS-VQE on a set of electronic structure Hamiltonians in the Jordan-Wigner mapping [JW28]. In order to distinguish CS-VQE from qubit tapering, we first tapered the Hamiltonians using symmetries as in [BGMT17, SCR+20], then implemented CS-VQE in order to remove even more qubits. The initial noncontextual approximation Hamiltonians were chosen via a greedy classical algorithm, as described in Section 4.3.2. The quantum parts of the procedures were simulated classically by directly evaluating the lowest eigenvalues of the quantum correction Hamiltonians. The results are given in Fig. 4.1, which shows the overall CS-VQE approximation errors versus the number of qubits used on the quantum computer, and in Fig. 4.2, which shows the number of terms that must be simulated on the quantum computer in order to reach chemical accuracy using CS-VQE. Code to

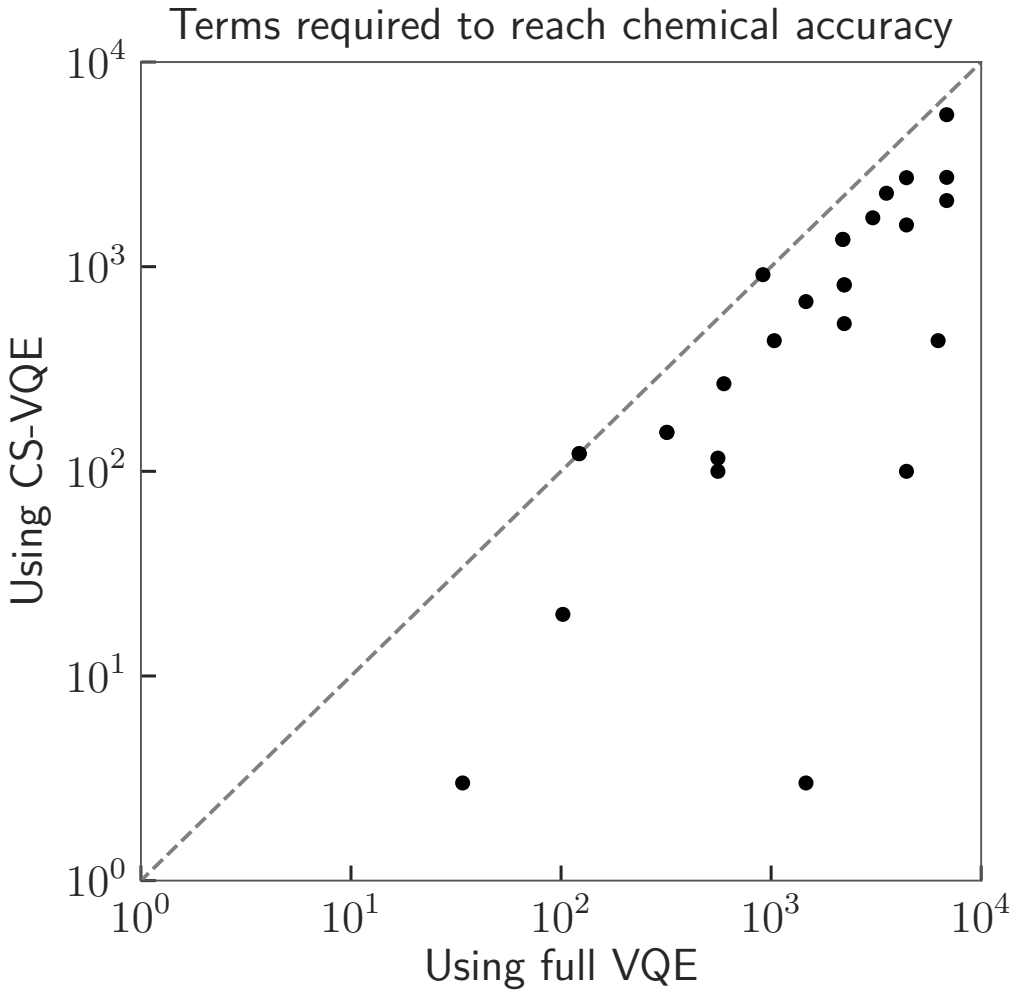


Figure 4.2: Number of terms simulated on the quantum processor required to reach chemical accuracy using CS-VQE versus using full VQE. The dashed line marks equality. All points represent either one, two, or three Hamiltonians.

reproduce our results is available[1]. Subsequent to our original CS-VQE paper, our collaborators have written an improved implementation[2] of CS-VQE, which should be used for any new applications of CS-VQE since it has improved runtimes and stability.

As noted at the end of Section 4.3.1, CS-VQE is sensitive to the order in which the qubits are moved from the noncontextual approximation to the quantum processor. In the calculations to obtain Figs. 4.1 and 4.2, we used a heuristic that

[1]Original CS-VQE source code: `https://github.com/wmkirby1/ContextualSubspaceVQE`

[2]Improved CS-VQE implementation (should be used for any new applications): `https://github.com/UCL-CCS/symmer`

begins with the noncontextual approximation, then adds qubits to the quantum correction two at a time, greedily choosing each pair to maximize the decrease in the ground state energy estimate. This method is informed by the structure of the noncontextual and contextual parts of the molecular Hamiltonians, and at the time of writing [KTL21] was the best heuristic we knew of that could be implemented efficiently without performing full VQE.

This heuristic involves running CS-VQE repeatedly, since for sufficiently large applications one would have to use the quantum processor to compute the quantum corrections on the way to choosing the set of qubits for the final quantum correction. However, these preliminary computations would only be necessary once the number of qubits chosen becomes unfeasible for classical simulation, and from Fig. 4.2 we see that the number of terms required to reach chemical accuracy can be many times smaller than the number of terms required to implement full VQE. Therefore, even the repeated runs of CS-VQE required for this heuristic can require fewer measurements overall than full VQE, and of course they also require fewer qubits.

In spite of this, it would be much more desirable to have a heuristic to determine the order without evaluating energies at all, so that the only VQE experiment to be performed would be the one in the final contextual subspace. We left this as an open question in [KHKL21] (upon which this chapter is based), and subsequent to its publication our collaborators Tim Weaving and Alexis Ralli have discovered a new family of heuristics that appear to satisfy all of our desiderata. These heuristics are based on using a classically-calculable weighting operator to score potential orders of the stabilizers by computing the $l1$-norms of the coefficients of terms in the operator that are permitted to vary by the input set of stabilizers. They have found numerically that using the coupled-cluster operator with single- and double-excitations yields subspaces that actually in general outperform those in our original paper, without requiring quantum resources to choose the subspace. These exciting new results are as yet unpublished.

4.4 Conclusion

In this chapter, we showed how to use a quantum computer to obtain a correction to a noncontextual approximation of a ground state energy. We then showed how to adjust the number of qubits used on the quantum computer in order to increase the accuracy of the hybrid approximation. This method, contextual subspace VQE or CS-VQE, is a true hybrid quantum-classical algorithm, in which the quantum resources used may be set to match whatever resources are available, and the classical approximation algorithm accounts for the remainder. The method is approximate, but variational, as is VQE itself. Exact methods will only achieve approximate results on NISQ devices due to their noisy character. CS-VQE allows the quantum resources used to be increased systematically until the desired precision is achieved, if possible.

Standard VQE is a heuristic algorithm: there are no analytic characterizations of its performance for general Hamiltonians, or even for special classes like electronic structure Hamiltonians, upon scaling the system size. This is also true for CS-VQE: its performance is sensitive to the specific problem to which it is applied. We do not analytically characterize the errors as a function of the number of qubits used on the quantum processor. However, the examples in Section 4.3.3 illustrate that CS-VQE performs well in many cases of interest going well beyond the scale of VQE implementations to date, so we hope that as the available quantum processors continue to grow, CS-VQE can be used to allow larger systems to be simulated using those processors.

The technique for restricting the quantum correction to the subspace consistent with the noncontextual ground state may appear reminiscent of using qubit tapering to exploit symmetries as described in [BGMT17, SCR$^+$20]. However, in CS-VQE the symmetries are intrinsic to the noncontextual ground state, rather than to the Hamiltonian (as in [BGMT17, SCR$^+$20]), and are thus under the experimenter's control. We illustrated this point in Section 4.3.3 by applying CS-VQE to Hamiltonians that were already tapered using the methods of [BGMT17, SCR$^+$20];

using CS-VQE we can eliminate additional qubits at will.

In this chapter we did not explore how to implement ansätze for the restricted VQE instance used by CS-VQE, instead finding the exact ground states of the contextual parts. However, standard ansatz classes, like unitary coupled-cluster (UCC) for electronic structure Hamiltonians [YCM+14, MRBAG16, RBM+18a], can be transformed into ansätze for CS-VQE by projecting the gates onto the contextual subspace, just as the contextual part of the Hamiltonian is restricted to the contextual subspace. This was studied in detail in the subsequent work [WRK+22].

One concern for standard VQE as well as for CS-VQE is that the ansatz may suffer from the *barren plateau problem* [MBS+18, UB21, CSV+21], where the gradient of the cost function (in this case expected energy) vanishes exponentially with the system size. It is hoped that for standard VQE, using physically-motivated ansätze like UCC may avoid the barren plateau problem, so since we can use projections of the same ansätze for CS-VQE, this same hope transfers to our case. However, even physically-motivated ansätze may be subject to *noise-induced barren plateaus* [WFC+21]: to the best of our knowledge, all variational quantum algorithms have the potential to fail in this way, including CS-VQE. Non-linear optimization and its attendant problems, including barren plateaus, may be avoided by the use of quantum imaginary time evolution (QITE) or similar methods [MST+20, MJE+19]. In our case, QITE could be applied directly to the contextual part of the Hamiltonian.

It is possible that some of the qubit and term reductions we obtained using CS-VQE have explanations in terms of chemistry. However, in such cases CS-VQE identifies and exploits such features using principles that are derived from the foundations of quantum mechanics, and are consequently agnostic any specific, high-level chemistry arguments. Identifying such chemical arguments would illustrate the role contextuality plays in chemistry, which would be of independent interest.

By using CS-VQE it is possible to reach chemical accuracy for ground state energies of numerous small molecules using many fewer qubits than would be required to implement full VQE on the tapered Hamitonians. The number of terms

and thus number of measurements required is also substantially reduced by using CS-VQE, since groups of terms become equivalent under the symmetry imposed by the noncontextual ground state. The number of measurements needed to obtain the quantum correction can be further reduced by the techniques described in [VYI20, YVI20, GAD+20, IYLV20, ZTK+20, HKP20, HBRM22, HKP21, SJM+21, Had21, HHR+21, WSHY21, YGI22, EFH+22], and this was studied in detail in [RWT+22] subsequent to publication of the work described in this chapter. Current VQE implementations are limited in both qubit count and number of measurements by the available hardware, so we expect CS-VQE to be of immediate practical value in accessing new molecular simulation applications on NISQ computers.

4.5 Appendix: Moving qubits from the noncontextual approximation to the quantum correction

We describe in detail how the noncontextual approximation is truncated to make room for improved quantum corrections. As in Section 4.2, we work in the rotated basis, denote by $\mathcal{H}_1$ the subspace of n_1 qubits acted upon by the noncontextual generators G', and denote by $\mathcal{H}_2$ the subspace of the n_2 remaining qubits, which are used to implement the quantum correction. Let $\mathcal{I}_2$ be the set of indices of these latter n_2 qubits (those not acted upon by G'), whose Hilbert space is the quantum search space $\mathcal{H}_2$. To increase the size of $\mathcal{H}_2$, we first need to select some subset of G' that we want to remove from H_{nc}. Since the elements of G' are single-qubit Z operators, this subset defines a set $\mathcal{I}_{\text{add}}$ of indices for qubits whose states are initially fixed by the noncontextual state, but that we will switch to simulating on the quantum processor. To begin with, from $\mathcal{S}'_{\text{nc}}$ (the terms in the noncontextual part of the Hamiltonian, in the rotated basis) we remove all terms that act on any of the qubits in $\mathcal{I}_{\text{add}}$, including the elements of G' that act on these qubits. The

remaining elements of G' form a new generating set $G'' \subset G'$ satisfying

$$|G''| = |G'| - |\mathcal{I}_{\text{add}}|. \tag{4.45}$$

Let the new noncontextual set of terms be denoted $\mathcal{S}''_{\text{nc}}$, and let

$$H''_{\text{nc}} \coloneqq \sum_{P \in \mathcal{S}''_{\text{nc}}} h_P P. \tag{4.46}$$

All terms that were removed from $\mathcal{S}'_{\text{nc}}$ to obtain $\mathcal{S}''_{\text{nc}}$ should be added to $\mathcal{S}'_{\text{c}}$ to obtain an expanded contextual set of terms $\mathcal{S}''_{\text{c}}$, whose corresponding Hamiltonian is

$$H''_{\text{c}} \coloneqq \sum_{P \in \mathcal{S}''_{\text{c}}} h_P P. \tag{4.47}$$

We can see that this removal operation preserves closure under inference of $\mathcal{S}''_{\text{nc}}$ by again thinking of $G' \cup \mathcal{A}'$ as stabilizers (up to some signs) for the contextual subspace. The elements of G' are single Pauli operators that are generators for the stabilizer group of the subspace. In order to increase the dimension of the stabilized subspace, therefore, for each element G'_i of G' that we remove we must also remove all elements of the stabilizer group that include G'_i as a factor. In other words, in this instance preserving closure under inference is equivalent to preserving closure of a stabilizer group.

We can now implement the quantum correction on H''_{nc} and H''_{c}, keeping the same noncontextual ground state that we began with, but only applying its value assignments to terms in H''_{nc}. Let n''_2 denote the new number of qubits used in the quantum correction procedure: then by (4.45),

$$n''_2 = n - |G''| = n - |G| + |\mathcal{I}_{\text{add}}| = n_2 + |\mathcal{I}_{\text{add}}, \tag{4.48}$$

where n_2 was the initial number of qubits used in the quantum correction procedure.

type:	fermionic form	types of Pauli operators
number	$h_{ii}a_i^\dagger a_i$	Z_i
Coulomb/exchange	$h_{ijji}a_i^\dagger a_j^\dagger a_j a_i$	$Z_i,\ Z_iZ_j$
single-excitation	$h_{ij}(a_i^\dagger a_j + a_j^\dagger a_i)$	$X_iZZ\cdots ZX_j,\ Y_iZZ\cdots ZY_j$
number-excitation	$h_{ikkj}(a_i^\dagger a_k^\dagger a_k a_j + a_j^\dagger a_k^\dagger a_k a_i)$	above + $X_iZZ\cdots ZX_jZ_k$, $Y_iZZ\cdots ZY_jZ_k$
double-excitation	$h_{ijkl}(a_i^\dagger a_j^\dagger a_k a_l + a_l^\dagger a_k^\dagger a_j a_i)$	$Q^{(X,Y)}ZZ\cdots ZQ^{(X,Y)}II$ $\cdots IQ^{(X,Y)}ZZ\cdots ZQ^{(X,Y)}$

Table 4.1: Types of electronic structure Hamiltonian terms, showing their fermionic forms and the resulting one or more Pauli terms. All indices i, j, k, l are assumed to be distinct: otherwise the distinct types can collapse into each other. In the last line, all products are tensor products, and each $Q^{(X,Y)}$ is either X or Y such that an even number are X (so an even number are Y as well).

4.6 Appendix: choosing noncontextual sub-Hamiltonians from molecular electronic-structure Hamiltonians

The electronic structure Hamiltonian may be represented in terms of Pauli operators via several fermion-to-qubit mappings. It may also be represented in terms of Majorana operators. From the perspective of CS-VQE, which of the above we use has no formal effect on the performance of the algorithm (although it may matter in practice on a noisy quantum computer), since JW, BK, and parity mappings all map Majorana terms one-to-one onto Pauli terms, and the resulting commutation relations and spectral properties are identical. Therefore, in this appendix I will use the Jordan-Wigner mapping so that qubits are equivalent to fermionic modes, bearing in mind that this choice is in fact without loss of generality.

4.6.1 Types of Pauli terms in electronic structure Hamiltonians

Table 4.1 gives the different types of Pauli terms that appear in the electronic structure Hamiltonian, as well as the fermionic term types that they come from.

We can derive Table 4.1 as follows. Fermionic number terms have the form

$$h_{ii}a_i^\dagger a_i, \tag{4.49}$$

which in the JW mapping becomes

$$h_{ii}\frac{1}{2}(1 - Z_i). \tag{4.50}$$

Hence, the nontrivial Pauli terms corresponding to fermionic number terms are single-qubit Z operators.

Coulomb/exchange terms have the form

$$h_{ijji}a_i^\dagger a_j^\dagger a_j a_i = h_{ijji}a_j^\dagger a_j a_i^\dagger a_i. \tag{4.51}$$

In the JW mapping, this becomes

$$h_{ijji}\frac{1}{4}(1 - Z_i)(1 - Z_j) = h_{ijji}\frac{1}{4}(1 - Z_i - Z_j + Z_iZ_j). \tag{4.52}$$

Single-excitation terms have the form

$$h_{ij}(a_i^\dagger a_j + a_j^\dagger a_i) = h_{ij}(a_i^\dagger a_j - a_i a_j^\dagger) = h_{ij}\left[\frac{1}{2}(a_i^\dagger - a_i)(a_j^\dagger + a_j) - \frac{1}{2}(a_i^\dagger + a_i)(a_j^\dagger - a_j)\right]. \tag{4.53}$$

In the JW mapping, this becomes

$$\frac{1}{2}h_{ij}(Y_iZZ\cdots ZY_j + X_iZZ\cdots ZX_j). \tag{4.54}$$

Number-excitation terms have the form

$$\begin{aligned} &h_{ikkj}(a_i^\dagger a_k^\dagger a_k a_j + a_j^\dagger a_k^\dagger a_k a_i) = h_{ikkj}(a_i^\dagger a_j - a_i a_j^\dagger)a_k^\dagger a_k \\ &= h_{ikkj}\left[\frac{1}{2}(a_i^\dagger - a_i)(a_j^\dagger + a_j) - \frac{1}{2}(a_i^\dagger + a_i)(a_j^\dagger - a_j)\right]a_k^\dagger a_k. \end{aligned} \tag{4.55}$$

In the JW mapping, this becomes

$$\frac{1}{4}h_{ikkj}(Y_iZZ\cdots ZY_j + X_iZZ\cdots ZX_j)(1-Z_k). \tag{4.56}$$

Double-excitation terms have the form

$$h_{ijkl}(a_i^\dagger a_j^\dagger a_k a_l + a_l^\dagger a_k^\dagger a_j a_i) = h_{ijkl}(a_i^\dagger a_j^\dagger a_k a_l + a_i a_j a_k^\dagger a_l^\dagger). \tag{4.57}$$

In the JW mapping, this becomes

$$\begin{aligned}
&\frac{1}{16}h_{ijkl}\Big[(X_i - iY_i)ZZ\cdots Z(X_j - iY_j)(X_k + iY_k)ZZ\cdots Z(X_l + iY_l) \\
&\qquad + (X_i + iY_i)ZZ\cdots Z(X_j + iY_j)(X_k - iY_k)ZZ\cdots Z(X_l - iY_l)\Big] \\
&= \frac{1}{8}h_{ijkl}\Big[X_iZZ\cdots ZX_jX_kZZ\cdots ZX_l - X_iZZ\cdots ZX_jY_kZZ\cdots ZY_l \\
&\qquad + X_iZZ\cdots ZY_jX_kZZ\cdots ZY_l + X_iZZ\cdots ZY_jY_kZZ\cdots ZX_l \\
&\qquad + Y_iZZ\cdots ZX_jX_kZZ\cdots ZY_l + Y_iZZ\cdots ZX_jY_kZZ\cdots ZX_l \\
&\qquad - Y_iZZ\cdots ZY_jX_kZZ\cdots ZX_l + Y_iZZ\cdots ZY_jY_kZZ\cdots ZY_l\Big].
\end{aligned} \tag{4.58}$$

The above is slightly imprecise, because exactly where the strings of Zs should fall depends on the ordering of i, j, k, l (the above implicitly assumes $i < j < k < l$ or $i > j > k > l$). However, since we only care about the form of the terms, we can note that all Pauli terms have the form

$$Q^{(X,Y)}ZZ\cdots ZQ^{(X,Y)}II\cdots IQ^{(X,Y)}ZZ\cdots ZQ^{(X,Y)}, \tag{4.59}$$

where all products are tensor products and each $Q^{(X,Y)}$ is either X or Y such that an even number of them are X (so an even number are Y as well).

4.6.2 Possible choices for the noncontextual sub-Hamiltonian

The possible types of Pauli terms given in Table 4.1 dramatically restrict the forms of noncontextual Hamiltonians we can see. We will list several "maximal" possibilities, where "maximal" means that adding any single additional term to one of these

Hamiltonians would make it contextual.

4.6.2.1 Diagonal terms + one single-excitation and associated number-excitations

This is one of two types of maximal noncontextual Hamiltonian we obtain if we insist that it include all of the diagonal Pauli operators, i.e., that it contain all number and Coulomb/exchange terms. The corresponding Pauli terms are single-qubit Zs acting on all qubits, and ZZs acting on all pairs of qubits. Suppose we add one single-excitation. This contains XX and YY, both acting on the same pair of qubits, as well as strings of Z operators acting on other qubits. Both of these anticommute with the same subset of the diagonal Pauli terms, and commute with the remainder. Hence, the diagonal Pauli terms that they anticommute with must form one clique, and the two terms coming from the single-excitation must be in the other clique.

Any other single-excitation contains XX and YY acting on some other pair of qubits. These will anticommute with some strictly different subset of the diagonal Pauli terms (this is easy to see among the single-qubit Zs). Therefore, by the same argument as above, in order for this new single-excitation to be included in the noncontextual part, this other new set of diagonal Pauli terms would have to form a distinct clique, but this is impossible because they commute with the diagonal clique defined by the first single-excitation. Hence, once one single-excitation is included, no others can be.

A similar argument rules out including any double-excitations and any number excitations besides those containing the same excitation as the single-excitation we have chosen. These can be included, since the resulting Pauli terms are the same as those corresponding to the single-excitation, but simply multiplied by some additional single Z. Hence, the maximal noncontextual Hamiltonian obtained by including all diagonal terms and one single-excitation includes all of the corresponding number-excitations as well, and nothing else. Note that the same argument applies if we instead add one number-excitation to the diagonal terms, and the re-

sulting noncontextual Hamiltonian is the same as if we had added the corresponding single-excitation. This is the type of noncontextual Hamiltonian suggested in the summary section above as the best choice for CS-VQE.

4.6.2.2 Diagonal terms + one double-excitation

Suppose we again insist that the noncontextual Hamiltonian contain all diagonal Pauli terms, but now add one double-excitation instead of one single-excitation. As for the single-excitation, since the double-excitation includes commuting terms of the form $XXXX$, $XXYY$, or $YYYY$ with additional strings of Zs, it defines a subset of the diagonal Pauli terms that anticommute with all of these and thus must form one of the cliques. The other clique can contain only the terms corresponding to the chosen double-excitation, since there is nothing analogous to number-excitations for double-excitations. No single-excitations or number-excitations can be included, because all of them would define different cliques among the diagonal Pauli terms, as discussed in Section 4.6.2.1. Thus the maximal noncontextual Hamiltonian obtained by including all diagonal Pauli terms plus one double-excitation is exactly that: no more terms can be included.

4.6.2.3 Two-cells

The above two subsections cover all possible maximal noncontextual Hamiltonians containing all diagonal Pauli terms. Hence, in order to obtain more possibilities, we must begin to restrict to subsets of the diagonal terms. As an example, suppose we partition the qubits up into pairs that we can refer to as cells. We then include terms corresponding to single-excitations acting on each cell, double-excitations where each excitation in the term acts on a single cell, and number Coulomb/exchange, and number-excitation operators acting only on one particular cell (in the number-excitation case, meaning that the single-excitation in it can act on any other cell but the number operator must act on the chosen 'special' cell). Without loss of generality we can label the qubits in that particular cell as 0 and 1 and let all the

cells be labeled by pairs of adjacent qubit indices, so the included Pauli terms are

$$
\begin{aligned}
&Z_0, Z_1, Z_0Z_1, \\
&X_0X_1, Y_0Y_1, X_2X_3, Y_2Y_3, X_4X_5, Y_4Y_4, ..., \\
&Z_0X_2X_3, Z_1X_2X_3, Z_0Y_2Y_3, Z_1Y_2Y_3, Z_0X_4X_5, Z_1X_4X_5, Z_0Y_4Y_4, Z_1Y_4Y_4, ..., \\
&X_0X_1X_2X_3, X_0X_1Y_2Y_3, Y_0Y_1X_2X_3, Y_0Y_1Y_2Y_3, \\
&X_0X_1Z_2Z_3X_4X_5, X_0X_1Z_2Z_3Y_4Y_5, Y_0Y_1Z_2Z_3X_4X_5, Y_0Y_1Z_2Z_3Y_4Y_5, ...
\end{aligned}
\tag{4.60}
$$

The resulting noncontextual structure is

$$
\begin{aligned}
\mathcal{Z} &= \{Z_0Z_1\} \cup \{X_2X_3, Y_2Y_3, X_4X_5, Y_4Y_5, X_6X_7, Y_6Y_7, ...\}^2, \\
C_1 &= \{Z_0, Z_1\} \cdot \{X_2X_3, Y_2Y_3, X_4X_5, Y_4Y_5, X_6X_7, Y_6Y_7, ...\}, \\
C_2 &= \{X_0X_1, Y_0Y_1\} \cdot \{X_2X_3, Y_2Y_3, X_4X_5, Y_4Y_5, X_6X_7, Y_6Y_7, ...\},
\end{aligned}
\tag{4.61}
$$

where products of sets denote the sets of all products of operators, one from each set (and thus the square of a set denotes the set of products of pairs in the set). This is a viable noncontextual subset of the terms that includes a much larger number of excitations and double-excitations than our previous subsets. However, for molecules near their equilibrium nuclear configurations the diagonal Pauli terms, corresponding to number and Coulomb/exchange fermionic terms, dominate the coefficient weight, which leads to the above subset yielding dramatically different spectra and eigenbases than the original molecular Hamiltonians it is derived from. Hence, this is not a good choice for CS-VQE. While a complete categorization of all possible forms of maximal noncontextual subsets in electronic structure Hamiltonians has not been carried, the above example illustrates how, once more than one single-excitation has been included, many number and Coulomb-exchange terms must be excluded. By the argument we have just made, we expect that excluding number and Coulomb-exchange terms will lead to stronger errors, at least for molecules near equilibrium, than excluding most single- and double-excitations, so in such cases we believe that one of the first two subsets above, which include all number and Coulomb-exchange

terms, is likely to be optimal.

Chapter 5

Variational quantum eigensolvers for sparse Hamiltonians

5.1 Introduction

Methods that approximate energy eigenvalues, such as VQE and quantum phase estimation Section 1.3.3, rely on efficient representations of the Hamiltonian. As discussed in Section 1.3.1, three such representations are widely used: linear combinations of Pauli operators (which can efficiently represent all local Hamiltonians), sparse Hamiltonians, and linear combinations of unitaries. The sparse Hamiltonian input model strictly generalizes linear combinations of Pauli operators, since as discussed in Section 1.3.1, each Pauli operator in the linear combination is one-sparse, so if there are N of them in total, the resulting Hamiltonian is at most N-sparse.

In variational quantum eigensolvers to date, the Hamiltonian input model has been restricted to linear combinations of Pauli operators (1.19), which we reproduce here for convenience:

$$H = \sum_{i=1}^{N} h_i P_i, \tag{5.1}$$

where the h_i are real coefficients. On the other hand, other types of quantum

simulation algorithms for sparse Hamiltonians have undergone extensive study (see Section 1.3.2 and [ATS03, CCD$^+$03, BACS07, Chi10, CK10, CK11, BC12, CW12, BCC$^+$14, BCK15, BCC$^+$15, LC17, LW18, LC19, BCS$^+$20]), culminating in algorithms with optimal (for time-independent Hamiltonians) [LC17, LC19] or near-optimal (for time-dependent Hamiltonians) [BCS$^+$20] scaling with all parameters. In these algorithms a d-sparse Hamiltonian is accessed via a pair of oracle unitaries O_L and O_V as defined in (1.21) and (1.22), which we also reproduce here for convenience:

$$O_L|x, i\rangle = |x, y(x, i)\rangle, \tag{5.2}$$

where x is the index of a row in H (i.e., a basis state in the chosen basis), $i \in [k]$, and $y(x, i)$ is the index of the ith nonzero entry in row x of H;

$$O_V|x, y, 0\rangle = |x, y, H_{xy}\rangle, \tag{5.3}$$

where H_{xy} is the value of entry x, y in H. The Hamiltonians are given via these oracle unitaries for the sake of modularity: the sparse Hamiltonians are very general, so oracle queries provide a standardized formalism for accessing them.

As discussed in Section 1.3.3, the VQE algorithm comprises two main components: a quantum subroutine for estimating the expectation value of a Hamiltonian of interest for some parametrized ansatz state, and a classical outer loop that updates the parameters of the ansatz in order to minimize the expected energy [PMS$^+$14]. The quantum subroutine is implemented by separately estimating expectation values of the terms in the Hamiltonian under some decomposition, which up to this point has always been the Pauli decomposition (5.1).

In this chapter, we extend VQE to sparse Hamiltonians, the possibility of which was briefly discussed in the appendix of [PMS$^+$14]. We decompose sparse Hamiltonians into linear combinations of self-inverse one-sparse Hermitian matrices. We then show how to estimate expectation values of these one-sparse terms using two ansatz state preparations and calls to the oracle unitaries defining the Hamiltonian

terms. We will show that our algorithm requires at most six oracle queries per measurement circuit.

The class of sparse Hamiltonians that admit description by oracles of the forms (5.2) and (5.3) is much broader than the class of local Hamiltonians, which admit efficient Pauli decompositions and are thus simulable by standard VQE. To prove that local Hamiltonians are a subset of sparse Hamiltonians with efficient oracle descriptions, it is enough to give oracle descriptions of the Pauli operators, which we do explicitly in Section 5.6.1. Hence for any local Hamiltonian, we could first decompose it into Pauli operators [JW28, BK02, SRL12], and then simulate each Pauli operator using sparse VQE. All electronic structure Hamiltonians that can be simulated using standard VQE can be simulated using sparse VQE in this way. This also provides an example of oracles with simple implementations that are appropriate for NISQ devices.

However, when a Hamiltonian has an efficient Pauli decomposition, it is not a good candidate for sparse VQE because the measurement scheme in Section 5.2 requires an extra qubit and an extra ansatz preparation compared to measuring the Pauli terms directly. The cases of real interest for sparse VQE are Hamiltonians that are sparse and admit efficient oracle implementations, but do not admit efficient Pauli decompositions.

One such case is a Hamiltonian that includes bosons and is represented in a *direct encoding* [KKG+22], in which the occupation of each mode is stored in binary in its own register of qubits. Bosonic creation and annihilation operators in this encoding are naturally represented in terms of Weyl-Heisenberg shift operators but not as Pauli operators because the occupations of modes can be larger than one. In Section 5.6.2, we give explicit implementations of oracles for this case. These implementations can be combined with the oracles for Pauli operators to handle Hamiltonians that act on both fermions and bosons.

The class of sparse Hamiltonians is very large, and we will not attempt to give an exhaustive list of all theories that can be addressed within it. However, two more examples are as follows. The first is quantum field theory in *compact*

encoding [KKG+22, KHKL21], in which only the occupations of occupied modes are stored, providing asymptotically optimal space efficiency. Oracles for field theories in compact encoding are explicitly constructed in [KHKL21]. The second example is the CI-matrix representation of quantum chemistry, for which oracles are explicitly constructed in [BBS+17].

The number of gates and depth of circuits required by the oracles in [KHKL21] and [BBS+17] are larger than those required for Pauli operators or for quantum field theory in the direct encoding. Implementation of these oracles in the NISQ era would require extensive error mitigation or significantly improved physical gates and qubits. However, sparse VQE will become possible before other sparsity-based simulation algorithms [BACS07, BC12, BCC+14, BCK15, BCC+15, LC17, LC19, BCS+20], because these require more coherent queries to the same oracles.

In Section 5.2 we describe the basic structure of VQE for Hamiltonians that can be decomposed into self-inverse one-sparse Hermitian terms that possess efficient circuit representations. In Section 5.3, we describe methods for obtaining such decompositions. Then in Section 5.4, we explain how to construct efficient circuit representations of the resulting terms. These methods permit the implementation of efficient VQE procedures for sparse Hamiltonians. We close the chapter with some discussion and directions for future work in Section 5.5. This chapter is based on [KL21].[1]

5.2 Sparse VQE

VQE was first used to estimate expectation values of the Hamiltonian [PMS+14]. However, many other quantities are of interest given an ansatz state that is a good approximation to the ground state or other energy eigenstate. For example, [KKG+22, KJK+21a, KJK+21b] study various properties of composite particles in interacting quantum field theory. Properties such as the invariant mass, mass radius, parton distribution function, and form factor are expectation values of

[1] ©2021 American Physical Society, reproduced per Author's Rights under the transfer of copyright agreement (`https://journals.aps.org/authors/transfer-of-copyright-agreement`).

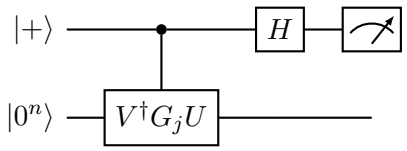


Figure 5.1: Hadamard test circuit realizing estimation of the real part of the matrix element $\langle 0|V^\dagger G_j U|0\rangle$.

corresponding operators, whereas quantities such as the decay constant are matrix elements between different states [KJK+21b]. We will therefore consider estimation of quantities $\langle\phi|\hat{O}|\psi\rangle$ for sparse operators $\hat{O}$ between ansatz states $|\phi\rangle = V|0\rangle$ and $|\psi\rangle = U|0\rangle$ prepared by quantum circuits U and V.

We begin with a Hermitian operator that we assume has an efficient decomposition into a sum of Hermitian, self-inverse, one-sparse terms G_j:

$$\hat{O} = \sum_{j=1}^{t} \alpha_j G_j \tag{5.4}$$

where α_j are real coefficients and the number of terms t is polylogarithmic in the dimension of the Hilbert space on which $\hat{O}$ acts. If $\hat{O}$ can be efficiently decomposed into Pauli operators, then the Pauli decomposition of $\hat{O}$ is an example of (5.4) because the Pauli operators are self-inverse and one-sparse. The terms G_j are both Hermitian and unitary, and we further assume that an efficient quantum circuit for each G_j is known. Any operator $\hat{O}$ of the form (5.4) is sparse, and the number of one-sparse terms t is an upper bound on the sparsity. In Section 5.3 we will explicitly show how to construct a decomposition as in (5.4) for any arbitrary sparse Hermitian operator. However, for the purpose of this Section it is enough to assume that it is possible, because the actual VQE implementation is agnostic to the method used to obtain the decomposition.

Given (5.4) we perform M Hadamard tests of the operators $V^\dagger G_j U$ via the circuit shown in Fig. 5.1. This circuit has a state register of n qubits initialized in

the all zeros state $|0^n\rangle$ and a single ancilla register initialized in the state

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}. \tag{5.5}$$

The first operation is application of $V^\dagger G_j U$ controlled on the ancilla qubit. The second operation is a single qubit Hadamard gate applied to the ancilla qubit. After application of this circuit, the probability of observing zero on the ancilla qubit is:

$$p(0) = \frac{1}{2}\left(1 + \mathrm{Re}\langle 0^n|V^\dagger G_j U|0^n\rangle\right). \tag{5.6}$$

To replace the real part by the imaginary part of $V^\dagger G_j U$ in (5.6), replace the initial $|+\rangle$ state of the ancilla by the state

$$|-i\rangle = \frac{|0\rangle - i|1\rangle}{\sqrt{2}}. \tag{5.7}$$

After M repetitions of the circuit in Fig. 5.1, one obtains n_0 zeros and n_1 ones from the measurement outcomes of the ancilla qubit. Some simple algebra shows that the quantity $(n_0 - n_1)/M$ is an estimate of $\mathrm{Re}\langle 0^n|V^\dagger G_j U|0^n\rangle$. We can therefore interpret ancilla outcome b as determining a random variable with value $(-1)^b$. The analysis of the variance of these estimates and hence the scaling of M for given $\hat{O}$ and precision ϵ proceeds exactly as for Pauli decompositions (given in [MRBAG16, RBM18b]), so the required M for precision ϵ scales as ϵ^{-2}.

In this section we have given the extension of VQE to matrices that can be efficiently decomposed into self-inverse one-sparse Hermitian terms described by efficient quantum circuits[2]. Estimation of a matrix element of a self-inverse term G_j between two ansatz states $U|0^n\rangle$ and $V|0^n\rangle$ is accomplished by controlled application of the ansatz circuits $U^\dagger$ and V as well as G_j. For estimation of expectation values we have $U = V$ and twice as many ansatz preparations are required as for Pauli

[2]Note that the linear combinations of unitaries input model (Section 1.3.1) is defined as a list of quantum circuits together with a list of coefficients, so by using a Hadamard test we can immediately implement VQE for LCU Hamiltonians as well, as long as we can implement controlled versions of block-box unitaries. While this is impossible in general, given circuit decompositions of those unitaries it is generally possible [AFCB14, FDDB14, GST20].

decomposition VQE. The necessity of these extra preparations is apparent when one notes the capability to also estimate matrix elements between distinct states. In the remainder of the chapter, we will focus on Hamiltonians for easier comparison to prior literature on sparse Hamiltonian simulation, but all of our results will apply to general sparse, Hermitian observables. We will discuss obtaining the necessary sparse decompositions in Section 5.3, and efficiently applying the resulting operators in Section 5.4.

5.3 Obtaining sparse decompositions

Access to a d-sparse Hamiltonian is provided by the oracles O_L and O_V as defined in (5.2) and (5.3). In the case that H is one-sparse we can simplify the action of O_L:

$$O_L|x,0\rangle = |x,y_x\rangle, \tag{5.8}$$

where y_x is defined to be the column-index of the single nonzero entry in row x (corresponding to $i = 0$).

Given a d-sparse Hamiltonian, we wish to decompose it into a polynomial number of one-sparse, self-inverse, Hermitian terms. In some cases of interest, most notably fermionic Hamiltonians in second-quantized form, we already have a decomposition into one-sparse Hermitian terms. A Hamiltonian expressed in second-quantized form is a polynomial of some set of ladder operators for various particles or modes. The basis for the Hilbert space is given by the occupation number (Fock) representation for each of the modes. Each term in the Hamiltonian is a monomial of ladder operators, which for fermions in the Fock basis is one-sparse since its action as a linear transformation is to map each single Fock state to some scaling of a single Fock state. Therefore, the fermionic Hamiltonian in the occupation number basis is at most d-sparse if it contains d terms, so assuming the number of terms is polynomial in the number of qubits, so is the sparsity.

Ladder operator monomials are in general not self-inverse, nor are they Hermitian. However, for each ladder operator monomial present in the fermionic Hamil-

tonian, its Hermitian conjugate must also be present, and each such pair together is one-sparse and Hermitian. One-sparseness follows because for a fermionic ladder operator monomial, any given state is mapped to zero by either the monomial or its conjugate (or both); this in fact extends to any Hamiltonian that contains fermionic ladder operators with nonidentity action in every term, even if bosonic operators are also present. To obtain a decomposition into one-sparse terms that are also self-inverse, we use the following Lemma:

Lemma 5.1. *Any one-sparse Hamiltonian $H^{(1)}$ may be expressed up to to L bits per real and imaginary part of each entry as a linear combination of*

$$4L = 4\left\lceil \log_2\left(\frac{\sqrt{2}\,\|H^{(1)}\|_{max}}{\gamma}\right)\right\rceil \tag{5.9}$$

one-sparse, self-inverse Hamiltonians G_j, where γ is the resulting error in max-norm. The O_L oracles for the G_j are the same as the O_L oracle for $H^{(1)}$, and the O_V oracle for any G_j may be computed using two queries to the O_V oracle for $H^{(1)}$.

Remark 5.1. Note that $\|H^{(1)}\|_{\max}$ denotes the max-norm of $H^{(1)}$, defined to be the maximum magnitude of any entry in $H^{(1)}$, which is upper-bounded by $\|H^{(1)}\|_\infty$ [CK10].

Proof. Let Λ denote the least power of two that is greater than $\|H^{(1)}\|_{\max}$, the maximum magnitude of any entry in $H^{(1)}$. Define $C^{\mathrm{Re},l}$ and $C^{\mathrm{Im},l}$ as the matrices of lth bits of the real and imaginary parts of $H^{(1)}$ (each of which is assumed to begin with the bit corresponding to $\Lambda/2$), so

$$H^{(1)} \approx \sum_{l=1}^{L} \frac{\Lambda}{2^{l+1}}\left(C^{\mathrm{Re},l} + iC^{\mathrm{Im},l}\right), \tag{5.10}$$

where the approximation is due to cutting the entries off at L bits. The error in max-norm in (5.10) is upper-bounded by

$$\gamma \leq \sqrt{2}\frac{\Lambda}{2^{L+1}} \leq \frac{\sqrt{2}\,\|H^{(1)}\|_{\max}}{2^L}, \tag{5.11}$$

since upon truncating to L bits both the real and imaginary part in any entry may incur an error at most $\frac{\Lambda}{2^{L+1}}$, giving a total error of at most $\sqrt{2}\frac{\Lambda}{2^{L+2}}$ in any entry, and $\Lambda \leq 2\|H^{(1)}\|_{\max}$.

Let j_i be the index of the single nonzero entry in row i of $H^{(1)}$ (or if there is no nonzero entry in row i, let j_i be whatever index is computed by applying the O_L oracle for $H^{(1)}$). In terms of this, let us entrywise define the operators $G^{\mathrm{Re},l,\pm}$ as follows:

$$\begin{aligned} G^{\mathrm{Re},l,\pm}_{ij_i} &= \begin{cases} 1 & \text{if } C^{\mathrm{Re},l}_{ij_i} = 1, \\ -1 & \text{if } C^{\mathrm{Re},l}_{ij_i} = -1, \\ \pm 1 & \text{if } C^{\mathrm{Re},l}_{ij_i} = 0 \end{cases} \\ &= \begin{cases} 1 & \text{if } \mathrm{Re}[H^{(1)}_{ij_i}] > 0 \text{ and bit } l \text{ of } \mathrm{Re}[H^{(1)}_{ij_i}] \text{ is 1}, \\ -1 & \text{if } \mathrm{Re}[H^{(1)}_{ij_i}] < 0 \text{ and bit } l \text{ of } \mathrm{Re}[H^{(1)}_{ij_i}] \text{ is 1}, \\ \pm 1 & \text{if bit } l \text{ of } \mathrm{Re}[H^{(1)}_{ij_i}] \text{ is 0}, \end{cases} \end{aligned} \tag{5.12}$$

and let all other entries in $G^{\mathrm{Re},l,\pm}$ be zero. Let the operators $G^{\mathrm{Im},l,\pm}$ be defined similarly in terms of the $C^{\mathrm{Im},l}$ or the imaginary parts of the entries in $H^{(1)}$. From these definitions it follows immediately that

$$C^{\mathrm{Re},l} = \frac{1}{2}\left(G^{\mathrm{Re},l,+} + G^{\mathrm{Re},l,-}\right) \tag{5.13}$$

and

$$C^{\mathrm{Im},l} = \frac{1}{2}\left(G^{\mathrm{Im},l,+} + G^{\mathrm{Im},l,-}\right) \tag{5.14}$$

for each l, so

$$H^{(1)} \approx \sum_{l=1}^{L} \frac{\Lambda}{2^{l+2}}\left(G^{\mathrm{Re},l,+} + G^{\mathrm{Re},l,-} + iG^{\mathrm{Im},l,+} + iG^{\mathrm{Im},l,-}\right), \tag{5.15}$$

with the error still given by (5.11). The operators $G^{\mathrm{Re},l,\pm}$ and $G^{\mathrm{Im},l,\pm}$ have nonzero entries exactly where $H^{(1)}$ does, so they are one-sparse, and all of their nonzero

entries are ± 1. Therefore, they are not only one-sparse, but self-inverse as well, so (5.15) is our desired decomposition. The G_j in the lemma statement are the $G^{\mathrm{Re},l,-}$ and $G^{\mathrm{Im},l,+}$, so j is a multi-index denoting the superscripts in the latter.

(5.15) contains $4L$ terms, and the error in max-norm is upper bounded by (5.11). This means that to achieve some target error γ, we require

$$L = \left\lceil \log_2 \left(\frac{\sqrt{2}\,\|H^{(1)}\|_{\max}}{\gamma} \right) \right\rceil \tag{5.16}$$

bits in the real and imaginary parts of each entry, so in terms of γ the number of terms in the decomposition is

$$4L = 4 \left\lceil \log_2 \left(\frac{\sqrt{2}\,\|H^{(1)}\|_{\max}}{\gamma} \right) \right\rceil, \tag{5.17}$$

as claimed.

Lastly, by construction the nonzero entries in $G^{\mathrm{Re},l,\pm}$ and $G^{\mathrm{Im},l,\pm}$ for each l appear in the same locations as the nonzero entries in $H^{(1)}$, so the O_L oracles for each $G^{\mathrm{Re},l,\pm}$ and $G^{\mathrm{Im},l,\pm}$ are the same as the O_L oracle for $H^{(1)}$. To compute the O_V oracle for a given $G^{\mathrm{Re},l,\pm}$, we first query the O_V oracle for the same entry in $H^{(1)}$, which returns the value $H^{(1)}_{ij}$ of that entry in an ancilla register. Then we can evaluate the corresponding entry in $G^{\mathrm{Re},l,\pm}$ using the formula in the second part of (5.12): this is a three-qubit operation in which a single qubit storing the sign of the entry may be flipped, controlled on both the qubit storing the sign $\mathrm{Re}[H^{(1)}_{ij}]$ and the qubit storing bit l of $\mathrm{Re}[H^{(1)}_{ij_i}]$. Finally, we uncompute $H^{(1)}_{ij}$ using a second query to the O_V oracle for $H^{(1)}$.

□

Before moving on with the section, it will be worthwhile to go through an example of the construction in the proof of Lemma 5.1. Suppose for simplicity that the input one-sparse Hamiltonian is scaled so that all entries have norm less than

1. First, split it into real and imaginary parts. Then for example if the real part is

$$C^{\mathrm{Re}} = \begin{pmatrix} 0 & 0 & 0.110 & 0 \\ 0 & -0.101 & 0 & 0 \\ 0.110 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.001 \end{pmatrix}, \tag{5.18}$$

its decomposition into one-sparse, self-inverse, Hermitian operators would be

$$\begin{pmatrix} 0 & 0 & 0.110 & 0 \\ 0 & -0.101 & 0 & 0 \\ 0.110 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.001 \end{pmatrix}$$

$$= \frac{1}{2} \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{C^{\mathrm{Re},1}} + \frac{1}{4} \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{C^{\mathrm{Re},2}} + \frac{1}{8} \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{C^{\mathrm{Re},3}}$$

$$= \frac{1}{4} \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{G^{\mathrm{Re},1,+}} + \frac{1}{4} \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}}_{G^{\mathrm{Re},1,-}} + \frac{1}{8} \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{G^{\mathrm{Re},2,+}} + \frac{1}{8} \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}}_{G^{\mathrm{Re},2,-}}$$

$$+ \frac{1}{16} \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{G^{\mathrm{Re},3,+}} + \frac{1}{16} \underbrace{\begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{G^{\mathrm{Re},3,-}}.$$

In words, the first step is to decompose into the matrices of first ($C^{\mathrm{Re},1}$), second ($C^{\mathrm{Re},2}$), and third bits ($C^{\mathrm{Re},3}$). This makes each row and column have norm either

zero or one. The second step is then to split each of the matrices of bits into two, so that each row and column has exactly one ± 1 in it. For example, $C^{\mathrm{Re},1}$ is replaced by $(G^{\mathrm{Re},1,+} + G^{\mathrm{Re},1,-})/2$, where $G^{\mathrm{Re},1,+}$ the entry in $C^{\mathrm{Re},1}$ that is nonzero in C^{Re} is replaced by $+1$ if it is zero in $C^{\mathrm{Re},1}$, and in $G^{\mathrm{Re},1,+}$ it is replaced by -1. This makes each row and column have norm one, as desired.

We also need to construct an oracle for any one of the terms $G^{\mathrm{Re},l,\pm}$ using oracle queries to C^{Re}. Let O_L and O_V denote the oracles for C^{Re}, and let $O_L^{\mathrm{Re},l,\pm}$ and $O_V^{\mathrm{Re},l,\pm}$ denote the oracles that we want to construct. As illustrated by the above example, $O_L^{\mathrm{Re},l,\pm} = O_L$ for all $l, \pm$ (i.e., the nonzero entries are in the same places in all of the $G^{\mathrm{Re},l,\pm}$ as in C^{Re}). For $O_V^{\mathrm{Re},l,\pm}$, we want to simulate its action on some pair of indices (x, y). For example, if $(x, y) = (0, 2)$ and we want to simulate the action of $O_V^{\mathrm{Re},2,+}$...

$$\begin{aligned} |x = 0, y = 2\rangle &\xrightarrow{O_V} |0, 2, +0.110\rangle \\ &\xrightarrow{\text{control on sign and 2nd bit after point}} |0, 2, +0.110, 1\rangle. \end{aligned} \tag{5.19}$$

The last qubit in the second line is the desired output of $O_V^{\mathrm{Re},2,+}$, so we would subsequently uncompute the action of O_V. Second example: if $(x, y) = (1, 1)$ and we want to simulate the action of $O_V^{\mathrm{Re},1,+}$...

$$\begin{aligned} |x = 1, y = 1\rangle &\xrightarrow{O_V} |0, 2, -0.101\rangle \\ &\xrightarrow{\text{control on sign and 1st bit after point}} |0, 2, -0.101, -1\rangle. \end{aligned} \tag{5.20}$$

Third example: if $(x, y) = (1, 1)$ and we want to simulate the action of $O_V^{\mathrm{Re},2,-}$...

$$\begin{aligned} |x = 1, y = 1\rangle &\xrightarrow{O_V} |0, 2, -0.101\rangle \\ &\xrightarrow{\text{control on sign and second bit after point}} |0, 2, -0.101, -1\rangle. \end{aligned} \tag{5.21}$$

Lemma 5.1 is constructive, so we can use the proof to decompose each Hermitian conjugate pair of ladder operator monomials into one-sparse, self-inverse terms. If N is the number of ladder operator monomials in the second-quantized fermionic

Hamiltonian, the number of conjugate pairs is at most $N/2$, so Lemma 5.1 provides a decomposition of H into a linear combination of at most

$$2N \left\lceil \log_2 \left(\frac{\sqrt{2}\,\|H\|_{\max}}{\gamma} \right) \right\rceil \tag{5.22}$$

one-sparse, self-inverse Hermitian terms, since the max-norm of each monomial is upper-bounded by $\|H\|_{\max}$.

Beyond the case of fermionic second-quantized Hamiltonians we consider an arbitrary d-sparse Hamiltonian that we only have oracle access to. This includes the case of second-quantized Hamiltonians with both fermionic and bosonic modes. In order to apply Lemma 5.1 we first decompose the Hamiltonian into one-sparse terms:

Lemma 5.2 ([BCC$^+$14], Lemma 4.4)**.** *If H is a d-sparse Hamiltonian, there exists a decomposition $H = \sum_{j=1}^{d^2} H_j$ where each H_j is Hermitian and one-sparse. An O_L query to any H_j can be simulated with two O_L queries to H, and an O_V query to any H_j can be simulated with one O_V query to H.*

The proof (in [BCC$^+$14]) is again constructive, so we can use Lemma 5.2 to obtain Hermitian one-sparse terms H_j, and then use Lemma 5.1 to approximately decompose each of these into Hermitian one-sparse, self-inverse terms. The resulting total number of one-sparse, self-inverse terms in the decomposition of H is at most

$$4d^2 \left\lceil \log_2 \left(\frac{\sqrt{2}\,\|H\|_{\max}}{\gamma} \right) \right\rceil . \tag{5.23}$$

Comparing (5.23) and (5.22), we see that in cases where either decomposition could be used, which one is preferable depends on half the number of ladder operator monomials ($N/2$) versus the squared sparsity (d^2). For example, the light-front Yukawa model studied in [KKG$^+$22] leads to a second-quantized Hamiltonian whose sparsity scales as $\Theta(N^{2/3})$. This is sublinear because each ladder operator monomial maps a large number of Fock states to zero in this model. However, even though the sparsity is asymptotically smaller than N, the squared sparsity is

$d^2 = \Theta(N^{4/3})$, so for this example it is still better to separately decompose each Hermitian conjugate pair of ladder operator monomials into one-sparse, self-inverse terms using Lemma 5.1.

We use the decomposition provided by Lemmas 5.1 and 5.2 because it results in terms that are one-sparse and unitary, and as we will see below, have entries ± 1 or $\pm i$; the cost is that the decomposition itself is approximate. However, the resulting terms can be implemented exactly using at most six oracle queries (see Section 5.4). Alternative decompositions exist that avoid approximations in the decompositions themselves (e.g., [CK11]), so it is possible that in future the method given above can be improved if the terms in such a decomposition can be implemented using few oracle queries.

5.4 Evolution under one-sparse unitary operators

The expectation value estimation method in Section 5.2 requires controlled applications of one-sparse, self-inverse, Hermitian operators; let G be such an operator. In practice, G will be one of the operators G_j obtained from Lemma 5.1. There is an extensive body of methods for simulating sparse Hamiltonians [ATS03, CCD$^+$03, BACS07, Chi10, CK10, CK11, BC12, CW12, BCC$^+$14, BCK15, BCC$^+$15, LC17, LW18, LC19, BCS$^+$20], any of which could be used to implement the controlled application of G. The fact that methods for simulation of time evolution generated by sparse Hamiltonians can also be used for simulation of sparse unitaries was first noted in [JW09]. However, because G is one-sparse and self-inverse, we can use a simpler method similar to the construction of the quantum walk operator in [BC12] (see the proof of Lemma 4 in [BC12]).

Using the oracles O_L and O_V for G, we can apply G as follows: let $|x\rangle_s$ be any input computational basis state, and let $|0\rangle_{a_1}|0\rangle_{a_2}$ be ancilla registers. The

steps to apply G are: first,

$$|x\rangle_s|0\rangle_{a_1}|0\rangle_{a_2} \xrightarrow{O_L} |x\rangle_s|y_x\rangle_{a_1}|0\rangle_{a_2} \tag{5.24}$$

$$\xrightarrow{O_V} |x\rangle_s|y_x\rangle_{a_1}|G_{xy_x}\rangle_{a_2}, \tag{5.25}$$

where y_x is the column-index of the single nonzero entry in row x of G, as in (5.8). From the proof of Lemma 5.1 above, it follows that $G_{xy_x} = \pm 1 \ \forall x, y_x$ or $G_{xy_x} = \pm i \ \forall x, y_x$. Whether the entries are ± 1 or $\pm i$ is determined by j (where $G = G_j$ for some G_j resulting from Lemma 5.1), which is evaluated in classical preprocessing. Therefore, $|G_{xy_x}\rangle_{a_2}$ need only be a single qubit determining the sign, and as our next step we can apply the entry G_{xy_x} exactly as a phase controlled by $|G_{xy_x}\rangle_{a_2}$, and then complete the implementation of G as follows:

$$\xrightarrow{\text{controlled phase}} G_{xy_x}|x\rangle_s|y_x\rangle_{a_1}|G_{xy_x}\rangle_{a_2} \tag{5.26}$$

$$\xrightarrow{O_V^{-1}} G_{xy_x}|x\rangle_s|y_x\rangle_{a_1}|0\rangle_{a_2} \tag{5.27}$$

$$\xrightarrow{\text{swap } s, a_1} G_{xy_x}|y_x\rangle_s|x\rangle_{a_1}|0\rangle_{a_2} \tag{5.28}$$

$$\xrightarrow{O_L^{-1}} G_{xy_x}|y_x\rangle_s|0\rangle_{a_1}|0\rangle_{a_2}, \tag{5.29}$$

where the last step follows because for a one-sparse, Hermitian operator, (5.8) implies

$$O_L|y_x, 0\rangle = |y_x, x\rangle. \tag{5.30}$$

The effect of these operations is to map

$$|x\rangle_s \ \rightarrow \ G_{xy_x}|y_x\rangle_s = G|x\rangle_s, \tag{5.31}$$

i.e., we have applied G to $|x\rangle_s$. This required four queries to the oracles: one query each to O_L, O_V, and their inverses. For ancillas, we required copying the computational register s in the register a_1 to apply the O_L oracle, and one additional qubit in the register a_2 to represent the sign of $G_{xy_x} = \pm 1, \pm i$. In both queries and

ancillas, these are the minimum requirements to apply the oracle unitaries at all.

Finally, recall that the one-sparse, self-inverse terms that we are estimating expectation values of were obtained via Lemma 5.1, above. However, the full Hamiltonian is first decomposed into one-sparse terms $H^{(1)}$ — either conjugate pairs of ladder-operator monomials or via Lemma 5.2 — which form the inputs to Lemma 5.1. From Lemma 5.1 we know that the O_L oracle for any of the G_j is identical to the O_L oracle for $H^{(1)}$. Also, from the proof of Lemma 5.1, we know that the O_V oracle for any of the G_j can be implemented by first applying the O_V oracle for G_j and then performing a single controlled operation (we would later undo both of these steps to apply O_V^{-1}). Hence the number of queries to each $H^{(1)}$ is still four, each of which will either be implemented directly (in the ladder operator monomial decomposition), or via one (for O_V) or two (for O_L) queries to the full Hamiltonian oracles using Lemma 5.2. This gives a total of at most six oracle queries.

5.5 Conclusion

In prior work, the only Hamiltonian input model used in VQE has been decomposition into Pauli operators. In this chapter we have extended VQE to the case of sparse Hamiltonians. We accomplished this by employing a variant of techniques previously considered applicable to future fault-tolerant quantum computers [BCC$^+$14]. For sparse Hamiltonians, we have demonstrated how VQE can be implemented via a decomposition into one-sparse, self-inverse Hermitian terms. As discussed in Section 5.1, simulation of second-quantized Hamiltonians in condensed matter, high energy and nuclear physics, and in compact representations of quantum chemistry are natural candidates for this sparse VQE method [AGDLHG05, BBS$^+$17, KKG$^+$22, KJK$^+$21a, KJK$^+$21b, KHKL21].

This chapter focused on VQE, but the results may also be used in the context of QAOA [FGG14]. QAOA to date treats classical objective functions that are sums of local clauses, of which 3-SAT and MAXCUT are canonical NP-complete examples. Classical objective operators are diagonal in the computational basis and

hence naturally one-sparse. The techniques here would allow extension to the case where the diagonal entries are given by more complicated classical functions. This broadens the space of examples within which to search for quantum advantage and also may provide practical advantages for problems with large locality such as the travelling salesman problem. We leave the investigation of these ideas to future work.

5.6 Appendix: oracle implementations for example applications

In this appendix, we give explicit implementations for the oracles used in sparse variational quantum eigensolvers (VQEs) for the special cases of Pauli operators and bosonic creation and annihilation operators. As described above, a Hamiltonian H can be simulated using sparse VQE when it is sparse and admits description by two oracles O_L and O_V as defined in (5.2) and (5.3), which respectively give the locations and values of the nonzero entries in H.

5.6.1 Oracles for Pauli operators

We first prove a result characterizing the locations and values of the nonzero entries in arbitrary Pauli operators, then use this to construct oracles for the Pauli operators.

Lemma 5.3. *For any Pauli operator $P = P_1 \otimes P_2 \otimes \cdots \otimes P_n$ (for single-qubit Pauli factors P_k), let a and b be length-n binary vectors defined bitwise as follows:*

$$a_k \coloneqq \begin{cases} 0 & \text{if } P_k = I \text{ or } Z, \\ 1 & \text{if } P_k = X \text{ or } Y, \end{cases} \tag{5.32}$$

and

$$b_k = \begin{cases} 0 & \text{if } P_k = I \text{ or } X, \\ 1 & \text{if } P_k = Y \text{ or } Z. \end{cases} \tag{5.33}$$

Equivalently,

$$P_k = i^{a_k b_k} X^{a_k} Z^{b_k}. \tag{5.34}$$

Then for any row index expressed as a binary number x, the location y of the nonzero entry in row x of P is

$$y = x \oplus a, \tag{5.35}$$

where $\oplus$ denotes bitwise XOR. The value of the nonzero entry in row x of P is

$$p_x = (-1)^{b \cdot x} (-i)^{a \cdot b}, , \tag{5.36}$$

where $\cdot$ is the dot product in $\mathbb{Z}_2^n$, e.g.,

$$b \cdot x = \sum_{k=1}^{n} b_k x_k. \tag{5.37}$$

Note that the binary vectors a and b are the symplectic binary representation of the Pauli operators as in [Got97, Got98].

Proof. We first prove (5.35). For any computational basis state $|i\rangle$,

$$\begin{aligned} P|i\rangle &= (P_1 \otimes P_2 \otimes \cdots \otimes P_n)(|i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle) \\ &= P_1|i_1\rangle \otimes P_2|i_2\rangle \otimes \cdots \otimes P_n|i_n\rangle \\ &= s|j_1\rangle \otimes |j_2\rangle \otimes \cdots \otimes |j_n\rangle = |j\rangle \end{aligned} \tag{5.38}$$

for some $s \in \{\pm 1, \pm i\}$. In other words, each bit y_k in y is equal to the corresponding bit x_k in x, acted upon by the corresponding single-qubit Pauli operator P_k in the tensor factorization of P. If P_k is X or Y, then the bit is flipped, i.e., $y_k = x_k \oplus 1$; otherwise, $y_k = x_k$. Hence,

$$y_k = x_k \oplus a_k, \tag{5.39}$$

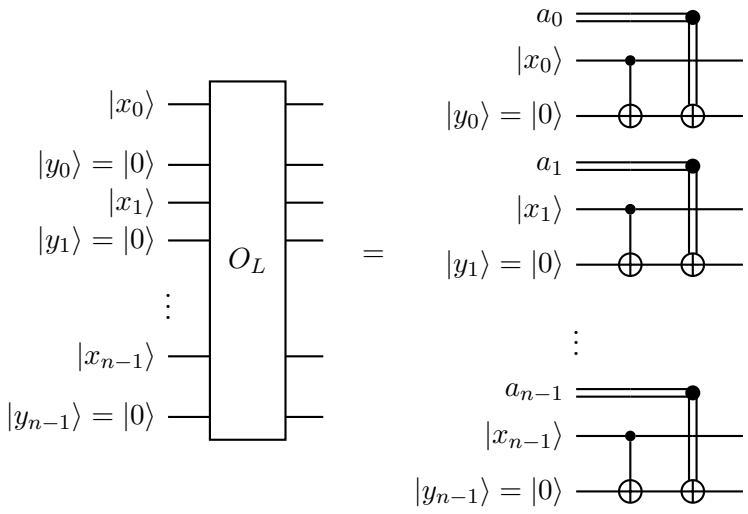


Figure 5.2: Circuit to implement the O_L oracle as given by (5.46), for the Pauli $P(a, b)$ and input $|x, y = 0\rangle$. x_k is the kth bit in x, y_k is the kth bit in y, and a_k is the kth bit in a.

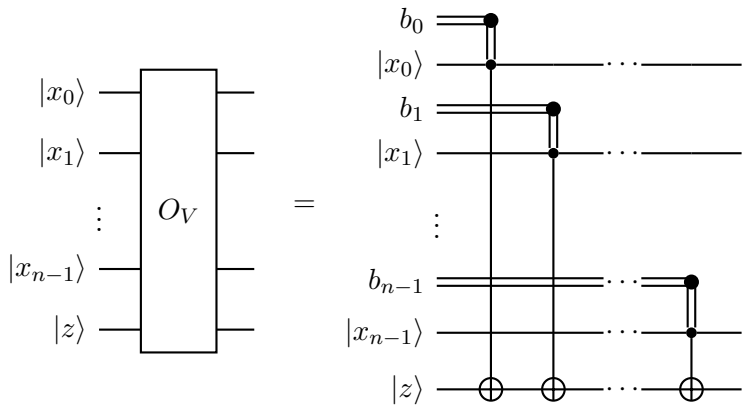


Figure 5.3: Circuit to implement the O_V oracle as given by (5.49), for the Pauli $P(a, b)$ and input $|x\rangle|z\rangle$. x_k is the kth bit in x, b_k is the kth bit in b, and $|z\rangle$ is a single qubit. We do not show the register encoding $|x \oplus a\rangle$ as in (5.49) because the action of the oracle does not depend on it.

where $a_k = 1$ if P_k is X or Y and $a_k = 0$ otherwise. But this means that when we combine the bits a_k together into the binary vector a, we have exactly the definition in (5.32), and our desired result (5.36) simply summarizes the bitwise relations (5.39) for all k.

We now prove (5.36). First, note that P can be expressed as

$$P = s(P_1^{(Z)} \otimes P_2^{(Z)} \otimes \cdots \otimes P_n^{(Z)})(P_1^{(X)} \otimes P_2^{(X)} \otimes \cdots \otimes P_n^{(X)}) \tag{5.40}$$

for some $s \in \{\pm 1, \pm i\}$, where each single-qubit Pauli matrix $P_k^{(X)}$ is either I or X, each single qubit Pauli matrix $P_k^{(Z)}$ is either I or Z. The value of s is determined by the fact that $ZX = iY$, so for each k such that $P_k^{(X)} = X$ and $P_k^{(Z)} = Z$ we pick up a factor of i that needs to be cancelled on the right-hand side of (5.40). The number of these factors is therefore $a \cdot b$, so $s = (-i)^{a \cdot b}$ to cancel them.

The action of P on a computational basis state $|x\rangle$ is

$$\begin{aligned} P|x\rangle &= (-i)^{a \cdot b}(P_1^{(Z)} \otimes P_2^{(Z)} \otimes \cdots \otimes P_n^{(Z)})|y\rangle \\ &= (-i)^{a \cdot b} P_1^{(Z)}|y_1\rangle \otimes P_2^{(Z)}|y_2\rangle \otimes \cdots \otimes P_n^{(Z)}|y_n\rangle. \end{aligned} \tag{5.41}$$

In this expression, the tensor factor for the kth qubit is

$$\begin{aligned} P_k^{(Z)}|y_k\rangle &= \begin{cases} |y_k\rangle & \text{if } P_k^{(Z)} = I, \\ (-1)^{y_k}|y_k\rangle & \text{if } P_k^{(Z)} = Z \end{cases} \\ &= \begin{cases} |y_k\rangle & \text{if } b_k = 0, \\ (-1)^{y_k}|y_k\rangle & \text{if } b_k = 1 \end{cases} \\ &= (-1)^{b_k y_k}|j_k\rangle. \end{aligned} \tag{5.42}$$

We can plug this back into (5.41) to obtain

$$P|x\rangle = s \prod_{k=1}^{n} (-1)^{b_k y_k}|y\rangle = (-1)^{b \cdot y}(-i)^{a \cdot b}|y\rangle. \tag{5.43}$$

Hence

$$(-1)^{b\cdot y}(-i)^{a\cdot b} = (-1)^{b\cdot(a\oplus x)}(-i)^{a\cdot b} \tag{5.44}$$

is the nonzero value in the xth column of P. We want the value in the xth row, but by (5.35) transposing P simply corresponds to mapping x to $x \oplus a$, so the nonzero value in the xth row of P is

$$(-1)^{b\cdot(x\oplus a\oplus a)}(-i)^{a\cdot b} = (-1)^{b\cdot x}(-i)^{a\cdot b}, \tag{5.45}$$

as desired.

□

Using Lemma 5.3, the oracles for a Pauli operator $P(a, b)$ are defined as follows:

$$O_L|x\rangle = |x, x \oplus a\rangle, \tag{5.46}$$

$$O_V|x, y\rangle|z\rangle = |x, y\rangle|z \oplus H_{xy}\rangle. \tag{5.47}$$

Note that there is no argument i in (5.46), unlike (5.2): this is because there is only one nonzero entry per row in a Pauli operator, so there is no need to index the nonzero entries. The binary vectors a and b are defined in terms of $P(a, b)$ as in (5.32) and (5.33), respectively, and

$$H_{xy} = \begin{cases} (-1)^{b\cdot x}(-i)^{a\cdot b} & \text{if } y = x \oplus a, \\ 0 & \text{otherwise.} \end{cases} \tag{5.48}$$

Recall that for sparse VQE, one only needs to evaluate O_V for states that are the output of O_L. Also, the factor of $(-i)^{a\cdot b}$ in the first case above is an overall phase that we can compute classically, so to encode the matrix element it is enough to compute $(-1)^{b\cdot x}$ on the quantum computer. Hence, in fact we will only ever need to compute

$$O_V|x, x \oplus a\rangle|z\rangle = |x, x \oplus a\rangle \left|z \oplus (-1)^{b\cdot x}\right\rangle, \tag{5.49}$$

where $|z\rangle$ is a single qubit that encodes ± 1 as

$$\begin{aligned} +1 &\mapsto |0\rangle, \\ -1 &\mapsto |1\rangle. \end{aligned} \tag{5.50}$$

O_L as given by (5.46) is implemented via a single layer of CNOTs and a single layer of classically controlled NOTs, as shown in Fig. 5.2. O_V as given by (5.49) is implemented via a sequence of CNOTs (which are also classically controlled), as shown in Fig. 5.3.

5.6.2 Oracles for Bosonic operators

As our second example of oracle implementations, we consider a Hamiltonian that acts on M bosonic modes, each of which is encoded in N qubits. For simplicity, assume that the cutoff Λ on the occupation of each mode is a power of two, so $N = \log_2 \Lambda$. Assume that the bosonic Hamiltonian is written as a polynomial in the creation and annihilation operators $a_p^\dagger$ and a_p, where p indexes the mode acted upon. We want to give oracles for each monomial in the Hamiltonian, since in that case we can use sparse VQE to estimate their expectation values, and hence the expectation value of the Hamiltonian.

These oracles will turn out to involve only binary arithmetic operations on the mode occupation encodings. Reversible arithmetic is an important subroutine in many quantum algorithms, so various compilers have been designed specifically for this purpose, see for example [JPK+14]. However, for completeness we will give details below of how to implement the specific operations required for the present case.

First consider the location oracle O_L. A monomial of creation and annihilation operators is one-sparse in the Fock (occupation number) basis, so given any input Fock state (encoded in qubits as above), we just need to identify the single outgoing Fock state under the action of the monomial. Assuming the monomial is normal ordered, we first have a sequence of annihilation operators, then a sequence

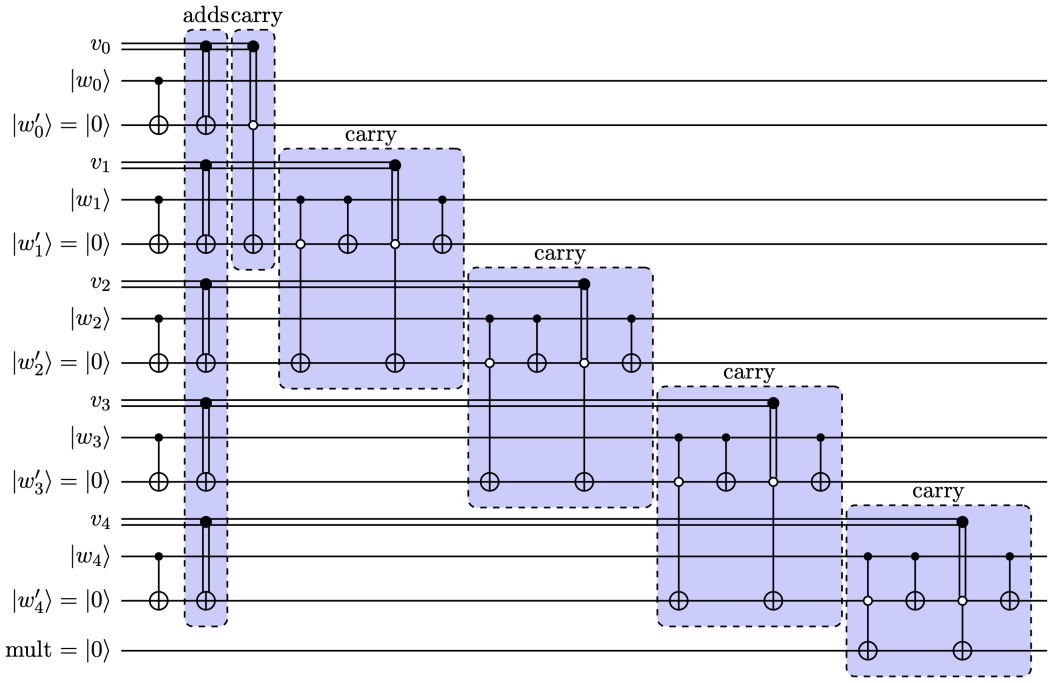


Figure 5.4: Circuit to implement $|w\rangle|w' = 0\rangle \mapsto |w\rangle|w' = w \oplus v\rangle$, for a classical binary number v and a binary number w stored in qubits. w represents the occupation of a mode in the input state x; w_{N-1} is the most significant bit and w_0 is the least. w' represents the occupation of the corresponding mode in the output state, which is initially set to $|0\rangle$. The circuit shown is for five bits in w and w'; to generalize to N bits, one should extend the 'adds' column and continue to cascade the 'carry' blocks as in the example above. If there is a carry from the most significant bit, it means that w' has reached or exceeded the occupation cutoff. Therefore, we flip the 'mult' qubit from $|1\rangle$ to $|0\rangle$ if there is a carry from the most significant bit: the matrix element will later be multiplied by the value of the 'mult' qubit, so this sets the matrix element to zero when the occupation reaches the cutoff.

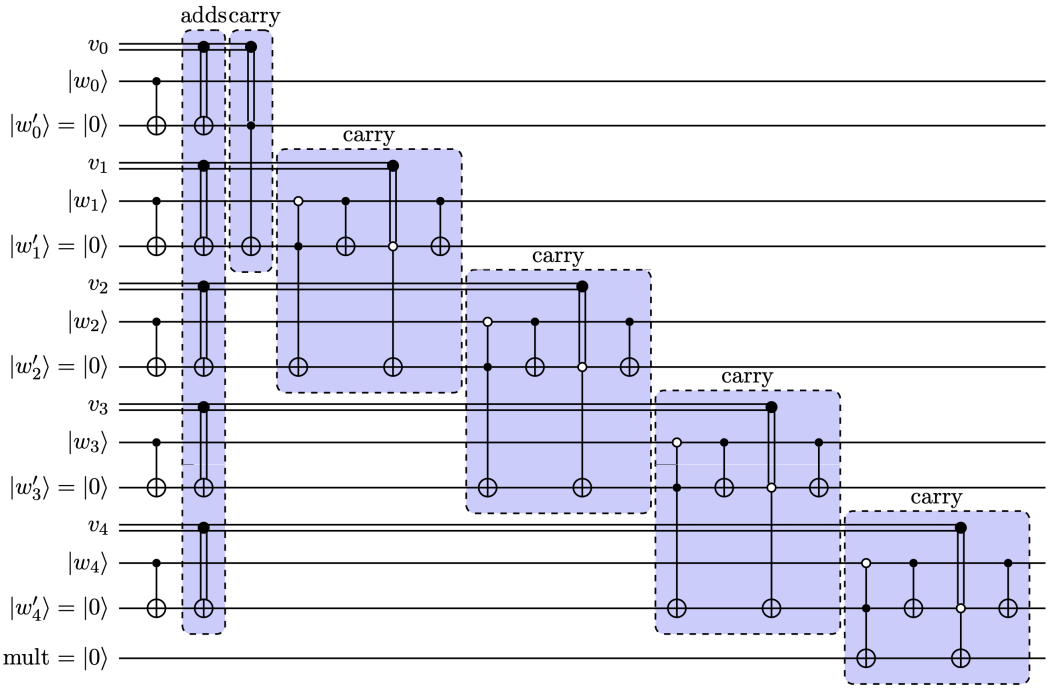


Figure 5.5: Circuit to implement $|w\rangle|w' = 0\rangle \mapsto |w\rangle|w' = w \ominus v\rangle$, for a classical binary number v and a binary number w stored in qubits. w represents the occupation of a mode in the input state x; w_{N-1} is the most significant bit and w_0 is the least. w' represents the occupation of the corresponding mode in the output state, which is initially set to $|0\rangle$. The circuit shown is for five bits in w and w'; to generalize to N bits, one should extend the 'adds' column and continue to cascade the 'carry' blocks as in the example above. If there is a carry from the most significant bit, it means that w' has reached or exceeded the occupation cutoff. Therefore, we flip the 'mult' qubit from $|1\rangle$ to $|0\rangle$ if there is a carry from the most significant bit: the matrix element will later be multiplied by the value of the 'mult' qubit, so this sets the matrix element to zero when the occupation reaches the cutoff.

of creation operators. For any one of these operators, the mode it acts upon is specified classically, so we just need to either decrement (for annihilation operators) or increment (for creation operators) the occupancy w of the mode. The occupancies of the modes are encoded as binary numbers, so these operations are just binary addition or subtraction of the number of copies of the operator acting on the mode. For a given monomial, we determine classically what the corresponding additions or subtractions for each mode are. On the quantum computer, then, we just need to implement these classically-controlled circuits. Circuits to implement these operations are given in Fig. 5.4 and 5.5.

We now describe the action of the increment circuit Fig. 5.4 on computational basis states: the action on generic states is obtained by linearity. The first column of CNOTs copies (in the computational basis) the input occupation $|w\rangle$ to the register $|w'\rangle$ that will end up storing the output occupation. The column labeled 'adds' then adds each bit of the classical number v to the corresponding bit in w'. The first 'carry' block then adds the bit carried from the zeroth place to w'_1: the carried bit from the zeroth place is 1 if and only if $v_0 = 1$ and $w'_0 = 0$ (in which case w_0 must be 1). Subsequent 'carry' blocks add the bit carried from each place to the next more significant place. Within each 'carry' block (after the first), there are two components. For a given place k, if $w_k = 1$ and $w'_k = 0$, then either $v_k = 1$ or there was a 1 carried from the previous place (but not both), and either way there is a 1 carried to the next place: this is the first multiply-controlled-NOT in the 'carry' block. If $w'_k = w_k$ and $v_k = 1$, then there must have been a 1 carried from the previous place, so there is a 1 carried to the next place: the remaining three gates in the 'carry' block implement this operation. However, if $w'_k = w_k$ and $v_k = 0$, then there cannot have been a 1 carried from the previous place, so there is no 1 carried to the next place. Similarly, if $w_k = 0$ and $w'_k = 1$, then either $v_k = 1$ or there was a 1 carried from the previous place (but not both), and either way there is no 1 carried to the next place. The multiply-controlled-NOTs in each 'carry' block are mutually exclusive, so at most one bit is carried to the next place. The logic in the decrement circuit Fig. 5.5 is similar.

From Fig. 5.4 and 5.5 we can see that each circuit for N bit occupations requires N NOTs (classically controlled), $4N-2$ CNOTs (including both control conditions), and $N-1$ Toffoli gates (controlled on $|0\rangle$ in the first control and $|1\rangle$ in the second control). One such circuit will be required for each distinct mode acted upon by the creation and annihilation operators in the Hamiltonian term (monomial), so if there are f creation and annihilation operators in the term then implementing O_L requires at most

$$(6N-3)f \tag{5.51}$$

gates and f ancilla qubits (the 'multiplier' qubits).

As discussed above, all that is required to implement the O_L oracle for a creation/annihilation operator monomial is a sequence of the occupation increment and decrement circuits given in Fig. 5.4 and 5.5. For the O_V oracle for the monomial, we have to calculate the value of the matrix element between some input state Fock state encoded in qubits $|x\rangle$ and its image $|y\rangle$ under the action of the oracle, which we will have already computed using O_L. The action of a single creation or annihilation operator acting on a bosonic mode with occupation w is given by

$$a^\dagger|w\rangle = \sqrt{w+1}|w+1\rangle, \quad a|w\rangle = \sqrt{w}|w-1\rangle, \tag{5.52}$$

assuming $w+1$ is still below the occupation number cutoff. Hence, the matrix element of a product of creation and annihilation operators is

$$\sqrt{\left(w^{(y)}_{n_{\text{out}}} w^{(y)}_{n_{\text{out}}-1} \cdots w^{(y)}_2 w^{(y)}_1\right)\left(w^{(x)}_{n_{\text{in}}} w^{(x)}_{n_{\text{in}}-1} \cdots w^{(x)}_2 w^{(x)}_1\right)}, \tag{5.53}$$

where the $w_i^{(y)}$ are the occupations of the modes in the output state $|y\rangle$ whose occupations are incremented by the monomial, and the $w_i^{(x)}$ are the occupations of the modes in the input state $|x\rangle$ prior whose occupations are decremented by the monomial. When all of the occupation number changes for single modes are plus one or minus one, the variables $w_i^{(x)}$ and $w_i^{(y)}$ in (5.53) can be read directly from

$|x\rangle$ and $|y\rangle$. When multiple particles are created in the same mode, the occupations appearing in the product should be those *immediately after* the corresponding creation operator is applied, and when multiple particles are annihilated in the same mode, the occupations appearing in the product should be those *immediately before* the corresponding annihilation operator is applied. Thus in either of these cases, additional addition or subtraction circuits as in Fig. 5.4 and 5.5 will be required to compute the variables in (5.53). If any of the 'multiplier' qubits computed by O_L are $|0\rangle$, then the whole matrix element should be 0.

To compute the value of the matrix element given by (5.53) once the variables have been determined, we require $n_{\text{in}}+n_{\text{out}}-1$ multiplications followed by a square-root operation. Reversible circuits for multiplication are given in e.g. [PRM18], the simplest of which requires $4W^2-3W$ Toffoli gates and linear ancilla space, where W is the number of qubits used to store the value of the matrix element in (5.53) prior to taking the square-root. Hence to implement all $n_{\text{in}}+n_{\text{out}}-1$ of the multiplications requires

$$(4W^2-3W)(n_{\text{in}}+n_{\text{out}}-1) \tag{5.54}$$

Toffoli gates and

$$W\big(n_{\text{in}}+n_{\text{out}}+O(1)\big) \tag{5.55}$$

ancilla qubits, because the multiplications are not in place and thus the partial products are stored at each step. Note that since we only use the matrix element to compute the phase to be applied to the state, as described in Section 5.4, the fact that the ancillas are not uncomputed during the multiplications is no problem because the entire matrix element will be uncomputed after controlling the phase. A reversible circuit for square-root is given in [SR11], requiring $\frac{5}{2}W^2+7W-2$ elementary gates and $\frac{1}{4}W^2+3W-2$ ancilla qubits. Hence the total gate cost of the matrix element oracle O_V is

$$O\big(fW^2\big) \tag{5.56}$$

and the number of ancilla qubits required is

$$O\big(fW + W^2\big), \tag{5.57}$$

where W is the number of bits in the matrix element and f is the number of creation and annihilation operators in the Hamiltonian term.

Chapter 6

Oracles for sparse Hamiltonians in compact mapping

6.1 Introduction

In this chapter we describe a framework for simulating second-quantized Hamiltonians on quantum computers. Hamiltonians in second-quantization are ubiquitous in quantum chemistry, many-body physics, and quantum field theory, all of which are target applications for quantum simulation. Fermionic Hamiltonians with fixed particle number admit simple encodings in the Pauli basis [JW28, BK02, SRL12] as discussed in Section 1.3.1, and these have been the focus of many quantum simulation experiments to date [AGDLHG05, LWG$^+$10, WBAG11, KWPO$^+$11, JWM$^+$12, YCM$^+$14, PMS$^+$14, WDB$^+$15, OBK$^+$16, SZZ$^+$17, PGS$^+$17, KMT$^+$17, MKSCdJ17, HMR$^+$18, SWG$^+$18, CRD$^+$18, BWM$^+$18, KTC$^+$19, PM19, NCP$^+$20, AAB$^+$20, MEAG$^+$20, MST$^+$20, HLB$^+$20, SHE20, TRJ$^+$20, SBW$^+$21, YHM$^+$22, SY21, CMP21, CG22, BHS$^+$22, TZC$^+$22]. However, second-quantized Hamiltonians are *sparse* — they have only polynomially-many nonzero entries per row or column — as long as they have polynomially-many terms. This makes them appropriate for simulation using methods developed for sparse Hamiltonians (see Section 1.3.2 and [ATS03, CCD$^+$03, BACS07, Chi10, CK10, CK11, BC12, CW12,

BCC$^+$14, BCK15, BCC$^+$15, LC17, LW18, LC19, BCS$^+$20]).

The second-quantized Hamiltonians we consider are given as polynomials in ladder operators acting on occupation number states (*Fock states*). The main idea is to extend the *compact encoding* previously studied in [AGDLHG05, TL13, KKG$^+$22], which only stores information about occupied modes in a given Fock state. The application of sparse simulation techniques to electronic structure Hamiltonians was previously studied in [BBK$^+$16, BBS$^+$17], and these papers use a special case of the compact encoding (which they call "compressed representation"). In Section 6.2.2, we will compare the overall cost of simulation using our algorithm to those of [BBK$^+$16, BBS$^+$17].

The compact encoding is to be contrasted with *direct encodings*, which store information about all physical modes, whether they are occupied or not. The Jordan-Wigner and Bravyi-Kitaev encodings commonly used in quantum algorithms for quantum chemistry are examples of direct encodings (see Section 1.3.1 and [JW28, BK02, SRL12]). Compact encodings are suitable for Hamiltonians that are sparse in the occupation number basis. In a sparse Hamiltonian, the number of nonzero elements in each row or column scales polynomially with the problem size, and therefore polylogarithmically with Hamiltonian dimension. The compact encoding permits efficient sparsity-based state preparation and time evolution methods [AGDLHG05, TL13, KKG$^+$22].

The methods we develop in this chapter are motivated by simulation of quantum field theory. In particular, we will focus on the case of Hamiltonians expressed in the plane wave momentum basis as our main example, since it illustrates the key techniques of our method. We use the fact that such Hamiltonians can be expressed as sums of *interactions*, where an interaction is a sum of ladder operator monomials that only differ in their momentum quantum numbers. The sum within each interaction runs over all assignments of momenta that conserve the total momentum (see Section 6.3.3 for details).

Although in Sections 6.2-6.7 we will define multi-particle states using the plane wave momentum basis, our method extends straightforwardly to Hamilto-

nians where the sums within interactions run over quantum numbers other than plane wave momenta, as long as the number of distinct interactions in the Hamiltonian is polynomial in the system parameters (such as momentum cutoffs). This will always be true when the Hamiltonian has only a polynomial number of second-quantized terms, since the number of interactions is upper bounded by the number of terms. These cases include a wide range of theories in quantum chemistry, condensed matter physics, and quantum field theory, including basis light-front quantization [VHL$^+$10, KJK$^+$21a, KJK$^+$21b]. How to extend our methods beyond the plane wave momentum basis is explained in Section 6.8. This chapter is based on [KHKL21].[1]

6.2 Main results for plane wave momentum basis

Algorithms for simulating general sparse Hamiltonians access the Hamiltonian via oracle unitaries that are queried (applied) to provide the locations and values of the nonzero Hamiltonian matrix elements (see (1.21) and (1.22) for the definitions of the oracles, and for more general background see Section 1.3.1 and [ATS03, CCD$^+$03, BACS07, Chi10, CK10, CK11, BC12, CW12, BCC$^+$14, BCK15, BCC$^+$15, LC17, LW18, LC19, BCS$^+$20]). If we want to apply such algorithms to a second-quantized Hamiltonian in compact encoding, then we have to provide two main additional components. First, we need to explicitly construct the oracle unitaries for the specific Hamiltonian of interest, as sequences of primitive gates. We will show how to do this in Sections 6.5 and 6.6, decomposing the oracle unitaries into qubit operations that are log-local in the problem parameters, which for us will be momentum cutoffs, since we focus on the example of the plane wave momentum basis. The log-local operations can then themselves be decomposed into primitive gates from any desired gate set with only polynomial overhead.

Second, the general sparse Hamiltonian methods assume that the oracles act directly upon row and column indices (encoded in qubit states) of the Hamil-

[1] ©2021 American Physical Society, reproduced per Author's Rights under the transfer of copyright agreement (`https://journals.aps.org/authors/transfer-of-copyright-agreement`).

tonian [ATS03, CCD$^+$03, BACS07, Chi10, CK10, CK11, BC12, CW12, BCC$^+$14, BCK15, BCC$^+$15, LC17, LW18, LC19, BCS$^+$20]. We instead want methods that act directly upon compact-encoded Fock states, because the physical meaning of such states can be directly read out, which ultimately permits efficient implementation of observables. However, unlike simply labeling the rows and columns of the Hamiltonian by sequential binary numbers, the set of bitstrings corresponding to compact-encoded Fock states is not simple to characterize or enumerate. These bitstrings label computational basis states that span the subspace of qubit Hilbert space that the Hamiltonian acts on. Therefore, we need to show that when we implement oracle unitaries that act directly on compact-encoded Fock states, the high-level simulation algorithms [ATS03, CCD$^+$03, BACS07, Chi10, CK10, CK11, BC12, CW12, BCC$^+$14, BCK15, BCC$^+$15, LC17, LW18, LC19, BCS$^+$20]that use the oracles as their building blocks will still work. This is explained in Section 6.4.

The overall asymptotic costs of our methods in both qubit and gate counts are summarized in Table 6.2, for the example of the plane wave momentum basis. The details of the costs are as follows. The number of qubits required to encode a Fock state in compact encoding is derived in Section 6.3.1, resulting in the expression in (6.22), which is asymptotically

$$\mathsf{Q} = O\left(I \log W + I \sum_{j=1}^{d} \log\left(\Lambda_j^{\max} - \Lambda_j^{\min}\right) \right), \tag{6.1}$$

where I is the maximum possible number of occupied modes in a Fock state, W is the maximum possible occupation of any mode, d is the number of spatial dimensions, and $\Lambda_j^{\min}$ and $\Lambda_j^{\max}$ are lower and upper momentum cutoffs in each dimension j. Hence fixing $|\Lambda_j^{\max}|, |\Lambda_j^{\min}| \leq \Lambda$ for some overall cutoff Λ results in the scaling given in Table 6.2. The expression in (6.1) assumes that the number of qubits required to encode the non-momentum quantum numbers is constant.

In our implementations the cost in log-local gates of the location oracle O_L (the oracle that gives the locations of nonzero matrix elements, defined in (1.21)) asymptotically dominates the cost of the matrix element value oracle O_V (defined

in (1.22)). These costs are derived in Section 6.5 and Section 6.6, respectively, and result in the expressions (6.84) and (6.93). The dominant cost is the former, which is

$$O\Big(I^h + \Lambda^{dg}\Big) \tag{6.2}$$

exactly as in Table 6.2, where h is the maximum number of annihilation operators in any interaction in the Hamiltonian and g is the maximum number of creation operators in any interaction in the Hamiltonian.

Finally, if our Hamiltonian is time-independent, then by using qubitization [LC19] the total number of oracle queries required to simulate time evolution is

$$O\left(\tau + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right), \tag{6.3}$$

where $\tau = k\|H\|_{\max}t$, k is the sparsity of the Hamiltonian H, t is the total evolution time, and ϵ is the error. Multiplying by the oracle cost (6.2) gives the overall asymptotic scaling of the number of log-local gates:

$$O\left[\left(\tau + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right)\left(I^h + \Lambda^{dg}\right)\right]. \tag{6.4}$$

If instead our Hamiltonian is time-dependent, then by using the method of [BCS+20] the total number of oracle queries required to simulate time evolution is

$$O\left(\tau\frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\right), \tag{6.5}$$

where now $\tau \coloneqq k\int_0^t \|H\|_{\max}dt$ (without loss of generality taking the starting time to be $t = 0$). Hence the overall log-local gate count for our algorithm is

$$O\left[\left(\tau\frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\right)\left(I^h + \Lambda^{dg}\right)\right]. \tag{6.6}$$

Suppressing the logarithmic components in either (6.4) or (6.6) gives the expression in Table 6.2.

Parameters:	
number of spatial dimensions	d
max number of occupied modes	I
max occupancy of a single mode	W
momentum cutoff	Λ
max incoming lines in any interaction	h
max outgoing lines in any interaction	g
sparsity	k
max-norm of Hamiltonian	$\|H\|_{\max}$
simulation time	t

Table 6.1: Glossary of parameter definitions.

Costs:	
qubits to encode Fock state	$O\Big(I \log\big(W\Lambda^d\big)\Big)$
log-local operations for oracle	$O\Big(I^h + \Lambda^{dg}\Big)$
total log-local operations	$\widetilde{O}\Big(k\|H\|_{\max} t\big(I^h + \Lambda^{dg}\big)\Big)$

Table 6.2: Summary of qubit and gate count costs for our algorithm, for a second-quantized Hamiltonian H in the plane wave momentum basis. The parameters are defined in Table 6.1.

6.2.1 Comparison to direct encoding

Recall that the goal of the compact encoding is to minimize the number of qubits required to simulate a second-quantized Hamiltonian. Direct encodings, which explicitly store information about every mode including the unoccupied modes, will require more qubits but afford simpler operations, as discussed in the introduction.

In direct encoding we store the occupation of every mode in a Fock state. Hence each mode can be assigned to a specific register of qubits, so it is not necessary to store the information identifying the mode in the qubit state. Therefore, the number of qubits required for a single mode is just $O(\log W)$, where as above W is the single-mode occupation cutoff. This is multiplied by the number of modes to give the total number of qubits required for the direct encoding:

$$O\left((\log W)\prod_{j=1}^{d}\left(\Lambda_j^{\max}-\Lambda_j^{\min}\right)\right), \tag{6.7}$$

where the number of modes is $O\left(\prod_{j=1}^{d}\left(\Lambda_j^{\max}-\Lambda_j^{\min}\right)\right)$ assuming the numbers of

species and non-momentum quantum numbers are constant. In other words, compared to the number (6.1) of qubits for the compact encoding, the direct encoding has linear rather than logarithmic scaling with the number of modes. Hence, when the maximum number I of distinct occupied modes is much smaller than the total number of modes, the compact encoding will be asymptotically advantageous in number of qubits.

The costs of oracle implementations for the direct encoding were evaluated in Section I.B of the Supplemental Material to [KL21]. As with the compact encoding, the cost of the location oracle dominates, coming out to

$$O\left((h+g)\log W\right) \tag{6.8}$$

Toffoli gates (using our notation). However, this is just the cost of the location oracle for a single creation or annihilation operator, so for a fair comparison to the cost of compact encoding, we should get the cost of the location oracle for a whole interaction in direct encoding. To get this, index the nonzero matrix elements in the same way as we index the second-quantized terms: as an ordered set of mode indices specifying the modes acted on by the term. Then the location oracle will end up just being creations or annihilations on single modes (each of which has the cost given in (8)) controlled on the corresponding mode indices. Hence, we only need to run over all possible indices for each creation and annihilation operator in the term independently, so the multiplicative overhead will end up being linear in the total number Λ^d of modes. This yields

$$O\left(\Lambda^d(h+g)\log W\right) \tag{6.9}$$

as the cost of the location oracle in direct encoding. As expected, in typical cases this will be smaller than the cost (6.2) of the location oracle in compact encoding, since this scales as Λ^{dg}.

These comparisons confirm the expected relation between direct and com-

pact encodings: they form a space-time tradeoff, with the direct encoding using more space to obtain shorter circuits, and the compact encoding saving space at the expense of longer circuits. Note, however, that both are efficient in the sense that their costs in both space and time are at worst polynomial in the problem parameters. The differences are in which scalings are logarithmic (or constant) versus polynomial.

6.2.2 Comparison to prior work on electronic-structure Hamiltonians

Previous work has demonstrated how to implement sparsity-based simulation of the electronic-structure problem in second-quantization [BBK$^+$16] and the configuration-interaction (CI) representation [TL13, BBS$^+$17]. These result in gate counts of $\widetilde{O}(N^5 t)$ and $\widetilde{O}(\eta^2 N^3 t)$, respectively, where N is the number of orbitals, η is the number of electrons, and t is the simulation time. The dependence on error is suppressed in these expressions, but is polylogarithmic in the inverse error.

We can compare our method to [TL13, BBK$^+$16, BBS$^+$17] by applying it to the electronic-structure problem. In this case, we can replace I (the maximum possible number of occupied modes) by η. The total number of modes in our method is $O(\Lambda^d)$, so we may replace Λ^d by N in our asymptotic expressions. If we apply our method to the CI-matrix, Eq. (20) in [BBS$^+$17] gives the sparsity as $O(\eta^2 N^2)$, and the discussion following Eq. (73) in [BBS$^+$17] shows that $\|H\|_{\max}$ is polylogarithmic in N. Finally, g and h are both two for the electronic-structure problem. Making all of these replacements in (6.4) and suppressing polylogarithmic factors gives

$$\widetilde{O}\Big(\eta^2 N^2 t(\eta^2 + N^2)\Big) = \widetilde{O}\Big(\eta^2 N^4 t\Big). \tag{6.10}$$

This is better than the scaling for the second-quantized algorithm of [BBK$^+$16], but worse by a factor of N than the CI algorithm of [BBS$^+$17]. The extra factor of N essentially comes from the fact that the algorithm of [BBS$^+$17] uses the Slater rules directly, which our algorithm does not take into account.

This is illustrative of what we expect to be a general pattern: while our algorithm is applicable to a broad range of second-quantized Hamiltonians, if special structure is known about some particular Hamiltonian it may be possible to design algorithms that are specific to that Hamiltonian and outperform ours. An interesting question for future work is to what extent it is possible to design general-purpose algorithms that are able to naturally take advantage of such problem-specific structure.

6.3 Compact encoding and Hamiltonians

In this section, we define the compact encoding, which maps Fock states to qubit states, and the input model for our second-quantized Hamiltonians. After this section, we will often say "Fock state" when we really mean "compact-encoded Fock state," since the latter is cumbersome. There will usually be no ambiguity in this, since qubit operators can only act upon compact-encoded Fock states, but whenever there is ambiguity we will explicitly state which we are talking about.

6.3.1 Compact encoding of Fock states

Throughout, when we refer to momenta we will mean dimensionless momenta, denoted by $\mathbf{n}$. These are related to the dimensionful momenta $\mathbf{p}$ as

$$p_j = \frac{2\pi}{L_j} n_j \tag{6.11}$$

where L_j is the box size for each component j and we take $\hbar = 1$. We also impose cutoffs Λ on the momenta, i.e., each component n_j must satisfy

$$\Lambda_j^{\min} \leq n_j \leq \Lambda_j^{\max} \tag{6.12}$$

for some cutoffs

$$\Lambda = (\ (\Lambda_1^{\min}, \Lambda_1^{\max}),\ (\Lambda_2^{\min}, \Lambda_2^{\max}), ...,\ (\Lambda_d^{\min}, \Lambda_d^{\max})\ ), \tag{6.13}$$

where d is the number of spatial dimensions. In equal-time quantization, it is generally the case that each $\Lambda_i^{\text{min}} < 0$ and each $\Lambda_i^{\text{max}} > 0$, while in light-front quantization there is some special axis z such that $\Lambda_z^{\text{min}} > 0$ (see Section 6.3.2 for details of light-front quantization).

A Fock state in compact encoding has the form:

$$|\mathcal{F}\rangle = |(q_1, \mathbf{n}_1, w_1), (q_2, \mathbf{n}_2, w_2), ..., (q_J, \mathbf{n}_J, w_J)\rangle, \tag{6.14}$$

where each w_i is the occupancy of the mode q_i with momentum $\mathbf{n}_i$ [KKG+22]. q_i is a collective label that specifies the particle up to its momentum; for example, q_i might determine whether the particle is a boson or a fermion [2], what species of boson or fermion it is (if multiple are present in the theory), whether it is a particle or an antiparticle, and any other properties like spin, flavor, color, etc. We store only occupied modes, so each occupancy $w_i \geq 1$.

Example 6.1. Suppose we have a 1+1D theory containing bosons and fermions whose only quantum number is momentum. We can let $q_i = 0$ label bosons, and $q_i = 1$ label fermions (and $q_i = 2$ label antifermions, but for simplicity we will not include these in the examples below). Then a few examples of compact-encoded Fock states are:

$$|(0, 2, 1)\rangle, \tag{6.15}$$

which encodes one boson with momentum 2.

$$|(0, 2, 2)\rangle \tag{6.16}$$

encodes two bosons with momentum 2.

$$|(0, 2, 3), (1, 5, 1)\rangle \tag{6.17}$$

[2] We leave consideration of exotic particle statistics to future work.

encodes three bosons with momentum 2 and one fermion with momentum 5.

$$|(0,2,3),(0,3,2),(1,5,1)\rangle \tag{6.18}$$

encodes three bosons with momentum 2, two bosons with momentum 3, and one fermion with momentum 5. If our theory also contained spin, for example, then we would expand the q_i labels to include this: e.g., $q_i = \{1, \uparrow\}$ means fermion with spin up, so

$$\big|(\{1,\uparrow\},2,1),\ (\{1,\downarrow\},2,1)\big\rangle \tag{6.19}$$

encodes one fermion with momentum 2 and spin up, and one fermion with momentum 2 and spin down.

We compact-encode a Fock state (6.14) in a qubit register of the form

$$|X_1, X_2, ..., X_I\rangle, \tag{6.20}$$

where I is the maximum possible number of occupied modes, and each X_i is a *mode register* capable of encoding a single mode $(q_i, \mathbf{n}_i, w_i)$. For a Fock state containing $J \leq I$ occupied modes we use the first J of the X_i to encode the modes. The encoded modes are ordered primarily by q_i, and secondarily by momentum. Note that in equal-time quantization the actual number of occupied modes can be unbounded, so we would have to impose a cutoff I by hand. In light-front quantization I is finite and determined by the harmonic resolution [KKG+22]. In chemistry, the particle number (i.e., the total occupation) is generally fixed, so I is equal to the particle number.

Given some maximum number I of occupied modes, either fixed by the theory or imposed by hand, we have to encode I mode registers. Each mode register must encode occupation of the mode (which we take to be upper bounded by some cutoff W), the non-momentum quantum numbers of the mode (which we assume to take a constant number of possible values, and hence to require some fixed number N_q of qubits), and the momentum of the mode. For d spatial dimensions indexed by j,

each component of momentum takes $\Lambda_j^{\max} - \Lambda_j^{\min} + 1$ values, so the momentum of a mode can be encoded in $\sum_{j=1}^{d} \left\lceil \log_2 \left(\Lambda_j^{\max} - \Lambda_j^{\min} + 1 \right) \right\rceil$ qubits. Hence the total number of qubits required to encode a mode is

$$N_q + \lceil \log_2 W \rceil + \sum_{j=1}^{d} \left\lceil \log_2 \left(\Lambda_j^{\max} - \Lambda_j^{\min} + 1 \right) \right\rceil , \tag{6.21}$$

so the total number of qubits to encode a Fock state that contains at most I occupied modes is

$$\mathsf{Q} = I \left(N_q + \lceil \log_2 W \rceil + \sum_{j=1}^{d} \left\lceil \log_2 \left(\Lambda_j^{\max} - \Lambda_j^{\min} + 1 \right) \right\rceil \right). \tag{6.22}$$

If for some dimension j, $\Lambda_j^{\max}$ and $\Lambda_j^{\min}$ are both positive (or both negative), the number of occupied modes I is bounded by some fraction of the total momentum in that dimension. If this is only true for a single dimension (i.e., all other dimensions can have both positive and negative momentum), then this case corresponds to light-front quantization [KKG+22]. Without loss of generality, suppose $\Lambda_1^{\min} > 0$. Let K denote the total momentum in dimension 1. In this case the maximum total number I of occupied modes in any Fock state is

$$I = \left\lfloor \frac{K}{\Lambda_1^{\min}} \right\rfloor , \tag{6.23}$$

since every particle must have momentum at least $\Lambda_1^{\min}$ in dimension 1. A Fock state in which the number of occupied modes achieves the maximum value in (6.23) is one containing $\lfloor K/\Lambda_1^{\min} \rfloor$ modes, all with momentum $\Lambda_1^{\min}$ or $\Lambda_1^{\min} + 1$ in dimension 1 such that the total momentum in dimension 1 is K, but with distinct other quantum numbers [KKG+22]. In the special case where $\Lambda_1^{\min} = 1$, (6.23) simplifies to show that the maximum number of occupied modes is K.

6.3.2 Case study: light-front quantization

Although the compact encoding is agnostic to the form of the theory to which it is applied, it turns out to be particularly advantageous for relativistic field theo-

ries in the light-front (LF) formulation [PB85, HV87, BPP98, VHL$^+$10, KKG$^+$22, KJK$^+$21a, KJK$^+$21b]. Here, we review light-front quantization and explain how compact encoding applies in this case. We will later return to the light-front example to illustrate methods.

We can think of LF quantization as taking the perspective of a massless observer moving at the speed of light in some direction, which we take to be the $-z$ direction (this is admittedly a noninertial reference frame, and therefore not actually a proper observer). The dimensionless discretized momenta along this axis take strictly positive values $n_z \in [1, K]$, where K is the total dimensionless LF momentum of the Fock state (also called the *harmonic resolution*). Importantly, this is also true for massless particles [BPP98]. In other words,

$$\Lambda_1^{\min} = 1, \quad \Lambda_1^{\max} = K, \tag{6.24}$$

where we take axis 1 to correspond to the z direction. Also, since it is the total LF momentum, K automatically imposes a cutoff on the number of excitations in a mode ($W = K$), as well as on the number of occupied modes in a Fock state ($I = \sqrt{2K}$ for $d = 1$ and $I = K$ for $d \geq 2$) [KKG$^+$22]. The momenta along axes transverse to the light-front direction have the same properties as in equal-time quantization.

The above points mean that for light-front quantization the general expression (6.22) for qubit count in compact encoding specializes to

$$\mathsf{Q}_{\text{compact}}^{\text{LF},d=1} = \sqrt{2K}\,(N_q + 2\lceil \log_2 K \rceil) \tag{6.25}$$

for one spatial dimension, or

$$\mathsf{Q}_{\text{compact}}^{\text{LF},d\geq 2} = K\left(N_q + 2\lceil \log_2 K \rceil + \sum_{j=2}^{d} \left\lceil \log_2 \left(\Lambda_j^{\max} - \Lambda_j^{\min} + 1\right)\right\rceil\right) \tag{6.26}$$

for $d > 1$ spatial dimensions.

We can compare this to the number of qubits required for the direct encoding,

as a special case of the general comparison between the two encodings given above. The total number of modes is

$$K\left(\prod_{j=2}^{d}\left(\Lambda_j^{\max}-\Lambda_j^{\min}+1\right)\right)q, \tag{6.27}$$

where $q = O(2^{N_q})$ is the number of possible values of the intrinsic quantum numbers, the number of possible values of the light-front momentum is K, and the number of possible values of the transverse momenta is $\prod_{j=2}^{d}\left(\Lambda_j^{\max}-\Lambda_j^{\min}+1\right)$. In a direct encoding we would encode the occupancy of each of these modes in $\lceil \log_2 K \rceil$ qubits (since K is an upper bound on the occupancy), so the total number of qubits for the direct encoding is

$$\mathsf{Q}_{\text{direct}}^{\text{LF}} = K\left(\prod_{j=2}^{d}\left(\Lambda_j^{\max}-\Lambda_j^{\min}+1\right)\right)q\,\lceil \log_2 K \rceil. \tag{6.28}$$

In other words, for an upper bound $\Lambda^{\perp}$ on the transverse momentum cutoffs, up to constant and logarithmic factors the number of qubits for the direct encoding is $\widetilde{\Theta}\left(K(\Lambda^{\perp})^{d-1}\right)$, while the number of qubits for the compact encoding is $\widetilde{\Theta}\left(K\right)$. This explains why LF quantization motivates development of compact encoding methods. In Section 6.7, we will analyze our oracle constructions for a number of field theories in both equal-time and light-front quantization.

6.3.3 Hamiltonian

A normal-ordered, second-quantized Hamiltonian is composed of terms with the form

$$\beta a_i^{\dagger} a_j^{\dagger} \cdots a_k^{\dagger} a_l a_m \cdots a_n, \tag{6.29}$$

where β is a coefficient, $a^{\dagger}$ and a are fermionic or bosonic creation and annihilation operators, and $i, j, ..., k, l, m, ..., n$ are labels for the particles being created and annihilated. In the remainder of this chapter, we will write creation and annihilation

operators as

$$a_{q_i}^{(\dagger)}(\mathbf{n}_i), \tag{6.30}$$

where $\mathbf{n}_i$ is the momentum of the created or annihilated particle and q_i is a collective label for its remaining quantum numbers (including species), as in the previous section.

We may visualize a term like (6.29) as a diagram with an incoming line for each annihilation operator and an outgoing line for each creation operator. We define an *interaction* to be a sum of such terms, with the momenta varying over all momentum-conserving combinations, but the other properties of the incoming and outgoing particles fixed. Thus we can visualize an interaction as a diagram without momentum specifications. An interaction whose diagram contains f external lines is called an *f-point interaction*. In the remainder of the chapter, when we refer to incoming or outgoing particles, we will mean the incoming or outgoing lines of the diagram of an interaction, which represent annihilation or creation operators, respectively. Note that although a diagram of this kind resembles a Feynman diagram, it does not represent a matrix element calculation, but instead is just a visualization of a collection of ladder operator monomials.

Example 6.2. In a 1+1D theory consider the 3-point interaction

$$\sum_{\mathbf{n}_1,\mathbf{n}_2,\mathbf{n}_3} a_2^\dagger(\mathbf{n}_3)a_1(\mathbf{n}_2)a_1(\mathbf{n}_1), \tag{6.31}$$

where the sum varies over all momenta such that $\mathbf{n}_1 + \mathbf{n}_2 = \mathbf{n}_3$. This interaction describes annihilation of a pair of particles of type '1' to form a single particle of type '2', and is represented by the diagram shown in Fig. 6.1. So, for example, one possible instance of the interaction would map two particles of type '1', both with momentum 2 (i.e., $\mathbf{n}_1 = \mathbf{n}_2 = 2$), to a particle of type '2' with momentum (i.e., $\mathbf{n}_3 = 4$). We can represent these as Fock states in compact encoding as in (6.14):

$$|(\text{`1'}, 2, 2)\rangle \;\rightarrow\; |(\text{`2'}, 4, 1)\rangle, \tag{6.32}$$

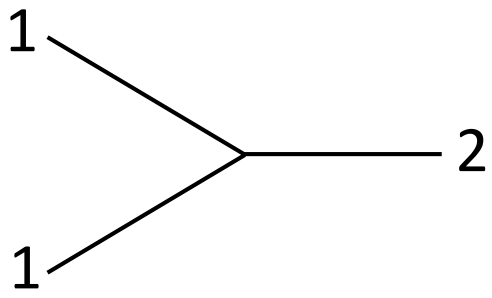


Figure 6.1: Diagram of the example 3-point interaction (6.31), which describes annihilation of a pair of particles of type '1' to form a single particle of type '2' (we read time from left to right).

where we recall that the first entry in each tuple encodes q_i (in this case '1' or '2'), the second entry encodes the momentum, and the third entry encodes the occupation. If instead the incoming momenta were $\mathbf{n}_1 = 1$ and $\mathbf{n}_2 = 3$, then the incoming and outgoing Fock states would instead be represented as

$$|(\text{`1'}, 1, 1), (\text{`1'}, 3, 1)\rangle \ \rightarrow \ |(\text{`2'}, 4, 1)\rangle. \tag{6.33}$$

If another, non-interacting mode were present (say, two particles of type '2' with momentum 5), then we would have

$$|(\text{`1'}, 1, 1), (\text{`1'}, 3, 1), (\text{`2'}, 5, 2)\rangle \rightarrow |(\text{`2'}, 4, 1), (\text{`2'}, 5, 2)\rangle, \tag{6.34}$$

where the additional mode $(\text{`2'}, 5, 2)$ representing the non-interacting particles is present on both sides. Note that since all momenta are positive in the above examples, they represent possible interaction instances in a 1+1D light-front field theory.

We can formally define an interaction as follows:

Definition 6.1. An *interaction* is specified by a set

$$\{(q_1, q_2, ..., q_g), (q_{g+1}, q_{g+2}, ..., q_f)\}, \tag{6.35}$$

together with a coefficient function β that maps sets of momenta for the incoming

and outgoing particles to coefficient values. The $q_1, q_2, ..., q_g$ specify the outgoing particles, and $q_{g+1}, q_{g+1}, ..., q_f$ specify the incoming particles, up to their momenta.

For an interaction as in Definition 6.1, the corresponding *interaction Hamiltonian* is

$$H_I \coloneqq \sum_{\{\mathbf{n}_i\}} \beta(\{\mathbf{n}_i\}) \left(\prod_{i=1}^{g} a_{q_i}^\dagger(\mathbf{n}_i) \right) \left(\prod_{i=g+1}^{f} a_{q_i}(\mathbf{n}_i) \right), \tag{6.36}$$

where the sum runs over all sets $\{\mathbf{n}_i\}$ that conserve total momentum, i.e., such that

$$\sum_{i=1}^{g} \mathbf{n}_i = \sum_{i=g+1}^{f} \mathbf{n}_i. \tag{6.37}$$

Any second-quantized Hamiltonian may be expressed as a sum of interaction Hamiltonians H_I of the form (6.36).

Notice that since the Hamiltonian must be Hermitian, for each interaction the Hamiltonian must also contain its Hermitian conjugate. For example, a Hamiltonian containing the interaction in Example 6.2 (Eq. (6.31)) must also contain the interaction

$$\sum_{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3} a_1^\dagger(\mathbf{n}_1) a_1^\dagger(\mathbf{n}_2) a_2(\mathbf{n}_3), \tag{6.38}$$

where again the sum varies over all momenta such that $\mathbf{n}_1 + \mathbf{n}_2 = \mathbf{n}_3$.

In this chapter, we will assume that the Hamiltonian, and thus the interactions included in it, are fixed up to momentum cutoffs. Each interaction sets particular values of f and g, so f and g can be treated as constants. We will focus on the scaling of our algorithms with momentum cutoffs, which specify the resolution at which we study the given Hamiltonian.

6.4 Sparse Hamiltonians

Sparse Hamiltonians and their simulation are introduced in detail in Section 1.3.1 and Section 1.3.2. We recall the key facts here. A Hamiltonian written as a matrix in a particular basis is said to be *sparse* if the number of nonzero elements in each row (or column) is polylogarithmic in the total Hilbert space dimension. Similarly,

the maximum number of nonzero elements in any row (or column) is called the *sparsity* of the Hamiltonian. A k-sparse Hamiltonian can be efficiently simulated by a quantum algorithm if the locations and values of its nonzero matrix elements can be accessed by a pair of oracles O_L and O_V defined in (1.21) and (1.22), respectively, which we repeat here for convenience:

$$\begin{aligned} O_L|x,i\rangle &= |x,y(x,i)\rangle, \\ O_V|x,y,0\rangle &= |x,y,H_{xy}\rangle, \end{aligned} \tag{6.39}$$

where x, y are row/column indices, $i \in [k]$, and $y(x,i)$ is the index of the ith nonzero entry in row x of H. The main goal in this chapter will be to construct explicit implementations of O_L and O_V for second-quantized Hamiltonians in compact encoding.

6.4.1 Sparse Hamiltonian Simulation for Compact-Encoded Fock States

In the usual definition of the sparse oracles (6.39), which is the definition used in quantum simulation algorithms for sparse Hamiltonians (see Section 1.3.2 and [ATS03, CCD$^+$03, BACS07, Chi10, CK10, CK11, BC12, CW12, BCC$^+$14, BCK15, BCC$^+$15, LC17, LW18, LC19, BCS$^+$20]), the oracles act upon states that encode row and column indices of the Hamiltonian. We instead want to use oracles that act upon Fock states (recall that throughout, when we say "Fock states" we mean "compact-encoded Fock states").

The state-of-the-art methods for simulation of sparse Hamiltonians introduced in Section 1.3.2 have in common the basic step that they use to access the Hamiltonian. This step is implementation of an isometry typically labeled T, which was originally proposed in [Chi10] as a component of a quantum walk, and first used explicitly in a Hamiltonian simulation technique in [BC12]. In our notation, T may be written:

$$T = \sum_{x=1}^{2^n} |x\rangle_a |\phi_x\rangle_{b,c} \langle x|_a \langle 0|_{b,c}, \tag{6.40}$$

where

$$|\phi_x\rangle_{b,c} \coloneqq \sqrt{\frac{r}{\|H\|_1}} \sum_{y=1}^{2^n} \sqrt{H_{xy}^*} |y\rangle_b |0\rangle_c + \sqrt{1 - \frac{r\sigma_x}{\|H\|_1}} |\zeta_x\rangle_b |1\rangle_c, \tag{6.41}$$

and a and b label registers of the same number of qubits, and c labels a single ancilla qubit (we will typically suppress these subscripts when doing so leads to no ambiguity). In (6.41), $r \in (0, 1]$ is a parameter,

$$\sigma_x \coloneqq \sum_{y=1}^{2^n} |H_{xy}|, \tag{6.42}$$

and $|\zeta_x\rangle$ is some linear combination of the $|y\rangle$. T as defined in (6.40) is not a complete definition of a unitary. The action of T on states of $|\cdot\rangle_{b,c}$ other than $|0\rangle_{b,c}$ does not affect the simulation algorithms using T, so that action may be taken to be defined by the circuit implementation of T.

The final ancilla qubit in (6.41) (i.e., the qubit labeled c) is present in order to ensure that the last term (proportional to $|\zeta_x\rangle|1\rangle$) is orthogonal to any of the first set of terms for any x, which is required by Eq. (25) in [Chi10]. Note that the final terms need not be orthogonal to each other for distinct values of x. In fact, it is the final term in $|x\rangle|\phi_x\rangle$ (rather than just $|\phi_x\rangle$) that corresponds to the state $|\perp_j\rangle$ in [Chi10]: this term must therefore take orthogonal values for distinct values of x (see Eq. (24) and the corresponding discussion in [Chi10]), but this is trivially satisfied, since the final term in $|x\rangle|\phi_x\rangle$ is proportional to $|x\rangle|\zeta_x\rangle|1\rangle$.

A complete description of the various ways that the isometry T can be used to construct Hamiltonian simulation algorithms is beyond the scope of this chapter, but it was originally introduced in order to construct the quantum walk operator

$$U = iS(2TT^\dagger - 1) \tag{6.43}$$

in [Chi10], where S is the operator that swaps registers a and b, and also swaps the ancilla qubit c with an additional ancilla qubit initially in the $|0\rangle$ state. Repeatedly applying the quantum walk step U yields a discrete approximation of the Hamiltonian evolution, up to unitary equivalence [Chi10, BC12].

We want to use our new oracles to implement a version of T (6.40) that acts on Fock states rather than on row indices. This will allow the sparse simulation methods to be implemented directly on Fock states.

One complication is the fact that for a k-sparse Hamiltonian, there are at most k nonzero entries in any row or column, but in general there can be fewer than k in some rows and columns. In such a row, some of the values of i, which runs from 0 to $k-1$ and is supposed to index the nonzero entries in a given row, cannot actually index nonzero entries, because there aren't enough nonzero entries in the row. There is no obvious natural mapping from these unused values of i to matrix entries. We will see that this situation arises very commonly for interaction Hamiltonians. We will define oracles that act on Fock states in a way that resolves this issue, and show that we can use these to recover the desired building blocks for the sparse simulation algorithms described above. In Sections 6.5 and 6.6, we will then explicitly construct implementations of these oracles.

Let $|\mathcal{F}\rangle$ be a Fock state, and let H be a k-sparse Hamiltonian. Then there are at most k states $|\mathcal{F}'_i\rangle$ whose Hamiltonian matrix elements with $|\mathcal{F}\rangle$ are nonzero: assume that they are indexed by elements of some set $\mathcal{I}_\mathcal{F} \subseteq [k]$. Let O_L and O_V now define oracles that act as follows:

$$O_L|\mathcal{F}\rangle|0\rangle|i\rangle = |\mathcal{F}\rangle|\Psi'(\mathcal{F},i)\rangle|a(\mathcal{F},i)\rangle, \tag{6.44}$$

$$O_V|\mathcal{F}\rangle|\mathcal{F}'\rangle = |\mathcal{F}\rangle|\mathcal{F}'\rangle\big|\langle\mathcal{F}'|H|\mathcal{F}\rangle\big\rangle, \tag{6.45}$$

where $i \in [k]$ and the functions $\Psi'(\mathcal{F},i)$ and $a(\mathcal{F},i)$ are defined by

$$\Psi'(\mathcal{F},i) = \begin{cases} \mathcal{F}'_i \text{ if } i \in \mathcal{I}_\mathcal{F}, \\ \mathcal{F} \text{ if } i \notin \mathcal{I}_\mathcal{F}, \end{cases} \tag{6.46}$$

$$a(\mathcal{F},i) = \begin{cases} 0 \text{ if } i \in \mathcal{I}_\mathcal{F}, \\ i \text{ if } i \notin \mathcal{I}_\mathcal{F}. \end{cases} \tag{6.47}$$

Without loss of generality, let $\mathcal{F}'_0 = \mathcal{F}$ whenever the matrix element of $\mathcal{F}$ with itself is nonzero. Thus the location oracle O_L may be alternatively expressed as

$$O_L|\mathcal{F}\rangle|0\rangle|i\rangle = \begin{cases} |\mathcal{F}\rangle|\mathcal{F}'_i\rangle|0\rangle \text{ if } i \in \mathcal{I}_\mathcal{F}, \\ |\mathcal{F}\rangle|\mathcal{F}\rangle|i\rangle \text{ if } i \notin \mathcal{I}_\mathcal{F}. \end{cases} \tag{6.48}$$

Next, we define an analog of $|\phi_x\rangle$ (6.41) in terms of Fock states:

$$|\phi_\mathcal{F}\rangle \coloneqq \sqrt{\frac{r}{\|H\|_1}} \sum_{|\mathcal{F}'\rangle} \sqrt{\langle\mathcal{F}'|H|\mathcal{F}\rangle}|\mathcal{F}'\rangle|a(\mathcal{F},\mathcal{F}')\rangle|0\rangle + \sqrt{1 - \frac{r\sigma_\mathcal{F}}{\|H\|_1}}|\zeta_\mathcal{F}\rangle|1\rangle, \tag{6.49}$$

where $a(\mathcal{F},\mathcal{F}')$ is some binary number that is zero when $\langle\mathcal{F}'|H|\mathcal{F}\rangle \neq 0$, $\sigma_\mathcal{F}$ is defined analogously to σ_x, i.e.,

$$\sigma_\mathcal{F} \coloneqq \sum_{|\mathcal{F}''\rangle} |\langle\mathcal{F}|H|\mathcal{F}''\rangle|, \tag{6.50}$$

and $|\zeta_\mathcal{F}\rangle$ is a linear combination of states of the form

$$|\mathcal{F}''\rangle|b\rangle, \tag{6.51}$$

where $|b\rangle$ is some binary number encoded in the same register as $a(\mathcal{F},\mathcal{F}')$. Any of these components that are not completely determined will be determined by the algorithm for constructing $|\phi_\mathcal{F}\rangle$, below. The fact that $a(\mathcal{F},\mathcal{F}') = 0$ whenever $\langle\mathcal{F}'|H|\mathcal{F}\rangle \neq 0$ ensures that for

$$Q(\mathcal{F}) \coloneqq \{|\mathcal{F}'\rangle \mid \langle\mathcal{F}'|H|\mathcal{F}\rangle \neq 0\}, \tag{6.52}$$

we may rewrite (6.49) as

$$|\phi_\mathcal{F}\rangle = \sqrt{\frac{r}{\|H\|_1}} \sum_{|\mathcal{F}'\rangle \in Q(\mathcal{F})} \sqrt{\langle\mathcal{F}'|H|\mathcal{F}\rangle}|\mathcal{F}'\rangle|0\rangle|0\rangle + \sqrt{1 - \frac{r\sigma_\mathcal{F}}{\|H\|_1}}|\zeta_\mathcal{F}\rangle|1\rangle. \tag{6.53}$$

Using $|\phi_{\mathcal{F}}\rangle$, we define a version of T for Fock states:

$$T \coloneqq \sum_{|\mathcal{F}\rangle} |\mathcal{F}\rangle|\phi_{\mathcal{F}}\rangle\langle\mathcal{F}|. \tag{6.54}$$

In Lemma 6.1, below, we show how T can be constructed using $O(1)$ queries to the oracles O_L and O_V as defined by (6.44) and (6.45). First, however, we will show that T as defined by (6.54) may replace the original version of T in (6.40), and the resulting operator acts on Fock states but otherwise reproduces all of the properties of the basic step used in [BC12].

To see this, first note that the first sum in (6.53) is an exact analog of the corresponding sum in (6.41), the definition of $|\phi_x\rangle$ used to construct the standard operator T as used in [BC12, BCK15, LC19, BCS$^+$20]. The only difference is the inclusion of an additional ancilla register in state $|0\rangle$ instead of just the single ancilla qubit in state $|0\rangle$.

The last term in (6.53) is also analogous to the corresponding term in (6.41), but here $|\zeta_{\mathcal{F}}\rangle$ includes the ancilla register $|b\rangle$ (see (6.51)) in addition to the register encoding Fock states, whereas $|\zeta_x\rangle$ in (6.41) is a linear combination of row indices only. However, in order to satisfy the orthogonality conditions discussed in the final paragraph of Section 1.3.2, we only require that for any $\mathcal{F}$, $\mathcal{F}'$, and $\mathcal{F}''$, $|\zeta_{\mathcal{F}}\rangle|1\rangle$ is orthogonal to $|\mathcal{F}''\rangle|a(\mathcal{F}', \mathcal{F}'')\rangle|0\rangle$; this is satisfied because of the final single qubit. These are the only conditions that T must satisfy [BC12].

Lemma 6.1 (Construction of T)**.** *Given an input state $|\mathcal{F}\rangle$, the operator T as defined in* (6.54) *can be implemented using $O(1)$ queries to the oracles O_L and O_V as defined in* (6.44) *and* (6.45)*. (This Lemma closely follows Lemma 4 in [BC12].)*

Proof. The operator T maps a Fock state $|\mathcal{F}\rangle$ to $|\mathcal{F}\rangle|\phi_{\mathcal{F}}\rangle$, for $|\phi_{\mathcal{F}}\rangle$ as defined by (6.53). Explicitly including ancillas, we assume an input state of the form

$$|\mathcal{F}\rangle|0\rangle|0\rangle|0\rangle. \tag{6.55}$$

We map this to $|\mathcal{F}\rangle|\phi_{\mathcal{F}}\rangle$ as follows:

1. Prepare a uniform superposition of the indices $i = 1, 2, ..., kd$ as:

$$\frac{1}{\sqrt{kd}} \sum_{i=1}^{k} d|\mathcal{F}\rangle|0\rangle|i\rangle|0\rangle. \tag{6.56}$$

2. Apply O_L as defined in (6.44) to the first three registers, obtaining

$$\frac{1}{\sqrt{kd}} \sum_{i=1}^{k} d|\mathcal{F}\rangle|\Psi'(\mathcal{F}, i)\rangle|a(\mathcal{F}, i)\rangle|0\rangle. \tag{6.57}$$

3. Controlled on $a(\mathcal{F}, i) = 0$, apply O_V as defined in (6.45) to the first two registers, to calculate $\langle\mathcal{F}_i'|H|\mathcal{F}\rangle$ in an ancilla register that is initially $|0\rangle$ (recall that $a(\mathcal{F}, i) = 0$ if and only if $\langle\mathcal{F}_i'|H|\mathcal{F}\rangle \neq 0$). Then, controlled on the resulting value $\langle\mathcal{F}_i'|H|\mathcal{F}\rangle$, rotate the single ancilla qubit (the final register) as

$$|0\rangle \;\mapsto\; \sqrt{kd\frac{r\langle\mathcal{F}_i'|H|\mathcal{F}\rangle}{\|H\|_1}}|0\rangle + \sqrt{1 - kd\frac{r|\langle\mathcal{F}_i'|H|\mathcal{F}\rangle|}{\|H\|_1}}|1\rangle. \tag{6.58}$$

   Finally, uncompute the ancilla register encoding $\langle\mathcal{F}_i'|H|\mathcal{F}\rangle$ using another controlled query to O_V. This step is identical to the corresponding step in the method described in the proof of Lemma 4 in [BC12].

To obtain the full state after these steps are complete, we insert (6.58) into (6.57), giving

$$\begin{aligned} &\sum_{i=1}^{k} d\sqrt{\frac{r\langle\mathcal{F}_i'|H|\mathcal{F}\rangle}{\|H\|_1}}|\mathcal{F}\rangle|\Psi'(\mathcal{F}, i)\rangle|a(\mathcal{F}, i)\rangle|0\rangle \\ &+ \sum_{i=1}^{k} d\sqrt{\frac{1}{kd} - \frac{r|\langle\mathcal{F}_i'|H|\mathcal{F}\rangle|}{\|H\|_1}}|\mathcal{F}\rangle|\Psi'(\mathcal{F}, i)\rangle|a(\mathcal{F}, i)\rangle|1\rangle. \end{aligned} \tag{6.59}$$

Using the fact that when $\langle\mathcal{F}_i'|H|\mathcal{F}\rangle \neq 0$, we have $\Psi'(\mathcal{F}, i) = \mathcal{F}_i'$ and $a(\mathcal{F}, i) = 0$ (see

(6.46) and (6.47)), we can rewrite the above expression as

$$|\mathcal{F}\rangle\Bigg(\sqrt{\frac{r}{\|H\|_1}}\sum_{|\mathcal{F}'\rangle\in Q(\mathcal{F})}\sqrt{\langle\mathcal{F}'|H|\mathcal{F}\rangle}|\mathcal{F}'\rangle|0\rangle|0\rangle \\ +\sum_{i=1}^{k}d\sqrt{\frac{1}{kd}-\frac{r|\langle\mathcal{F}_i'|H|\mathcal{F}\rangle|}{\|H\|_1}}|\Psi'(\mathcal{F},i)\rangle|a(\mathcal{F},i)\rangle|1\rangle\Bigg), \tag{6.60}$$

Thus as noted in [BC12], the expression in parentheses in (6.60) is equal to $|\phi_{\mathcal{F}}\rangle$ as given by (6.53), for $|\zeta_{\mathcal{F}}\rangle$ given by

$$|\zeta_{\mathcal{F}}\rangle=\frac{1}{\sqrt{1-\frac{r\sigma_{\mathcal{F}}}{\|H\|_1}}}\times\sum_{i=1}^{k}d\sqrt{\frac{1}{kd}-\frac{r|\langle\mathcal{F}_i'|H|\mathcal{F}\rangle|}{\|H\|_1}}|\Psi'(\mathcal{F},i)\rangle|a(\mathcal{F},i)\rangle. \tag{6.61}$$

Hence, we have implemented T using one query to O_L and two queries to O_V, as in [BC12]. □

In Sections 6.5 and 6.6 we describe how to implement O_L and O_V, respectively, in the compact mapping, for general interactions as described in Definition 6.1. These implementations, together with the construction in Lemma 6.1, give us access to the optimal sparse Hamiltonian simulation technique afforded by qubitization [LC19], as well as the other nearly-optimal techniques [BCK15, BCS$^+$20].

In this chapter, we focus on the application to simulating time evolution, which is the goal of all of the sparse simulation papers we have cited [ATS03, CCD$^+$03, BACS07, Chi10, CK10, CK11, BC12, CW12, BCC$^+$14, BCK15, BCC$^+$15, LC17, LW18, LC19, BCS$^+$20]. In Chapter 5 [KL21], we demonstrated how to approximate ground state energies of sparse Hamiltonians using variational quantum eigensolvers. Fock state oracles of the forms given in (6.44), (6.45) apply in this setting, which would allow us to implement VQE for second-quantized Hamiltonians in compact encoding. The measurement scheme is given in Chapter 5, and for an ansatz we could for example implement a version of Unitary Coupled Cluster [RBM$^+$18a] via oracle-based time evolutions generated by the Hamiltonian terms.

Because of the complexity of implementing the oracles, which we will present below in Sections 6.5 and 6.6, the circuit depths required for such an algorithm will

be substantially longer than those used in VQE implementations appropriate for existing quantum computers, and will probably require fault tolerance. However, sparse VQE will still become possible before simulation of time evolution, because it requires only a constant number of oracle queries (at most six [KL21]) per variational circuit, whereas simulation of time evolution requires numbers of oracle queries that scale with the problem parameters, as in (1.35) for example.

6.4.2 Sparsity of general interactions in the Fock basis

For the simulation methods described above to be efficient, we require the interactions to be sparse. Consider an interaction specified as in (6.35), in d dimensions with cutoffs as in (6.13). The corresponding interaction Hamiltonian is given by (6.36).

To obtain an upper bound on the sparsity we may assume that the $f - g$ incoming particles in the interaction may be taken from any of the modes in the input state (recall that f is the total number of external lines in the interaction, and g is the number of outgoing lines). In general, this upper bound is not tight, because it requires all modes in the input state to be the same up to momentum and to match all incoming lines in the interaction, but it will suffice to show that the sparsity is polynomial in the momentum cutoffs. In terms of the maximum number of modes I in the input state (as in Section 6.3.1), this upper bound is

$$\left(\!\!\binom{I}{f-g}\!\!\right) = O\left(I^{f-g}\right), \tag{6.62}$$

where the left-hand side denotes I choose $f - g$ with replacement. As discussed in Section 6.3.3, the intrinsic quantum numbers of the outgoing particles are fixed by the interaction, but their momenta can take any values consistent with total momentum conservation. For simplicity, we may upper bound this by counting all

outgoing momentum assignments consistent with the cutoffs, of which there are

$$O\left(\prod_{j=1}^{d}\left(\Lambda_j^{\max}-\Lambda_j^{\min}\right)^{g-1}\right), \tag{6.63}$$

since we assign a momentum to each of the g outgoing particles, but one of these is fixed by momentum conservation.

Hence, an upper bound on the total number of connected states to any given initial state under the interaction, and thus on the sparsity, is the product of (6.62) and (6.63):

$$k \leq O\left(I^{f-g}\prod_{j=1}^{d}\left(\Lambda_j^{\max}-\Lambda_j^{\min}\right)^{g-1}\right) \leq O\left(\Lambda_{\max}^{d(f-1)}\right). \tag{6.64}$$

The second upper bound is obtained by replacing I with the total number of modes as a function of $\Lambda_{\max}$, the maximum momentum cutoff. This illustrates that even in this worst case, the sparsity is polynomial in the momentum cutoffs, for fixed dimension d and number f of external lines in the interaction. The number of qubits Q is linear in I up to logarithmic factors, so assuming I is polynomial in the momentum cutoffs, Q is also polynomial in the momentum cutoffs, and thus the sparsity is polynomial in Q.

6.5 Location oracle

In this section, we describe how to efficiently implement the oracle O_L, whose action is given by (6.44). In the first subsection, we describe some examples that illustrate all of the main techniques required for the general method. In the second subsection, we describe the general method. This description refers extensively to the details explained in the examples in the first subsection, so we strongly encourage the reader to begin with these. Finally, in the third subsection we analyze the general method.

6.5.1 Examples

For an input state $|\mathcal{F}\rangle|i\rangle$, where $|\mathcal{F}\rangle$ is some (compact-encoded) Fock state and $i \in [k]$ for sparsity k, the action of the location oracle should be as follows: in an ancilla register, compute $|\mathcal{F}'_i\rangle$, the ith Fock state whose matrix element with $|\mathcal{F}\rangle$ is nonzero, and also uncompute $|i\rangle$. Note that this action is assuming that i does in fact index a nonzero matrix element (see Section 6.4); the later examples will illustrate the possible situations where this may not be the case.

Example 6.3. Consider a boson field on which our interaction is the number operator,

$$H_I = \sum_{\mathbf{n}} a^\dagger(\mathbf{n}) a(\mathbf{n}). \tag{6.65}$$

In the notation of Definition 6.1, if we let '0' denote 'boson', we would express this interaction as $\{(\text{'0'}), (\text{'0'})\}$ (meaning one boson in, one boson out), and the coefficient function is just $\beta(\mathbf{n}) = 1$. The interaction Hamiltonian H_I maps any Fock state to itself, rescaled by a coefficient given by the total number of particles. Hence, this interaction is diagonal and one-sparse (it maps each Fock state to at most one other Fock state). Therefore, for any input $|\mathcal{F}\rangle|i = 0\rangle$ (since i takes only one value for a one-sparse interaction), the output of the location oracle is just $|\mathcal{F}\rangle$, i.e.,

$$O_L|\mathcal{F}\rangle|0\rangle = |\mathcal{F}\rangle|\mathcal{F}\rangle. \tag{6.66}$$

Comparing this expression to (6.44), the reader will notice that we are suppressing the output $|a(\mathcal{F}, i)\rangle$, but for the current example $a(\mathcal{F}, i)$ is always 0.

Example 6.4. Let us still consider a boson field, but now with the three-point interaction

$$H_I = \sum_{\mathbf{n}_1, \mathbf{n}_2} a^\dagger(\mathbf{n}_1 + \mathbf{n}_2) a(\mathbf{n}_2) a(\mathbf{n}_1) \tag{6.67}$$

i.e., two incoming bosons annihilate to form a single outgoing boson whose momentum is the sum of the incoming momenta. In the notation of Definition 6.1, we would express this interaction as $\{(\text{'0'}), (\text{'0'}, \text{'0'})\}$ (meaning two bosons in, one boson out),

and the coefficient function is still just $\beta(\mathbf{n}_1, \mathbf{n}_2) = 1$. Even though this example appears only slightly more complicated than the number operator in Example 6.3, it in fact will introduce almost all of the considerations we will require for completely general interactions. We break the implementation of O_L up into steps.

Step 1 (identify incoming modes). Given the input Fock state $|\mathcal{F}\rangle$, we need to use the input index $|i\rangle$ to determine the output Fock state $|\mathcal{F}'_i\rangle$. As a first step, we use i to choose the two modes in $|\mathcal{F}\rangle$ that will contribute the incoming bosons in the interaction. For maximum possible number of occupied modes I (as in Section 6.3.1), we let i index all pairs J_i of mode indices: i.e.,

$$\left\{J_i \mid i = 1, 2, ..., I^2\right\} = \left\{(j_1, j_2) \mid j_1, j_2 = 1, 2, ..., I\right\} \tag{6.68}$$

(the pair could come from the same mode). So the first step in implementing O_L is to compute J_i in an ancilla register, as a function of i.

Step 2 (remove incoming bosons). Next, we find the momenta of the two modes indexed by J_i, and make sure that they have sufficient occupation to provide the incoming particles. We can combine this with decrementing the occupation of those modes, as the first step towards constructing $|\mathcal{F}'_i\rangle$. So, prior to beginning this step, we copy $|\mathcal{F}\rangle$ to a second qubit register that will become the output register containing $|\mathcal{F}'_i\rangle$ at the end of the implementation. Note that this copying of $|\mathcal{F}\rangle$ is allowed because it is just copying in the (compact-encoded) Fock basis, which can be implemented via qubitwise CNOTs from the input $|\mathcal{F}\rangle$ to the new copy register assuming this is initially in the all-zeroes state.

On the copy of $|\mathcal{F}\rangle$, we first find the j_1th mode (where $(j_1, j_2) = J_i$), and check whether it has nonzero occupation. If it does not, then we need some way to record the fact that i does not index a valid nonzero matrix element. The way we do this is to maintain an ancilla register called the "flag" register, whose value is initially zero and should remain zero at the end of the implementation of O_L if and only if i indexes a valid nonzero matrix element. So, to check whether the j_1th mode has nonzero occupation, we add one to the flag register controlled on

j_1th mode having zero occupation. Now we may proceed as if the mode does have nonzero occupation, knowing that if the flag register is nonzero at the end of the implementation we will simply reverse the whole procedure.

Next we decrement the occupation of the j_1th mode by one, and record its momentum in an ancilla register that encodes $\mathbf{n}_1$, the momentum of the first incoming boson. If the occupation of the mode is now zero, we should also set the remaining qubits encoding the mode to all zeroes (which we can do by applying CNOTs controlled on the corresponding qubits in the original version of $|\mathcal{F}\rangle$ — recall that we are currently operating on the copy). Also controlled on the occupation of the current mode being zero, we store its index in the first entry of an ancilla register E, which we will use later.

We now repeat this whole procedure (including checking the occupation) for the j_2th mode, recording its momentum in another ancilla register that encodes $\mathbf{n}_2$. If this mode is left empty, we store its index in the second entry of the ancilla register E. Note that if both modes are the same, i.e., $j_1 = j_2$, then at this second step we will add one to the flag register if the initial occupation of that mode was not at least two, since one boson has already been removed from the mode.

Step 3 (reorder modes). Once we are done with steps 1 and 2, we have $\mathbf{n}_1$ and $\mathbf{n}_2$ recorded in ancilla registers, and we have decremented the corresponding mode occupations in the copy of $|\mathcal{F}\rangle$. However, it is possible that up to two modes in the copy of $|\mathcal{F}\rangle$ may have been left empty after their occupations were decremented. Since the compact encoding only stores occupied modes, and encodes them in the first J mode registers X_i (see (6.20)), we need to move any empty modes to the end of the encoding.

To do this, we use the ancilla register E that we introduced above. This contains two entries E_1 and E_2; $E_1 = j_1$ if the j_1th mode was left empty after removing the incoming particles to the interaction, and otherwise E_1 remains in its initial state (which should be chosen to be different from any of the values encoding indices). E_2 was similarly set according to the j_2th mode.

Using these, we reorder the modes as follows. Iterate over $j = 1, 2, ..., I-1$

(the mode indices). For each j, swap the jth register and the $(j+1)$th register controlled on $E_1 \leq j$ (and on E_1 actually encoding a valid mode index). If the j_1th mode is emptied, the first time the above control condition will be satisfied is when $j = E_1 = j_1$, so this will swap the emptied j_1th mode register with the (j_1+1)th mode register. Next we move to $j = j_1 + 1 \geq E_1$, so the (j_1+1)th mode register (which now contains the empty mode) gets swapped with the (j_1+2)th mode register, and so forth until the empty mode has been moved to the last mode register.

We now repeat this procedure for E_2 in order to move the j_2th mode to the end of the mode registers if it was emptied. The only caveat with this step is that, if both the j_1th and j_2th modes were emptied and $j_1 < j_2$, then after the first swapping sequence (for $E_1 = j_1$), the empty mode that was initially in position j_2 is now in position $j_2 - 1$, since the j_1th mode was swapped out from before it. Therefore, before repeating the procedure for E_2, we should subtract one from E_2 controlled on $E_1 < j_2$.

Step 4 (insert outgoing boson). After we have completed all of the above steps, the copy of $|\mathcal{F}\rangle$ has had the two incoming bosons to the interaction removed, and the modes have been reordered if necessary so that the J occupied modes are encoded in the first J mode registers. The momenta $\mathbf{n}_1, \mathbf{n}_2$ of the incoming bosons are also recorded in ancilla registers. All that remains is to insert the new boson with momentum $\mathbf{n}_1 + \mathbf{n}_2$.

To do this, we first iterate over the already occupied modes, checking whether each one has momentum $\mathbf{n}_1 + \mathbf{n}_2$ and incrementing its occupation if so. We also flip a single ancilla qubit from $|0\rangle$ to $|1\rangle$ if we find such a mode, to record the fact that we have inserted the new boson. If at the end of the iterations, this ancilla qubit is still $|0\rangle$, then the new boson needs to be inserted as a new mode (so the following operations should be controlled on this). In this case, we first find the location where the new mode should be inserted: to do this, we iterate over $j = 1, 2, ..., I$, checking whether $\mathbf{n}_1 + \mathbf{n}_2$ is greater than the momentum of the $(j-1)$th mode and less than the momentum of the jth mode. This will only be true for a single value

of j, which we can call j', so we record j' in an ancilla register.

Then we iterate over the mode indices in reverse order, i.e., $j = I-1, ..., 1$, for each j swapping the jth and $(j+1)$th mode registers controlled on $j' \leq j$. Since the maximum possible number of occupied modes is I and we are about to insert a new mode, prior to this sequence of swaps the Ith mode register is guaranteed to be unencoded. Hence, when $j = I-1$, we swap the $(I-1)$th and Ith mode registers, moving the unencoded register to location $I-1$. We then proceed to $j = I-2$, swapping the $(I-2)$th and $(I-1)$th mode registers and thus moving the unencoded register to location $I-2$, and so forth. The last swap we perform is when $j = j'$, so once we are done with the full sequence of swaps the unencoded register is in location j'. Therefore, we can simply set its momentum to $\mathbf{n}_1 + \mathbf{n}_2$ and set its occupation to one, and we are done.

Step 5 (uncompute ancillas). Once all of the above operations are complete, the copy of $|\mathcal{F}\rangle$ has been transformed into the desired output Fock state $|\mathcal{F}'_i\rangle$, so all that remains is to uncompute the ancillas. This could be done simply by copying $|\mathcal{F}'_i\rangle$ (in the Fock basis) and then exactly reversing all of the prior operations, but we also want to uncompute $|i\rangle$, the input index. In order to accomplish this, we uncompute $|i\rangle$ using the register encoding J_i. But now we cannot uncompute J_i using i, since this value has been uncomputed, so we instead need to uncompute J_i using the registers it was used to compute. The details of this are tedious, but since $|\mathcal{F}\rangle$ and $|\mathcal{F}'_i\rangle$ together contain enough information to determine the values of all of the ancillas that were used to compute $|\mathcal{F}'_i\rangle$, we can use those values to uncompute the ancillas via similar operations to those used to compute $|\mathcal{F}'_i\rangle$. These details can be found in the code[3] that compiles our oracle constructions, and do not increase the asymptotic costs of the oracles.

Also, if we reach the end of the procedure and the flag register is nonzero, then we know that the index i did not in fact correspond to a valid matrix element. In this case, the desired output as given in (6.44) is $|\mathcal{F}\rangle|\mathcal{F}\rangle|i\rangle$. Hence, we want to keep

[3]Python code to compile the O_L oracle is available on GitHub at `https://github.com/wmkirby1/quspice`.

$|i\rangle$ (which is $|a(\mathcal{F}, i)\rangle$ in this case), completely reverse the rest of the computation, and then just copy $|\mathcal{F}\rangle$ itself to the output Fock state register. This completes the implementation of O_L for this example.

Example 6.5. Let us again consider a boson field, but now with the four-point interaction

$$H_I = \sum_{\mathbf{n}_1,\mathbf{n}_2,\mathbf{n}_3,\mathbf{n}_4} a^\dagger(\mathbf{n}_4)a^\dagger(\mathbf{n}_3)a(\mathbf{n}_2)a(\mathbf{n}_1), \tag{6.69}$$

where the sum runs over all momentum conserving combinations, i.e., $\mathbf{n}_1 + \mathbf{n}_2 = \mathbf{n}_3 + \mathbf{n}_4$. Many of the steps to implement the O_L oracle for this interaction are the same as in Example 6.4, so instead of going through the entire procedure again, we will just describe what needs to change.

Steps 1 through 3, in which we identify the incoming modes, decrement their occupations, and reorder the modes if some of them are left empty, are the same as in Example 6.4. However, the incoming momenta no longer uniquely determine the outgoing momenta, since for a given value of $\mathbf{n}_1 + \mathbf{n}_2$ there are multiple values of $\mathbf{n}_3$ and $\mathbf{n}_4$ that satisfy momentum conservation. Therefore, our input index i needs to do more work than just to specify J_i (the incoming mode indices). In particular, after the total incoming momentum

$$\mathbf{Q} \coloneqq \mathbf{n}_1 + \mathbf{n}_2 \tag{6.70}$$

has been determined, we need to use i to determine how this momentum should be split up between the outgoing bosons.

To do this, we use a classically-precomputed lookup table $A(\mathbf{Q}, i)$ that maps any possible value of the total momentum $\mathbf{Q}$ together with an index value i to a partition of $\mathbf{Q}$ among the outgoing particles. For example, if we are in a 1+1D light-front field theory (see Section 6.3.2) with momentum cutoffs $\Lambda^{\min} = 1$, $\Lambda^{\max} =$

$K = 5$, then we could take $A(\mathbf{Q}, i)$ to be

$$\begin{aligned} A = \{ & \\ & (2,0) \mapsto \{1,1\}, \\ & (3,0) \mapsto \{2,1\}, \\ & (4,0) \mapsto \{3,1\}, \\ & (4,1) \mapsto \{2,2\}, \\ & (5,0) \mapsto \{4,1\}, \\ & (5,1) \mapsto \{3,2\} \\ \}. & \end{aligned} \tag{6.71}$$

In other words, for each fixed value of $\mathbf{Q}$, i indexes the possible partitions of $\mathbf{Q}$ into two parts satisfying the momentum cutoffs, and $A(\mathbf{Q}, i)$ returns these values. So, for example, if $\mathbf{Q} = 4$ and $i = 1$, then

$$A(\mathbf{Q}, i) = A(4, 1) = \{2, 2\}. \tag{6.72}$$

So, given the actual value of $\mathbf{Q}$, which is stored in some ancilla register, and the value of i, which is one of the quantum inputs, we need to compute $A(\mathbf{Q}, i)$ in an ancilla register. To do this, we classically iterate over the possible values $(\mathbf{Q}', i')$, for each one setting the ancilla register to $A(\mathbf{Q}', i')$ controlled on $(\mathbf{Q}, i) = (\mathbf{Q}', i')$. When this iteration is complete, we will have the outgoing momenta stored in this ancilla register.

Since we are now using i both to specify J_i and $A(\mathbf{Q}, i)$, we need to keep these independent. To do this, if $A(\mathbf{Q}, i)$ requires at most a distinct values of i, then we can let $A(\mathbf{Q}, i)$ be a function of i mod a and J_i be a function of $\lfloor i/a \rfloor$. For example, the instance of $A(\mathbf{Q}, i)$ given in (6.71) only requires two distinct values of i, so for this case we could let

$$A(\mathbf{Q}, i) \mapsto A(\mathbf{Q}, i \bmod 2) \tag{6.73}$$

and

$$J_i \mapsto J_{\lfloor i/2 \rfloor}. \tag{6.74}$$

A simple special case is when $a = 10^b$ for some positive integer b, in which case $i \bmod a$ is the last b digits of i and $\lfloor i/a \rfloor$ is all remaining digits of i.

The only additional consideration is that, as illustrated in (6.71), depending on the value of $\mathbf{Q}$ not all of the values of $i \bmod a$ may be used to specify outgoing momentum assignments via $A(\mathbf{Q}, i)$. If $i \bmod a$ takes one of these unused values, then this is just another instance of i not indexing a valid matrix element, so we should add one to the flag register. This would happen, for example, if we obtained the inputs $\mathbf{Q} = 2$, $i \bmod 2 = 1$ for $A(\mathbf{Q}, i \bmod 2)$ as given in (6.71).

Once we have specified the two outgoing momenta $\mathbf{n}_3$ and $\mathbf{n}_4$, inserting them in the outgoing Fock state just requires applying step 4 in Example 6.4 twice. Step 5 is then also the same as for Example 6.4, and that completes the implementation of O_L for this example.

Example 6.6. Let us now, finally, consider an interaction including fermions as well as bosons:

$$H_I = \sum_{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3, \mathbf{n}_4} a_1^\dagger(\mathbf{n}_4) a_0^\dagger(\mathbf{n}_3) a_1(\mathbf{n}_2) a_0(\mathbf{n}_1), \tag{6.75}$$

where subscript 0 indicates boson and subscript 1 indicates fermion, and the sum runs over all momentum conserving combinations, i.e., $\mathbf{n}_1 + \mathbf{n}_2 = \mathbf{n}_3 + \mathbf{n}_4$. Hence this interaction is an incoming boson and fermion, and an outgoing boson and fermion.

Many of the elements of the implementation of O_L are the same as in the previous two examples. One change is that the orders of the values of J_i and $A(\mathbf{Q}, i)$ now matter, since the two incoming modes are now distinguishable, as are the two outgoing modes. Also, we must now check that for $J_i = (j_1, j_2)$, the j_1th mode is bosonic and the j_2th mode is fermionic, adding one to the flag register if either is not. The remainder of identifying and removing the incoming particles, reordering the modes, and computing the outgoing momenta are the same as in the prior examples.

When inserting the outgoing fermion we must also alter the procedure. When

we iterate over the modes to check whether any match the new fermion to be inserted, instead of incrementing its occupancy if we find a mode that matches (as we would for a boson), we add one to the flag register, because fermionic modes cannot have occupancy greater than one. We then proceed with inserting the fermion as a new mode in exactly the same way as for bosons, and complete the rest of the procedure exactly as in Example 6.4. Note that when computing the matrix element oracle O_V, we will have to additionally treat fermions and bosons differently because of their different commutation relations, but for the location oracle this is not relevant.

6.5.2 General method

We assume that the input is in the form $|\mathcal{F}\rangle|i\rangle$, where $|\mathcal{F}\rangle$ is some compact-encoded Fock state, and $i \in [k]$ where k is the sparsity. The f-point interaction is specified as in Definition 6.1, i.e., as a set of g outgoing lines identified by $(q_1, q_2, ..., q_g)$, and a set of $f-g$ incoming lines identified by $(q_{g+1}, q_{g+2}, ..., q_f)$. The momentum cutoffs are as in (6.13): for each momentum $\mathbf{n}$, each component n_j must satisfy

$$\Lambda_j^{\min} \leq n_j \leq \Lambda_j^{\max}, \tag{6.76}$$

where j runs over the dimensions. All of the main ideas for the implementation of O_L were introduced by the examples in Section 6.5.1, mostly in Example 6.4, so we simply indicate how to appropriately generalize these ideas in order to describe the method for arbitrary interactions.

Step 1 (identify incoming modes). This step is the same as in Examples 6.4 and 6.5, except that the possible sets of incoming modes now have size $f-g$:

$$J_i \in \{1, 2, ..., I\}^{f-g}. \tag{6.77}$$

We therefore have to check that for $J_i = (j_1, j_2, ..., j_{f-g})$, the j_kth mode in $|\mathcal{F}\rangle$ matches the identifying information q_{g+k} of the kth incoming line in the interaction.

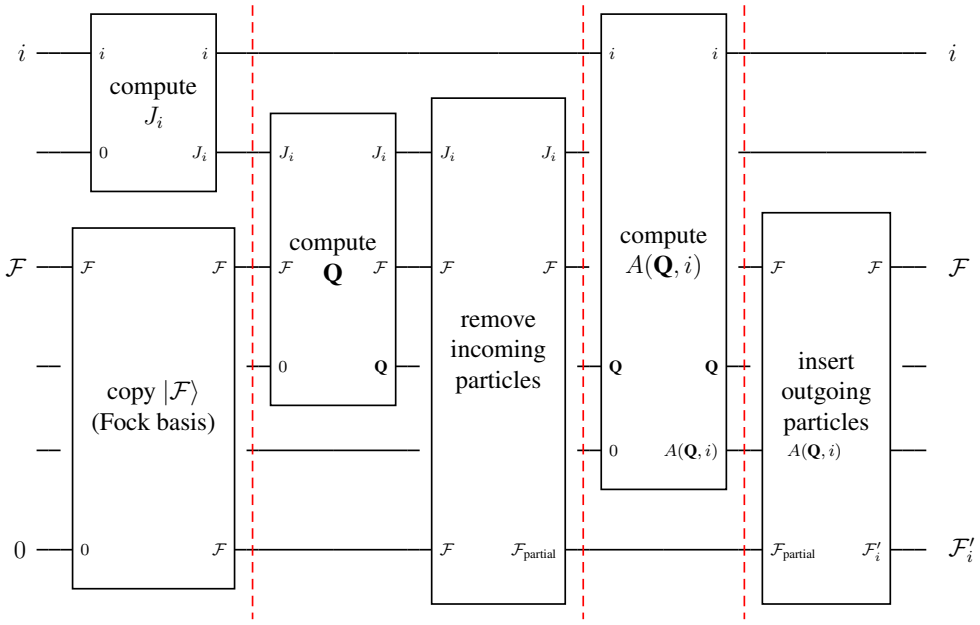


Figure 6.2: Schematic of the circuit to implement the location oracle. Labeled wires are input and output registers, and unlabeled wires are ancilla registers (each initially in the all 0s state). The inputs and outputs are $|\mathcal{F}\rangle$ (the incoming Fock state), $|i\rangle$ (the sparsity index), and $|\mathcal{F}'_i\rangle$ (the outgoing Fock state), all as in (6.44). The intermediate quantities computed are J_i (the list of modes in $|\mathcal{F}\rangle$ from which the incoming particles are taken, as in (6.77)), $\mathbf{Q}$ (the total incoming and outgoing momentum), and $A(\mathbf{Q}, i)$ (the list of momenta of the outgoing particles, as in (6.78)). The circuit hides some additional ancillas, does not show uncomputation of the ancillas, and shows the action when i is a valid index for an outgoing state. At the points marked by red dashed lines we have to check whether this is true, as described in detail in the text.

This generalizes the step in Example 6.6 where we check that the j_1th mode is bosonic and the j_2th mode is fermionic.

Step 2 (remove incoming particles and reorder modes). This step is implemented exactly as in the examples, just with more repetitions of the removal procedure as we decrement the occupation in modes j_1, j_2, through j_{f-g}. We store the momenta $\mathbf{n}_{g+1}, \mathbf{n}_{g+2}, ..., \mathbf{n}_f$ of the incoming lines in ancilla registers, and compute their sum $\mathbf{Q}$. Reordering modes is also exactly as in the examples; the list E of emptied modes must now contain $f - g$ entries.

Step 3 (compute outgoing momenta). The classical lookup table $A(\mathbf{Q}, i)$ maps values of $\mathbf{Q}$ and i to ordered sets of g outgoing momenta:

$$A(\mathbf{Q}, i) = (\mathbf{n}_1, \mathbf{n}_2, ..., \mathbf{n}_g) \tag{6.78}$$

such that

$$\sum_{k=1}^{g} \mathbf{n}_k = \mathbf{Q}. \tag{6.79}$$

As in Example 6.5, we classically iterate over the possible values $\mathbf{Q}'$ and i' of $\mathbf{Q}$ and i, and implement a quantum operation that encodes $A(\mathbf{Q}', i')$ in an ancilla register controlled on $(\mathbf{Q}', i') = (\mathbf{Q}, i)$.

The only difference is that now $\mathbf{Q}$ and i can take more values. $\mathbf{Q}$ can be any momentum that is the sum of $f - g$ momenta consistent with the cutoffs $\Lambda_j^{\min}, \Lambda_j^{\max}$ in each dimension j. For a given $\mathbf{Q}$, outgoing momenta can be any set satisfying (6.79), so i must provide enough distinct values to distinguish these assignments for whichever value of $\mathbf{Q}$ gives the most of them. We will provide a detailed analysis of this below.

Step 4 (insert outgoing particles). This step is the same as in the examples in Section 6.5.1, except that we must now repeat the insertion procedure g times, once for each of the g outgoing modes. The momenta of the outgoing modes are given by $\mathbf{n}_1, \mathbf{n}_2, ..., \mathbf{n}_g$, which we computed in step 3, and their identifying information (particle types and quantum numbers) is given by $q_1, q_2, ..., q_g$ in the

interaction specification.

Step 5 (uncompute ancillas). This step is the same as in the examples in Section 6.5.1. This completes the implementation of O_L for a general interaction. A schematic for the circuit is shown in Fig. 6.2.

6.5.3 Analysis

We will analyze the above algorithm in terms of the number of log-local operations required. The specific log-local operations of interest are actions on constant numbers of mode registers X_j, either in encoded states or in ancilla registers. These are log-local because each mode register contains logarithmically-many qubits in the momentum and occupation number cutoffs (see (6.21)). The log-local operations we used are all controlled arithmetic operations. The problem of compiling such operations into primitive gates can be addressed independently, and is well-studied (see for example [JPK$^+$14]). The choice of primitive gate set to compile into is also hardware-specific. Hence, we express our gate counts in terms of the log-local operations.

We analyze each of the steps outlined in the previous section. Step 1 requires controlling on the possible values of J_i, leading to a number of log-local operations that scales with the number of possible values of J_i. By (6.77), the number of possible values of J_i is upper bounded by I^{f-g} (recall that I is the maximum possible number of occupied modes), so

$$O\left(I^{f-g}\right) \tag{6.80}$$

is an upper bound on the number of log-local operations required to implement step 1. Recall that f and g are constant, so (6.80) is polynomial in I.

Step 2 requires finding the modes whose indices match indices in J_i, decrementing their occupations, and reordering the modes: these are implemented via a constant number of simultaneous iterations over the $f - g$ entries in J_i, and over

the I modes in the copy of $|\mathcal{F}\rangle$. Thus

$$O\left(I(f-g)\right) \tag{6.81}$$

is an upper bound on the number of log-local operations required to implement step 2.

Step 3 requires controlling on the pairs of possible values of $\mathbf{Q}$ and i that give distinct values of $A(\mathbf{Q}, i)$. The number of such pairs is the same as the number of possible distinct values of $A(\mathbf{Q}, i)$. These values have the form (6.78), so the number of possible values is upper bounded by the number of possible values for each entry, raised to power g. Each entry is the momentum of a single particle, so if we take $\Lambda_{\max}$ to be the maximum momentum cutoff (in magnitude) over all dimensions, the number of possible values for each entry in $A(\mathbf{Q}, i)$ is $O(\Lambda_{\max}^d)$ (recall that d is the spatial dimension). Hence,

$$O\left(\Lambda_{\max}^{dg}\right) \tag{6.82}$$

is an upper bound on the number of distinct values of $A(\mathbf{Q}, i)$, and thus also an upper bound on the number of log-local operations required to implement step 3.

Step 4 requires a constant number of simultaneous iterations over the g outgoing modes (determined by the value of $A(\mathbf{Q}, i)$ and $q_1, q_2, ..., q_g$ as specified by the interaction), and over the I modes in the copy of $|\mathcal{F}\rangle$. Thus

$$O\left(Ig\right) \tag{6.83}$$

is an upper bound on the number of log-local operations required to implement step 4.

Step 5, uncomputing the ancillas, at worst doubles the cost of the full algorithm, so we may ignore it in the scaling. The costs of steps 2 and 4 are subsumed by the costs of steps 1 and 3, so the total number of log-local operations required

to implement the location oracle is

$$O\left(I^{f-g} + \Lambda_{\max}^{dg}\right). \tag{6.84}$$

Hence, the number of log-local operations required to implement the location oracle is polynomial in the momentum cutoff $\Lambda_{\max}$, the number I of mode registers, and the number of qubits (since this is linear in I and logarithmic in the other parameters — see (6.22)).

6.6 Matrix element (value) oracle

The oracle O_V defined in (6.45) calculates a matrix element of the interaction Hamiltonian H_I (given by (6.36)) to some desired precision. The quantum input is a pair of compact-encoded Fock states $|\mathcal{F}\rangle$ and $|\mathcal{F}'\rangle$, taken to be the incoming and outgoing states in the interaction, respectively. As described in the proof of Lemma 6.1, above, O_V is only implemented when $|\mathcal{F}'\rangle = |\mathcal{F}'_i\rangle$ for some i (recall that $|\mathcal{F}'_i\rangle$ is the ith connected state to $|\mathcal{F}\rangle$), so we may assume that the matrix element of $|\mathcal{F}\rangle$ and $|\mathcal{F}'\rangle$ is nonzero (or that it is zero and that fact is recorded by the register $a(\mathcal{F}, i)$ being nonzero — see (6.47)).

The value of the matrix element is given by its coefficient $\beta(\{\mathbf{n}_i\})$ as in (6.36) multiplied by any factors coming from the ladder operators. As usual, applying a creation operator to a mode containing w particles contributes a factor of $\sqrt{w+1}$, while applying an annihilation operator contributes a factor of $\sqrt{w}$. In order to enforce antisymmetrization of fermions and antifermions, each (anti)fermionic ladder operator also contributes a factor of ± 1 determined by the parity of the number of particles of the same type encoded in mode registers preceding the mode register acted upon by the ladder operator (in the canonical ordering established in Section 6.3.1).

Consider a general interaction, with incoming lines $q_{g+1}, q_{g+2}, ..., q_f$ and outgoing lines $q_1, q_2, ..., q_g$. This interaction connects Fock states $|\mathcal{F}\rangle$ and $|\mathcal{F}'\rangle$ when

there is some assignment of momenta $\{\mathbf{n}_i\}$ to the incoming and outgoing lines that conserves momentum, i.e., $\sum_{i=1}^{g} \mathbf{n}_i = \sum_{i=g+1}^{f} \mathbf{n}_i$, such that

$$\langle \mathcal{F}' | \left(\prod_{i=1}^{g} a_{q_i}^\dagger(\mathbf{n}_i) \right) \left(\prod_{i=g+1}^{f} a_{q_i}(\mathbf{n}_i) \right) |\mathcal{F}\rangle \neq 0. \tag{6.85}$$

This is the case if and only if $\{(q_i, \mathbf{n}_i) \mid i = 1, 2, ..., g\}$ are the extra particles in $|\mathcal{F}'\rangle$ (and not in $|\mathcal{F}\rangle$), and $\{(q_i, \mathbf{n}_i) \mid i = g+1, g+2, ..., f\}$ are the extra particles in $|\mathcal{F}\rangle$ (and not in $|\mathcal{F}'\rangle$). When this condition holds,

$$\langle \mathcal{F}' | \left(\prod_{i=1}^{g} a_{q_i}^\dagger(\mathbf{n}_i) \right) \left(\prod_{i=g+1}^{f} a_{q_i}(\mathbf{n}_i) \right) |\mathcal{F}\rangle = \pm \sqrt{\left(\prod_{i=1}^{g} w_i' \right) \left(\prod_{i=g+1}^{f} w_i \right)}, \tag{6.86}$$

where the $\pm$ is set by fermion/antifermion antisymmetrization, each w_i (for $i = g+1, g+2, ..., f$) is the occupation of the mode $(q_i, \mathbf{n}_i)$ in

$$\left(\prod_{j=i+1}^{f} a_{q_j}(\mathbf{n}_j) \right) |\mathcal{F}\rangle, \tag{6.87}$$

and each w_i' (for $i = 1, 2, ..., g$) is the occupation of the mode $(q_i, \mathbf{n}_i)$ in

$$\left(\prod_{j=i}^{g} a_{q_j}^\dagger(\mathbf{n}_j) \right) \left(\prod_{j=g+1}^{f} a_{q_j}(\mathbf{n}_j) \right) |\mathcal{F}\rangle. \tag{6.88}$$

In other words, if multiple creation or annihilation operators act on the same mode, for each the corresponding w_i or w_i' should be the occupation of the mode immediately before the annihilation or after the creation.

Example 6.7. Consider $a_0(2)^\dagger a_0(1)^2$, i.e., annihilation of two identical bosons with momentum one followed by creation of a boson with momentum two. If the input state is

$$|\mathcal{F}\rangle = |(0, 1, 5)\rangle, \tag{6.89}$$

i.e., five bosons of momentum one and nothing else, then the output state is

$$|\mathcal{F}'\rangle = |(0,1,3),(0,2,1)\rangle, \tag{6.90}$$

i.e., three bosons of momentum one and one boson of momentum two. Hence $w'_1 = 1$ (since the created momentum-two boson is in its own mode), $w_2 = 4$, and $w_3 = 5$ (since the momentum-one mode has occupation 5 when the first boson is annihilated and occupation 4 when the second boson is annihilated). Thus for this example, (6.86) becomes

$$\langle \mathcal{F}'|a(2)^\dagger a(1)^2|\mathcal{F}\rangle = \sqrt{w'_1 w_2 w_3} = \sqrt{20}. \tag{6.91}$$

Similarly, the value of the parity factor ± 1 in (6.86) is the product of the parity factors due to the ladder operators at the times when they are applied. The coefficient β is a function of the $\mathbf{n}_i$, so the complete value of the matrix element is

$$\pm\beta(\{\mathbf{n}_i\})\sqrt{\left(\prod_{i=1}^{g} w'_i\right)\left(\prod_{i=g+1}^{f} w_i\right)}. \tag{6.92}$$

Recall that this is all assuming that $|\mathcal{F}'\rangle$ is connected to $|\mathcal{F}\rangle$ by the interaction, and that $\{\mathbf{n}_i\}$ is the corresponding assignment of momenta to the external lines in the interaction. But as pointed out in the first paragraph of this section, we may assume that we only have to evaluate the matrix element for pairs of states that are the output of the location oracle, and hence are connected. Therefore, computing the matrix element of $|\mathcal{F}\rangle, |\mathcal{F}'\rangle$ requires two steps:

1. Find the momenta of the extra particles in each state, the occupations of the corresponding modes (accounting for the case when multiple bosons in the same mode are created or annihilated), and the parities of the preceding modes for particles of the same type (for fermions and antifermions).

2. Evaluate (6.92).

When we apply the location oracle to determine $|\mathcal{F}'_i\rangle$ given $|\mathcal{F}\rangle$ and $i\rangle$, we

obtain the first step above along the way. In particular, the set of indices J_i (6.77) identifies the set of extra particles in $|\mathcal{F}\rangle$, and $A(\mathbf{Q}, i)$ is the set of momenta of the extra particles in $|\mathcal{F}_i'\rangle$. The occupations of the corresponding modes in $|\mathcal{F}_i'\rangle$ are identified when the new particles are inserted to construct $|\mathcal{F}_i'\rangle$. The parities for fermions and antifermions can be obtained by simply counting the numbers of preceding modes with the same particle type (the particle types are defined by the interaction), since the positions of the modes that the ladder operators act on are specified explicitly by J_i (for the incoming particles), and in the course of inserting the outgoing particles. Therefore, by the time we have obtained $|\mathcal{F}_i'\rangle$ in the course of implementing O_L, we can also complete step 1 of implementing O_V above. We can execute O_V as many times as desired by implementing step 2 above, as long as we do so prior to uncomputing the ancillas used to compute O_L.

In order to implement the quantum walk operator T (see (6.54)), we require two applications of O_V, one to compute the matrix element, and another to uncompute the matrix element after performing a rotation controlled on it (see step 3 in the proof of Lemma 6.1, above). There is no problem in putting off uncomputing the ancillas used in the computation of O_L until after the controlled rotation has been executed. Thus, we can perform both applications of O_V simply by executing step 2 above, with the inputs given by these ancillas. In other words, we can include all necessary applications of O_V in our implementation of O_L, without needing to recompute the extra particles in each state $|\mathcal{F}\rangle, |\mathcal{F}'\rangle$.

The implementation of step 2 above, i.e., the actual evaluation of the matrix element as in (6.92), depends on the specific functional form of $\beta(\{\mathbf{n}_i\})$. However, we can make some general statements. The matrix element expression (6.92) is a function of $2f$ variables, $\{\mathbf{n}_i\}$ and w_i', w_i. Each of the $\mathbf{n}_i$ is a d-dimensional vector whose entries are constrained by the cutoffs (6.13), so if $\Lambda_{\max}$ is the maximum magnitude of any cutoff, $\mathbf{n}_i$ takes $O(\Lambda_{\max}^d)$ values and is encoded in $O(d \log \Lambda_{\max})$ qubits for each i. Each of the w_i and w_i' is a positive integer upper bounded by W, where W is the occupation number cutoff, so each can be encoded in $O(\log W)$ qubits. Elementary arithmetic operations can be implemented as sequences of NOT,

CNOT, and Toffoli gates with depth polynomial in the number of qubits of the inputs [VBE96, JPK+14].

Thus, assuming that the matrix element can be expressed as a fixed combination of elementary arithmetic operations, evaluating it requires

$$O\big(d\, f\, \mathrm{polylog}(\Lambda_{\max}) + f\, \mathrm{polylog}(W)\big) \tag{6.93}$$

NOT, CNOT, and Toffoli gates. In other words, for fixed interactions in fixed dimension, the entire O_V can be executed by using the ancilla values computed during implementation of O_L, with gate count overhead that is polylogarithmic in the momentum and occupation number cutoffs.

6.7 Example applications

In this section, we explain how to simulate several example models in 1+1D using the tools we have described above. For each model, we also provide a comparison of the number of gates required for equal-time versus light-front formulations of relativistic quantum field theory.

The gates we are counting are log-local operations, i.e., operations on constant numbers of registers encoding single modes. As noted above, we do not compile these operations all the way into primitive gates, because optimizing such compilation is an independent problem and is itself the subject of extensive study (see for example [JPK+14]), as well as being hardware-specific. The gate counts we provide are also only for implementing the location oracle and obtaining the inputs to the coefficient function, since as explained in Section 6.6, once these steps are complete computing the value of the matrix element requires a number of additional gates that is polylogarithmic in the momentum and occupation cutoffs (see (6.93)).

6.7.1 Free boson and fermion theory

We begin by examining free theories for bosons or fermions before moving on to interacting theories. The Hamiltonians we consider are linear combinations of number operators, and thus diagonal and 1-sparse. More general free fermion and boson Hamiltonians can be cast into this diagonal form. An oracle call only entails computing the diagonal matrix element given the initial state $|\mathcal{F}\rangle$. Clearly, applying sophisticated quantum simulation methods to free theories is overkill. However, we discuss these theories because they are the simplest examples, and because these terms occur in interacting theories where nontrivial methods are necessary.

The location oracle for a diagonal Hamiltonian simply copies any input Fock state to the output register:

$$O_L : |\mathcal{F}\rangle|0\rangle \to |\mathcal{F}\rangle|\mathcal{F}\rangle. \tag{6.94}$$

Thus it can be implemented by a single layer of CNOTs, one to copy the state of each qubit (in the computational basis).

In light-front quantization, the Hamiltonian for a free boson of mass m_B in 1+1D is

$$H = m_B^2 \sum_{n=1}^{K} \frac{1}{n} a_n^\dagger a_n, \tag{6.95}$$

where the different values of n are light-front momenta, and K is the total light-front momentum (harmonic resolution; see Section 6.3.2). The coefficient function for the Hamiltonian is therefore

$$\beta_{\{(0),(0)\}}(n) = \frac{m_B^2}{n}, \tag{6.96}$$

where '0' denotes boson. Thus in this case the operations required to compute the matrix element are just those to compute a reciprocal.

Recall that our interactions as in (6.35) (in Definition 6.1) are specified as a pair of lists $\{(q_1, ..., q_g), (q_{g+1}, ..., q_f)\}$, where $(q_1, ..., q_g)$ are the outgoing particles and $(q_{g+1}, ..., q_f)$ are the incoming particles: thus in the present example $\{(0), (0)\}$ means one incoming boson and one outgoing boson. The Hamiltonian can be rewrit-

ten in terms of (6.96) as

$$H = \sum_{n=1}^{K} \beta_{\{(0),(0)\}}(n) a_n^\dagger a_n. \tag{6.97}$$

The matrix element oracle for the free boson field Hamiltonian is

$$\begin{aligned} &O_V : |\mathcal{F}\rangle|\mathcal{F}\rangle \to |\mathcal{F}\rangle|\mathcal{F}\rangle \left| \sum_{n=1}^{K} \beta_{\{(0),(0)\}}(n) w_n \right\rangle, \\ &O_V : |\mathcal{F}\rangle|\mathcal{F}'\rangle \to |\mathcal{F}\rangle|\mathcal{F}'\rangle|0\rangle, \end{aligned} \tag{6.98}$$

where w_n is the occupation of the mode with light-front momentum n in $|\mathcal{F}\rangle$, and $|\mathcal{F}'\rangle \neq |\mathcal{F}\rangle$.

The Hamiltonian for the Dirac field in 1+1D light-front quantization is

$$H = m_F^2 \sum_{n=1}^{K} \frac{1}{n} (b_n^\dagger b_n + d_n^\dagger d_n), \tag{6.99}$$

where m_F is the fermion/antifermion mass. The coefficient function for each interaction is

$$\beta_{\{(1),(1)\}}(n) = \beta_{\{(2),(2)\}}(n) = \frac{m_F^2}{n}, \tag{6.100}$$

where '1' denotes fermion and '2' denotes antifermion. Rewriting the Hamiltonian in terms of these gives

$$H = \sum_{n=1}^{K} \left(\beta_{\{(1),(1)\}}(n) b_n^\dagger b_n + \beta_{\{(2),(2)\}}(n) d_n^\dagger d_n \right). \tag{6.101}$$

The matrix element oracles for the two interactions in the Dirac field Hamiltonian are thus identical to the matrix element oracle for the free boson field, replacing the coefficient functions and occupation numbers with those corresponding to fermions and antifermions for the first and second interactions, respectively.

In equal-time quantization, the free Hamiltonian in second-quantized form looks similar to that in light-front quantization. The only difference is that the sum

runs over positive and negative momenta, and the coefficient function is given by

$$\beta_{\{(i),(i)\}}(n) = \frac{1}{\sqrt{m^2+n^2}} = \frac{1}{\omega_n}, \tag{6.102}$$

where $i = 0$, 1, or 2 for the boson, fermion, or antifermion interactions, and m is the mass of the particle (scaled by the box size L). Thus in this case the operations required to compute the matrix element are a square, a sum, a square-root, and a reciprocal.

6.7.2 $\lambda\phi^4$ theory

In light-front quantization, the $\lambda\phi^4$ theory in 1+1D has the Hamiltonian [HV87]

$$H = H_0 + H_I, \tag{6.103}$$

where

$$H_0 = \sum_n \frac{1}{n} a_n^\dagger a_n \left(m^2 + \frac{\lambda}{4\pi} \frac{1}{2} \sum_k \frac{1}{k} \right) \tag{6.104}$$

and

$$H_I = \frac{1}{4}\frac{\lambda}{4\pi} \sum_{klmn} \frac{a_k^\dagger a_l^\dagger a_m a_n}{\sqrt{klmn}} \delta_{m+n,k+l} + \frac{1}{6}\frac{\lambda}{4\pi} \sum_{klmn} \left(\frac{a_k^\dagger a_l a_m a_n + a_k^\dagger a_l^\dagger a_m^\dagger a_n}{\sqrt{klmn}} \right) \delta_{k,m+n+l}, \tag{6.105}$$

where λ is the coupling constant, and H_0, H_I are the free and interacting parts of the Hamiltonian, respectively. The sums are over light-front momenta in the range $[1, K]$. We can treat the free part of the Hamiltonian by the methods of Section 6.7.1. In this section we focus on the interacting part of the Hamiltonian, given in (6.105).

H_I is composed of three interactions, corresponding to the ladder operator monomials $a_k^\dagger a_l a_m a_n$, $a_k^\dagger a_l^\dagger a_m a_n$, and $a_l^\dagger a_m^\dagger a_n^\dagger a_k$ (summed over the momenta). In our interaction notation as in Definition 6.1, these are written $\{(0),(0,0,0)\}$, $\{(0,0),(0,0)\}$, and $\{(0,0,0),(0)\}$, respectively. Reading off from (6.105), the coeffi-

cient functions are given by

$$\beta_{\{(0,0),(0,0)\}}(k,l,m,n) = \frac{\lambda}{16\pi\sqrt{klmn}}, \tag{6.106}$$

$$\beta_{\{(0),(0,0,0)\}}(k,l,m,n) = \frac{\lambda}{24\pi\sqrt{klmn}}, \tag{6.107}$$

and

$$\beta_{\{(0,0,0),(0)\}}(k,l,m,n) = \frac{\lambda}{24\pi\sqrt{klmn}}. \tag{6.108}$$

Note that the delta functions that enforce momentum conservation are not included in the coefficient functions, because momentum conservation is enforced at an earlier step in the algorithm than computation of matrix elements. These are the entirety of the inputs needed to specify our oracle implementations. Rewriting the Hamiltonian in terms of the coefficient functions gives

$$\begin{aligned} H_I = &\sum_{klmn} \beta_{\{(0,0),(0,0)\}}(k,l,m,n) a_k^\dagger a_l^\dagger a_m a_n \\ &+ \sum_{klmn} \beta_{\{(0,0,0),(0)\}}(k,l,m,n) a_k^\dagger a_l a_m a_n \\ &+ \sum_{klmn} \beta_{\{(0),(0,0,0)\}}(k,l,m,n) a_k^\dagger a_l^\dagger a_m^\dagger a_n, \end{aligned} \tag{6.109}$$

where the sums run over momentum-conserving combinations of $k,l,m,n \in \{1,2,...,K\}$ for total light-front momentum K.

In equal-time quantization, the interacting part of the $\lambda\phi^4$ Hamiltonian in 1+1D is

$$\begin{aligned} H_I = \frac{\lambda}{4!} \sum_{k,l,p,f} \frac{1}{\sqrt{16\omega_p\omega_l\omega_k\omega_f}} \Big[& a_p a_k a_l a_f \delta_{-f-l,k+p} + a_p^\dagger a_k^\dagger a_l^\dagger a_f^\dagger \delta_{l+f,-k-p} \\ & + 4a_f^\dagger a_p a_k a_l \delta_{f,k+l+p} + 6a_k a_l \delta_{f,p}\delta_{k,-l} \\ & + 6a_l^\dagger a_f^\dagger a_p a_k \delta_{l+f,k+p} + 6a_k^\dagger a_l^\dagger \delta_{f,p}\delta_{k,-l} \\ & + 4a_k^\dagger a_l^\dagger a_f^\dagger a_p \delta_{l+f+k,p} \Big]. \end{aligned} \tag{6.110}$$

From this, we can read off the coefficient functions:

$$\begin{aligned}
\beta_{\{(0,0,0,0),()\}}(k,l,f,p) &= \frac{1}{\sqrt{16\omega_p\omega_l\omega_k\omega_f}},\\
\beta_{\{(),(0,0,0,0)\}}(k,l,f,p) &= \frac{1}{\sqrt{16\omega_p\omega_l\omega_k\omega_f}},\\
\beta_{\{(0,0,0),(0)\}}(k,l,f,p) &= \frac{4}{\sqrt{16\omega_p\omega_l\omega_k\omega_f}},\\
\beta_{\{(0,0),()\}}(k,l) &= \sum_{f,p}\frac{6}{\sqrt{16\omega_p\omega_l\omega_k\omega_f}}\delta_{f,p},\\
\beta_{\{(0,0),(0,0)\}}(k,l,f,p) &= \frac{6}{\sqrt{16\omega_p\omega_l\omega_k\omega_f}},\\
\beta_{\{(),(0,0)\}}(k,l) &= \sum_{f,p}\frac{6}{\sqrt{16\omega_p\omega_l\omega_k\omega_f}}\delta_{f,p},\\
\beta_{\{(0),(0,0,0)\}}(k,l,f,p) &= \frac{4}{\sqrt{16\omega_p\omega_l\omega_k\omega_f}}.
\end{aligned} \tag{6.111}$$

Note that the coefficient functions for the interactions having only two external lines ($\beta_{\{(),(0,0)\}}$ and $\beta_{\{(0,0),()\}}$) are only functions of the momenta of those lines (k and l). For the sake of brevity, we will omit rewriting the Hamiltonian explicitly in terms of the coefficient functions going forward, as we hope this correspondence has been made clear from the examples above. As before, the coefficient functions together with their associated interactions are the inputs required to define the Hamiltonian oracles.

The log-local gate counts for an explicit implementation of the oracles for $\lambda\phi^4$ in both light-front and equal-time quantization are given in Fig. 6.3. The gate counts are given as a function of K. The equal-time counts are obtained by choosing single-particle momentum cutoffs $[-\Lambda,\Lambda]$ for $\Lambda = \lceil K/2\rceil - 1$, so that the total number of lattice points in the equal-time simulation would be

$$2\Lambda + 1 = 2\lceil K/2\rceil - 1 = \begin{cases} K-1 & \text{for even } K,\\ K & \text{for odd } K, \end{cases} \tag{6.112}$$

for a fair comparison to K lattice points in the light-front simulation. For the equal-time counts, we also have to impose a cutoff on the number I of distinct occupied

modes (which is *a priori* arbitrary), which we choose to be K, again to provide a conservative comparison to the light-front simulation (in which the number of distinct occupied modes is in fact smaller still at $O(\sqrt{K})$ [KKG+22]).

The number of log-local operations to implement the location oracle is given by (6.84): $O\left(I^{f-g} + \Lambda_{\max}^{dg}\right)$. Since the matrix-element oracle requires only logarithmically many additional gates, implementing it does not change the asymptotic scaling, as noted above. Since $I = K$ for equal-time, this number of log-local operations becomes $O(K^4)$ due to the interactions that have four incoming particles ($f - g = 4$) in (6.110). For light-front, $I = \sqrt{K}$ so the term I^{f-g} no longer dominates, but the second term gives $O(K^3)$ due to the interactions that have three outgoing particles ($g = 3$) in (6.105) (since $\Lambda_{\max} = K$ in this case).

These costs are indeed what we see in the log-log plot Fig. 6.3, which shows the exact log-local gate counts for light-front and equal-time. To illustrate the asymptotic behaviors, Fig. 6.3 also plots K^3 and $K^4/5$. Since we expect the light-front cost to be $O(K^3)$ and the equal-time cost to be $O(K^4)$, as discussed above, the slopes of the data should approach those of the plotted lines on the log-log plot, which is what we see.

The extra operations required to compute the matrix elements for light-front quantization are those required to evaluate (6.106), (6.107), and (6.108), namely products, square-roots, and reciprocals. The extra operations required to compute the matrix elements for light-front quantization are those required to evaluate (6.111) for the frequencies ω_n defined as in (6.102). Hence they require squares, sums, products, square-roots, and reciprocals.

6.7.3 Massive Yukawa model

For the massive Yukawa model in 1+1D light-front quantization, we only write the interaction Hamiltonian [PB85]. This is composed of the so-called *vertex*, *seagull*, and *fork* terms:

$$H_I = H_V + H_S + H_F. \tag{6.113}$$

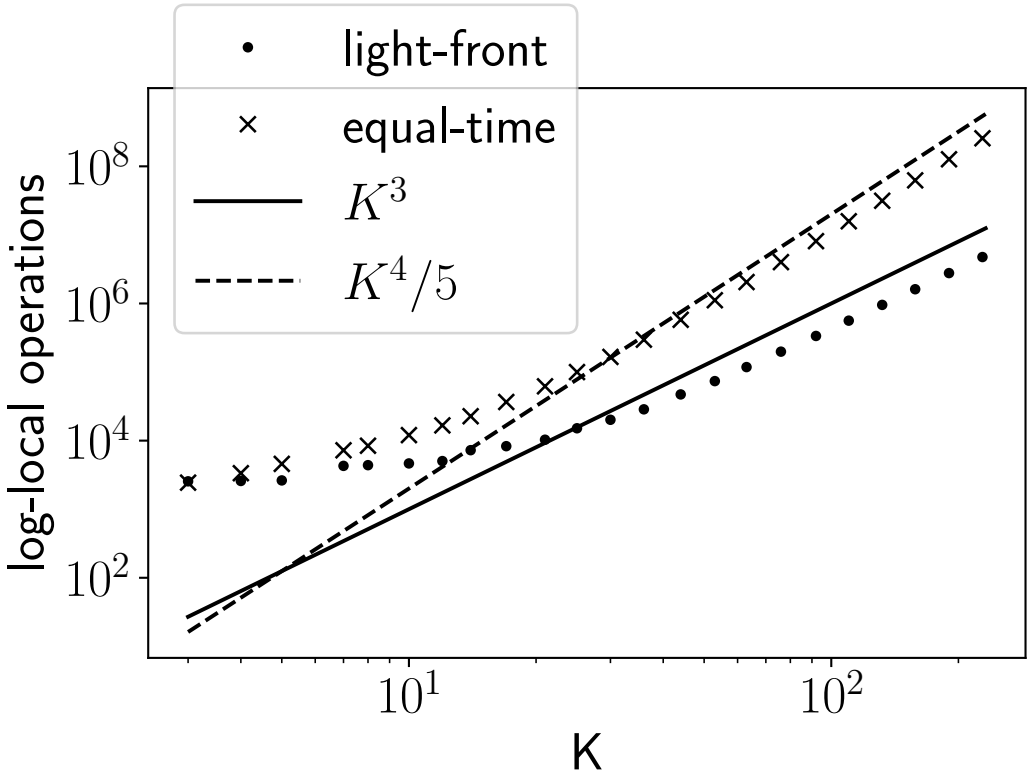


Figure 6.3: Gate counts (in log-local operations) to implement oracles for $\lambda\phi^4$ theory in 1+1D. The equal-time cutoffs are $[-\Lambda, \Lambda]$, for Λ defined in terms of K by (6.112). The exact gate counts for light-front and equal-time quantization are given by the points and crosses, respectively. The solid line K^3 and dashed line $K^4/5$ are included to illustrate that the datapoints are indeed converging to their expected asymptotic scalings of $O(K^3)$ for light-front and $O(K^4)$ for equal-time. Python code to generate the data for this plot is available on GitHub at `https://github.com/wmkirby1/quspice/blob/main/Plots_for_2105.10941.ipynb`.

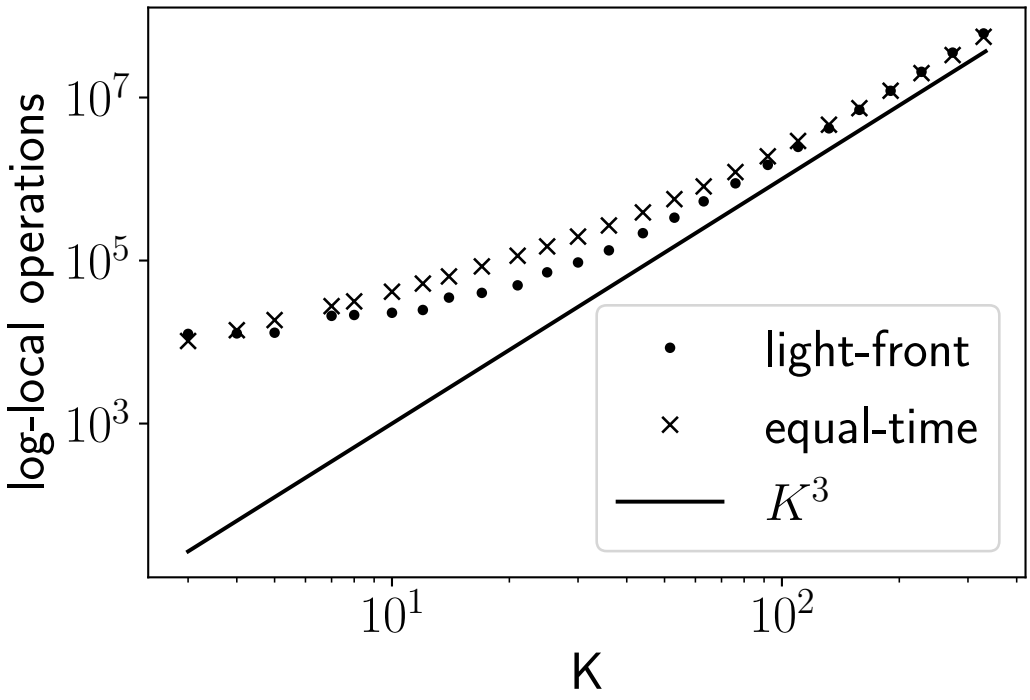


Figure 6.4: Gate counts (in log-local operations) to implement oracles for the massive Yukawa model. The equal-time cutoffs are $[-\Lambda, \Lambda]$, for Λ defined in terms of K by (6.112). The exact gate counts for light-front and equal-time quantization are given by the points and crosses, respectively. The line K^3 is included to illustrate that the datapoints are indeed converging to their expected asymptotic scaling of $O(K^3)$. Python code to generate the data for this plot is available on GitHub at `https://github.com/wmkirby1/quspice/blob/main/Plots_for_2105.10941.ipynb`.

The first term, H_V, is

$$H_V = gm_F \sum_{k,l,m} \Bigg[\frac{2}{(k+l)\sqrt{l}} (b_k^\dagger b_m a_l^\dagger + b_m^\dagger b_k a_l) + \frac{2}{(k+l)\sqrt{l}} (d_k^\dagger d_m a_l^\dagger + d_m^\dagger d_k a_l) + \frac{2}{(k-m)\sqrt{m}} (b_k d_l a_m^\dagger + d_l^\dagger b_k^\dagger a_m) \Bigg] \delta_{k+l,m}, \tag{6.114}$$

where the $a^{(\dagger)}$ are boson ladder operators, $b^{(\dagger)}$ are fermion ladder operators, and $d^{(\dagger)}$ are antifermion ladder operators. The resulting coefficient functions for H_V are

$$\begin{aligned} &\beta_{\{(1),(1,0)\}}(k,l,m) = \beta_{\{(1,0),(1)\}}(k,l,m) \\ &= \beta_{\{(2,2),(0)\}}(k,l,m) = \beta_{\{(2,0),(2)\}}(k,l,m) = \frac{2gm_F}{(k+l)\sqrt{l}}, \end{aligned} \tag{6.115}$$

$$\beta_{\{(1,2),(0)\}}(k,l,m) = \beta_{\{(0),(1,2)\}}(k,l,m) = \frac{2gm_F}{(k-m)\sqrt{m}}, \tag{6.116}$$

where in the subscripts, '0' denotes boson, '1' denotes fermion, and '2' denotes antifermion.

The second term in (6.113), H_S, is

$$H_S = g^2 \sum_{k,l,m,n} \Bigg[\frac{1}{m-k} (d_k b_l a_m^\dagger a_n^\dagger + b_l^\dagger d_k^\dagger a_n a_m) + \frac{2}{k-n} b_k^\dagger b_l a_m^\dagger a_n + \frac{2}{k-n} d_k^\dagger d_l a_m^\dagger a_n \Bigg] \frac{\delta_{k+l,m+n}}{\sqrt{mn}}, \tag{6.117}$$

resulting in the following coefficient functions:

$$\beta_{\{(1,0),(1,0)\}}(k,l,m,n) = \beta_{\{(2,1),(2,1)\}}(k,l,m,n) = \frac{2g^2}{(k-n)\sqrt{mn}}, \tag{6.118}$$

$$\beta_{\{(2,1),(0,0)\}}(k,l,m,n) = \beta_{\{(0,0),(2,1)\}}(k,l,m,n) = \frac{g^2}{(m-k)\sqrt{mn}}. \tag{6.119}$$

The third and final term in (6.113), H_F, is

$$H_F = g^2 \sum_{k,l,m,n} \Bigg[\frac{1}{(k+l)\sqrt{lm}} (b_k^\dagger b_n a_l^\dagger a_m^\dagger + b_n^\dagger b_k a_m a_l) + \frac{1}{(k+l)\sqrt{lm}} (d_k^\dagger d_n a_l^\dagger a_m^\dagger + d_n^\dagger d_k a_m a_l) + \frac{2}{(k-n)\sqrt{ln}} b_k^\dagger d_m^\dagger a_l^\dagger a_n + \frac{2}{(k-n)\sqrt{ln}} d_m b_k a_n^\dagger a_l \Bigg] \delta_{k+l+m,n}, \tag{6.120}$$

resulting in the following coefficient functions:

$$\begin{aligned} &\beta_{\{(1)(0,0,1)\}}(k,l,m,n) = \beta_{\{(0,0,1),(1)\}}(k,l,m,n) \\ &= \beta_{\{(2)(0,0,2)\}}(k,l,m,n) = \beta_{\{(0,0,2),(2)\}}(k,l,m,n) = \frac{g^2}{(k+l)\sqrt{lm}}, \end{aligned} \tag{6.121}$$

$$\beta_{\{(0),(0,1,2)\}}(k,l,m,n) = \beta_{\{(0,1,2),(0)\}}(k,l,m,n) = \frac{2g^2}{(k-n)\sqrt{ln}}. \tag{6.122}$$

In 1+1D equal-time quantization, the interacting part of the Yukawa Hamiltonian with free field expansion is:

$$\begin{aligned} H_I = \sum_{l,k,p} \frac{1}{\sqrt{2\omega_k}} \frac{1}{\sqrt{2\omega_p}} \frac{1}{\sqrt{2\omega_l}} \sum_{\gamma,s} \Big[& c_l^{s\dagger} c_k^\gamma a_p \bar{\mu}^s(l) \mu^\gamma(k) \delta_{l,k+p} + c_l^{s\dagger} d_k^{\gamma\dagger} a_p \bar{\mu}^s(l) \nu^\gamma(k) \delta_{l+k,p} \\ & + d_l^s c_k^\gamma a_p \bar{\nu}^s(l) \mu^\gamma(k) \delta_{l+k+p,0} - d_k^{\gamma\dagger} d_l^s a_p \bar{\nu}^s(l) \nu^\gamma(k) \delta_{k,l+p} \\ & + c_l^{s\dagger} c_k^\gamma a_p^\dagger \bar{\mu}^s(l) \mu^\gamma(k) \delta_{k,l+p} + c_l^{s\dagger} d_k^{\gamma\dagger} a_p^\dagger \bar{\mu}^s(l) \nu^\gamma(k) \delta_{l+k+p,0} \\ & + d_l^s c_k^\gamma a_p^\dagger \bar{\nu}^s(l) \mu^\gamma(k) \delta_{l+k,p} - d_k^{\gamma\dagger} d_l^s a_p^\dagger \bar{\nu}^s(l) \nu^\gamma(k) \delta_{l,k+p} \Big], \end{aligned} \tag{6.123}$$

where μ is the fermion spinor, ν is the antifermion spinor, and s and γ are the spin indices. This leads to the following coefficient functions:

$$\beta_{\{(0,1),(1\}}(k,l,p,s,\gamma) = \beta_{\{(1),(0,1)\}}(k,l,p,s,\gamma) = \frac{1}{\sqrt{8\omega_k\omega_p\omega_l}} \bar{\mu}^s(l) \mu^\gamma(k), \tag{6.124}$$

$$\beta_{\{(0),(1,2)\}}(k,l,p,s,\gamma) = \beta_{\{(),(0,1,2)\}}(k,l,p,s,\gamma) = \frac{1}{\sqrt{8\omega_k\omega_p\omega_l}} \bar{\mu}^s(l) \nu^\gamma(k), \tag{6.125}$$

$$\beta_{\{(1,2),(0)\}}(k,l,p,s,\gamma) = \beta_{\{(0,1,2),()\}}(k,l,p,s,\gamma) = \frac{1}{\sqrt{8\omega_k\omega_p\omega_l}}\bar{\nu}^s(l)\mu^\gamma(k), \quad (6.126)$$

$$\beta_{\{(0,2),(2)\}}(k,l,p,s,\gamma) = \beta_{\{(2),(0,2)\}}(k,l,p,s,\gamma) = \frac{1}{\sqrt{8\omega_k\omega_p\omega_l}}\bar{\nu}^s(l)\nu^\gamma(k). \quad (6.127)$$

Note that we have expanded our set of arguments of the coefficient functions to include the spin indices s, γ. If we instead wished to obtain interactions exactly as defined in Definition 6.1, we could let each of the above coefficient functions split into four functions, one for each of the pairs of values for the spin indices, but this would just become unwieldy, and there is no harm in including the spin indices as arguments.

The log-local gate counts for implementing the oracles for the Yukawa interaction in both light-front and equal-time quantization are given in Fig. 6.4, with the equal-time cutoff $\Lambda = \lceil K/2 \rceil - 1$ as for the ϕ^4 theory, above. Recall that (6.84) gives the scaling of the number of log-local operations required to implement the oracles. Since $\Lambda = O(K)$ as we just discussed, and $I < O(K)$ in both light-front and equal-time, from (6.84) we see that in both light-front and equal-time the most costly interactions to simulate are those with three outgoing particles. This gives a cost in log-local operations of $O(K^3)$, which is indeed what we see in Fig. 6.4.

6.8 Beyond the plane wave momentum basis

We have demonstrated how, given a second-quantized Hamiltonian in the plane wave momentum representation of a field theory, we can implement the oracle unitaries necessary to apply sparsity-based simulation methods. Our methods extend to any second-quantized Hamiltonian containing a fixed number of interactions, even if it is not expressed in the plane wave momentum basis. All that is required is that for each interaction, it is possible to efficiently enumerate all possible sets of outgoing particles given a particular set of incoming particles, and to efficiently compute the matrix element given the incoming and outgoing particles. In the plane wave momentum representation, we used momentum conservation for the former task: given a set of incoming particles, we can add up their momenta to

obtain the total transferred momentum, and then enumerate all possible allocations of this momentum amongst the outgoing particles.

However, more generally the sets of outgoing particles can always be enumerated in polynomial time as long as the number of outgoing particles is fixed and there are only polynomially-many possible states for each particle. Here polynomial means polynomial in whatever problem parameter(s) governs the asymptotic scaling, which might no longer be the momentum cutoffs for a problem not in the plane wave momentum basis. If g is the number of outgoing particles and P is an upper bound on the number of states that each outgoing particle may take, then P^g is an upper bound on the number of distinct sets of outgoing particles from a particular set of incoming particles. We could apply this argument to the plane wave momentum basis case, and it would lead to an efficient algorithm, but with worse scaling than the one we presented above, since it would overcount the possible outgoing states. This illustrates that using momentum conservation at the level of enumerating outgoing states was really an additional constraint that we imposed in order to save resources, rather than an intrinsically necessary part of the algorithm.

Hence there is no problem with extending our algorithm to a non-momentum basis as long as it is possible to efficiently calculate the matrix element between two Fock states. If there is no conserved quantity that constrains the outgoing particles, then we can enumerate all of the possible sets of outgoing particles as described above. If there is a conserved quantity (or more than one), then just as for momentum conservation we can compute its value for the incoming particles and then only enumerate sets of outgoing particles that conserve it. But to reiterate, either of these approaches is efficient; choosing whether or not to exploit a conserved quantity simply changes the details of the scaling. Hence, although our main presentation focused on the plane wave momentum basis, we can apply our methods to a wide variety of theories expressed in other bases, in quantum chemistry, condensed matter physics, and quantum field theory, including basis light-front quantization [VHL$^+$10, KJK$^+$21a, KJK$^+$21b].

6.9 Conclusion

In this chapter, we presented implementations of the Hamiltonian oracles for second-quantized Hamiltonians of theories including bosons and fermions. We focused on the plane wave momentum basis, but the methods we described generalize to any second-quantized Hamiltonian as long as it only contains polynomially-many terms (monomials in the creation and annihilation operators), and as long as the coefficients of the terms can be computed efficiently. These oracle implementations are the necessary inputs to any of the large collection of quantum simulation techniques for sparse Hamiltonians [ATS03, CCD$^+$03, BACS07, Chi10, CK10, CK11, BC12, CW12, BCC$^+$14, BCK15, BCC$^+$15, LC17, LW18, LC19, BCS$^+$20]. The generality of our algorithms means that for some specific field theories, it is likely possible to develop algorithms that are tailored to the structure of the theories and outperform our methods (see Section 6.2.2 for examples). However, our goal was to provide a general-purpose tool, and thus to establish that for second-quantized Hamiltonians satisfying only the modest constraints stated above, efficient quantum simulation by optimal sparsity-based methods is possible.

Chapter 7

Conclusion

The fields of quantum algorithms and quantum information contain many disparate areas of study and computational techniques. On the quantum computing side in particular, since the field is still relatively young, many of the connections and cross-applications both from the more mature field of quantum information and between areas within quantum algorithms are underdeveloped. A common theme of all of the topics in this thesis is the project of building such connections and cross-applications.

We began by studying contextuality with an eye towards applying it to quantum algorithms. Contextuality, as discussed in the introduction and in detail in Chapters 2 and 3, is a feature of quantum mechanics that distinguishes it from classical mechanics, and thus distinguishes quantum computing from classical computing. It has been heavily studied in the abstract, from the perspective of quantum information theory, but its known practical applications to quantum computing are few. In the first three chapters of this thesis, we developed techniques for identifying contextuality in the measurement schemes that appear in variational quantum eigensolvers, classically modeling those schemes when contextuality is not present, and finally using contextuality to separate a VQE instance into classical and non-classical parts, with only the latter being simulated on the quantum computer. This is a natural progression whose general form we believe could be applicable in other scenarios, including finding other applications of contextuality itself in quantum

algorithms: identify an appropriate quantum algorithm, analyze it from the perspective of a foundational concept in quantum theory, and extend or modify the algorithm to better exploit the structure illuminated by that analysis.

In Chapter 5, we again focused on the variational quantum eigensolver algorithm, but this time with the goal of expanding its domain of applicability. In particular, as discussed in Section 5.1, the linear combination of Pauli operators input model that prior to our work had been the only input model used in VQE can be limiting, especially when applying VQE to quantum field theories. In Chapter 5 we showed how VQE can be extended to take as input sparse Hamiltonians, which are a much broader class. By writing these Hamiltonians in second-quantized form and using one of two possible sequences of decompositions we are able to express them as linear combinations of one-sparse, self-inverse, Hermitian terms whose expectation values can be estimated via Hadamard tests using queries to oracles that encode the original Hamiltonian. This extension of VQE is much too depth-intensive to implement on existing quantum computers, and most likely will require fault-tolerant quantum computers in order to be effective. However, even once fault tolerance is achieved, VQE will be one of the least expensive simulation algorithms we can run and therefore is likely to remain an application of early fault-tolerant quantum computers, so we may hope that in this setting the sparse VQE algorithm can prove useful.

Our contextuality test, and consequently the algorithms in Chapters 3 and 4, apply only to sets of Pauli operators or other sets of observables whose product and commutation relations are isomorphic to the Pauli operators (such as Majorana operators). This suggests future work on extending efficient testing of contextuality to more general sets of observables. A natural algorithmic motivation for this is sparse VQE, which ultimately decomposes sparse Hamiltonians into linear combinations of one-sparse, self-inverse observables that are measured on the quantum computer. Pauli operators are one-sparse and self-inverse as well as being subject to additional properties, so observables that are only required to be one-sparse and self-inverse provide a strict generalization that is nonetheless still restricted enough that it may

be tractable.

An initial challenge in doing this will be to appropriately generalize the notion of inference, since unlike for Pauli operators, whether or not an inference is possible given measurement of more general two-outcome observables can depend on the specific measurement outcomes instead of only on the global commutativity of the observables. Determining whether or not a given set of two-outcome observables is contextual is equivalent to determining whether or not a set of boolean equations is satisfiable, with one equation corresponding each inference relation. For the Pauli operators, we implicitly demonstrated by giving our efficiently testable criterion for contextuality in Chapter 2 that testing satisfiability of the corresponding sets of boolean equations is always efficient, but generic boolean satisfiability problems are NP-complete [Coo71, Lev73]. Hence, one should approach the problem of generalizing beyond Pauli operators with a measure of caution, since it could be that for some other class of operators of interest, testing for contextuality turns out to be NP-complete. However, the possibility of generalizing our criterion for contextuality to sparse Hamiltonians is tantalizing given the prospect of constructing a version of contextual subspace VQE for sparse Hamiltonians as well as applying analysis of contextuality to the rich field of sparse Hamiltonian dynamics simulation, so I still feel that it is a research direction worth pursuing.

Finally, in Chapter 6 we gave a gate-level implementation of the sparse Hamiltonian oracles for second-quantized Hamiltonians in a very general class, including all mixed bosonic and fermionic Hamiltonians in lattice field theories provided they contain only polynomially-many second-quantized terms. This chapter is the most purely methods-oriented in this thesis, but again bridges a gap between high-level, theoretical and asymptotic analyses of quantum simulation algorithms, which often are expressed simply in terms of asymptotic numbers of queries to the sparse Hamiltonian oracles, and low-level gate decompositions of quantum circuits. The resulting circuits are, partly because of their generality, the most depth-intensive of the algorithms discussed in this thesis, and will certainly not be applicable until the fault-tolerant era. However, by completing the link between the asymptotic analy-

ses that are usually the endpoints of quantum simulation papers aimed at that era and actual circuit-level constructions, our oracle implementations provide proof-of-principle of such explicit decompositions. They also set the stage for future work that can hopefully optimize such implementations for more specific physical models either directly or by utilizing general methods that adapt to take advantage of problem specific structure, paving the way to future high-accuracy quantum simulation of quantum field theory.

One final research direction that I believe is important is construction of improved block encodings. We mentioned block encoding in Section 1.3.2 as the required input model for qubitization [LC19], which provides optimal scaling for simulation of time-evolution, but it is also the required input model for my paper on a quantum Lanczos method [KMM22] (see Section 1.3.3), which provides efficient ground state energy approximation given an initial state with $1/\mathrm{poly}(n)$ overlap with the low-lying energy subspace. While block encodings can be implemented using the sparse Hamiltonian oracles, and hence our method in Chapter 6 can in principle be used as the input model for qubitization, doing this involves significant overhead even though it is asymptotically efficient. It would be desirable to do something like what we have done for the sparse oracles in Chapter 6, but for block encodings instead, i.e., to construct explicit block encodings for generic second-quantized Hamiltonians that would ideally be more efficient than those obtained from the sparse oracles. A few papers have presented explicit block encoding constructions [LC19, SW20, CLBY22, CB22], but I still feel that the area is under-researched, and there are probably useful results yet to be found. This would be a natural follow-up to the work presented in Chapter 6 [KHKL21].

Bibliography

[AA17] Yosi Atia and Dorit Aharonov. Fast-forwarding of hamiltonians and exponentially precise measurements. *Nature Communications*, 8(1):1572, 2017.

[AAB+20] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Sergio Boixo, Michael Broughton, Bob B. Buckley, David A. Buell, Brian Burkett, Nicholas Bushnell, Yu Chen, Zijun Chen, Benjamin Chiaro, Roberto Collins, William Courtney, Sean Demura, Andrew Dunsworth, Edward Farhi, Austin Fowler, Brooks Foxen, Craig Gidney, Marissa Giustina, Rob Graff, Steve Habegger, Matthew P. Harrigan, Alan Ho, Sabrina Hong, Trent Huang, William J. Huggins, Lev Ioffe, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Cody Jones, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Seon Kim, Paul V. Klimov, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Pavel Laptev, Mike Lindmark, Erik Lucero, Orion Martin, John M. Martinis, Jarrod R. McClean, Matt McEwen, Anthony Megrant, Xiao Mi, Masoud Mohseni, Wojciech Mruczkiewicz, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Hartmut Neven, Murphy Yuezhen Niu, Thomas E. O'Brien, Eric Ostby, Andre Petukhov, Harald Putterman, Chris Quintana, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Doug Strain, Kevin J. Sung, Marco Szalay, Tyler Y. Takeshita, Amit Vainsencher, Theodore White, Nathan

Wiebe, Z. Jamie Yao, Ping Yeh, and Adam Zalcman. Hartree-fock on a superconducting qubit quantum computer. *Science*, 369(6507):1084–1089, 2020.

[AB11] Samson Abramsky and Adam Brandenburger. The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics*, 13(11):113036, 2011.

[ABM17] Samson Abramsky, Rui Soares Barbosa, and Shane Mansfield. Contextual fraction as a measure of contextuality. *Phys. Rev. Lett.*, 119:050504, Aug 2017.

[AC17] Barbara Amaral and Marcelo Terra Cunha. On geometrical aspects of the graph approach to contextuality. *arXiv preprint*, 2017.

[AE11] Dorit Aharonov and Lior Eldar. On the complexity of commuting local hamiltonians, and tight conditions for topological order in such systems. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 334–343, 2011.

[AE15] Dorit Aharonov and Lior Eldar. The commuting local hamiltonian problem on locally expanding graphs is approximable in np. *Quantum Information Processing*, 14(1):83–101, Jan 2015.

[AFCB14] Mateus Araújo, Adrien Feix, Fabio Costa, and Časlav Brukner. Quantum circuits cannot control unknown operations. *New Journal of Physics*, 16(9):093026, sep 2014.

[AGDLHG05] Alán Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, 2005.

[AHH+22] Ismail Yunus Akhalwaya, Yang-Hui He, Lior Horesh, Vishnu Jejjala, William Kirby, Kugendran Naidoo, and Shashanka Ubaru. Represen-

tation of the fermionic boundary operator. *Phys. Rev. A*, 106:022407, Aug 2022.

[AM22] Anurag Anshu and Tony Metger. Concentration bounds for quantum states and limitations on the qaoa from polynomial approximations. *arXiv preprint, arXiv:2209.02715*, 2022.

[AQB+13] Mateus Araújo, Marco Túlio Quintino, Costantino Budroni, Marcelo Terra Cunha, and Adán Cabello. All noncontextuality inequalities for the n-cycle scenario. *Phys. Rev. A*, 88:022118, Aug 2013.

[ASYHL+20] David R. M. Arvidsson-Shukur, Nicole Yunger Halpern, Hugo V. Lepage, Aleksander A. Lasek, Crispin H. W. Barnes, and Seth Lloyd. Quantum advantage in postselected metrology. *Nature Communications*, 11(1):3775, 2020.

[ATS03] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, STOC '03, pages 20–29, New York, NY, USA, 2003. Association for Computing Machinery.

[BACS07] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. *Communications in Mathematical Physics*, 270(2):359–371, 2007.

[Bar82] F Barahona. On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241–3253, oct 1982.

[BBK+16] Ryan Babbush, Dominic W Berry, Ian D Kivlichan, Annie Y Wei, Peter J Love, and Alán Aspuru-Guzik. Exponentially more precise

quantum simulation of fermions in second quantization. *New Journal of Physics*, 18(3):033032, mar 2016.

[BBS+17] Ryan Babbush, Dominic W Berry, Yuval R Sanders, Ian D Kivlichan, Artur Scherer, Annie Y Wei, Peter J Love, and Alán Aspuru-Guzik. Exponentially more precise quantum simulation of fermions in the configuration interaction representation. *Quantum Science and Technology*, 3(1):015006, dec 2017.

[BC12] Dominic W. Berry and Andrew M. Childs. Black-box hamiltonian simulation and unitary implementation. *Quantum Information and Computation*, 12(1-2):29–62, January 2012.

[BCC+14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse hamiltonians. *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 283–292, 2014.

[BCC+15] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Phys. Rev. Lett.*, 114:090502, Mar 2015.

[BCG+21] Costantino Budroni, Adán Cabello, Otfried Gühne, Matthias Kleinmann, and Jan-Åke Larsson. Kochen-specker contextuality. *arXiv preprint, arXiv:2102.13036*, 2021.

[BCK15] D. W. Berry, A. M. Childs, and R. Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 792–809, Oct 2015.

[BCS+20] Dominic W. Berry, Andrew M. Childs, Yuan Su, Xin Wang, and Nathan Wiebe. Time-dependent Hamiltonian simulation with L^1-norm scaling. *Quantum*, 4:254, April 2020.

[Bel64] John S Bell. On the einstein podolsky rosen paradox. *Physics*, 1(3):195, 1964.

[Bel66] John S. Bell. On the problem of hidden variables in quantum mechanics. *Rev. Mod. Phys.*, 38:447–452, July 1966.

[BGK18] Sergey Bravyi, David Gosset, and Robert König. Quantum advantage with shallow circuits. *Science*, 362(6412):308–311, 2018.

[BGKT20] Sergey Bravyi, David Gosset, Robert König, and Marco Tomamichel. Quantum advantage with noisy shallow circuits. *Nature Physics*, 16(10):1040–1045, 2020.

[BGMT17] Sergey Bravyi, Jay M. Gambetta, Antonio Mezzacapo, and Kristan Temme. Tapering off qubits to simulate fermionic hamiltonians. *arXiv preprint, arXiv:1701.08213*, 2017.

[BGMZ22] Joao Basso, David Gamarnik, Song Mei, and Leo Zhou. Performance and limitations of the qaoa at constant levels on large sparse hypergraphs and spin glass models. *arXiv preprint, arXiv:2204.10306*, 2022.

[BHS+22] Unpil Baek, Diptarka Hait, James Shee, Oskar Leimkuhler, William J. Huggins, Torin F. Stetina, Martin Head-Gordon, and K. Birgitta Whaley. Say no to optimization: A non-orthogonal quantum eigensolver. *arXiv preprint, arXiv:2205.09039*, 2022.

[BK02] Sergey B. Bravyi and Alexei Yu. Kitaev. Fermionic quantum computation. *Annals of Physics*, 298(1):210–226, 2002.

[BL08] Jacob D. Biamonte and Peter J. Love. Realizable hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A*, 78:012352, Jul 2008.

[BMW+15] Ryan Babbush, Jarrod McClean, Dave Wecker, Alán Aspuru-Guzik, and Nathan Wiebe. Chemical basis of trotter-suzuki errors in quantum chemistry simulation. *Phys. Rev. A*, 91:022311, Feb 2015.

[BPP98] Stanley J. Brodsky, Hans-Christian Pauli, and Stephen S. Pinsky. Quantum chromodynamics and other field theories on the light cone. *Physics Reports*, 301(4):299–486, 1998.

[BRS12] Stephen D. Bartlett, Terry Rudolph, and Robert W. Spekkens. Reconstruction of gaussian quantum mechanics from liouville mechanics with an epistemic restriction. *Phys. Rev. A*, 86:012103, Jul 2012.

[BV05] Sergey Bravyi and Mikhail Vyalyi. Commutative version of the k-local hamiltonian problem and common eigenspace problem. *Quantum Information and Computation*, 5(3):187–215, 2005.

[BWM+18] Ryan Babbush, Nathan Wiebe, Jarrod McClean, James McClain, Hartmut Neven, and Garnet Kin-Lic Chan. Low-depth quantum simulation of materials. *Phys. Rev. X*, 8:011044, Mar 2018.

[CB22] Daan Camps and Roel Van Beeumen. Fable: Fast approximate quantum circuits for block-encodings. *arXiv preprint, arXiv:2205.00081*, 2022.

[CCD+03] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 59–68, New York, NY, USA, 2003. Association for Computing Machinery.

[CCY93] A. Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 352–361, 1993.

[CG22] Cristian L. Cortes and Stephen K. Gray. Quantum krylov subspace algorithms for ground- and excited-state energy estimation. *Phys. Rev. A*, 105:022417, Feb 2022.

[Chi10] Andrew M. Childs. On the relationship between continuous- and discrete-time quantum walk. *Communications in Mathematical Physics*, 294(2):581–603, 2010.

[CHSH69] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, Oct 1969.

[CK10] Andrew M Childs and Robin Kothari. Limitations on the simulation of non-sparse hamiltonians. *Quantum Information & Computation*, 10(7):669–684, 2010.

[CK11] Andrew M. Childs and Robin Kothari. Simulating sparse hamiltonians with star decompositions. In Wim van Dam, Vivien M. Kendon, and Simone Severini, editors, *Theory of Quantum Computation, Communication, and Cryptography*, pages 94–103, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[CK22] Riley W. Chien and Joel Klassen. Optimizing fermionic encodings for both hamiltonian and hardware. *arXiv preprint, arXiv:2210.05652*, 2022.

[CKB15] Adán Cabello, Matthias Kleinmann, and Costantino Budroni. Necessary and sufficient condition for quantum state-independent contextuality. *Phys. Rev. Lett.*, 114:250402, Jun 2015.

[CKH21] Yi-Hsiang Chen, Amir Kalev, and Itay Hen. Quantum algorithm for time-dependent hamiltonian simulation by permutation expansion. *PRX Quantum*, 2:030342, Sep 2021.

[CLBY22] Daan Camps, Lin Lin, Roel Van Beeumen, and Chao Yang. Explicit quantum circuits for block encodings of certain sparse matrices. *arXiv preprint, arXiv:2203.10236*, 2022.

[CLSS21] Chi-Ning Chou, Peter J. Love, Juspreet Singh Sandhu, and Jonathan Shi. Limitations of local quantum algorithms on random max-k-xor and beyond. *arXiv preprint, arXiv:2108.06049*, 2021.

[CMGB09] Craig R. Clark, Tzvetan S. Metodi, Samuel D. Gasster, and Kenneth R. Brown. Resource requirements for fault-tolerant quantum simulation: The ground state of the transverse ising model. *Phys. Rev. A*, 79:062314, Jun 2009.

[CMN+18] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018.

[CMP21] Jeffrey Cohn, Mario Motta, and Robert M. Parrish. Quantum filter diagonalization with compressed double-factorized hamiltonians. *PRX Quantum*, 2:040352, Dec 2021.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, page 151–158, New York, NY, USA, 1971. Association for Computing Machinery.

[COS19] Andrew M. Childs, Aaron Ostrander, and Yuan Su. Faster quantum simulation by randomization. *Quantum*, 3:182, September 2019.

[CRD+18] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi. Computation of molecular spectra on a quantum processor with an error-resilient algorithm. *Phys. Rev. X*, 8:011021, Feb 2018.

[CS19] Andrew M. Childs and Yuan Su. Nearly optimal lattice simulation by product formulas. *Phys. Rev. Lett.*, 123:050503, Aug 2019.

[CST+21] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Phys. Rev. X*, 11:011020, Feb 2021.

[CSV+21] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1):1791, 2021.

[CSW14] Adán Cabello, Simone Severini, and Andreas Winter. Graph-theoretic approach to quantum correlations. *Phys. Rev. Lett.*, 112:040401, Jan 2014.

[CW12] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Information and Computation*, 12(11-12):901–924, 2012.

[DA18] Cristhiano Duarte and Barbara Amaral. Resource theory of contextuality for arbitrary prepare-and-measure experiments. *Journal of Mathematical Physics*, 59(6):062202, 2018.

[DJ92] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992.

[DK21] Charles Derby and Joel Klassen. A compact fermion to qubit mapping part 2: Alternative lattice geometries. *arXiv preprint, arXiv:2101.10735*, 2021.

[DKBC21] Charles Derby, Joel Klassen, Johannes Bausch, and Toby Cubitt. Compact fermion to qubit mappings. *Phys. Rev. B*, 104:035118, Jul 2021.

[DLT22] Yulong Dong, Lin Lin, and Yu Tong. Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices. *PRX Quantum*, 3:040305, Oct 2022.

[DMH+18] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski. Cloud quantum computing of an atomic nucleus. *Phys. Rev. Lett.*, 120:210501, 2018.

[DN05] Christopher M. Dawson and Michael A. Nielsen. The solovay-kitaev algorithm. *Quantum Information and Computation*, 0(0):1–15, 2005.

[DP85] David Deutsch and Roger Penrose. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

[dS17] Nadish de Silva. Graph-theoretic strengths of contextuality. *Phys. Rev. A*, 95:032108, Mar 2017.

[DXP+10] Jiangfeng Du, Nanyang Xu, Xinhua Peng, Pengfei Wang, Sanfeng Wu, and Dawei Lu. Nmr implementation of a molecular hydrogen quantum simulation with adiabatic state preparation. *Phys. Rev. Lett.*, 104:030502, 2010.

[EFH+22] Andreas Elben, Steven T. Flammia, Hsin-Yuan Huang, Richard Kueng, John Preskill, Benoît Vermersch, and Peter Zoller. The randomized measurement toolbox. *arXiv preprint, arXiv:2203.11374*, 2022.

[ELN21] Ethan N. Epperly, Lin Lin, and Yuji Nakatsukasa. A theory of quantum subspace diagonalization. *arXiv preprint, arXiv:2110.07492*, 2021.

[FDDB14] Nicolai Friis, Vedran Dunjko, Wolfgang Dür, and Hans J. Briegel. Implementing quantum control for unknown subroutines. *Phys. Rev. A*, 89:030303, Mar 2014.

[FE08] Christopher Ferrie and Joseph Emerson. Frame representations of quantum mechanics and the necessity of negativity in quasi-probability representations. *Journal of Physics A: Mathematical and Theoretical*, 41(35):352001, jul 2008.

[FE09] Christopher Ferrie and Joseph Emerson. Framed hilbert space: hanging the quasi-probability pictures of quantum theory. *New Journal of Physics*, 11(6):063040, jun 2009.

[Fer11] Christopher Ferrie. Quasi-probability representations of quantum theory with applications to quantum information science. *Reports on Progress in Physics*, 74(11):116001, oct 2011.

[Fey82] Richard P. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21(6):467–488, 1982.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint, arXiv:1411.4028*, 2014.

[FGGS00] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint, arXiv:quant-ph/0001106*, 2000.

[Fin82] Arthur Fine. Hidden variables, joint probability, and the bell inequalities. *Phys. Rev. Lett.*, 48:291–295, Feb 1982.

[FRB18] Markus Frembs, Sam Roberts, and Stephen D Bartlett. Contextuality as a resource for measurement-based quantum computation beyond qubits. *New Journal of Physics*, 20(10):103011, oct 2018.

[GaB20] Ernesto F. Galvão and Daniel J. Brod. Quantum and classical bounds for two-state overlaps. *Phys. Rev. A*, 101:062110, Jun 2020.

[GAD+20] Pranav Gokhale, Olivia Angiuli, Yongshan Ding, Kaiwen Gui, Teague Tomesh, Martin Suchara, Margaret Martonosi, and Frederic T. Chong. $o(n^3)$ measurement cost for variational quantum eigensolver on molecular hamiltonians. *IEEE Transactions on Quantum Engineering*, 1:1–24, 2020.

[GEB+19] Marc Ganzhorn, Daniel J Egger, Panagiotis Barkoutsos, Pauline Ollitrault, Gian Salis, Nikolaj Moll, Marco Roth, A Fuhrer, P Mueller, Stefan Woerner, et al. Gate-efficient simulation of molecular eigenstates on a quantum computer. *Physical Review Applied*, 11(4):044092, 2019.

[GHH+14] A. Grudka, K. Horodecki, M. Horodecki, P. Horodecki, R. Horodecki, P. Joshi, W. Kłobus, and A. Wójcik. Quantifying contextuality. *Phys. Rev. Lett.*, 112:120401, Mar 2014.

[GHW04] Kathleen S. Gibbons, Matthew J. Hoffman, and William K. Wootters. Discrete phase space based on finite fields. *Phys. Rev. A*, 70:062101, Dec 2004.

[GL00] Daniel Gottesman and Hoi-Kwong Lo. From quantum cheating to quantum security. *Physics Today*, 53(11):22–27, 2000.

[Got97] Daniel Gottesman. Stabilizer codes and quantum error correction. *arXiv preprint, arXiv:quant-ph/9705052*, 1997.

[Got98] Daniel Gottesman. The heisenberg representation of quantum computers. *arXiv preprint, arXiv:quant-ph/9807006*, 1998.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. Association for Computing Machinery.

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 193–204, New York, NY, USA, 2019. Association for Computing Machinery.

[GST20] Zuzana Gavorová, Matan Seidel, and Yonathan Touati. Topological obstructions to implementing quantum if-clause. *arXiv preprint, arXiv:2011.10031*, 2020.

[Had21] Charles Hadfield. Adaptive pauli shadows for energy estimation. *arXiv preprint, arXiv:2105.12207*, 2021.

[HBB+09] B L Higgins, D W Berry, S D Bartlett, M W Mitchell, H M Wiseman, and G J Pryde. Demonstrating heisenberg-limited unambiguous phase estimation without adaptive measurements. *New Journal of Physics*, 11(7):073023, jul 2009.

[HBRM22] Charles Hadfield, Sergey Bravyi, Rudy Raymond, and Antonio Mezzacapo. Measurements of quantum hamiltonians with locally-biased classical shadows. *Communications in Mathematical Physics*, 391(3):951–967, 2022.

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.

[HHR+21] Stefan Hillmich, Charles Hadfield, Rudy Raymond, Antonio Mezzacapo, and Robert Wille. Decision diagrams for quantum measurements with shallow circuits. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 24–34, 2021.

[HKP20] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020.

[HKP21] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Efficient estimation of pauli observables by derandomization. *Phys. Rev. Lett.*, 127:030503, Jul 2021.

[HLB+20] William J Huggins, Joonho Lee, Unpil Baek, Bryan O'Gorman, and K Birgitta Whaley. A non-orthogonal variational quantum eigensolver. *New Journal of Physics*, 22(7):073009, jul 2020.

[HMR+18] Cornelius Hempel, Christine Maier, Jonathan Romero, Jarrod McClean, Thomas Monz, Heng Shen, Petar Jurcevic, Ben P. Lanyon, Peter Love, Ryan Babbush, Alán Aspuru-Guzik, Rainer Blatt, and Christian F. Roos. Quantum chemistry calculations on a trapped-ion quantum simulator. *Phys. Rev. X*, 8:031022, Jul 2018.

[HP18] Stuart Hadfield and Anargyros Papageorgiou. Divide and conquer approach to quantum hamiltonian simulation. *New Journal of Physics*, 20(4):043003, apr 2018.

[HR90] J Huyghebaert and H De Raedt. Product formula methods for time-dependent schrodinger problems. *Journal of Physics A: Mathematical and General*, 23(24):5777, dec 1990.

[HV87] A. Harindranath and J. P. Vary. Solving two-dimensional φ^4 theory by discretized light-front quantization. *Phys. Rev. D*, 36:1141–1147, Aug 1987.

[HWVE14] Mark Howard, Joel Wallman, Victor Veitch, and Joseph Emerson. Contextuality supplies the 'magic' for quantum computation. *Nature*, 510:351 EP –, 06 2014.

[HZS+22] Karol Horodecki, Jingfang Zhou, Maciej Stankiewicz, Roberto Salazar, Paweł Horodecki, Robert Raussendorf, Ryszard Horodecki, Ravishankar Ramanathan, and Emily Tyhurst. The rank of contextuality. *arXiv preprint, arXiv:2205.10307*, 2022.

[IB98] C. J. Isham and J. Butterfield. Topos perspective on the kochen-specker theorem: I. quantum states as generalized valuations. *International Journal of Theoretical Physics*, 37(11):2669–2733, Nov 1998.

[INE+22] Yohei Ibe, Yuya O. Nakagawa, Nathan Earnest, Takahiro Yamamoto, Kosuke Mitarai, Qi Gao, and Takao Kobayashi. Calculating transition amplitudes by variational quantum deflation. *Phys. Rev. Research*, 4:013173, Mar 2022.

[IYLV20] Artur F. Izmaylov, Tzu-Ching Yen, Robert A. Lang, and Vladyslav Verteletskyi. Unitary partitioning approach to the measurement problem in the variational quantum eigensolver method. *Journal of Chemical Theory and Computation*, 16(1):190–195, 01 2020.

[JKLP18] Stephen P. Jordan, Hari Krovi, Keith S. M. Lee, and John Preskill. BQP-completeness of scattering in scalar quantum field theory. *Quantum*, 2:44, January 2018.

[JLP12] Stephen P. Jordan, Keith S. M. Lee, and John Preskill. Quantum algorithms for quantum field theories. *Science*, 336(6085):1130–1133, 2012.

[JLP14a] Stephen P. Jordan, Keith S. M. Lee, and John Preskill. Quantum algorithms for fermionic quantum field theories. *arXiv preprint, arXiv:1404.7115*, 2014.

[JLP14b] Stephen P. Jordan, Keith S. M. Lee, and John Preskill. Quantum computation of scattering in scalar quantum field theories. *Quantum Information and Computation*, 14:1014–1080, 2014.

[JMSS22] Chris Jones, Kunal Marwaha, Juspreet Singh Sandhu, and Jonathan Shi. Random max-csps inherit algorithmic hardness from spin glasses. *arXiv preprint, arXiv:2210.03006*, 2022.

[JPK+14] Ali JavadiAbhari, Shruti Patil, Daniel Kudrow, Jeff Heckey, Alexey Lvov, Frederic T. Chong, and Margaret Martonosi. Scaffcc: A framework for compilation and analysis of quantum computing programs. In *Proceedings of the 11th ACM Conference on Computing Frontiers*, CF '14, New York, NY, USA, 2014. Association for Computing Machinery.

[JSW93] Klaus Jansen, Petra Scheffler, and Gerhard Woeginger. The disjoint cliques problem. In *Proc. Fund. Comput. Theory, Lecture Notes in Computer Science*, volume 710, pages 319–328. Springer International Publishing, 1993.

[JW28] P. Jordan and E. Wigner. Über das paulische äquivalenzverbot. *Z. Phys.*, 47:631–651, 1928.

[JW09] Stephen P Jordan and Pawel Wocjan. Efficient quantum circuits for arbitrary sparse unitaries. *Physical Review A*, 80(6):062301, 2009.

[JWM+12] N Cody Jones, James D Whitfield, Peter L McMahon, Man-Hong Yung, Rodney Van Meter, Alán Aspuru-Guzik, and Yoshihisa Yamamoto. Faster quantum chemistry simulation on fault-tolerant

quantum computers. *New Journal of Physics*, 14(11):115023, nov 2012.

[Kan66] Shmuel Kaniel. Estimates for some computational techniques in linear algebra. *Mathematics of Computation*, 20(95):369–378, 1966.

[KFHM22] William Kirby, Bryce Fuller, Charles Hadfield, and Antonio Mezzacapo. Second-quantized fermionic operators with polylogarithmic qubit and gate complexity. *PRX Quantum*, 3:020351, Jun 2022.

[KGE14] Martin Kliesch, Christian Gogolin, and Jens Eisert. *Lieb-Robinson Bounds and the Simulation of Time-Evolution of Local Observables in Lattice Systems*, pages 301–318. Springer International Publishing, Cham, 2014.

[KGL+22] Oriel Kiss, Michele Grossi, Pavel Lougovski, Federico Sanchez, Sofia Vallecorsa, and Thomas Papenbrock. Quantum computing of the ^{6}Li nucleus via ordered unitary coupled clusters. *Phys. Rev. C*, 106:034325, Sep 2022.

[KH21] Amir Kalev and Itay Hen. Quantum Algorithm for Simulating Hamiltonian Dynamics with an Off-diagonal Series Expansion. *Quantum*, 5:426, April 2021.

[KHKL21] William M. Kirby, Sultana Hadi, Michael Kreshchuk, and Peter J. Love. Quantum simulation of second-quantized hamiltonians in compact encoding. *Phys. Rev. A*, 104:042607, Oct 2021.

[Kit95] Alexei Y. Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint, arXiv:quant-ph/9511026*, 1995.

[Kit97] A. Yu Kitaev. Quantum computations: Algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191, Dec 1997.

[KJK+21a] Michael Kreshchuk, Shaoyang Jia, William M. Kirby, Gary Goldstein, James P. Vary, and Peter J. Love. Light-front field theory on current quantum computers. *Entropy*, 23(5), 2021.

[KJK+21b] Michael Kreshchuk, Shaoyang Jia, William M. Kirby, Gary Goldstein, James P. Vary, and Peter J. Love. Simulating hadronic physics on noisy intermediate-scale quantum devices using basis light-front quantization. *Phys. Rev. A*, 103:062601, Jun 2021.

[KKG+22] Michael Kreshchuk, William M. Kirby, Gary Goldstein, Hugo Beauchemin, and Peter J. Love. Quantum simulation of quantum field theory in the light-front formulation. *Phys. Rev. A*, 105:032418, Mar 2022.

[KKR06] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

[KL17] Lucas Kocia and Peter Love. Discrete wigner formalism for qubits and noncontextuality of clifford gates on qubit stabilizer states. *Phys. Rev. A*, 96:062134, Dec 2017.

[KL18] Lucas Kocia and Peter Love. Measurement contextuality and planck's constant. *New Journal of Physics*, 20(7):073020, jul 2018.

[KL19a] William M. Kirby and Peter J. Love. Contextuality test of the nonclassicality of variational quantum eigensolvers. *Phys. Rev. Lett.*, 123:200501, Nov 2019.

[KL19b] Lucas Kocia and Peter Love. The non-disjoint ontic states of the grassmann ontological model, transformation contextuality, and the single qubit stabilizer subtheory. *Journal of Physics A: Mathematical and Theoretical*, 52(9):095303, feb 2019.

[KL20] William M. Kirby and Peter J. Love. Classical simulation of noncontextual pauli hamiltonians. *Phys. Rev. A*, 102:032418, Sept 2020.

[KL21] William M. Kirby and Peter J. Love. Variational quantum eigensolvers for sparse hamiltonians. *Phys. Rev. Lett.*, 127:110503, Sep 2021.

[KLY15] Shelby Kimmel, Guang Hao Low, and Theodore J. Yoder. Robust calibration of a universal single-qubit gate set via robust phase estimation. *Phys. Rev. A*, 92:062315, Dec 2015.

[KMM22] William Kirby, Mario Motta, and Antonio Mezzacapo. Exact and efficient lanczos method on a quantum computer. *arXiv preprint, arXiv:2208.00567*, 2022.

[KMT+17] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549:242–246, 2017.

[KMvB+19] C. Kokail, C. Maier, R. van Bijnen, T. Brydges, M. K. Joshi, P. Jurcevic, C. A. Muschik, P. Silvi, R. Blatt, C. F. Roos, and P. Zoller. Self-verifying variational quantum simulation of lattice models. *Nature*, 569(7756):355–360, 2019.

[KMZC+22] Katherine Klymko, Carlos Mejuto-Zaera, Stephen J. Cotton, Filip Wudarski, Miroslav Urbanek, Diptarka Hait, Martin Head-Gordon, K. Birgitta Whaley, Jonathan Moussa, Nathan Wiebe, Wibe A. de Jong, and Norm M. Tubman. Real-time evolution for ultracompact hamiltonian eigenstates on quantum hardware. *PRX Quantum*, 3:020323, May 2022.

[KR03] Julia Kempe and Oded Regev. 3-local hamiltonian is qma-complete. *Quantum Information and Computation*, 3(3):258–64, 2003.

[KS67] S. Kochen and E. Specker. The problem of hidden variables in quantum mechanics. *J. Math. Mech.*, 17(1):59–87, 1967.

[KSV02] A.Y. Kitaev, A. Shen, and M.N. Vyalyi. *Classical and Quantum Computation*. Graduate studies in mathematics. American Mathematical Society, 2002.

[KTC+19] Abhinav Kandala, Kristan Temme, Antonio D. Córcoles, Antonio Mezzacapo, Jerry M. Chow, and Jay M. Gambetta. Error mitigation extends the computational reach of a noisy quantum processor. *Nature*, 567(7749):491–495, 2019.

[KTL21] William M. Kirby, Andrew Tranter, and Peter J. Love. Contextual Subspace Variational Quantum Eigensolver. *Quantum*, 5:456, May 2021.

[KWB18] Angela Karanjai, Joel J. Wallman, and Stephen D. Bartlett. Contextuality bounds the efficiency of classical simulation of quantum processes. *arXiv preprint*, 2018.

[KWPO+11] Ivan Kassal, James D. Whitfield, Alejandro Perdomo-Ortiz, Man-Hong Yung, and Alán Aspuru-Guzik. Simulating chemistry using quantum computers. *Annual Review of Physical Chemistry*, 62(1):185–207, 2011. PMID: 21166541.

[KZG+09] G. Kirchmair, F. Zähringer, R. Gerritsma, M. Kleinmann, O. Gühne, A. Cabello, R. Blatt, and C. F. Roos. State-independent experimental test of quantum contextuality. *Nature*, 460:494 EP –, 07 2009.

[Lan50] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4), 1950.

[LC17] Guang Hao Low and Isaac L. Chuang. Optimal hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.*, 118:010501, Jan 2017.

[LC19] Guang Hao Low and Isaac L. Chuang. Hamiltonian Simulation by Qubitization. *Quantum*, 3:163, July 2019.

[LD20] Matthew Leifer and Cristhiano Duarte. Noncontextuality inequalities from antidistinguishability. *Phys. Rev. A*, 101:062113, Jun 2020.

[Lev73] L. A. Levin. Universal sequential search problems. *Probl. Peredachi Inf.*, 9:115–116, 1973.

[LGZ16] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature Communications*, 7(1):10138, 2016.

[LL93] Arjen K. Lenstra and Hendrik W.Jr. Lenstra, editors. *The Development of the Number Field Sieve*. Springer-Verlag, Berlin, Heidelberg, New York, 3 edition, 1993.

[Llo96] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.

[LLY20] Henry Lamm, Scott Lawrence, and Yukari Yamauchi. Parton physics on a quantum computer. *Phys. Rev. Research*, 2:013272, Mar 2020.

[LMZ+18] F. M. Leupold, M. Malinowski, C. Zhang, V. Negnevitsky, A. Cabello, J. Alonso, and J. P. Home. Sustained state-independent quantum contextual correlations from a single ion. *Phys. Rev. Lett.*, 120:180401, May 2018.

[LT20] Lin Lin and Yu Tong. Near-optimal ground state preparation. *Quantum*, 4:372, December 2020.

[LT22] Lin Lin and Yu Tong. Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers. *PRX Quantum*, 3:010318, Feb 2022.

[LW18] Guang Hao Low and Nathan Wiebe. Hamiltonian simulation in the interaction picture. *arXiv preprint, arXiv:1805.00675*, 2018.

[LWE19] Piers Lillystone, Joel J. Wallman, and Joseph Emerson. Contextuality and the single-qubit stabilizer subtheory. *Phys. Rev. Lett.*, 122:140405, Apr 2019.

[LWG+10] Benjamin P Lanyon, James D Whitfield, Geoff G Gillett, Michael E Goggin, Marcelo P Almeida, Ivan Kassal, Jacob D Biamonte, Masoud Mohseni, Ben J Powell, Marco Barbieri, Alan Aspuru-Guzik, and A. G. White. Towards quantum chemistry on a quantum computer. *Nature Chemistry*, 2(2):106–111, 2010.

[LYTT21] Erik Lötstedt, Kaoru Yamanouchi, Takashi Tsuchiya, and Yutaka Tachikawa. Calculation of vibrational eigenenergies on a quantum computer: Application to the fermi resonance in co 2. *Physical Review A*, 103(6):062609, 2021.

[MBS+18] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812, 2018.

[MEAG+20] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. Quantum computational chemistry. *Rev. Mod. Phys.*, 92:015003, Mar 2020.

[Mer90] N. David Mermin. Simple unified form for the major no-hidden-variables theorems. *Phys. Rev. Lett.*, 65:3373–3376, Dec 1990.

[Mer93] N. David Mermin. Hidden variables and the two theorems of john bell. *Rev. Mod. Phys.*, 65:803–815, Jul 1993.

[MHM+17] Christine Muschik, Markus Heyl, Esteban Martinez, Thomas Monz, Philipp Schindler, Berit Vogell, Marcello Dalmonte, Philipp Hauke, Rainer Blatt, and Peter Zoller. U(1) wilson lattice gauge theories in digital quantum simulators. *New Journal of Physics*, 19(10):103020, Oct 2017.

[MJE+19] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C. Benjamin, and Xiao Yuan. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Information*, 5(1):75, 2019.

[MK18] Shane Mansfield and Elham Kashefi. Quantum advantage from sequential-transformation contextuality. *Phys. Rev. Lett.*, 121:230401, Dec 2018.

[MKSCdJ17] Jarrod R. McClean, Mollie E. Kimchi-Schwartz, Jonathan Carter, and Wibe A. de Jong. Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states. *Phys. Rev. A*, 95:042308, Apr 2017.

[MMS+19] Sam McArdle, Alexander Mayorov, Xiao Shan, Simon Benjamin, and Xiao Yuan. Digital quantum simulation of molecular vibrations. *Chem. Sci.*, 10:5725–5735, 2019.

[MPJ+19] Alexander J McCaskey, Zachary P Parks, Jacek Jakowski, Shirley V Moore, Titus D Morris, Travis S Humble, and Raphael C Pooser. Quantum chemistry as a benchmark for near-term quantum computers. *npj Quantum Information*, 5(1):1–8, 2019.

[MRBAG16] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, feb 2016.

[MST+20] Mario Motta, Chong Sun, Adrian T. K. Tan, Matthew J. O'Rourke, Erika Ye, Austin J. Minnich, Fernando G. S. L. Brandão, and Garnet Kin-Lic Chan. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nature Physics*, 16(2):205–210, 2020.

[NC01] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2001.

[NCP+20] Yunseong Nam, Jwo-Sy Chen, Neal C. Pisenti, Kenneth Wright, Conor Delaney, Dmitri Maslov, Kenneth R. Brown, Stewart Allen, Jason M. Amini, Joel Apisdorf, Kristin M. Beck, Aleksey Blinov, Vandiver Chaplin, Mika Chmielewski, Coleman Collins, Shantanu Debnath, Kai M. Hudek, Andrew M. Ducore, Matthew Keesan, Sarah M. Kreikemeier, Jonathan Mizrahi, Phil Solomon, Mike Williams, Jaime David Wong-Campos, David Moehring, Christopher Monroe, and Jungsang Kim. Ground-state energy estimation of the water molecule on a trapped-ion quantum computer. *npj Quantum Information*, 6(1):33, 2020.

[NMF19] Ken M. Nakanishi, Kosuke Mitarai, and Keisuke Fujii. Subspace-search variational quantum eigensolver for excited states. *Phys. Rev. Research*, 1:033062, Oct 2019.

[NPdJB21] Benjamin Nachman, Davide Provasoli, Wibe A. de Jong, and Christian W. Bauer. Quantum algorithm for high energy physics simulations. *Phys. Rev. Lett.*, 126:062001, Feb 2021.

[OBK+16] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley,

C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, and J. M. Martinis. Scalable quantum simulation of molecular energies. *Phys. Rev. X*, 6:031007, Jul 2016.

[OR20] Cihan Okay and Robert Raussendorf. Homotopical approach to quantum contextuality. *Quantum*, 4:217, January 2020.

[OT08] Roberto Oliveira and Barbara M. Terhal. The complexity of quantum spin systems on a two-dimensional square lattice. *Quantum Information and Computation*, 8(10):0900–0924, 2008.

[OTR18] Cihan Okay, Emily Tyhurst, and Robert Raussendorf. The cohomological and the resource-theoretic perspective on quantum contextuality: common ground through the contextual fraction. *arXiv preprint*, 2018.

[Pai71] Christopher Conway Paige. *The computation of eigenvalues and eigenvectors of very large sparse matrices.* PhD thesis, University of London, 1971.

[PB85] Hans-Christian Pauli and Stanley J. Brodsky. Solving field theory in one space and one time dimension. *Phys. Rev. D*, 32:1993–2000, Oct 1985.

[Per91] Asher Peres. Two simple proofs of the kochen-specker theorem. *J. Phys. A: Math. Theor.*, 24(4):L175, 1991.

[PGS+17] S. Paesani, A. A. Gentile, R. Santagati, J. Wang, N. Wiebe, D. P. Tew, J. L. O'Brien, and M. G. Thompson. Experimental bayesian quantum phase estimation on a silicon photonic chip. *Phys. Rev. Lett.*, 118:100503, 2017.

[PM19] Robert M. Parrish and Peter L. McMahon. Quantum filter diagonalization: Quantum eigendecomposition without full quantum phase estimation. *arXiv preprint, arXiv:1909.08925*, 2019.

[PMS+14] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5:4213 EP –, July 2014.

[Pre18] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[PRM18] Alex Parent, Martin Roetteler, and Michele Mosca. Improved reversible and quantum circuits for Karatsuba-based integer multiplication. In Mark M. Wilde, editor, *12th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2017)*, volume 73 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Rau13] Robert Raussendorf. Contextuality in measurement-based quantum computation. *Phys. Rev. A*, 88:022322, Aug 2013.

[Rau19] Robert Raussendorf. Cohomological framework for contextual quantum computations. *Quantum Information and Computation*, 19(13-14):1141–1170, 2019.

[RBD+17] Robert Raussendorf, Dan E. Browne, Nicolas Delfosse, Cihan Okay, and Juan Bermejo-Vega. Contextuality and wigner-function negativity in qubit quantum computation. *Phys. Rev. A*, 95:052334, May 2017.

[RBM+18a] Jonathan Romero, Ryan Babbush, Jarrod R McClean, Cornelius Hempel, Peter J Love, and Alán Aspuru-Guzik. Strategies for quan-

tum computing molecular energies using the unitary coupled cluster ansatz. *Quantum Science and Technology*, 4(1):014008, Oct 2018.

[RBM18b] Nicholas C Rubin, Ryan Babbush, and Jarrod McClean. Application of fermionic marginal constraints to hybrid quantum algorithms. *New Journal of Physics*, 20(5):053020, 2018.

[RBVT+20] Robert Raussendorf, Juani Bermejo-Vega, Emily Tyhurst, Cihan Okay, and Michael Zurel. Phase-space-simulation method for quantum computation with magic states on qubits. *Phys. Rev. A*, 101:012350, Jan 2020.

[RGD+22] Gokul Subramanian Ravi, Pranav Gokhale, Yi Ding, William M. Kirby, Kaitlin N. Smith, Jonathan M. Baker, Peter J. Love, Henry Hoffmann, Kenneth R. Brown, and Frederic T. Chong. Cafqa: A classical simulation bootstrap for variational quantum algorithms. *arXiv preprint, arXiv:2202.12924*, 2022.

[RH14] Ravishankar Ramanathan and Pawel Horodecki. Necessary and sufficient condition for state-independent contextual measurement scenarios. *Phys. Rev. Lett.*, 112:040404, Jan 2014.

[RKR+21] Antonio E. Russo, William M. Kirby, Kenneth M. Rudinger, Andrew D. Baczewski, and Shelby Kimmel. Consistency testing for robust phase estimation. *Phys. Rev. A*, 103:042609, Apr 2021.

[RSKanK12] Ravishankar Ramanathan, Akihito Soeda, Paweł Kurzyński, and Dagomir Kaszlikowski. Generalized monogamy of contextual inequalities from the no-disturbance principle. *Phys. Rev. Lett.*, 109:050404, Aug 2012.

[RWT+22] Alexis Ralli, Tim J. Weaving, Andrew Tranter, William M. Kirby, Peter J. Love, and Peter V. Coveney. Unitary partitioning and the

contextual subspace variational quantum eigensolver. *arXiv preprint, arXiv:2207.03451*, 2022.

[Saa80] Y. Saad. On the rates of convergence of the lanczos and the block-lanczos methods. *SIAM Journal on Numerical Analysis*, 17(5):687–706, 1980.

[SBMS+19] Ramiro Sagastizabal, Xavier Bonet-Monroig, Malay Singh, M Adriaan Rol, CC Bultink, Xiang Fu, CH Price, VP Ostroukh, N Muthusubramanian, A Bruno, et al. Experimental error mitigation via symmetry verification in a variational quantum eigensolver. *Physical Review A*, 100(1):010302, 2019.

[SBW+21] Yuan Su, Dominic W. Berry, Nathan Wiebe, Nicholas Rubin, and Ryan Babbush. Fault-tolerant quantum simulations of chemistry in first quantization. *PRX Quantum*, 2:040332, Nov 2021.

[Sch11] Norbert Schuch. Complexity of commuting hamiltonians on a square lattice of qubits. *Quantum Information and Computation*, 11:901, 2011.

[SCR+20] Kanav Setia, Richard Chen, Julia E. Rice, Antonio Mezzacapo, Marco Pistoia, and James D. Whitfield. Reducing qubit requirements for quantum simulations using molecular point group symmetries. *Journal of Chemical Theory and Computation*, 16(10):6091–6097, 10 2020.

[SHE20] Nicholas H. Stair, Renke Huang, and Francesco A. Evangelista. A multireference quantum krylov algorithm for strongly correlated electrons. *Journal of Chemical Theory and Computation*, 16(4):2236–2245, 04 2020.

[Sho94] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[SJM+21] Ariel Shlosberg, Andrew J. Jena, Priyanka Mukhopadhyay, Jan F. Haase, Felix Leditzky, and Luca Dellantonio. Adaptive estimation of quantum observables. *arXiv preprint, arXiv:2110.15339*, 2021.

[SM19] Scott E Smart and David A Mazziotti. Quantum-classical hybrid algorithm using an error-mitigating n-representability condition to compute the mott metal-insulator transition. *Physical Review A*, 100(2):022517, 2019.

[SN20] J.J. Sakurai and J. Napolitano. *Modern Quantum Mechanics*. Cambridge University Press, 2020.

[Som05] Rolando D. Somma. Quantum computation, complexity, and many-body physics. *arXiv preprint, arXiv:quant-ph/0512209*, 2005.

[Som16] Rolando D. Somma. A trotter-suzuki approximation for lie groups with applications to hamiltonian simulation. *Journal of Mathematical Physics*, 57(6):062202, 2016.

[Spe05] R. W. Spekkens. Contextuality for preparations, transformations, and unsharp measurements. *Phys. Rev. A*, 71:052108, May 2005.

[Spe07] Robert W. Spekkens. Evidence for the epistemic view of quantum states: A toy theory. *Phys. Rev. A*, 75:032110, Mar 2007.

[Spe08] Robert W. Spekkens. Negativity and contextuality are equivalent notions of nonclassicality. *Phys. Rev. Lett.*, 101:020401, Jul 2008.

[Spe16] Robert W. Spekkens. *Quasi-Quantization: Classical Statistical Theories with an Epistemic Restriction*, pages 83–135. Springer Netherlands, Dordrecht, 2016.

[SR11] Sayeeda Sultana and Katarzyna Radecka. Reversible implementation of square-root circuit. In *2011 18th IEEE International Conference on Electronics, Circuits, and Systems*, pages 141–144, 2011.

[SRL12] Jacob T. Seeley, Martin J. Richard, and Peter J. Love. The bravyi-kitaev transformation for quantum computation of electronic structure. *The Journal of Chemical Physics*, 137(22):224109, 2012.

[SS18] David Schmid and Robert W. Spekkens. Contextual advantage for state discrimination. *Phys. Rev. X*, 8:011015, Feb 2018.

[SSW18] David Schmid, Robert W. Spekkens, and Elie Wolfe. All the noncontextuality inequalities for arbitrary prepare-and-measure experiments with respect to any fixed set of operational equivalences. *Phys. Rev. A*, 97:062103, Jun 2018.

[SSW+21] David Schmid, John H. Selby, Elie Wolfe, Ravi Kunjwal, and Robert W. Spekkens. Characterization of noncontextuality in the framework of generalized probabilistic theories. *PRX Quantum*, 2:010331, Feb 2021.

[Ste19] Mark Steudtner. *Methods to simulate fermions on quantum computers with hardware limitations*. PhD thesis, Leiden University, 2019.

[Suz85] Masuo Suzuki. Decomposition formulas of exponential operators and lie exponentials with some applications to quantum mechanics and statistical physics. *Journal of Mathematical Physics*, 26(4):601–612, 1985.

[Suz91] Masuo Suzuki. General theory of fractal path integrals with applications to many-body theories and statistical physics. *Journal of Mathematical Physics*, 32(2):400–407, 1991.

[SW18a] Kanav Setia and James D. Whitfield. Bravyi-kitaev superfast simulation of electronic structure on a quantum computer. *The Journal of Chemical Physics*, 148(16):164104, 2018.

[SW18b] Mark Steudtner and Stephanie Wehner. Fermion-to-qubit mappings with varying resource requirements for quantum simulation. *New Journal of Physics*, 20(6):063010, June 2018.

[SW20] Mark Steudtner and Stephanie Wehner. Estimating exact energies in quantum simulation without toffoli gates. *Phys. Rev. A*, 101:052329, May 2020.

[SWG+18] Raffaele Santagati, Jianwei Wang, Antonio A. Gentile, Stefano Paesani, Nathan Wiebe, Jarrod R. McClean, Sam Morley-Short, Peter J. Shadbolt, Damien Bonneau, Joshua W. Silverstone, David P. Tew, Xiaoqi Zhou, Jeremy L. O'Brien, and Mark G. Thompson. Witnessing eigenstates for quantum simulation of hamiltonian spectra. *Science Advances*, 4(1), 2018.

[SY21] Kazuhiro Seki and Seiji Yunoki. Quantum power method by a superposition of time-evolved states. *PRX Quantum*, 2:010333, Feb 2021.

[SZZ+17] Yangchao Shen, Xiang Zhang, Shuaining Zhang, Jing-Ning Zhang, Man-Hong Yung, and Kihwan Kim. Quantum implementation of the unitary coupled cluster for simulating molecular electronic structure. *Physical Review A*, 95(2):020501, 2017.

[TCC+22] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, 986:1–128, 2022. The Variational Quantum Eigensolver: a review of methods and best practices.

[TD00] Barbara M. Terhal and David P. DiVincenzo. Problem of equilibration and the computation of correlation functions on a quantum computer. *Phys. Rev. A*, 61:022301, Jan 2000.

[Tha08] Mechthild Thalhammer. High-order exponential operator splitting methods for time-dependent schrödinger equations. *SIAM Journal on Numerical Analysis*, 46(4):2022–2038, 2008.

[TL13] Borzu Toloui and Peter Love. Quantum algorithms for quantum chemistry based on the sparsity of the ci-matrix. *arXiv preprint*, 2013.

[TRJ+20] Tyler Takeshita, Nicholas C. Rubin, Zhang Jiang, Eunseok Lee, Ryan Babbush, and Jarrod R. McClean. Increasing the representation accuracy of quantum simulations of chemistry without extra quantum resources. *Phys. Rev. X*, 10:011004, Jan 2020.

[TZC+22] Nikolay V. Tkachenko, Yu Zhang, Lukasz Cincio, Alexander I. Boldyrev, Sergei Tretiak, and Pavel A. Dub. Quantum davidson algorithm for excited states. *arXiv preprint, arXiv:2204.10741*, 2022.

[UAS+21] Shashanka Ubaru, Ismail Yunus Akhalwaya, Mark S. Squillante, Kenneth L. Clarkson, and Lior Horesh. Quantum topological data analysis with linear depth and exponential speedup. *arXiv preprint, arXiv:2108.02811*, 2021.

[UB21] Alexey Uvarov and Jacob Biamonte. On barren plateaus and cost function locality in variational quantum algorithms. *Journal of Physics A: Mathematical and Theoretical*, 2021.

[VBE96] Vlatko Vedral, Adriano Barenco, and Artur Ekert. Quantum networks for elementary arithmetic operations. *Phys. Rev. A*, 54:147–153, Jul 1996.

[VC05] F Verstraete and J I Cirac. Mapping local hamiltonians of fermions to local hamiltonians of spins. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09012–P09012, sep 2005.

[vE07] S. J. van Enk. A toy model for quantum mechanics. *Foundations of Physics*, 37(10):1447–1460, Oct 2007.

[VFGE12] Victor Veitch, Christopher Ferrie, David Gross, and Joseph Emerson. Negative quasi-probability as a resource for quantum computation. *New Journal of Physics*, 14(11):113011, nov 2012.

[VHL+10] J. P. Vary, H. Honkanen, Jun Li, P. Maris, S. J. Brodsky, A. Harindranath, G. F. de Teramond, P. Sternberg, E. G. Ng, and C. Yang. Hamiltonian light-front field theory in a basis function approach. *Phys. Rev. C*, 81:035205, 3 2010.

[Vor63] N. Vorob'yev. Markov measures and markov extensions. *Theory of Probability & Its Applications*, 8(4):420–429, 1963.

[Vor67] N. Vorob'yev. Coalition games. *Theory of Probability & Its Applications*, 12(2):251–266, 1967.

[VYI20] Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F. Izmaylov. Measurement optimization in the variational quantum eigensolver using a minimum clique cover. *The Journal of Chemical Physics*, 152(12):124114, 2020.

[WB03] Pawel Wocjan and Thomas Beth. The 2-local hamiltonian problem encompasses np. *International Journal of Quantum Information*, 01(03):349–357, 2003.

[WBAG11] James D. Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. Simulation of electronic structure hamiltonians using quantum computers. *Molecular Physics*, 109(5):735–750, 2011.

[WBC+14] Dave Wecker, Bela Bauer, Bryan K. Clark, Matthew B. Hastings, and Matthias Troyer. Gate-count estimates for performing quantum chemistry on small quantum computers. *Phys. Rev. A*, 90:022305, Aug 2014.

[WDB+15] Ya Wang, Florian Dolde, Jacob Biamonte, Ryan Babbush, Ville Bergholm, Sen Yang, Ingmar Jakobi, Philipp Neumann, Alán Aspuru-Guzik, James D Whitfield, and Jörg Wrachtrup. Quantum simulation of helium hydride cation in a solid-state spin register. *ACS Nano*, 9(8):7769–7774, 2015.

[WFC+21] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 12(1):6961, 2021.

[WHW+15] Dave Wecker, Matthew B. Hastings, Nathan Wiebe, Bryan K. Clark, Chetan Nayak, and Matthias Troyer. Solving strongly correlated electron models on a quantum computer. *Phys. Rev. A*, 92:062318, Dec 2015.

[WRK+22] Tim J. Weaving, Alexis Ralli, William M. Kirby, Andrew Tranter, Peter J. Love, and Peter V. Coveney. A stabilizer framework for contextual subspace vqe and the noncontextual projection ansatz. *arXiv preprint, arXiv:2204.02150*, 2022.

[WSHY21] Bujiao Wu, Jinzhao Sun, Qi Huang, and Xiao Yuan. Overlapped grouping measurement: A unified framework for measuring quantum states. *arXiv preprint, arXiv:2105.13091*, 2021.

[XC19] Zhen-Peng Xu and Adán Cabello. Necessary and sufficient condition for contextuality from incompatibility. *Phys. Rev. A*, 99:020103, Feb 2019.

[YB12] Jijiang Yan and Dave Bacon. The k-local pauli commuting hamiltonians problem is in p. *arXiv preprint*, 2012.

[YCM+14] M. H. Yung, J. Casanova, A. Mezzacapo, J. McClean, L. Lamata, A. Aspuru-Guzik, and E. Solano. From transistor to trapped-ion computers for quantum chemistry. *Scientific Reports*, 4:3589 EP –, 01 2014.

[YGI22] Tzu-Ching Yen, Aadithya Ganeshram, and Artur F. Izmaylov. Deterministic improvements of quantum measurements with grouping of compatible operators, non-local transformations, and covariance estimates. *arXiv preprint, arXiv:2201.01471*, 2022.

[YHM+22] Nobuyuki Yoshioka, Hideaki Hakoshima, Yuichiro Matsuzaki, Yuuki Tokunaga, Yasunari Suzuki, and Suguru Endo. Generalized quantum subspace expansion. *Phys. Rev. Lett.*, 129:020502, Jul 2022.

[YVI20] Tzu-Ching Yen, Vladyslav Verteletskyi, and Artur F. Izmaylov. Measuring all compatible operators in one series of single-qubit measurements using unitary transformations. *Journal of Chemical Theory and Computation*, 16(4):2400–2409, 04 2020.

[ZTK+20] Andrew Zhao, Andrew Tranter, William M. Kirby, Shu Fay Ung, Akimasa Miyake, and Peter J. Love. Measurement reduction in variational quantum algorithms. *Phys. Rev. A*, 101:062322, Jun 2020.