

EIGENSTATE PREPARATION ON QUANTUM COMPUTERS

By

Joey Bonitati

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Physics and Computational Mathematics, Sciences and Engineering—Doctor of Philosophy

2024

ABSTRACT

This thesis investigates quantum algorithms for eigenstate preparation, with a primary focus on solving eigenvalue problems such as the Schrödinger equation by utilizing near-term quantum computing devices. These problems are ubiquitous in several scientific fields, but more accurate solutions are specifically needed as a prerequisite for many quantum simulation tasks. To address this, we establish three methods in detail: quantum adiabatic evolution with optimal control, the Rodeo Algorithm, and the Variational Rodeo Algorithm.

The first method explored is adiabatic evolution, a technique that prepares quantum states by simulating a quantum system that evolves slowly over time. The adiabatic theorem can be used to ensure that the system remains in an eigenstate throughout the process, but its implementation can often be infeasible on current quantum computing hardware. We employ a unique approach using optimal control to create custom gate operations for superconducting qubits and demonstrate the algorithm on a two-qubit IBM cloud quantum computing device.

We then explore an alternative to adiabatic evolution, the Rodeo Algorithm, which offers a different approach to eigenstate preparation by using a controlled quantum evolution that selectively filters out undesired components in the wave function stored on a quantum register. We show results suggesting that this method can be effective in preparing eigenstates, but its practicality is predicated on the preparation of an initial state that has significant overlap with the desired eigenstate. To address this, we introduce the novel Variational Rodeo Algorithm, which replaces the initialization step with dynamic optimization of quantum circuit parameters to increase the success probability of the Rodeo Algorithm. The added flexibility compensates for instances in which the original algorithm can be unsuccessful, allowing for better scalability.

This research seeks to contribute to a deeper understanding of how quantum algorithms can be employed to attain efficient and accurate solutions to eigenvalue problems. The overarching goal is to present ideas that can be used to improve understanding of nuclear physics by providing potential quantum and classical techniques that can aid in tasks such as the theoretical description of nuclear structures and the simulation of nuclear reactions.

Copyright by
JOEY BONITATI
2024

ACKNOWLEDGEMENTS

I acknowledge the intellectual support of my advisor Dean Lee, as well as our various collaborators at MSU (Jacob Watkins, Ashe Hicks, Gabriel Given, Zhengrong Qian, Xilin Zhang, Yuanzhuo Ma, Max Bee-Lindgren, Kenneth Choi) and LLNL (Kyle Wendt, Sofia Quaglioni, Toño Coello Perez).

I acknowledge the financial support from MSU and through grants from the DOE and NSF.

I acknowledge the professional support from my undergraduate REU advisor Filomena Nunes at MSU.

I acknowledge the emotional support from my immediate and extended family, who love to celebrate my accomplishments so that I don't have to.

Above all, I acknowledge my love, Sarah, for supporting me in many ways through the final years of my PhD program.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER 1 INTRODUCTION	1
1.1 The Scientific Method	1
1.2 History of Quantum Physics	5
1.3 Motivation for Research in Quantum Many-Body Theory	10
CHAPTER 2 BACKGROUND	15
2.1 Quantum Information and Computation	15
2.2 Eigenvalue Problems and Methods for Solving Them	28
2.3 Applications of Eigenstate Preparation	43
CHAPTER 3 ADIABATIC EVOLUTION WITH OPTIMAL CONTROL	49
3.1 The Adiabatic Theorem	49
3.2 Adiabatic Evolution of Two-Spin Systems	51
3.3 Two-Qubit Quantum Adiabatic Evolution Implementation	53
3.4 Results and Discussion	59
CHAPTER 4 RODEO ALGORITHM ANALYSIS AND DEMONSTRATION	75
4.1 Rodeo algorithm	75
4.2 Demonstration of the Rodeo Algorithm	81
CHAPTER 5 VARIATIONAL RODEO ALGORITHM	95
5.1 Algorithm	95
5.2 Mathematical Comparison of Cost Functions	97
5.3 Simulation and Results	101
5.4 Future Work	107
CHAPTER 6 SUMMARY	110
6.1 Adiabatic Evolution with Optimal Control	110
6.2 Rodeo Algorithm	110
6.3 Variational Rodeo Algorithm	111
BIBLIOGRAPHY	112

LIST OF TABLES

Table 3.1	Calibration data for <i>ibmq_belem</i> , <i>casablanca</i> , <i>ibmq_lima</i> , and <i>ibmq_manila</i> . The relaxation and dephasing times, T_1 and T_2 , are provided for qubits 0 and 1 of each system throughout the simulation. The implementation times for the CNOT gate and the decomposition of the short-time propagators are denoted as τ_{CNOT} and τ_U , respectively.	60
Table 4.1	Overlap of $ \psi_F\rangle$ with energy eigenvector $ E_j\rangle$ after N cycles of the rodeo algorithm using Gaussian random values for t_n with $\sigma = 5$ and $E = E_j$	84
Table 4.2	Results obtained using RA for the energy eigenvalues of $H^{(0)}$ and the expectation values of $H^{(1)}$ corresponding to the eigenstates of $H^{(0)}$. For reference, exact values are also provided.	92
Table 4.3	Results for eigenstate preparation without measurement error mitigation.	92
Table 4.4	Results for eigenstate preparation with measurement error mitigation.	93

LIST OF FIGURES

Figure 2.1	The Bloch sphere representing the state of a single qubit as coordinates on a sphere [104]	17
Figure 2.2	A basic quantum circuit diagram for creating a Bell state. The initial state on the left is $ 00\rangle$, the gates are applied from left to right, leading to the final state $\frac{1}{2}(00\rangle + 11\rangle)$	21
Figure 3.1	Evolution of the fidelity between the device state and the ground state of $H(t)$ for various values of a) the evolution time T and b) the number of time steps n as a function of the parameter $s \equiv t/T$. The adiabatic evolution implemented on quantum devices utilizes $T = 20$ and $n = 20$, as this combination results in high fidelity while minimizing the number of gates required for device implementation.	66
Figure 3.2	Decomposition of the k -th unitary transformation in the sequence of Equation (3.12) [circuit a)] into CNOT and U3 gates [circuit b)]. Each U3 gate depends on three Euler angles. The quantum circuit resulting from this decomposition can be directly implemented on IBMQ systems.	67
Figure 3.3	Decomposition of the first short-time propagator in Eq. (3.12) [circuit a)] into CNOT and U3 gates [circuit b)]. Circuit c) shows how the same type of decomposition can be used to decompose the CNOT gate into elementary gates.	67
Figure 3.4	Schematic depiction of a) the model two-qubit processor, consisting of two capacitively coupled superconducting transmons controlled with microwave pulses, and b) the energy spectrum of a superconducting transmon in which the lowest two levels, shown as solid lines, are the computational qubit.	68
Figure 3.5	First 100 ns of control pulses with lengths a) $\tau = 2500$ ns, b) $\tau = 400$ ns, and c) $\tau = 120$ ns, realizing the first unitary in the sequence of Equation (3.12). The correlation between the amplitude of the control pulse with its length establishes a lower bound for the implementation time of two-qubit gates on current quantum devices.	69
Figure 3.6	Adiabatic evolution of a) instantaneous fidelity $F(t)$ and b) expectation value $\langle \psi(t) H_T \psi(t) \rangle$, obtained through the application of the unitary sequence from Equation (3.12) decomposed into elementary gates on four IBM quantum processors: <i>ibmq_belem</i> (blue triangles), <i>ibmq_casablanca</i> (orange diamonds), <i>ibmq_lima</i> (green squares), and <i>ibmq_manila</i> (red circles). The significant discrepancies between the experimental results and the ideal quantum processor simulations (dotted lines), conducted using Qiskit's Aer simulator, can be attributed to decoherence due to long implementation times.	70

Figure 3.7 Combination of simultaneous U3 gates [circuit a] into two-qubit $U^{(2)}$ gates [circuit b]. The CNOT and $U3^{(2)}$ gates in circuit b) are executed with control pulses, the total durations of which closely correspond to the implementation time of circuit a) on IBMQ systems.	71
Figure 3.8 A comparison of the emulated outputs of the adiabatic evolution of Eq. (3.12) and the results obtained from <i>ibmq_belem</i> , <i>ibmq_casablanca</i> , <i>ibmq_lima</i> , and <i>ibmq_manila</i> systems. Panels a), c), e) and g) display the progression of the instantaneous fidelity F . Classical emulations implementing the adiabatic evolution through either a circuit of elementary gates or custom gates of length τ_U (represented by green triangles and orange diamonds, respectively) achieve target state fidelities comparable to those attained by IBMQ systems (indicated by blue crosses). Emulated outputs utilizing shorter gates (denoted by orange squares and circles) reached approximately 95% fidelity, facilitating enhanced extraction of the target state's spectroscopic information, as demonstrated by the evolution of $\langle H_T \rangle$ in panels b), d), f) and h).	72
Figure 3.9 Mean error in measured occupation probabilities as a function of the number of shots N . A higher number of shots reduces statistical noise and provides an estimation of the measurement error of the quantum computer. The error in occupation probabilities from these simulations is capped at $\delta c _{\text{exp}}^2 = 0.016$, which constitutes only a minor portion of the error seen in calculated fidelities and expectation values.	73
Figure 3.10 Evolution of a) fidelity and b) expectation value $\langle H_T \rangle$ using Qiskit's Aer simulator with a noise model based on <i>ibmq_belem</i> . Solid blue lines represent the expected results when incorporating all error sources considered by Aer, while the blue shaded areas show the probable values at each time interval. Dotted black lines indicate the ideal evolution in a noiseless system. Orange triangles come from singular value decomposition filtering of the data supporting the blue distributions. The closeness between the dotted lines and the orange triangles strongly implies that dissipative processes are the primary source of noise in IBMQ simulations, even at the early stages of the evolution.	74
Figure 4.1 One cycle of the Rodeo algorithm. Additional copies of this circuit can be appended to increase its efficacy. If mid-circuit measurements are allowed, only one ancilla is needed for the arena, otherwise additional ancilla qubits may be used, with one ancilla qubit participating in each cycle.	77
Figure 4.2 (color online) Initial-state spectral function for the Heisenberg model. Using the RA, we depict the initial-state spectral function for the Heisenberg spin chain with 3 (thin blue line), 6 (thick green line), and 9 (medium red line) cycles. The average was taken over 20 sets of Gaussian random values for t_n with $\sigma = 5$. For reference, the exact initial-state spectral function is presented with black open circles.	83

Figure 4.3 (color online) Logarithm of the wave function error versus the total propagation time for the Heisenberg model. We plot $\log \Delta$, versus the total propagation time, T , for the Heisenberg model. We show results for the rodeo algorithm, phase estimation, and adiabatic evolution. We also show the asymptotic estimates $\log F_A$ and $\log F_G$.	85
Figure 4.4 Sequential scans of the energy. Each bin represents a distinct RA circuit for target energy E and width parameter σ . The color and shading indicates the success probability $P_{0^N}(E)$. Centered around each of the peaks from the first scan, a second scan is performed using with a large value of σ and better energy resolution. This is then repeated for the third scan.	88
Figure 4.5 Energy scans for $H_{\text{obj}}(0)$. The results were obtained from experiments on IBMQ’s Casablanca device. The dashed lines show predicted outcomes from classical success probability calculations. The initial scan also includes noiseless quantum device simulation results.	89
Figure 4.6 Eigenvalues E_1 (top) and E_2 (bottom) vs. ϕ . The plot shows RA data (circles), RA quadratic fit (solid line) with 3σ error bands (shaded), exact values (squares), and exact quadratic fit (dashed line).	91
Figure 5.1 A circuit diagram of a QAOA circuit with objective Hamiltonian H_f , mixer Hamiltonian H_i , and tunable parameters γ and β . Each time evolution gate has its own adjustable parameter, typically restricted to be in the range $(0, 2\pi)$.	102
Figure 5.2 An analysis of the final states $ \psi_f\rangle$ of VRA simulations on random 6-qubit (top row) and 10-qubit (bottom row) systems, repeated for different values of E , increasing linearly along each x -axis. The y -axes represent the exact eigenstates ordered by their eigenvalues scaled by energy (left column) and level number (right column). The overlap of the final state of the QAOA circuit with each eigenvector is represented by the relative darkness of each shaded rectangle. Note that many rectangles are too light to be seen due to extremely low overlap. For each VRA optimization, a red circle is placed at the eigenstate with the highest overlap	104
Figure 5.3 A comparison of the VRA (left) and VQE (right) cost functions for BFGS optimizations of a 20 parameter 6 qubit QAOA circuit. The shared x axes represent the BFGS iteration number from 1 to 100. The y axes in each row represent properties of the output state $ \psi_f\rangle$ of the QAOA circuit, namely energy (top), ground state overlap (middle), and RA success probability (bottom). Note the different scales on the y -axes. 10 random sets of 20 parameters each were generated and used as the initial circuit parameters in separate optimizations for both methods, with each set represented by a different color on the plots.	106

CHAPTER 1

INTRODUCTION

The overarching goal of the research presented in this dissertation is to discover more efficient computational methods to aid in the theoretical understanding of nuclear physics. The nucleus of an atom is so small that, in order to properly understand it, it is necessary to consider the laws of quantum mechanics. It is notoriously difficult to computationally analyze such systems, so the concept of quantum simulation, the approach of using one quantum system to emulate another, has recently gained popularity [109]. A likely first step of a quantum simulation algorithm will be to prepare a quantum computer in a state that is analogous in some way to the initial state of the quantum system of interest. Often, this will be one of the characteristic states, called "eigenstates", of the system. This critical step of "eigenstate preparation" is not straightforward to perform, and it has enough significant applications in other areas to be interesting enough on its own.

To properly build up the motivation for this particular problem, this chapter provides a brief history of the development of quantum and nuclear physics, starting with the scientific method, a topic that is often misunderstood and underappreciated. More detailed background on quantum computation and eigenvalue problems is given in Chapter 2.

1.1 The Scientific Method

The scientific method is the primary way in which all kinds of scientists, including physicists, make discoveries about things that can happen in the universe. There has been a growing effort to increase awareness of the scientific method among cultures and individuals to achieve a so-called "science literacy" [32]. General ignorance of science may be a factor in the proliferation of false information, an issue that has been exacerbated by social media and recent rapid developments in generative AI, which has led the World Economic Forum to declare the spread of misinformation and disinformation to be the most serious threat facing the world in 2024 [120]. Furthermore, it is helpful to understand how the seemingly abstract field of quantum physics actually emerged from discoveries made using the scientific method, which will be explained in Section 1.2. For these reasons, it is pertinent to explore how science, particularly physics, evolved over history into what

it is today.

1.1.1 History of the Scientific Method

In short, the scientific method is the way that people can use a combination of logic and observation to learn new things. The idea that we can learn about the universe by observing it from within was first established in writing by the ancient Greek philosopher Aristotle [8]. His method of obtaining knowledge relies on two types of logical reasoning: induction and deduction. Induction (or inductive reasoning) involves deriving broad generalizations from observations, whereas deduction (or deductive reasoning) involves invoking the rules of logic to make valid conclusions based on given premises. A common type of generalization used in induction is that if you repeatedly observe a particular outcome in a certain set of circumstances, then that outcome will occur again the next time those circumstances are present; for example, if you notice that whenever you drink milk, you get an upset stomach, then you could conclude that the next time you drink milk, you will get an upset stomach again. A common type of deduction used in the scientific method is the mathematical manipulation of equations; for example, the equation $x + 1 = 2$ implies that $x = 1$ (using the axioms of algebra). Consider the following example of how Aristotle's scientific method can be applied:

1. Start by making observations of a physical phenomenon
2. Using these observations and some intuition, design an equation that can make predictions for future observations
3. (Deduction) If possible, use algebraic manipulation on this equation to discover more equations that can make different predictions
4. (Induction) Validate the predictions from steps (2) and (3) with more observations.

After Aristotle, the scientific method was refined further by philosophers and mathematicians such as Descartes and Isaac Newton. An important part of the scientific method that was gradually added to Aristotle's initial idea is that, in order to be considered scientifically rigorous, the predic-

tions and hypotheses produced must be testable. In order to be considered testable, a statement must be falsifiable, meaning that it is possible for it not to be true, and it must be practically feasible to set up a reproducible experiment that can refute it. Furthermore, to avoid confirmation bias, all predictions and hypotheses must be assumed to be false until there is sufficient reason to believe otherwise. If an idea is not testable, then it cannot be considered scientific. This is not to say that such ideas cannot be true; it simply means that they are outside the scope of what is considered science.

Similarly, just because an idea is scientific does not mean that it is true. In fact, all scientific predictions are by definition not provable with complete certainty! Since the scientific method relies on induction, it is subject to the "problem of induction" formulated by the philosopher David Hume [55]. Hume pointed out that inferences based on induction always assume the principle of induction, which requires that the past can be used to predict the future. Hume argues that this is not necessarily true, so every prediction about the future based on past observations has a chance of being wrong. When using the scientific method, one must always acknowledge the possibility that a prediction could be wrong, perhaps even for some unknown reason. Since all predictions are initially assumed to be false, this means that scientific hypotheses cannot be verified beyond all doubt. Thus, the phrase "scientific fact", which may sometimes be used in misinformation campaigns to create a false sense of authority, is actually an oxymoron!

Despite the lack of complete certainty inherent in science, it is obvious that the scientific method is still useful for making predictions. For example, physicists at NASA were able to understand the complicated trajectories of space shuttles well enough to land humans on the moon. Thus, in order to practice science, one must ignore the problem of induction to some extent and acknowledge that, even though inductive reasoning is flawed, it is necessary. After all, the assumption that the laws of physics will be the same in the future as they were in the past, while not necessarily valid, is not a difficult assumption to believe. If we make this assumption, then we can at least infer that when we make repeated observations of a phenomenon, it is somewhat probable that we will make the same observations in the future. This is enough to justify the empirical testing of hypotheses in

physics, which, along with intuition and mathematical deduction, leads to a better understanding of the universe in which we live.

1.1.2 From Classical Physics to Modern Physics

Today, physics is generally classified into two branches: classical physics and modern physics. Classical physics deals with phenomena occurring under normal conditions, whereas modern physics addresses extraordinary situations, including velocities approaching the speed of light and scales of distance similar to the size of an atom. Prominent fields within modern physics include nuclear physics, special relativity, and quantum mechanics. These new branches all deal with situations outside the realm of everyday human experiences, but there is no denying that they have had noticeable effects on society; for example, advances in these sciences have made new inventions possible, such as nuclear power plants, GPS tracking, and quantum computers.

For most of history, the study of physics has been related to the phenomena we experience directly as humans. This makes sense because empirical hypotheses must be testable and the most obvious way to test things is through direct observation with the senses. However, human sensory organs are limited in their ability to gather information; the naked eye is easily fooled by optical illusions, for example. Thus, scientific progress naturally advances alongside the development of better tools and measuring devices. For example, Isaac Newton's theory of gravity was likely based on intuition that came from his human senses; perhaps it was conceived when an apple fell on his head, as the story goes. This theory was only testable thanks to the immense amount and accuracy of astronomical data collected with meticulously calibrated tools by Tycho Brahe and Johannes Kepler on the motion of planets across the night sky. It was verified further when the telescope was invented, making it possible for humans to not only see the planets but even track the movement of the moons orbiting them.

Improvements in our ability to measure things allow us to draw stronger conclusions about the accuracy of our predictions and hypotheses. The more data a theory has supporting it, the more likely it is to be true. More accurate measurements also lead to more accurate predictions. For example, predictions about the motion of objects in space are constantly improving, thanks to more

accurate estimations of the gravitational constant in Newton’s law of gravity. This constant is still being tested and updated frequently, and in 2022, its relative uncertainty was only 2.2×10^{-5} [84].

Possibly even more important than the improved predictive accuracy provided by new technology is the capability to observe phenomena that would otherwise be undetectable. For example, the discovery of the Higgs boson, a fundamental particle associated with the mechanism that gives mass to other particles, was only possible because of indirect observations provided by the Large Hadron Collider (LHC). This discovery, like many others in particle physics, would have been impossible with direct sensory observation. Even sense-enhancing tools such as microscopes are incapable of assisting with observations on such a small scale. The shift in empiricism from observation through the senses to indirect observations beyond the grasp of humans was pivotal in the transition to modern physics in the early 1900s.

Observations in modern physics, unlike those in classical physics, often conflict with human intuition. As physicists began to explore realms beyond normal experiences, they uncovered phenomena which were unexplained by classical equations. New theories were necessary to describe these phenomena and, thanks to the availability of advanced equipment, they were testable, making them distinctly different from previous hypotheses pertaining to the same subjects. For example, modern atomic theory, which evolved to include electrons, protons, and neutrons, differs greatly from ancient Greek atomism, which was based on philosophy rather than empirical evidence. It is important to note that despite our theories progressing beyond our everyday human experiences, intuition, and comprehension, their main purpose continues to be rooted in explaining observations.

1.2 History of Quantum Physics

To help set the context for the rest of this thesis, it is fitting to discuss how quantum physics emerged from empirical observations. Quantum physics originated in the late 19th and early 20th centuries when researchers started exploring the properties of light and matter on atomic and subatomic scales.

1.2.1 Quantization

One of the first major departures from classical physics occurred in 1900 when Max Planck attempted to explain black-body radiation, a type of electromagnetic radiation emitted by objects, e.g. infrared light given off by humans or visible light from the Sun. Classical theories, including Newtonian mechanics and Maxwell's equations, failed to properly predict the color spectrum of light that would be emitted from an object. To explain this, Planck came up with the idea that energy is quantized, meaning that it is emitted or absorbed in small indivisible amounts, called 'quanta.' In 1905, Albert Einstein used a similar idea to come up with the photoelectric effect, which describes how materials absorb light in the form of quantized packets called photons. These were the only theories that properly explained many experimental observations that had been puzzling physicists at the time, but their consequence was that a new type of physics had to be formalized; thus, quantum physics became a new field of study.

The idea of quantization proved to be important for other theories in the following years. For example, in 1913, Niels Bohr introduced a new model of the atom, postulating that electrons occupy fixed orbits around the nucleus with quantized angular momentum. The transitions of electrons between these quantized energy levels would cause photons to be emitted at specific frequencies, so this model provided a satisfactory explanation for the observed emission spectrum from hydrogen atoms. However, this raised further questions about the nature of subatomic particles because it implied that electrons would only be observed at certain energy levels and never in-between, while somehow being able to transition between these energy levels. This mysterious instantaneous transition became known as "quantum tunneling".

Experiments continued to be performed to probe the nature of the newly discovered electrons and photons. Initially, theories were formulated on the idea that they were particles that occupy definite amounts of space. But in the famous double-slit experiment performed by Thomas Young on light and Davisson and Germer on beams of electrons, their behavior closely resembled that of waves. In particular, they were able to exhibit constructive and destructive interference, phenomena which only make sense for waves and not for particles. This implied that for photons and electrons,

the models of "wave" and "particle" were appropriate descriptions in different contexts, a concept called wave-particle duality. This meant that a comprehensive description of such small physical things could not be achieved with a single label such as "particle", "wave", or any known model in classical physics. This provided further evidence that an entirely new branch of science was necessary.

1.2.2 Schrödinger's Equation and Cat

In 1926, Erwin Schrödinger introduced his wave equation, marking a significant advancement in quantum mechanics by providing a way to predict the behavior of physical systems at the atomic and subatomic levels. Schrödinger based his equation on the idea that physics at the quantum level is probabilistic in nature. His wave equation models quantum mechanical systems not as particles or waves but as "wave functions", mathematical objects that can be used to calculate the probability distribution of a particle's position and momentum. Solutions to the wave equation correspond to different states in which a system can exist.

In its simplest form, the time-independent Schrödinger equation can be written as follows:

$$\hat{H}\psi = E\psi \tag{1.1}$$

Where ψ (often written as $|\psi\rangle$) is the wave function of the system, E is the energy of the system, and \hat{H} is the Hamiltonian operator (often written as H), a mathematical construct analogous to the classical Hamiltonian in William Rowan Hamilton's formation of classical mechanics, which equals the total kinetic and potential energy of a system. Since ψ is represented by a vector and E a scalar, the Schrödinger equation written in this form is an eigenvalue equation, a type of equation frequently used in linear algebra. The solutions for ψ and E in this equation are called eigenvectors (or eigenstates) and eigenvalues, respectively. When a Hamiltonian in this equation corresponds to a particular physical system, the eigenstates and eigenvalues correspond to the different quantized states in which that system can be observed.

An interesting consequence of this model being a linear equation is that it inherits the property from linear algebra that any sum of eigenvectors is also a valid solution to the Schrödinger equation.

This suggests that new wave functions can be formed by taking sums of other wave functions. In the process, they may interfere constructively or destructively, like waves (hence the name "wave" equation). Since such sums of wave functions are valid solutions to the equation, they also correspond to physical states in which the corresponding system can exist. These states are known as superpositions, so named because they are analogous to the classical phenomenon of wave superposition. When wave functions are combined in this way, the resulting state might not be an eigenstate of the Hamiltonian, which means that superposition states do not necessarily correspond to states that can be physically observed. This discrepancy between theoretical possible states and actual observable states is called the measurement problem, and despite the success of Schrödinger's equation in accurately predicting the outcomes of experiments, it is still the subject of much debate due to its counterintuitive nature.

In order to illustrate the measurement problem, Schrödinger came up with a thought experiment about a hypothetical cat in a closed box. A radioactive atom could be placed next to a vial of poisonous gas in the box so that if a radioactive decay occurs, the gas would be released and the cat would die. There would then be two possible observations that could be made when the box is open: either the atom will have decayed and the cat will be dead, or the atom will not have decayed and the cat will be alive. Such a setup, in which the observation of one component (i.e. the atom) reveals information about another component (i.e. the cat), is referred to as an entangled system. Since atoms exhibit quantum effects, the state of the atom can be described as a wave function, and its two observable states are "decayed" and "not decayed". By the superposition principle, it is possible for the atom to be in a combined state, a superposition, that looks like the sum of these two observable states. Since the state of the cat is entangled with the state of the atom, the cat could also be in a similar state that looks like the sum of its "alive" and "dead" states. This highlights the problem, as it seems impossible for cats to exist in such a state of superposition; they should strictly be either alive or dead.

A popular interpretation of quantum mechanics is the Copenhagen interpretation, which is based on the idea that a system in a superposition state immediately changes into one of the

possible observable states for that system at the moment when a measurement (or observation) is made. Under this interpretation, there is no meaningful way to interpret the superposition state that the cat experiences in Schrödinger's thought experiment. Its state before the box is opened is often paradoxically described as "both alive and dead", which by the definitions of "alive" and "dead" is not logically possible. Other interpretations exist, such as the many worlds interpretation (MWI), which explains the measurement problem by postulating that all possible outcomes of quantum measurements actually occur in alternate universes. Theories like the Copenhagen interpretation and MWI are not falsifiable and thus are philosophical in nature rather than scientific, which can also be seen in how they lead to the same empirical predictions for the outcomes of experiments.

The apparent paradox in the Copenhagen interpretation prompted many scientists, including Albert Einstein and Louis de Broglie, to search for alternative theories that consider new or hidden variables in order to avoid the measurement problem. The presence of such variables could allow the interpretation that quantum systems only physically exist in their observable states, so the apparent probabilistic nature of quantum mechanics would be explained by a lack of complete information by observers. However, it was shown by John Stewart Bell that in any case where such a hidden variable exists, a certain mathematical inequality, now called a Bell inequality, would necessarily be satisfied. Experimental evidence has shown that this inequality is not satisfied for systems exhibiting quantum entanglement, suggesting that quantum systems obey an entirely different kind of logic than classical systems. This discovery led to the foundation of quantum information theory, and the 2022 Nobel Prize in Physics was jointly awarded to Alain Aspect, John Clauser, and Anton Zeilinger for their experiments observing entangled photons violating Bell inequalities [2]. Since logical contradictions are impossible to avoid when interpreting the meaning of entangled superposition states, the straightforward but paradoxical explanation of Schrödinger's cat being "both alive and dead simultaneously" may be the best interpretation we can grasp.

1.2.3 Heisenberg's Matrix Mechanics

Around the same time that Schrödinger introduced wave mechanics, Werner Heisenberg, Max Born, and Pascual Jordan introduced another mathematical formalism to describe quantum me-

chanics, called matrix mechanics. In their formalism, they modeled measurable characteristics (called observables) of quantum systems as matrices, mathematical objects often used in linear algebra. Matrices have characteristic eigenvalues, which can be interpreted as the possible values that are measurable for an observable, and corresponding eigenvectors, which represent the state of the quantum system when that measurement is made. In matrix mechanics, two observables can be combined by multiplying them together. However, matrices have the property that they are not necessarily commutative, meaning that the order in which they are multiplied may affect the result of that multiplication (i.e. $AB \neq BA$). Some observables, such as position and momentum, do not commute with each other, which implies that the order in which they are measured matters for the outcomes of those measurements. This allowed Heisenberg to mathematically derive the uncertainty principle, which states that certain pairs of observables like position and momentum cannot be known simultaneously for a quantum system.

Schrödinger's wave mechanics and Heisenberg's matrix mechanics turned out to be equivalent descriptions of quantum physics, and they were both accurate at predicting the outcomes of experiments. These testable theories were conceived on the basis of empirical evidence and repeatedly withstand rigorous experimentation, which makes them firmly rooted in the scientific method. As more powerful tools are created to probe the extremes of nature, these theories continue to evolve to help us explain and understand the universe in which we live.

1.3 Motivation for Research in Quantum Many-Body Theory

The main objective of the research presented in this dissertation is to enhance the computational methods used to understand quantum systems, especially nuclear systems. The most advanced supercomputers today face many challenges in simulating even relatively simple systems. Therefore, to gain a better understanding of the nature at the nuclear level, it makes sense to explore alternative computing paradigms such as quantum computing. This thesis will concentrate on quantum computing algorithms for a critical aspect of this problem, namely, preparing eigenstates of quantum Hamiltonians. These algorithms will be elaborated on in subsequent sections, but first, we will examine some intriguing problems in nuclear physics and, more broadly, quantum many-body the-

ory, as well as the limitations of current classical methods to justify using this relatively unexplored computing paradigm.

1.3.1 The Strong Force

Our observations of the universe reveal that nature is governed by four fundamental forces: gravity, electromagnetic force, weak force, and strong force. Although gravity and electromagnetism are generally the most familiar, the weak and strong forces are just as critical for the universe's operations. These latter forces dominate on the scale of the atomic nucleus; the weak force is responsible for radioactive processes such as nuclear fission and fusion, while the strong force binds together hadrons, e.g. protons and neutrons, as well as their constituent quarks.

The significance of the strong force is clear, as atoms would not exist without it (making the universe far less interesting!) However, the precise characteristics of the strong force are not yet fully understood. Experimental physicists are actively scientifically testing theories with particle accelerators, such as the heavy-ion accelerator at the Facility for Rare Isotope Beams in East Lansing, to explore the nature of matter at a nuclear level. Meanwhile, theoretical physicists are developing equations to describe the same properties, but there is still no definitive consensus between experimental results and theoretical predictions. To continue increasing our understanding of the strong force, it is likely that more accurate theoretical predictions are needed to align with the experimental results. To this end, theoretical physicists often explore and develop new computational techniques to take advantage of the vast computing resources available today.

1.3.2 Quantum Chromodynamics

One of the most well-supported and developed hypotheses to describe the strong force is the theory of quantum chromodynamics (QCD). QCD explains the observation that hadrons are made of exactly three quarks each by introducing the concept of color charge, where each quark has one of the three properties labeled red, blue, and green. Analogously to the way positive and negative electric charges attract, these colors attract each other in such a way that they tend to form triplets in the case of baryons (e.g. protons and neutrons) and pairs in the case of mesons (e.g. pions).

QCD is theoretically capable of predicting all the properties of strongly interacting particles,

but extracting information from it requires solving mathematical equations which are often very challenging or impossible to solve. The conditions of QCD are incompatible with perturbation theory, a technique that is often useful for simplifying equations describing quantum systems, so the most common technique for approximately solving these equations is the non-perturbative theory of lattice QCD. In lattice QCD, space and time are modeled as discrete rather than continuous, a simplification that allows for approximate solutions to be reached. Its equations are well defined and are therefore known to have solutions, but the process of finding such solutions is still too computationally intensive, even for the largest modern supercomputers, making lattice QCD unusable for understanding even relatively small nuclei.

1.3.3 Nuclear Physics

The further understanding of nuclear systems can lead to important progress in areas of scientific and technological interest. In astrophysics, more accurate modeling of neutron stars, dense celestial objects composed primarily of neutrons, could lead to a better fundamental understanding of how matter behaves under extreme conditions. In nuclear reaction theory, better understanding of the mechanisms in nuclear fusion reactions could lead to breakthroughs in clean energy production, reducing the world's dependence on fossil fuels. There is also motivation in nuclear structure theory because a better understanding of the nuclear landscape could lead to the discovery of novel isotopes which can be used in medical imaging and radiotherapy. These examples highlight the global scale of importance in the advancement of knowledge of nuclear physics.

Unlike the more general theory of QCD which explains forces between quarks, the objective of nuclear physics is to describe the nature of the strong force at just the level of the atomic nucleus, which allows for some simplifications. Quark interactions are prominent at higher energies ($\tilde{1000}$ MeV), while most interactions on the scale of the whole nucleus are important at lower energies ($\tilde{100}$ MeV or $\tilde{10}$ MeV). This discrepancy in scale makes it possible to introduce an effective theory, allowing more accurate predictions at the expense of abstracting the theory further from its physical interpretation. By focusing on systems with lower energy, one can model only the interactions between hadrons and ignore the many smaller interactions among the quarks that make them up.

An example of an effective theory that is useful in the low-energy regime for nuclear physics is Chiral Effective Field Theory (cEFT), which is closely related to the underlying theory of QCD. In cEFT, an assumption is made that particles exhibit chiral symmetry, a property which implies that quarks are massless. This is not the case in QCD in general, but it is approximately true for low-energy systems involving lighter quarks, such as those found in protons and neutrons. This provides a framework for modeling interactions among hadrons in nuclear systems and has been successfully applied to few-nucleon systems [34]. A relatively new approach combines cEFT with a lattice approximation to take advantage of the same computational techniques that were developed to study lattice QCD [72]. This type of approach, which seeks to describe nuclear systems by solving the Schrödinger equation for the nuclear Hamiltonian, is called an *ab initio* method. Other popular *ab initio* methods in nuclear physics include the no-core shell model [85], Green's function Monte Carlo [89], and coupled cluster [50] methods.

Despite the simplifications introduced from the low-energy regime in nuclear physics compared to QCD, many challenges remain to fully understand nuclear phenomena. Even when considering just protons and neutrons, the interaction equations are still complex and generally unsolvable, requiring the use of numerical simulations to find approximate solutions. However, known simulation methods are often inadequate to provide solutions at a useful level of accuracy because of the high computational complexity associated with modeling most nuclear systems.

The complexity of these numerical simulations can be elucidated by the size and number of terms in the quantum Hamiltonian for a nuclear system. The number of interaction terms in a Hamiltonian for any group of interacting particles scales, at minimum, with the number of pairs of particles in the system of interest, which increases quadratically with the number of particles. However, the strong force is known to include many-body interactions, such as the three-body interactions observed in experiments involving the Helium-3 isotope [100]. The number of terms in the Hamiltonian that describe these interactions scales cubically with the number of particles. More importantly, every possible quantum state of the system must be in the domain of the Hamiltonian, so its dimensionality increases exponentially with the number of particles in the system. Consequently, the significant

memory demand associated with storing quantum states often results in simulations running out of available memory.

In light of the vast scientific and technological implications, the search for more sophisticated computational techniques in nuclear physics is a meaningful endeavor. The better understanding of nuclear systems facilitated by these advancements will not only enhance our understanding of fundamental physical laws but may also lead to breakthroughs in diverse fields such as astrophysics, clean energy, and medical technology. Given the inherent complexities and computational challenges outlined in modeling nuclear interactions, it should be worthwhile to invest in innovative computational strategies, including alternative paradigms like quantum computing.

CHAPTER 2

BACKGROUND

The first part of this chapter describes quantum information and computation, a prerequisite for understanding algorithms on quantum computers. The remainder of the chapter discusses eigenvalue problems, traditional methods to solve them on both classical and quantum computers, and several interesting applications that involve solving eigenvalue problems.

2.1 Quantum Information and Computation

Quantum information and computation is a rapidly advancing field at the forefront of modern physics. It studies the computational capabilities of quantum systems, which behave fundamentally differently from classical systems due to quantum effects like superposition and entanglement. This section explores the fundamentals of quantum computation, and specific algorithms relevant to eigenvalue problems are described in more detail in Section 2.2.

In classical computing, the basic unit of information is the bit, a theoretical unit that can exist in one of two distinct states ("binary" states). In silicon-based computers, bits are usually stored in transistors, which act as electrical switches. The two possible states of a transistor correspond to different voltage levels, typically labeled 0 and 1. All data on computers are encoded using these bits and can be manipulated through various electrical components to execute programs. The operations performed using bits adhere to the logical principles established in classical information theory.

Quantum computing explores the consequences of replacing classical bits with their quantum alternatives, called quantum bits or "qubits." A qubit is defined as a quantum system that can be observed in one of two distinct states. As a quantum system, a qubit can therefore also exist in a superposition of its two states, and multiple qubits can be entangled together so that the information stored in them becomes correlated. Even though the binary nature of qubits makes them analogous to classical bits, their quantum nature leads to behavior that has no direct analogy in classical systems. The manipulations that can be done with qubits therefore adhere to an entirely different set of logical principles established in the different framework of quantum information theory.

2.1.1 Qubits

The state of a qubit is represented in the Dirac notation as

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

where $|\psi\rangle$ is the wave function of the qubit, $|0\rangle$ and $|1\rangle$ represent its two observable states, and the coefficients α and β are complex numbers that hold information about the relative probability of measuring the qubit in their respective states. The states $|0\rangle$ and $|1\rangle$ are usually written in vector form as $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ to show that they are orthonormal. These states together form the "computational basis."

Information can be extracted from a qubit through a measurement process, but the measurement result by definition can only be one of the two states $|0\rangle$ or $|1\rangle$. Any information held in superposition states is necessarily destroyed when measurement occurs, a feature called wave function collapse. This type of measurement process (also called "projective" measurement) follows Born's rule, which states that the results $|0\rangle$ or $|1\rangle$ occur with probabilities $|\alpha|^2$ and $|\beta|^2$, respectively, and immediately afterwards the qubit state is reset to the measured state[83].

Theoretically, if one could restrict α and β so that their corresponding probabilities are either 0 or 1, they would be equivalent to classical bits and therefore capable of performing all computations that are possible on a classical computer. In the remainder of this section, we will examine how the capacity of qubits to be in superposed and entangled conditions results in novel and unique computational methods.

2.1.1.1 The Bloch Sphere

The coefficients α and β in Equation 2.1 are only connected to physical meaning by the condition that the square of their magnitudes represents a probability. Since complex numbers have real-valued magnitudes, these coefficients may have imaginary components. When combined with the requirement that these probabilities add up to 1, as dictated by the probability axioms, the space defined by the possible values of α and β corresponds to points on a spherical surface, called the

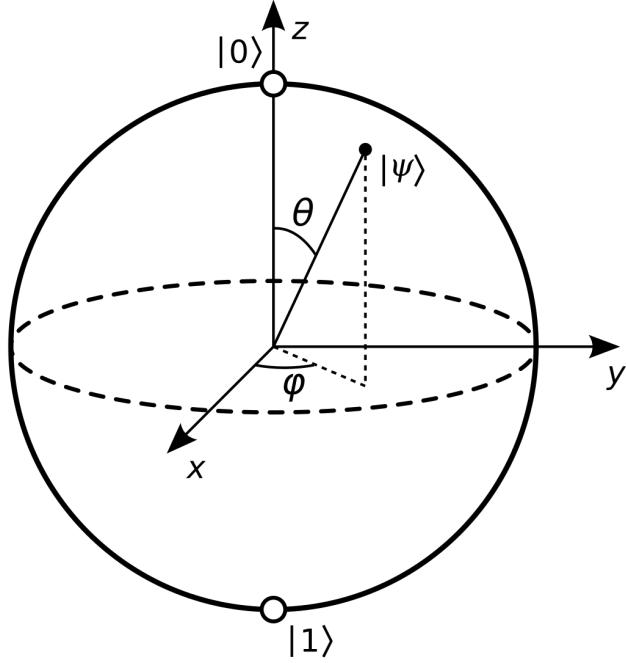


Figure 2.1 The Bloch sphere representing the state of a single qubit as coordinates on a sphere [104]

Bloch sphere (shown in Figure 2.1).

The two observable states $|0\rangle$ and $|1\rangle$ appear at the north and south poles of the Bloch sphere, and each other point on the sphere corresponds to a distinct pure superposition of those two states. Furthermore, the points inside the Bloch sphere, rather than on its surface, represent potential mixed states of the qubit. These mixed states characterize the state of a qubit when its wave function is unknown, a scenario that frequently arises in practice because of unpredictable interactions of quantum systems with their surroundings.

The Bloch sphere is also useful for visualizing operations that can be performed on a qubit. Due to the conservation of probability, all operations that transform a quantum system from one state to another are necessarily unitary operations; the quantum states before and after must have the probabilities of all their possible measurements sum to 1. For a single qubit, the set of all possible transformations is therefore equivalent to the special unitary Lie group $SU(2)$, which is homomorphic to the group of rotations around a sphere, $SO(3)$ [35]. This allows for any operation on a qubit to be visualized as a rotation around the Bloch sphere.4z5

2.1.1.2 Combining Multiple Qubits

Not many interesting computations can be done with a single qubit, so understanding the behavior of multiple qubits is crucial to unlocking the full potential of quantum computation. The combined state for a collection of qubits is represented by a tensor product of the individual states of the qubits. For example, the state of a system of two qubits, each described by a wave function $|\psi_1\rangle$ and $|\psi_2\rangle$, is given by:

$$|\psi_1\rangle \otimes |\psi_2\rangle = |\psi_1\psi_2\rangle. \quad (2.2)$$

The state of this two-qubit system can be expressed as a superposition of the four states:

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

which is referred to as a "product basis". For example, a general two-qubit state $|\Psi\rangle$ can be written as:

$$|\Psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle, \quad (2.3)$$

where α_{00} , α_{01} , α_{10} , and α_{11} are complex coefficients that satisfy the normalization condition:

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1. \quad (2.4)$$

A set of qubits taken together is called the quantum register. The set of possible pure states for a quantum register is a complex linear vector space (called a Hilbert space) spanned by its product basis states. In general, the state of an n -qubit system can be represented as a vector in a 2^n -dimensional Hilbert space [83].

2.1.1.3 Entanglement

Entanglement occurs when the state of one qubit cannot be described independently of the state of another. This phenomenon leads to correlations between measurements of the qubits that are stronger than what is possible in classical systems.

A well-known example of an entangled state is the Bell state, which for two qubits can be written as:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \quad (2.5)$$

In this state, a measurement of the first qubit will yield information about the second qubit, and vice versa. For example, if the first qubit is measured as $|0\rangle$, the second qubit will also be $|0\rangle$, and if the first qubit is $|1\rangle$, the second qubit will be $|1\rangle$. This type of state is considered maximally entangled because a measurement on one qubit provides complete information about the states of both qubits.

Entanglement enables quantum teleportation, a phenomenon in which information about the state of a qubit can be obtained without access to the physical qubit itself. Measurement of one qubit in an entangled pair, for example, causes the wave function of the other qubit to partially collapse (or fully collapse into a basis state if they are maximally entangled). Since entangled qubits can theoretically remain in an entangled state even when separated, there are some situations in which information can be transmitted in fundamentally different ways than when it is stored classically.

The idea that entangled states store information differently is illustrated in many pedagogical examples that have been developed in the new field of quantum game theory. One particularly striking example is the quantum prisoners' dilemma. In the well-known prisoners' dilemma problem, two players each have to decide between options A and B without communicating. If both choose A, they both receive mild punishment; if they both choose B, they both receive harsh punishment; if both players choose different options, the one who chose A receives much harsher punishment, and the one who chose B receives a reward. In the classical scenario, the optimal strategy for players acting in self-interest is to always choose option B, making the case where both players receive harsh punishment a so-called Nash equilibrium [91].

In the quantum prisoners' dilemma, the players are each given exactly one qubit from a pair of entangled qubits and told that the two possible measurements of the qubit correspond to options A and B. They are then allowed to perform any kind of unitary manipulations on their qubit to

influence the relative probability of choosing options A and B. The players are given full information on how the entangled state was prepared, so they could theoretically undo their entanglement and choose their unitary operations to yield 100% chance of choosing one of the options, similar to the classical case. However, the quantum information stored in their qubits starts out slightly correlated, meaning that manipulations on one qubit may affect the wave function of the other qubit, so the players technically share information despite not being able to communicate. It can be shown that, depending on how entangled the qubit is, the Nash equilibrium can be different for this scenario than in the classical case. In such scenarios, it can be shown that there exists a stochastic strategy for players that produces a better outcome on average than the optimal strategy for the classical case [33].

This example helps to illustrate the role of quantum entanglement in information theory. In the quantum prisoners' dilemma, entanglement provides a form of shared information that can be used to inform decisions. This shows how entanglement can be used as a resource, effectively "storing" information in the form of correlations between qubits, rather than actually storing information in the classical sense such as in memory registers.

2.1.2 Quantum Gates and Circuits

In classical computation, information is processed through logic gates (such as the AND and OR gates) that operate on classical bits, changing their values according to predefined rules. Similarly, quantum circuits employ quantum gates to manipulate the quantum states of qubits.

An n -qubit quantum register can be represented as a vector in a 2^n dimensional Hilbert space, so the quantum gates that may act on that quantum register can be represented by unitary matrices in the Lie group $SU(2^n)$. Since unitary matrices are reversible, quantum gates preserve all information within the quantum register. This fundamentally distinguishes them from classical logic gates; for example, the AND gate has two input bits and one output bit, making it impossible to infer the state of the bits before the operation from the output alone.

Multiple quantum gates form a quantum circuit that can then be used as instructions to execute a quantum algorithm. Quantum circuits are often represented visually via quantum circuit diagrams.

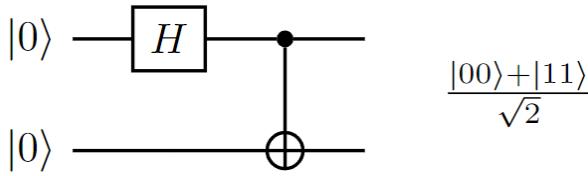


Figure 2.2 A basic quantum circuit diagram for creating a Bell state. The initial state on the left is $|00\rangle$, the gates are applied from left to right, leading to the final state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

The gates in these diagrams are applied sequentially from left to right on the quantum register, starting from an initial state, usually $|0\rangle^{\otimes N}$ (where all qubits are in the $|0\rangle$ state). These operations culminate in a final quantum state which, upon measurement, provides the outcome of the quantum computation.

$$\begin{aligned}
 U3(\theta, \phi, \lambda) &= R_z(\phi)R_x\left(-\frac{\pi}{2}\right)R_z(\theta)R_x\left(\frac{\pi}{2}\right)R_z(\lambda) \\
 &= \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda} \sin\left(\frac{\theta}{2}\right) \\ e^{i\phi} \sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)} \cos\left(\frac{\theta}{2}\right) \end{pmatrix}.
 \end{aligned} \tag{2.6}$$

In a quantum circuit diagram, each qubit is depicted as a horizontal line. Gates can be represented by boxes, and some types of entangling gates can have lines connecting multiple qubits. The meter symbol, often the rightmost symbol on a qubit line, represents a measurement operation.

Because quantum operations are linear, the behavior of a quantum gate is fully characterized by its effect on a collection of basis states. Hence, the operation of each single-qubit gate can be described by its effect on the computational basis, while the operation of each multi-qubit gate can be understood through its effect on the corresponding product basis. The following are some gates that are frequently used in quantum computing algorithms.

The Hadamard Gate (H) is a single-qubit gate that is typically used as the default method for creating superposition states from basis states. The Hadamard gate is represented in the quantum circuit as a boxed "H", as in Figure 2.1.2 and can be written in matrix form as:

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (2.7)$$

This results in a rotation of π radians about the axis $\frac{(\hat{x}+\hat{z})}{2}$ on the Bloch sphere. The output of a Hadamard gate acting on a qubit in either the $|0\rangle$ or $|1\rangle$ state results in an even superposition state on the equator of the Bloch sphere. These states are $\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$ and $\mathbf{H}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$, which are orthonormal and form their own basis in a 2-dimensional Hilbert space. Consequently, \mathbf{H} can be viewed as a basis transformation that interchanges the \hat{x} and \hat{z} axes of the Bloch sphere. Additionally, it is involutory, meaning that it serves as its own inverse. This makes it useful for switching back and forth between the bases $\{|0\rangle, |1\rangle\}$ and $\{|+\rangle, |-\rangle\}$.

Pauli Gates (X, Y, Z) perform rotations of qubit states around different axes on the Bloch sphere.

They can be written in matrix form as:

$$\mathbf{X} = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \mathbf{Y} = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \mathbf{Z} = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The X gate, also known as the quantum bit-flip gate, has the effect $|0\rangle \rightarrow |1\rangle$ and $|1\rangle \rightarrow |0\rangle$, making it the quantum analogue of the classical NOT gate for the computational basis. Similarly, the Y and Z gates cause a quantum state to rotate around the \hat{y} and \hat{z} axes of the Bloch sphere, respectively. They are all involutory and thus serve as their own inverses. A useful property of Pauli matrices is that any 2×2 matrix can be represented as a linear combination of the Pauli matrices and the identity matrix.

Rotation Gates (R_x, R_y, R_z) are operators that can each be written as the matrix exponential of a Pauli gate, i.e. $R_j = e^{-i\sigma_j\theta/2}$, $j \in (x, y, z)$. These operators represent partial rotations around the \hat{x} , \hat{y} , and \hat{z} axes of the Bloch sphere, respectively. The parameter θ dictates how many radians the state rotates, with a value of π making the rotation equivalent to a Pauli gate operation.

Phase Shift (P) gate is written in matrix form as:

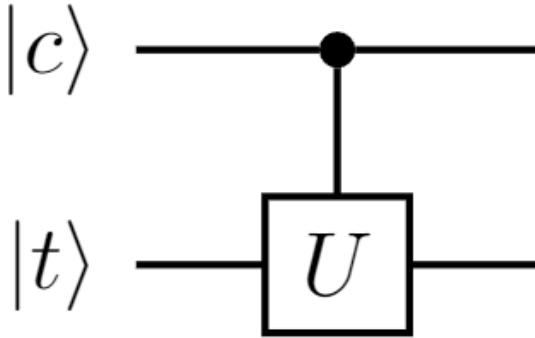
$$\mathbf{P}(\phi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} \quad (2.8)$$

The phase shift operation only affects the $|1\rangle$ state, while leaving the $|0\rangle$ state unchanged. It is equivalent to the R_z gate when $\phi = \pi$. This gate allows for precise manipulation of qubit phases, sometimes creating conditions in which wave functions destructively interfere, which is beneficial for some quantum algorithms and error correction schemes.

Controlled Gates are multi-qubit gates that operate conditionally based on the state of a control qubit. The most commonly used example is the **CNOT** or "controlled-NOT" gate, which can be represented in matrix form as:

$$\mathbf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (2.9)$$

The CNOT gate flips the state of the target qubit if and only if the control qubit is in the $|1\rangle$ state. Specifically, it performs the transformation $|c, t\rangle \rightarrow |c, c \oplus t\rangle$, where $|c\rangle$ and $|t\rangle$ are the control and target qubits, respectively, and \oplus denotes the XOR operation. It is shown on circuit diagrams as a line connecting a solid circle on the control qubit line with a hollow circle on the target qubit line. It is often used as a simple way to create entangled states, as in Figure 2.1.2. To draw other controlled gates, the hollow circle in the diagram can be replaced with any other gate; for example, the general "controlled-U" gate is shown in Figure 2.1.2. For example, if $U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}$, this corresponds to the matrix:



$$\mathbf{CU} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix} \quad (2.10)$$

Toffoli Gate (CCNOT) is a three-qubit gate, also known as the controlled-controlled-not gate. It extends the concept of the CNOT gate by flipping the state of the target qubit only if both control qubits are in the $|1\rangle$ state. The Toffoli gate can be represented in matrix form as:

$$\mathbf{Toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (2.11)$$

In terms of state transformation, the Toffoli gate performs $|c_1, c_2, t\rangle \rightarrow |c_1, c_2, t \oplus (c_1 \cdot c_2)\rangle$, where $|c_1\rangle$ and $|c_2\rangle$ are the control qubits, and $|t\rangle$ is the target qubit. The Toffoli gate is often used in quantum computing to create complex entangled states and implement error correction codes. It

is instrumental in the construction of certain quantum circuits, such as those required for arithmetic operations and fault-tolerant quantum computation.

2.1.2.1 Universality of Quantum Gates

In the context of quantum computation, a set of gates is considered universal if any unitary operation can be approximated to arbitrary accuracy using only gates in that set. This means that any quantum algorithm can be implemented using sequences of these gates. A commonly used set of gates are the rotation gates, the phase shift gate, and the CNOT gate [116]. The Toffoli gate is notable for being sufficient by itself for universal classical computation, and the set of the Toffoli gate and the Hadamard gate is universal for quantum computation [3]. Any physical quantum computing device that can implement a set of universal quantum gates is considered a universal quantum computer. Universal quantum computers are important because they can theoretically be used to execute any kind of quantum algorithm.

2.1.2.2 Extracting information from Quantum Computers

In quantum computers, qubits exist in superpositions which collapse into classical states when measured. Thus, the determination of the quantum state of a quantum register must be done statistically, requiring multiple circuit executions and repeated measurements. Many useful computations on a quantum computer require an iterative process of preparing a quantum register in a particular state, measuring the classical state into which it collapses, and updating a statistical analysis to estimate an expectation value of the quantum state. The precision of this estimation increases with the number of repetitions due to the Law of Large Numbers [98].

The two states that form the computational basis, $|0\rangle$ and $|1\rangle$, are the eigenvectors of σ_z . Measurements on a quantum computer can, therefore, be used straightforwardly to estimate the expectation value of σ_z in the state of the quantum register. To measure the expectation value of σ_x in the state of a qubit, one can apply a Hadamard gate prior to measurement. Similarly, the expectation value of σ_y can be obtained by first executing a $\frac{\pi}{2}$ phase shift gate, followed by a Hadamard gate, before measurement. These three types of measurements are called Pauli measurements. They can be thought of as measurements in the bases **X**, **Y**, and **Z** of a qubit, and the gates applied before

measurement can be thought of as rotations into their respective basis, starting from the **Z** basis.

Quantum state tomography, the process of approximately reconstructing a quantum state based on measurements, can be performed by using the three Pauli measurements on an ensemble of identical quantum registers. This approximation can be made more efficient by using different measurement techniques other than Pauli measurements [44]. Some algorithms are designed to directly estimate desired quantities without the need for tomography; for example, Grover's search algorithm uses only one measurement on each qubit at the end of a circuit [46].

In the context of eigenvalue problems on quantum computers, it is often desirable to estimate the expectation value of a Hamiltonian with respect to the wave function stored in a quantum register. This value, written as $\langle \psi | H | \psi \rangle$ can be thought of as the "energy" of the quantum state ψ within the quantum system described by H . This expectation value can be obtained straightforwardly when the Hamiltonian is expressed as a sum of tensor products of Pauli matrices, i.e. $H = \sum c_j (\sigma_0 \otimes \sigma_1 \otimes \cdots \otimes \sigma_n)_j$ where each σ_i is in the set $\{\sigma_x, \sigma_y, \sigma_z, I\}$ and n is the number of qubits in the quantum register. It is theoretically possible to represent any arbitrary Hamiltonian in this form because these strings of tensor products form a basis for the Hilbert space of the Hamiltonian. Due to the linearity of quantum mechanics, the energy can then be expressed as the sum of the expectation values of these Pauli strings, which can be estimated by taking the corresponding Pauli measurements of each qubit for each Pauli string and summing them together.

2.1.3 Quantum Hardware and Limitations

In the ongoing quest for quantum advantage, a diverse array of quantum hardware architectures have been developed, each with its unique attributes and inherent limitations. The types of quantum computers available in the current era can be divided into two categories: universal quantum computers and analog quantum computers. Universal quantum computers manipulate physical quantum states to perform gate operations and execute arbitrary quantum algorithms. For example, superconducting quantum computers manipulate quantum states found in Josephson junctions[68], and trapped-ion quantum computers manipulate the electronic states of ions held in place by electromagnetic fields [16]. Analog quantum computers forego the circuit model in order to

leverage quantum effects of certain physical systems to perform some subset of quantum computing operations more efficiently. For example, to more efficiently solve optimization problems, quantum annealers take advantage of quantum tunneling effects [119], and neutral atom arrays take advantage of condensed matter physics in Rydberg atoms [14].

Despite the promising capabilities of these quantum hardware architectures, they are not without limitations. Superconducting quantum computers, while scalable and compatible with existing semiconductor fabrication techniques, are highly sensitive to environmental noise. This sensitivity results in short coherence times and necessitates operation at cryogenic temperatures, presenting significant engineering challenges. Trapped-ion quantum computers offer longer coherence times due to their isolation from the environment, but face challenges in scaling up while maintaining connectivity between qubits, as the interaction between ions decreases with distance, making high-fidelity operations difficult as the system size expands. Analog quantum computers, including quantum annealers and Rydberg atom array computers, offer unique advantages, but also face their own set of challenges. Quantum annealers have limited application scope and are susceptible to thermal noise, leading to potential inaccuracies in solutions. Rydberg atom array computers, while offering high connectivity between qubits and long coherence times, face difficulties in precisely controlling the interactions between atoms and potential issues with scalability.

While researchers are working to mitigate the limitations of quantum hardware, it is still worthwhile to make use of the currently available technology by developing algorithms that can produce interesting results despite high decoherence times and environmental noise. This is the essence of the Noisy Intermediate-Scale Quantum (NISQ) era, a term coined by John Preskill to describe the current state of quantum computing technology [92]. NISQ devices, which may possess between 50 and a few hundred qubits, are not yet error-corrected and suffer significant operational errors. However, they represent the best quantum computing resources currently available; therefore, there is a pressing need for quantum algorithms that can operate effectively within the constraints of these devices. Such algorithms must be robust to noise and capable of providing a quantum advantage even with a limited number of qubits. This is a challenging

task, as many quantum algorithms proposed thus far require error correction or a large number of qubits to provide a significant advantage over classical computers. For this reason, there is a growing interest in hybrid algorithms that combine techniques from currently available quantum and classical hardware to surpass the limitations on both sides.

2.1.3.1 Optimal Control for Quantum Gates

Quantum computing algorithms are generally implemented using one- and two-qubit gates, and a many-qubit algorithm can be executed precisely as a sequence of these gates [103, 12]. However, the number of gates required for complex algorithms can become very large, and the short coherence times of NISQ devices impose a strict limit on the runtime of an algorithm before the output becomes dominated by noise. In order to optimize the performance of some algorithms, it is possible to create custom gates customized to the specific hardware and algorithm using optimal control theory [54]. Gates on physical hardware are often constructed from analog signals (e.g. microwave pulses on superconducting transmons), making this approach similar to hybrid digital-analog algorithms.

2.1.3.2 Qudits

Qudits extend the concept of qubits to quantum systems with more than two possible states, allowing for more complex state representations in a higher-dimensional state space. Some physical systems used to create qubits in quantum computers are amenable to having more than two possible states when measured. For example, superconducting qubits use the energy levels of a transmon to create basis states for quantum computing, and it is possible for a transmon state to be in a superposition of more than two distinct states. This can allow more quantum information to be stored in the same amount of hardware, possibly leading to more efficient algorithms when combined with optimal control techniques to implement multilevel quantum gates [118].

2.2 Eigenvalue Problems and Methods for Solving Them

The general form of an eigenvalue problem is as follows:

$$A\mathbf{v} = \lambda\mathbf{v} \tag{2.12}$$

Where A is a matrix, \mathbf{v} is an unknown vector, and λ is an unknown scalar. The solutions for \mathbf{v} and λ are called eigenvectors and eigenvalues, respectively.

Eigenvalue problems are prevalent in various disciplines, including nuclear physics, chemistry, and optimization. Eigenvalues and eigenvectors can yield important insights into underlying systems, such as the energy levels of a quantum system or optimal solutions to real-world problems. However, tackling these issues, especially for large and complicated systems, can be computationally demanding. In nuclear physics and chemistry, relevant eigenvalue problems take the form of solving the Schrödinger equation for a nuclear or molecular Hamiltonian. In combinatorial optimization, eigenvalue problems arise in graph theory, where the largest eigenvalue of a graph's adjacency matrix reveals structural information like maximum degree.

In this section, we will discuss various known computational techniques in both the classical and quantum computing regimes that aid in solving for eigenvalues and eigenvectors.

2.2.1 Classical Computational Techniques

Classical techniques for solving eigenvalue problems include direct diagonalization and iterative methods. Numerical diagonalization involves the transformation of matrices into diagonal forms to reveal eigenvalues using algorithms such as the QR or Jacobi methods [10]. These straightforward methods, while effective for small to medium-sized matrices, become computationally intensive for very large or dense matrices, requiring $O(N^3)$ operations for a $N \times N$ matrix. Considering that the size of the quantum Hamiltonian matrix scales exponentially with the number of particles in the quantum system, this becomes intractable for many reasonably-sized problems. For example, modeling nuclei requires 4^A states to fully describe the spin-isospin component, where A is the number of nucleons. Thus, solving the Schrödinger equation for most nuclear Hamiltonians requires the use of iterative methods, which typically compute only a few eigenvalues and eigenvectors rather than the full spectrum.

In this section, we will explore some iterative methods for eigenvalue problems, relating each method to how it can be used in nuclear physics.

2.2.1.1 The Power Method

The Power Method is one of the simplest iterative techniques for finding the dominant eigenvalue and its corresponding eigenvector of a matrix. The method involves repeatedly multiplying an arbitrary non-zero vector by the matrix and normalizing the result. Mathematically, this process can be expressed as:

$$\mathbf{v}_{k+1} = \frac{A\mathbf{v}_k}{\|A\mathbf{v}_k\|}$$

where \mathbf{v}_k is the vector at iteration k , and A is the matrix whose eigenvalues are desired. After many iterations, \mathbf{v}_k converges to the eigenvector associated with the largest eigenvalue of A .

In nuclear physics, the Power Method can be used to estimate the ground state energy of a system, which corresponds to the most extremal eigenvalue of the quantum Hamiltonian. Although the Power Method is simple and easy to implement, it converges slowly, especially if the dominant eigenvalue is not well separated from the other eigenvalues. Thus, its primary use is as a pedagogical tool to explain how iterative methods can converge on a single eigenvalue more efficiently than direct diagonalization.

2.2.1.2 The Lanczos Algorithm

In many nuclear physics applications, the Hamiltonian matrix is sparse, which means that most of its elements are zero. This sparsity arises because interactions in nuclear systems are typically local, involving only a limited number of neighboring particles or basis states. As a result, each row and column of the Hamiltonian matrix contains only a few nonzero elements. This sparsity makes certain iterative algorithms, such as the Lanczos algorithm, more effective.

The Lanczos algorithm is an iterative method that is particularly effective in determining a few eigenvalues and eigenvectors of large sparse matrices [51]. It simplifies the original matrix to a significantly smaller tridiagonal matrix, which is easier to solve. The steps of the Lanczos algorithm can be outlined as follows:

1. Begin with an initial vector \mathbf{v}_1 (often chosen randomly or based on some heuristic) and normalize it. Set $\beta_0 = 0$ and $\mathbf{v}_0 = 0$.

2. For $j = 1, 2, \dots, k$:

- a) Compute $\mathbf{w}_j = A\mathbf{v}_j - \beta_{j-1}\mathbf{v}_{j-1}$.
- b) Compute $\alpha_j = \mathbf{v}_j^T \mathbf{w}_j$.
- c) Update \mathbf{w}_j to $\mathbf{w}_j = \mathbf{w}_j - \alpha_j \mathbf{v}_j$.
- d) Orthogonalize \mathbf{w}_j with respect to all previous \mathbf{v}_i (usually done implicitly).
- e) Compute $\beta_j = \|\mathbf{w}_j\|$.
- f) Normalize \mathbf{w}_j to get the next vector $\mathbf{v}_{j+1} = \frac{\mathbf{w}_j}{\beta_j}$.

3. Construct the tridiagonal matrix T with diagonal entries α_j and off-diagonal entries β_j .

This process generates an orthogonal basis for the Krylov subspace $\mathcal{K}_k(A, \mathbf{v}_1)$, and the solutions for the eigenvalue problem for the matrix T can be used to approximate the eigenvalues and eigenvectors of the original matrix A . The algorithm is particularly useful because it significantly reduces the size of the problem compared to direct diagonalization, making it computationally efficient for large sparse matrices.

2.2.1.3 Monte Carlo

Monte Carlo algorithms, named after the city of Monte Carlo in Monaco known for gambling, use repeated random number generation to estimate quantities of interest. For example, they can be used to estimate the value of an integral of a function without requiring the knowledge of an analytical solution for that integral. For a function of one variable $f(x)$, the integral represents the area under the curve, which can be approximated for a particular interval by choosing random points within the domain and range of $f(x)$ and checking if each point is above or below $f(x)$. In the context of eigenvalue problems, Monte Carlo techniques can be used to estimate the properties of large matrices or to solve integrals that arise in the computation of matrix elements.

Although the Monte Carlo method is capable of handling large systems, it encounters the notorious "sign problem" when applied to quantum systems. This issue arises from the necessity to sample from a distribution that includes both positive and negative values, leading to cancellations

that cause the statistical error to grow exponentially with the size of the system. The sign problem is particularly an issue in simulations involving fermions due to the antisymmetrization requirement of their wave functions, which results in probability distributions that are not always positive definite.

2.2.1.4 Eigenvector Continuation

Several methods have been developed to overcome the sign problem of the Monte Carlo method when used in the context of lattice simulations for fermionic systems [30, 1, 70, 41]. One such method that has been successful recently is Eigenvector continuation (EC) [40].

EC is a variational method for finding extremal eigenvalues and eigenvectors of a Hamiltonian. It involves projecting the changing extremal eigenvector of a Hamiltonian onto a smaller subspace, reducing the dimensionality. This projection speeds up the estimation, making it versatile for various problems. By projecting the Hamiltonian onto a subspace of eigenvectors from selected training parameters and solving the generalized eigenvalue problem, we get a low-dimensional approximation of the true eigenvector. The main advantage of EC is its ability to extrapolate to regions which are inaccessible by direct calculation, similar to the technique of analytic continuation in the complex plane.

2.2 Quantum Computational Techniques

In this section, we will discuss various techniques that have been developed that utilize quantum computing resources to solve eigenvalue problems.

2.2.2.1 Variational Quantum Eigensolver

The Variational Quantum Eigensolver (VQE) algorithm is a hybrid quantum-classical method specifically designed to mitigate the challenges of quantum computation posed by noise and other physical limitations in the NISQ era [76]. Primarily used to find the ground-state energy of a quantum mechanical system, the algorithm combines the computational power of quantum systems with the robustness of classical optimization algorithms.

The central idea of VQE is the variational principle of quantum mechanics, which is the observation that the expectation value of the Hamiltonian, $\langle \psi | H | \psi \rangle$, for any state $|\psi\rangle$ is always greater than or equal to the ground state energy E_0 of the system. The VQE algorithm takes

advantage of this principle to approximate the ground state.

The VQE algorithm starts by preparing a quantum state $|\psi(\theta)\rangle$, also known as an ansatz, which is parameterized by a set of classical variables θ . This state is typically prepared using a specialized quantum circuit on a quantum computer. The selection of ansatz can influence the algorithm's performance, and it is typically constructed to reflect some of the known physics of the problem. For example, a commonly used ansatz is the Unitary Coupled Cluster ansatz, which is notably effective for simulating molecular systems because it can include electronic correlations [6]. Another example is the Hardware Efficient Ansatz, created to exploit the specific features of quantum hardware for more efficient implementation [61].

The algorithm proceeds by estimating the expectation value $\langle\psi(\theta)|H|\psi(\theta)\rangle$ of the Hamiltonian using the quantum computer. The calculated expectation value acts as an objective function for a classical optimization process that iteratively adjusts the parameters θ . This optimization loop continues to lower the expectation value until a predefined stopping condition is met. These procedures form a quantum-classical cycle: The quantum computer sets up the state and computes the expectation value, while the classical computer updates the parameters.

This algorithm can be used to determine the lowest eigenvalue of a Hamiltonian, which frequently corresponds to the ground-state energy of a quantum system of interest. This is of interest for nuclear and molecular Hamiltonians because physical systems naturally gravitate towards their ground state, often making them the most important states to understand. However, often the higher-energy eigenstates are also of interest, but since VQE is based on the variational principle, it is incapable of solving for such excited states.

2.2.2.2 Quantum Adiabatic Evolution

One approach to prepare an arbitrary state on a universal quantum computer with arbitrary precision (or "fidelity") is to simulate the process of adiabatic evolution starting from an easily accessible state. This method is based on the adiabatic theorem in quantum mechanics, which states that a system will stay in its instantaneous eigenstate as long as the perturbation influencing it changes gradually enough and there is a gap between the eigenvalue and the rest of the spectrum.

For example, if a quantum system is initialized in the ground state of a basic Hamiltonian, and the time evolution of the state of the quantum register is simulated while the Hamiltonian is gradually transformed into a complex one with the desired ground state, the system will remain in the ground state throughout the evolution.

Although adiabatic evolution can theoretically generate any eigenstate of a quantum Hamiltonian, its practical implementation on NISQ devices is challenging. Achieving a desired state with high precision may necessitate a large number of quantum gates, resulting in longer circuit execution times and a higher probability of qubits decohering due to undesired interactions with the environment. This can cause the evolution to deviate significantly from its intended trajectory toward the target state.

A more detailed description of the Quantum Adiabatic Evolution algorithm is given in Chapter 3, along with a demonstration of an implementation with custom time evolution gates created with optimal control.

2.2.2.3 Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm[36] and the very similar Quantum Alternating Operator Ansatz[49] (both called QAOA) are quantum computing algorithms closely related to VQE and Quantum Adiabatic Evolution. Like VQE, QAOA is a hybrid quantum-classical algorithm designed to take advantage of the computational advantages of quantum systems while mitigating their limitations. It uses the same variational approach as VQE but with a particular ansatz that is physically motivated by the adiabatic theorem, making it also closely related to quantum adiabatic evolution.

QAOA begins with a simple quantum state, typically a uniform superposition of all possible solutions, achieved by applying a Hadamard gate to each qubit. The algorithm then applies a sequence of unitary evolutions to this state, which perform the action of the time evolution operator $U = e^{-iHt}$, alternating H between two Hamiltonians, the cost (or "problem") Hamiltonian H_f and the mixer Hamiltonian H_i . The values of t in the time evolution operators are given by two sets of parameters, β for the time evolutions on H_i and γ for the time evolution on H_f . These evolution

operations are alternated for p "layers" (also called the "depth") to form the ansatz. Increasing the depth theoretically leads to higher fidelity of state preparation at the expense of increased risk of decoherence due to longer circuit execution times.

Similarly to VQE, the expectation value of the cost Hamiltonian with respect to the state of the quantum register at the end of the circuit is typically used as the cost function for a classical minimization to update the parameters β and γ . This forms a quantum-classical feedback loop, iteratively optimizing the parameters until the final state with the lowest energy value is found. By the variational principle, this state approaches the ground state of the cost Hamiltonian.

Similarly to the adiabatic evolution algorithm, QAOA can be thought of as an alternative way to discretize the adiabatic process into a sequence of unitary evolution operations. The difference is that, instead of a continuous and slow evolution between two Hamiltonians, QAOA performs a series of abrupt changes between those two Hamiltonians. This approach allows QAOA to approximate the adiabatic path in a stepwise manner, potentially achieving the same result with fewer resources and in a shorter time.

2.2.2.4 Quantum Phase Estimation

The Quantum Phase Estimation (QPE) algorithm is a fundamental quantum algorithm used to estimate the eigenvalues of a unitary operator. In the context of nuclear physics, unitary operators of interest often represent the time evolution operator $U = e^{-iHt}$ (which has the same eigenstates as the Hamiltonian H). Given an initial quantum state that has a non-zero overlap with a particular eigenstate of this unitary operator, the QPE algorithm estimates the phase (and thus the corresponding eigenvalue) with high precision.

The QPE algorithm works as follows:

1. **Initial State Preparation:** The algorithm begins with an initial state composed of two registers, each: an ancillary register prepared in a superposition state using the Hadamard gates, and a system register initialized in a state $|\psi\rangle$ that has a nonzero overlap with an eigenstate of the unitary operator U .

2. **Controlled Unitary Operations:** Controlled applications of the unitary operator U are performed, where each qubit in the ancillary register controls the application of U on the system register, repeated 2^j times, where j is the index of the ancilla qubit. This step encodes the phase information into the ancillary register.
3. **Quantum Fourier Transform (QFT):** After the controlled unitary operations, a Quantum Fourier Transform (QFT) is applied to the ancillary register. This step transforms the phase information from the time domain to the frequency domain, allowing the extraction of the phase.
4. **Measurement:** Finally, the ancillary register is measured, yielding an estimate of the phase ϕ . This phase is related to the eigenvalue λ of the unitary operator U by $U|\psi\rangle = e^{2\pi i \phi}|\psi\rangle$.

In addition to the standard QFT-based QPE, there are several iterative versions of the QPE algorithm that can offer advantages in terms of circuit depth and resource requirements. These iterative methods, such as Kitaev's iterative phase estimation [66], involve a sequence of measurements and classical feedback to refine the estimate of the phase ϕ .

Iterative QPE estimates the bits of the phase ϕ sequentially. This method uses fewer qubits, reducing the overall complexity of the quantum circuit. The technique involves initially focusing on the most significant bit, where the algorithm determines each bit of the phase through a series of controlled operations and measurements. The outcome of each measurement is then used to adjust the following controlled operations, iteratively refining the phase estimate. The accuracy of the phase estimation can be tuned by the number of iterations, potentially enabling eigenvalues to be estimated with higher precision.

2.2.2.5 Quantum Amplitude Estimation

Quantum Amplitude Estimation (QAE) is a quantum algorithm that extends the principles of QPE to estimate the amplitude of a specific quantum state within a superposition. It has been shown to provide a quadratic speedup over classical Monte Carlo methods in terms of the number of samples required to achieve a given precision [21].

The QAE algorithm works as follows:

1. **Initial State Preparation:** The algorithm begins by preparing an initial quantum state. This involves two main steps:
 - a) *State Preparation:* A unitary operator A is applied to the initial state $|0\rangle$ to prepare the state $A|0\rangle = \sqrt{1-a}|0\rangle + \sqrt{a}|1\rangle$, where a is the amplitude to be estimated.
 - b) *Ancillary Qubit:* An ancillary qubit is prepared in the superposition state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
2. **Quantum Operator Construction:** Define the Grover operator $Q = AS_0A^\dagger S_\chi$, where S_0 and S_χ are reflection operators. S_0 inverts the sign of the amplitude of the $|0\rangle$ state, and S_χ inverts the sign of the amplitude of the states marked by the oracle function χ .
3. **Amplitude Amplification:** Apply the Grover operator Q iteratively. The number of applications Q^k increases the amplitude of the target state quadratically with each iteration. The amplitude of the target state becomes $\sin((2k+1)\theta)$, where θ is related to the amplitude a by $\sin^2(\theta) = a$.
4. **Phase Estimation:** Use the QPE algorithm to estimate the eigenvalues of the Grover operator Q . This involves applying controlled applications of Q and performing a Quantum Fourier Transform (QFT) on the ancillary register to extract the phase θ .
5. **Measurement and Classical Post-processing:** Measure the ancillary qubits to obtain an estimate of the phase θ . Using classical post-processing, convert the phase θ into an estimate of the amplitude a . The relationship $a = \sin^2(\theta)$ allows for the extraction of the amplitude from the measured phase.

The number of quantum operations required for this method scales as $O(1/\epsilon)$, where ϵ is the desired precision, compared to the scaling $O(1/\epsilon^2)$ for the number of classical operations in Monte Carlo methods, representing a quadratic improvement in time complexity. This efficiency

makes QAE a powerful tool for problems involving probability estimation and expectation value calculations in quantum computing.

There are various modifications and enhancements to the standard QAE algorithm. One notable example is the Maximum Likelihood Amplitude Estimation (MLAE)[107] which refines the amplitude estimate using maximum likelihood methods. Instead of directly using the phase estimates to determine the amplitude, the measurement data is fed into a likelihood function that models the probability of observing the data given different possible amplitudes. The likelihood function can incorporate various sources of noise and uncertainty, potentially increasing the robustness and accuracy of the estimate.

2.2.2.6 Rodeo Algorithm

The Rodeo algorithm (RA) is an innovative quantum computing approach based on the suppression of the probability amplitudes of all eigenvectors except the one of interest. Applicable to any quantum Hamiltonian, its recursive nature allows for exponential convergence with increasing cycles and can be utilized for full-energy spectrum computation and the preparation of any energy eigenstate.

A detailed explanation and a demonstration of the effectiveness of RA are explored as the main subject of Chapter 4. A novel variational method combining RA and QAOA to prepare eigenstates more efficiently is the subject of Chapter 5.

2.2.3 Classical Optimization Techniques

As noted in Section 2.2.2, the preparation of eigenstates on quantum computers is frequently achieved through variational methods such as VQE, QAOA, or the novel Variational Rodeo Algorithm discussed in Chapter 5. These techniques require classical optimization in tandem with quantum circuit operations. Solving such optimization problems can be particularly challenging largely due to the high dimensionality of the problem space, leading to complications such as local minima and barren plateaus.

In this section, we will explore several classical optimization methods that are commonly used in combination with these variational quantum algorithms. These methods are theoretically

interchangeable in many applications of variational methods, although some work more efficiently than others. It is not necessary to understand how they work in order to successfully apply them to variational algorithms, but a short description of each approach is included for the curious reader.

2.2.3.1 Gradient Descent

Gradient Descent is a key optimization algorithm, used to find the global minimum of an objective function. It iteratively adjusts the parameters in the direction of the steepest decrease, defined by the negative gradient at the current point. The update rule is:

$$\theta_{t+1} = \theta_t - \eta \nabla f(\theta_t)$$

where η is the learning rate and $\nabla f(\theta_t)$ is the gradient at θ_t . Despite its simplicity, this method has limitations in high-dimensional spaces, such as in quantum variational algorithms. The convergence rate depends on selecting an appropriate learning rate. It can get trapped in local minima, saddle points, or areas of near-zero gradients that hinder optimization, known as barren plateaus.

2.2.3.2 Newton's Method

Newton's method is a numerical root-finding algorithm adapted for optimization. It uses the second-order Taylor expansion to find minima of functions. The update rule is:

$$\theta_{t+1} = \theta_t - H^{-1} \nabla f(\theta_t)$$

where H is the Hessian matrix of second-order partial derivatives. This method can converge faster than Gradient Descent near the optimum due to second-order information. However, its practicality in quantum optimization is limited. Computing the Hessian and its inverse is costly in high-dimensional spaces. If the Hessian isn't positive definite, convergence to a saddle point or divergence may occur. Due to these difficulties, combined with the complexity of cost functions in quantum optimization, this method is rarely used in practice, but it is still useful in the pedagogical sense to show how optimization techniques can incorporate second-order corrections to the gradient.

2.2.3.3 Broyden-Fletcher-Goldfarb-Shanno (BFGS) Algorithm

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is a popular quasi-Newton method for unconstrained optimization [22]. Unlike Newton's method, which requires computing the

Hessian matrix, BFGS approximates the inverse Hessian using gradients. The parameter update for θ is:

$$\theta_{t+1} = \theta_t - \eta B_t \nabla f(\theta_t)$$

where B_t is iteratively updated using gradients and parameters from previous iterations. BFGS is efficient in high-dimensional spaces and avoids the computational burden of full Hessian calculations, making it suitable for variational quantum algorithms. It often converges faster than Gradient Descent, especially when the cost function is smooth [47].

BFGS is effective in optimizing parameters for variational quantum algorithms such as VQE and QAOA, navigating high-dimensional landscapes more effectively than simpler gradient methods. Its adaptive Hessian refinement helps mitigate issues like barren plateaus by using curvature data to direct the optimization.

However, BFGS can be sensitive to the initial parameters and the nature of the cost function. Some hybrid methods have been proposed that combine BFGS with other strategies to improve robustness and convergence [57, 9].

2.2.3.4 Nelder-Mead Method

The Nelder-Mead method is a derivative-free optimization algorithm that works well for functions that are not smooth or lack derivatives [82]. This makes it a common choice for variational algorithms in which the derivative of the quantum circuit with respect to its circuit parameters may be difficult to compute. It uses a simplex, a geometric figure of $n + 1$ vertices in n -dimensional space, to iteratively search for the minimum. The algorithm performs the following operations at each step to move the simplex towards the minimum:

1. **Reflection:** Reflect the worst vertex through the centroid of the remaining vertices.
2. **Expansion:** If reflection improves the value of the function, expand further in that direction.
3. **Contraction:** If reflection does not improve, contract the simplex around the best vertex.
4. **Shrinkage:** If contraction fails, shrink the entire simplex towards the best vertex.

This method is advantageous in high-dimensional optimization where gradient information is unavailable or unreliable. However, it can be slow and may not converge to the global minimum, especially in high-dimensional spaces common in variational quantum algorithms on large sets of qubits.

2.2.3.5 Conjugate Gradient Method

The Conjugate Gradient method is an optimization algorithm that is particularly effective for large-scale linear systems originally invented to minimize quadratic functions [97]. It is an iterative method that improves upon Gradient Descent by considering the previous search direction, creating conjugate directions for faster convergence. The update rule is:

$$\theta_{t+1} = \theta_t + \alpha_t p_t$$

$$p_{t+1} = -\nabla f(\theta_{t+1}) + \beta_t p_t$$

where p_t is the conjugate direction, α_t is the step size, and β_t is computed to ensure conjugacy. This method converges in at most n steps for n -dimensional quadratic functions, but is more efficient for other functions as well. Its efficiency and relatively low memory requirement make it suitable for high-dimensional problems encountered in variational quantum algorithms, although some problems tackled by these approaches are nonlinear, which may lead to a decreased efficiency of the CG method compared to quadratic functions [75].

2.2.3.6 Simulated Annealing

Simulated Annealing is a probabilistic optimization technique inspired by the annealing process in metallurgy [63]. It aims to find a global minimum by allowing occasional uphill moves to escape local minima. The algorithm uses a temperature parameter that starts high and gradually cools down. The steps are:

1. **Initialization:** Start with an initial solution and temperature.
2. **Perturbation:** Modify the current solution slightly to generate a new candidate solution.

3. **Acceptance Criterion:** Accept the new solution if it improves the objective function. If not, accept it with a probability that decreases with temperature.

4. **Cooling Schedule:** Gradually reduce the temperature according to a predefined schedule.

The probability of accepting worse solutions decreases over time, allowing the algorithm to explore the solution space widely at high temperatures and focus on local improvements at low temperatures. This is particularly useful for problems with many local minima, such as those in variational quantum algorithms. However, it may not be obvious how to properly choose the cooling schedule so that it leads to better convergence.

2.2.3.7 Adam Optimization Algorithm

The Adam (Adaptive Moment Estimation) optimization algorithm, an advanced version of stochastic gradient descent, combines the adaptive learning rate of AdaGrad and the momentum of RMSProp [62]. The update rules are:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla f(\theta_t)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla f(\theta_t))^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

m_t and v_t are the first and second moment estimates of the gradients, β_1 and β_2 are the decay rates, η is the learning rate, and ϵ is a small constant to prevent division by zero.

Adam's adaptive learning rate is effective for training models with noisy and sparse gradients, which are common in high-dimensional optimization like variational quantum algorithms. It adjusts step sizes to facilitate efficient convergence compared to conventional methods. Its momentum reduces noise and fluctuations in gradient updates, speeding up optimization and preventing getting stuck in local minima or saddle points, which is useful for cost functions with complicated landscapes.

However, while Adam is highly effective for machine learning tasks due to its robustness and adaptive nature, it is not used as often for variational methods. Currently, machine learning cost functions have much higher dimensionality than those generated from quantum circuits, meaning that methods like Adam which are optimized to work well in such regimes may not be as effective as other optimization algorithms in variational methods. Many variational circuits have cost functions with gradients that can be computed analytically, such as with the stochastic parameter shift rule [11], which may allow them to take advantage of simpler methods.

2.3 Applications of Eigenstate Preparation

Though the main objective of this research is to improve computational methods to solve eigenvalue problems to better understand nuclear systems, there are many other practical applications that accompany these advances. This section will describe several other fields that share an interest in better eigenstate preparation techniques.

2.3.1 Quantum Simulation

A prime example of a case where quantum computation has the potential to efficiently address problems deemed intractable for classical computing systems is the simulation of the dynamics of large quantum systems. The resources required to solve this issue through classical computing grow exponentially in relation to the size of the system N . Conversely, a universal quantum computer can achieve a solution with resources that scale linearly with N , under the condition that local interactions drive the system's evolution [73, 38]. Before starting the simulation of a system, it is often beneficial to prepare a quantum register in a specific eigenstate of a particular Hamiltonian that represents that system. Quantum systems of practical and theoretical interest to simulate include systems in nuclear physics, chemistry, QCD, and many other fields.

2.3.2 Condensed Matter Physics

Condensed matter physics encompasses a wide array of models that describe the collective behavior of many-body systems. Eigenstate preparation plays a crucial role in understanding these models as it allows for a detailed examination of ground states and excited states, facilitating insights into various physical phenomena.

One prominent example is the transverse field Ising model, which serves as a prototypical system for studying quantum phase transitions [39]. Preparation of the system in its ground state enables exploration of critical behavior and scaling properties. The Heisenberg model, another fundamental model in condensed matter physics, describes interactions in magnetic systems; its eigenstates yield insights into magnetic excitations and spin dynamics [13]. In the context of strongly correlated electron systems, the Fermi-Hubbard model is a key model for understanding high-temperature superconductivity and Mott insulator transitions [102]. Preparing eigenstates of this Hamiltonian allows for the analysis of electron correlations and pairing mechanisms. Similarly, the Bose-Hubbard model describes bosonic particles in an optical lattice and is instrumental in studying superfluid-insulator transitions [43]. Its eigenstates illuminate quantum phase transitions and coherence properties in bosonic systems.

2.3.3 Quantum Chemistry

In quantum chemistry, eigenstate problems appear in the form of solving the Schrödinger equation for electronic and vibrational structure Hamiltonians of molecules. Accurate eigenstate preparation can allow for calculation of particular molecular properties, reaction pathways, and potential energy surfaces, which are fundamental to understanding chemical reactions and designing new materials.

For electronic structure calculations, obtaining the ground state and low-lying excited states of a molecular Hamiltonian enables the determination of properties such as ionization energies, electron affinities, and dipole moments. Furthermore, the full spectrum of eigenvalues provides insight into absorption spectra and photoelectron spectroscopy [108].

Vibrational structure calculations, which involve the preparation of eigenstates of the vibrational Hamiltonian, are important for understanding molecular vibrations and infrared spectra. These calculations aid in the interpretation of experimental data and in the prediction of spectroscopic signatures [20].

One of the promising applications of quantum eigensolvers in quantum chemistry is in drug discovery [23, 99, 25]. The process of discovering new drugs involves identifying molecules that

can interact with biological targets in specific ways. This requires a detailed understanding of molecular interactions and the ability to predict how different molecules will behave. If quantum computers can be used to simulate the electronic structures of complex molecules more accurately than classical computers, it may allow them to accelerate the process of identifying potential drug candidates.

2.3.4 Nuclear Physics

Quantum eigenstate preparation holds significant promise in advancing our understanding and solving complex problems in nuclear physics [24]. This approach is particularly important when the nuclear many-body problem is addressed, where the interactions among multiple nucleons within an atomic nucleus are modeled and studied. Using quantum computing, scientists aim to achieve more accurate and efficient solutions than classical computing can provide.

2.3.4.1 Nuclear Structure

One of the primary applications of quantum eigenstate preparation in nuclear physics is the determination of nuclear structures. This involves calculating the properties of nuclei, such as binding energies, excited states, and transition probabilities. Quantum computing can enable the simulation of nuclear shell model wave functions, for example, which represent the different energy states of a nucleus [93]. This can be used to understand the ground and excited states of various isotopes, potentially leading to insights into nuclear stability or the discovery of new medical isotopes.

2.3.4.2 Nuclear Reactions

Simulating nuclear reactions typically involves computing reaction cross sections and analyzing resonance phenomena by preparing the initial and final states of the interacting nuclei. Accurate simulations of nuclear reactions have applications ranging from energy production in nuclear reactors to the synthesis of elements in stellar environments. A better understanding of the eigenstates that represent states of interacting particles before and after a nuclear reaction could lead to more efficient designs of nuclear reactors and better predictions of reaction rates in astrophysical processes.

2.3.4.3 Quantum Chromodynamics and Nuclear Matter

Quantum Chromodynamics (QCD), the theory describing the strong interaction that binds quarks and gluons in protons and neutrons, poses significant computational challenges. Better simulations of QCD could enable us to understand the behavior of nuclear matter under extreme conditions, such as those found in neutron stars or heavy-ion collisions. They can also provide insights into phase transitions in nuclear matter, such as the transition from hadronic matter to quark-gluon plasma, a state of matter believed to have existed shortly after the Big Bang.

2.3.5 Combinatorial Optimization

Combinatorial optimization problems are a class of problems in which the objective is to find an optimal solution from a finite set of discrete options. These problems are ubiquitous in various fields such as logistics, scheduling, network design, and more. Many combinatorial optimization problems are classified as NP-hard, meaning that no known polynomial-time algorithm can solve all instances of these problems. Quantum algorithms offer promising approaches to address these challenges more efficiently than classical algorithms.

Importantly, all NP-hard problems can be reduced to the problem of finding the ground state of the transverse-field Ising model [113]. This reduction highlights why quantum algorithms, particularly those designed to find ground states of Hamiltonians, are powerful tools for solving combinatorial optimization problems. Here, we will focus on the different types of combinatorial optimization problems and how they can be formulated for quantum computation.

2.3.5.1 Binary Variable Optimization

Binary variable optimization problems involve variables that can take on one of two possible values (usually 0 or 1). These problems are fundamental in fields such as computer science, operations research, and artificial intelligence. Notable examples include MaxCut, QMaxCut, and Max-k-SAT.

MaxCut MaxCut is a classic NP-hard problem where the goal is to partition the vertices of a graph into two disjoint subsets such that the number of edges between the subsets is maximized.

The formulation of this problem on a quantum computer involves identifying a Hamiltonian whose ground state corresponds to the optimal partition. MaxCut is relevant in network design to optimize communication or transportation networks, circuit layout design to minimize the number of crossing wires, and clustering in machine learning, where the goal is to group similar items while separating dissimilar ones.

QMaxCut QMaxCut extends the MaxCut problem to quantum graphs, where edges represent quantum interactions. The formulation involves designing a Hamiltonian that encapsulates the quantum correlations in the graph, enabling the quantum algorithm to find the optimal partition that maximizes these correlations. QMaxCut can be applied in quantum communication networks to optimize the quantum entanglement distribution, and in quantum error correction where optimization of the network can improve fault tolerance.

Max-k-SAT Max-k-SAT is an extension of the Boolean satisfiability problem (SAT) and is also NP-hard. The objective is to determine the assignment of binary variables that satisfies the maximum number of clauses, where each clause has exactly k literals. Formulating this problem on a quantum computer involves constructing a Hamiltonian whose ground state represents the assignment that maximizes clause satisfaction. Max-k-SAT is relevant in artificial intelligence for constraint satisfaction problems, in software verification to check the correctness of programs, and in scheduling where various constraints must be satisfied simultaneously.

2.3.5.2 Discrete Variable Optimization

Discrete variable optimization problems involve variables that can take on a finite set of discrete values. Examples include Max-k-Cut and the Traveling Salesman Problem (TSP), both of which are fundamental in optimization theory and have numerous practical applications.

Max-k-Cut Max-k-Cut generalizes the MaxCut problem to partitioning the vertices of a graph into k disjoint subsets, with the aim of maximizing the number of edges between the subsets. Formulating this problem for quantum computation requires identifying a Hamiltonian whose

ground state represents the optimal partition into k different subsets. Max-k-Cut is applicable in telecommunications for optimizing the layout of cellular networks, in logistics for distributing resources efficiently, and in parallel computing for workload balancing across multiple processors.

Traveling Salesman Problem (TSP) The Traveling Salesman Problem (TSP) is one of the most studied NP-hard problems, where the objective is to find the shortest possible route that visits a set of cities exactly once and returns to the origin. Formulating the TSP on a quantum computer involves constructing a Hamiltonian whose ground state corresponds to the shortest route. TSP is critical in logistics for optimizing delivery routes, in manufacturing for minimizing the movement of robotic arms on assembly lines, and in DNA sequencing where finding the optimal path through a series of genes can significantly speed up the process.

CHAPTER 3

ADIABATIC EVOLUTION WITH OPTIMAL CONTROL

In this chapter, we discuss a noise-tolerant strategy for executing arbitrary sequences of unitary transformations with custom quantum gates. This strategy is then used to implement an adiabatic evolution algorithm designed to be run on two superconducting qubits on an IBM Quantum (IBMQ) system. These gates are created on a classical computer using optimal control theory by representing a two-qubit processor as a pair of capacitively coupled superconducting transmons driven by microwave pulses and solving the associated Lindblad master equation. This adiabatic evolution algorithm is tested by emulating specific two-qubit processors available on IBMQ. The results are then compared with those of an equivalent algorithm decomposed into native one- and two-qubit gates and executed on the same IBMQ processors. By reducing the execution time compared to the circuits using native gates, the emulations can prepare a target state with up to 95% fidelity, a significant improvement over IBMQ executions which had fidelities ranging from 65% to 85% .

3.1 The Adiabatic Theorem

This section details the mathematics behind the adiabatic theorem, the underlying mechanism behind adiabatic evolution.

Controllable quantum systems offer the capability to simulate a system that evolves from an initial Hamiltonian H_0 to an arbitrary "target" Hamiltonian H_T that encodes the dynamics of a system of interest. This evolution can be represented by a time-dependent Hamiltonian:

$$H(t) = f(t)H_0 + g(t)H_T, \quad (3.1)$$

where $f(t)$ and $g(t)$ are interpolation functions satisfying the conditions:

$$\begin{aligned} f(0) &= 1 - g(0) = 1 \\ \text{and} \\ f(T) &= 1 - g(T) = 0. \end{aligned} \quad (3.2)$$

These conditions correspond to the imposing the boundary conditions $H(0) = H_0$ and $H(T) = H_T$.

The adiabatic theorem states that a quantum system initially in the k^{th} eigenstate of H_0 will reach a state arbitrarily close to the k^{th} eigenstate of H_T after a sufficiently long evolution time T , provided the k^{th} eigenvalue is continuous throughout evolution and does not cross other levels [19]. The time T required to prepare the quantum system within a particular desired error can be estimated in terms of the parameter $s \equiv t/T$ to be of the following order [5, 18, 17]:

$$T \sim O \left(\max_s (||\partial_s H(s)||) / \Delta^2 \right), \quad (3.3)$$

Here, $\Delta = \min_s (|\varepsilon_k(s) - \varepsilon_{k\pm 1}(s)|)$ is the minimum energy gap between the k^{th} eigenvalue and any other eigenvalue throughout the evolution. The error is bounded above by ϵ which is defined as follows:

$$\epsilon = ||P(1) - P_k(1)||, \quad (3.4)$$

where $P(s)$ and $P_k(s)$ are the projectors onto the evolved state and the k^{th} eigenstate of H at time s , respectively.

The challenge posed by the quadratic dependence of the total evolution time, T , on the inverse of the minimum energy gap, $1/\Delta$, is significant for the implementation of adiabatic evolution as a quantum state preparation technique on both current and near-future quantum devices. The long evolution times that can be necessitated by small energy gaps lead to prolonged implementation times that increase the chances that the physical devices will decohere due to interaction with the environment.

Despite these difficulties, there is a strong incentive to improve the performance of adiabatic evolution as even a slightly faulty implementation can be used in conjunction with alternative eigenstate preparation methods such as phase estimation [64] and the rodeo algorithm [28, 94], which have the prerequisite of substantial overlap between the initial state and the target eigenstate. Notably, even a noisy or imperfect adiabatic evolution can yield a significant enhancement in the initial-state overlap, thereby substantially improving the performance of the subsequent quantum-

state preparation algorithm. For example, an increase in the initial state overlap from 0.1% to 5% would result in a fifty-fold improvement in the efficiency of the rodeo algorithm due to its time complexity scaling inversely with this overlap.

3.2 Adiabatic Evolution of Two-Spin Systems

In this section, we investigate the preparation of a two-spin system in the ground state of the following Hamiltonian:

$$H_T = -\sigma_1^x \sigma_2^x + \sigma_1^y \sigma_2^y + \frac{1}{2} \sigma_1^z \sigma_2^z - \sum_{i=1}^2 \sigma_i^z, \quad (3.5)$$

by initializing the system in the ground state of

$$H_0 = \sum_{i=1}^2 \sigma_i^x, \quad (3.6)$$

and carrying out the adiabatic evolution governed by the time-dependent Hamiltonian in Eq. (3.1) with interpolation functions

$$f(t) = \cos^2(\pi t/2T) \quad g(t) = 1 - f(t). \quad (3.7)$$

We note that the ground state of H_0 can be represented as a linear combination of the uncoupled two-spin states.

$$|\phi(0)\rangle = \frac{1}{2} (|\downarrow\downarrow\rangle - |\downarrow\uparrow\rangle - |\uparrow\downarrow\rangle + |\uparrow\uparrow\rangle). \quad (3.8)$$

Furthermore, the ground state of H_T can be represented as follows:

$$|\phi(T)\rangle = \mathcal{N} \left[\left(-1 + \sqrt{2} \right) |\downarrow\downarrow\rangle + |\uparrow\uparrow\rangle \right], \quad (3.9)$$

where \mathcal{N} is a constant to enforce the normalization of $|\phi(T)\rangle$. The eigenvalue of this state is $E_T = -2.328$. In a similar manner, the instantaneous ground state $|\phi(t)\rangle$ of $H(t)$ at any particular value of t can be determined by solving the time-independent Schrödinger equation $H(t) |\phi(t)\rangle = E_t |\phi(t)\rangle$. The solution to the time-dependent Schrödinger equation $i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle$ will approximate $|\phi(t)\rangle$ with a fidelity given by

$$F(t) = |\langle \phi(t) | \psi(t) \rangle|. \quad (3.10)$$

For an adequately long evolution time T , the system's state at time T will closely resemble the ground state of H_T , as shown by the exact analysis of the evolution in Figure 3.1(a). In this example, even for a relatively short time of $T = 16$, the infidelity $1 - F(T)$ at the end of the evolution is upper bounded by 10^{-3} . In the following analysis, we set $T = 20$. This choice is to reach an ideal balance to minimize both the run-time and the associated decoherence errors encountered during the implementation of this adiabatic evolution on current quantum hardware.

The state resulting from adiabatic evolution can be represented through the unitary-time evolution operator corresponding to a time-dependent Hamiltonian, as shown in Eq. (3.11):

$$|\psi(T)\rangle = \mathcal{U}(0, T) |\psi(0)\rangle . \quad (3.11)$$

To implement adiabatic evolution on a quantum device, the evolution must be done digitally, so the time evolution operation must be divided into discrete steps. To accomplish this, the evolution operator $\mathcal{U}(0, T)$ can be approximated by the product of n short-time propagators associated with n instantaneous Hamiltonians, as demonstrated in Equation 3.12:

$$\mathcal{U}(0, T) \approx \prod_{k=1}^n U(t_k) = \prod_{k=1}^n e^{-iH(t_k)\Delta t}, \quad (3.12)$$

where $\Delta t = T/n$. The error introduced due to the discretization of the evolution time is proportional to the time derivative of the Hamiltonian, $dH(t)/dt$, and is well-behaved as long as $H(t_k)$ provides a suitable approximation for the average value of $H(t)$ within the time interval $[t_{k-1}, t_k]$ [90]. Therefore, the discretization error can be effectively controlled by adjusting the number of steps n into which the evolution time is divided, as shown in Figure 3.1 (b), where the fidelity of the final state is evaluated with respect to the solutions $|\psi(t)\rangle$ derived from the unitary time evolution of Equation (3.11) using the approximation in Equation(3.12).

Although increasing the number of steps n reduces the discretization error, we must also consider the noise-induced error that occurs when the circuit is executed on a physical device. This error increases with the number of gates in the circuit implementation, which increases with the number

of steps n . In Figure 3.1(b), we show how the discretization error becomes negligible when n is as low as 20. Therefore, we adopt $n = 20$ in the subsequent analysis.

3.3 Two-Qubit Quantum Adiabatic Evolution Implementation

To implement a quantum simulation of the adiabatic evolution discussed in Section 3.2, it is necessary to map the spin degrees of freedom to the quantum processor's states and convert the short-time propagators $U(t_k)$ from Eq. (3.12) into quantum gates. The uncoupled states of the two-spin system can be straightforwardly mapped to the computational states of a two-qubit system. Specifically, the states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ are used to represent the two-spin states $|\downarrow\downarrow\rangle$, $|\downarrow\uparrow\rangle$, $|\uparrow\downarrow\rangle$, and $|\uparrow\uparrow\rangle$, respectively. To convert short-time propagators into quantum gates, two approaches are considered for the main comparison in this chapter: (i) a standard decomposition of each propagator into a circuit of elementary gates and (ii) a direct implementation using a single custom two-qubit gate.

3.3.1 Implementation with Elementary Gate Decomposition

The first approach takes advantage of the fact that any unitary operation involving two qubits, such as the short-time propagators $U(t_k)$ in Eq. (3.12), can be represented using three CNOT gates controlled by the first qubit and eight one-qubit U3 gates [103, 111, 114], as illustrated in Fig. 3.2.

The eight U3 gates can be written in terms of x and z one-qubit rotations as follows:

$$\begin{aligned} U3(\theta, \phi, \lambda) &= R_z(\phi)R_x\left(-\frac{\pi}{2}\right)R_z(\theta)R_x\left(\frac{\pi}{2}\right)R_z(\lambda) \\ &= \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda}\sin\left(\frac{\theta}{2}\right) \\ e^{i\phi}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix}. \end{aligned} \quad (3.13)$$

The combination of these techniques for decomposing an arbitrary two-qubit unitary operation into elementary gates is built in to Qiskit in the function `quantum_info.two_qubit_cnot_decompose`. As a specific example, the decomposition of the first short-time propagator in Equation (3.12) is shown in Figure 3.3

This decomposition is readily implementable on cloud-based quantum computing platforms. The efficacy of this technique was assessed by performing the adiabatic evolution on various

IBMQ systems [56]. The quantum circuit embodying the corresponding algorithm was constructed utilizing the open-source quantum information software kit (Qiskit) [7], which facilitated the decomposition of each short-time propagator into elementary gates using the built-in function `quantum_info.two_qubit_cnot_decompose`. The results of these digital quantum simulations on IBMQ processors are discussed in Section 3.4.1.

3.3.1.1 Error Mitigation with Confusion Matrix

Simulations of NISQ devices include several types of error, including coherence error and measurement error, some of which can be easily mitigated by basic techniques. In this work, measurement errors are mitigated by inverting the confusion matrix (also known as the "error matrix") [96]. The occupation probabilities measured in an experiment, arranged in the vector $|c_{\text{exp}}^2\rangle$, are related to the true ones, $|c_{\text{true}}^2\rangle$, through the confusion matrix P in the following way:

$$\begin{aligned} ||c_{\text{exp}}^2\rangle &= P ||c_{\text{true}}^2\rangle \\ \text{or} \\ ||c_{\text{true}}^2\rangle &= P^{-1} ||c_{\text{exp}}^2\rangle, \end{aligned} \tag{3.14}$$

Assuming that the measurement errors for qubits 1 and 2 are independent, the confusion matrix can be expressed using the single qubit measurement probabilities p_{ij} as follows:

$$\begin{pmatrix} (1 - p_{10})_1(1 - p_{10})_2 & (1 - p_{10})_1(p_{01})_2 & (p_{01})_1(1 - p_{10})_2 & (p_{01})_1(p_{01})_2 \\ (1 - p_{10})_1(p_{10})_2 & (1 - p_{10})_1(1 - p_{01})_2 & (p_{01})_1(p_{10})_2 & (p_{01})_1(1 - p_{01})_2 \\ (p_{10})_1(1 - p_{10})_2 & (p_{10})_1(p_{01})_2 & (1 - p_{01})_1(1 - p_{10})_2 & (1 - p_{01})_1(p_{01})_2 \\ (p_{10})_1(p_{10})_2 & (p_{10})_1(1 - p_{01})_2 & (1 - p_{01})_1(p_{10})_2 & (1 - p_{01})_1(1 - p_{01})_2 \end{pmatrix} \tag{3.15}$$

Prior to the execution of each circuit, we carry out two calibration experiments by initializing the device in the states $|ket00\rangle$ and $|11\rangle$ states. The results of these experiments correspond to the first and fourth columns of the confusion matrix. These results allow us to determine the single qubit measurement probabilities p_{ij} which are then used to form the complete confusion matrix.

3.3.2 Modeling the Device Hamiltonian

The alternative strategy involves a realistic representation of a physical quantum device to actualize each short-time propagator in Eq. (3.12) with a single custom gate. We emulate a two-qubit processor as two capacitively connected superconducting transmons regulated by microwave pulses, as illustrated schematically in Fig. 3.4 a).

The Hamiltonian for a superconducting transmon, when expressed with respect to the quantity of Cooper pairs (n) and the magnetic flux (ϕ), can be formulated up to the fourth order in ϕ . This representation is given by [67]:

$$\begin{aligned} H = & 4E_C n^2 - E_J \cos(\phi) \\ \approx & 4E_C n^2 - E_J + \frac{E_J}{2} \phi^2 - \frac{E_J}{24} \phi^4, \end{aligned} \quad (3.16)$$

where the energies stored in the capacitor and the Josephson junction are denoted as E_C and E_J , respectively. To proceed further, we introduce the creation and annihilation operators for the transmon, with which n and ϕ can be defined as follows:

$$n = i \left(\frac{E_J}{32E_C} \right)^{\frac{1}{4}} (a^\dagger - a) \quad \phi = \left(\frac{2E_C}{E_J} \right)^{\frac{1}{4}} (a^\dagger + a), \quad (3.17)$$

In terms of these operators, the Hamiltonian takes the form,

$$H \approx \omega a^\dagger a - \frac{\alpha}{6} (a^\dagger + a)^4, \quad (3.18)$$

where we defined $\omega \equiv (8E_C E_J)^{1/2}$ and $\alpha \equiv E_C/2$. According to the Baker-Campbell-Hausdorff formula [52], it can be shown that for a transformation $U = \exp(-i\Omega t a^\dagger a)$, the following is true:

$$U a_1 \cdots a_l U^\dagger = e^{(-i(m-n)\Omega t)} a_1 \cdots a_l, \quad (3.19)$$

where $a_1 \cdots a_l$ is a chain of creation and annihilation operators ($a_i \in \{a^\dagger, a\}$), and m and n are the number of creation and annihilation operators in the chain, respectively.

Under this transformation, ignoring constant and fast-rotating terms with $|m - n| > 1$, the Hamiltonian of the transmon takes the form

$$\begin{aligned} H \rightarrow H' &= UHU^\dagger + i\dot{U}U^\dagger \\ &\approx (\omega + \alpha + \Omega) a^\dagger a - \alpha a^\dagger a a^\dagger a. \end{aligned} \quad (3.20)$$

The Hamiltonian of two capacitively coupled transmons controlled by microwave pulses can be approximately written as follows [60, 69]

$$\begin{aligned} H &\approx \sum_{i=1}^2 \left(4E_{C_i} n_i^2 + \frac{E_{J_i}}{2} \phi_i^2 - \frac{E_{J_i}}{24} \phi_i^4 \right) + \frac{8E_{C_1}E_{C_2}}{E_{C_g}} n_1 n_2 \\ &\quad + 2 \sum_{i=1}^2 \eta_i \left[\epsilon_I^i(t) \sin(\Omega_i t) - \epsilon_Q^i(t) \cos(\Omega_i t) \right] n_i \\ &= \sum_{i=1}^2 \left[\omega_i a_i^\dagger a_i - \frac{\alpha_i}{6} (a_i^\dagger + a_i)^4 \right] + g(a_1^\dagger - a_1)(a_2^\dagger - a_2) \\ &\quad + 2i \sum_{i=1}^2 \left[\epsilon_I^i(t) \sin(\Omega_i t) - \epsilon_Q^i(t) \cos(\Omega_i t) \right] (a_i^\dagger - a_i), \end{aligned} \quad (3.21)$$

where the term proportional to $g \equiv 8E_{C_1}E_{C_2}/E_{C_g}\eta_1\eta_2$ with $\eta_i \equiv (32E_{C_i}/E_{J_i})^{1/4}$ describes the crosstalk interaction between the transmons due to their capacitive coupling. Under the transformation

$$U = \exp \left(-i\Omega_1 t a_1^\dagger a_1 - i\Omega_2 t a_2^\dagger a_2 \right), \quad (3.22)$$

By substituting $\Omega_i = -\omega_i - \alpha_i$, assuming that $\Omega_1 \approx \Omega_2$ and dropping constant and fast-rotating terms, the above Hamiltonian takes the form

$$\begin{aligned} H &\approx - \sum_{i=1}^2 \alpha_i a_i^\dagger a_i a_i^\dagger a_i - g \left(a_1^\dagger a_2 + a_1 a_2^\dagger \right) \\ &\quad + \sum_{i=1}^2 \left[\epsilon_I^i(t) (a_i^\dagger + a_i) - i\epsilon_Q^i(t) (a_i^\dagger - a_i) \right], \end{aligned} \quad (3.23)$$

For simplicity, we rewrite this Hamiltonian as follows.

$$H_{\text{QPU}}(t) = H_{\text{d}} + H_{\text{c}}(t), \quad (3.24)$$

where

$$H_d \approx - \sum_{i=1}^2 \alpha a_i^\dagger a_i a_i^\dagger a_i - g (a_1^\dagger a_2 + a_2^\dagger a_1) \quad (3.25)$$

constitutes the drift Hamiltonian of the undisturbed device, and

$$H_c(t) = \sum_{i=1}^2 \left[\epsilon_I^i(t) (a_i^\dagger + a_i) - i \epsilon_Q^i(t) (a_i^\dagger - a_i) \right] \quad (3.26)$$

represents the time-variant Hamiltonian delineating the governance of the quantum processor through irradiation with resonant microwave pulses.

3.3.3 Implementation with Custom Gates

We can now use the Hamiltonians from Section 3.3.2 to aid in an optimal control scheme to devise custom gates for transmon qubits. For the anharmonicity of both transmons, we use $\alpha = 200$ MHz, and for the strength of the crosstalk interaction strength between the transmons due to capacitive coupling, we use $g = 3$ MHz. The time-dependent amplitudes $\epsilon_I^i(t)$ and $\epsilon_Q^i(t)$ correspond respectively to the in-phase and quadrature tunable pulse sequences that control transmon i . We consider the first three energy levels of each transmon. The computational two-qubit states are defined by the subspace of states with zero and one quanta per transmon. By explicitly including states with two quanta in at least one of the transmons, we account for higher-energy states that can be populated due to gate error and decoherence. This inclusion also enables our control pulses to reduce leakage by blocking transitions to these states, analogously to the Derivative Removal by Adiabatic Gate (DRAG) algorithm [79].

To compute the custom two-qubit gates that realize each of the short-time propagators in Eq. (3.12), we determine the pulse sequences $\epsilon_I^i(t)$ and $\epsilon_Q^i(t)$ that solve the optimization problem as follows:

$$\begin{aligned} U_{\text{QPU}}(t_k) &\simeq \mathcal{U}_{\text{QPU}}(0, \tau) \\ &= \mathcal{T} \exp \left(-\frac{i}{\hbar} \int_0^\tau [H_d + H_c(\tau')] d\tau' \right), \end{aligned} \quad (3.27)$$

Here, $U_{\text{QPU}}(t_k)$ represents the short-time propagator $U(t_k)$ embedded in the Hilbert space spanned by the two-transmon states, and $\mathcal{T} \exp$ represents the time-ordered exponential. Using the gradient ascent pulse engineering algorithm (GRAPE) [81], we can find the solution to Equation (3.27) with acceptable accuracy by minimizing the objective function.

$$\Phi = 1 - \frac{F_{\text{gate}}^2}{2} + \chi \frac{\exp(\bar{\epsilon}^{2n}) - 1}{\exp(1) - 1}, \quad (3.28)$$

where

$$F_{\text{gate}} = \left| \frac{\text{tr}(U_{\text{QPU}}^\dagger \mathcal{U}_{\text{QPU}})}{\dim_{\text{QPU}}} \right|, \quad (3.29)$$

with \dim_{QPU} representing the dimension of the considered Hilbert space. In the objective function, $\bar{\epsilon}$ represents the root-mean-squared amplitude of the control pulse normalized to ϵ_{cut} and can be written as follows:

$$\bar{\epsilon} = \frac{1}{\epsilon_{\text{cut}}} \sqrt{\frac{1}{\tau} \sum_{i=1}^2 \sum_{j \in \{\text{I}, \text{Q}\}} \int_0^\tau \epsilon_j^i(\tau')^2 d\tau'} \quad (3.30)$$

The gate fidelity, F_{gate} , appearing in the second term on the right-hand side of Equation (3.28), serves as a measure of how accurately the pulse-controlled gate reproduces the desired unitary operation. The final term imposes a penalty on large amplitudes by means of the parameters ϵ_{cut} (cutoff amplitude) and n (cutoff harshness), with their significance determined by the parameter χ . This penalty term prevents high-amplitude solutions where the Hamiltonian approximation used in the optimization of Equation (3.27) becomes invalid and where the hardware implementing the control pulses may not function as intended.

The loss term is structured such that a zero-amplitude pulse corresponds to a loss of zero, and a pulse with a root mean square amplitude equal to ϵ_{cut} contributes χ to the total loss. Figure 3.5 shows the first 100 ns of separately optimized control pulses, each realizing the first short-time propagator in the adiabatic evolution algorithm, using three different total pulse lengths τ (2500 ns, 400 ns, and 120 ns) as per Equation (3.27), with a sampling rate of 8 samples per ns.

The pulse length $\tau = 2500$ ns approximately matches the implementation time τ_U of the short-time propagators on the *ibmq_belem* system, enabling a comparison between our emulated output and runs on that system. The pulse length $\tau = 400$ ns corresponds to the standard implementation time of CNOT gates for IBMQ systems. The pulse length $\tau = 120$ ns was the shortest pulse of our investigation that kept the gate infidelity $1 - F_{\text{gate}}$, below 10^{-4} . The parameters in the objective function of Equation (3.28) were $\epsilon_{\text{cut}} = 30$ MHz, $n = 3$ and $\chi = 10^{-3}$ for all minimization attempts.

As the control pulse length decreases, the root-mean-squared amplitude and the relative influence of its high-frequency components increase. This relationship between the control pulse amplitude and its length sets a lower bound for two-qubit gate implementation times, and, consequently, the overall implementation time for the entire evolution on the types of quantum devices that follow this qubit model. This is because the quantum computer's Hamiltonian approximations (used to solve Equation (3.27)) and the hardware controlling it both operate optimally within a specific energy regime.

In order to investigate the performance of adiabatic evolution with custom gates as a state preparation method, we used classical emulations to predict the output of a two-transmon processor by solving the Lindblad master equations for its density matrix. The analysis of this method is given in Section 3.4.2.

3.4 Results and Discussion

In this section, we present and discuss the results of our investigation of the two possible implementations of quantum adiabatic evolution to prepare the ground state of a target Hamiltonian.

3.4.1 Adiabatic Evolution with Elementary Gates on IBMQ

In this section, we evaluate the feasibility of implementing adiabatic evolution, as described in Equation (3.12), on IBMQ systems by performing experiments using circuits composed only of elementary gates. The calibration data for the IBMQ systems at the time of this study can be found in Table 3.1.

The quantum circuits were initialized by preparing the IBMQ processors in the ground state of the initial Hamiltonian H_0 [refer to Equation (3.6)] through the application of the two-qubit Pauli

$X^{(2)} = \sigma_1^x \sigma_2^x$ and Hadamard $H^{(2)} = H_1 H_2$ gates to the default initial state $|00\rangle$, resulting in the following initial wave function for the quantum register:

$$|\psi(0)\rangle = H^{(2)} X^{(2)} |00\rangle = \frac{1}{2} (|00\rangle - |01\rangle - |10\rangle + |11\rangle), \quad (3.31)$$

The rest of the adiabatic evolution circuit was constructed using elementary quantum gates derived from the decomposition of the n short-time propagators, as discussed in Section 3.3.

By truncating this adiabatic evolution circuit after k short-time propagator operations, the final state corresponds to the instantaneous wave function $|\psi(t_k)\rangle$. We used Pauli measurements on these truncated circuits to estimate the expectation value of the target Hamiltonian $\langle H_T \rangle(t) = \langle \psi(t) | H_T | \psi(t) \rangle$ at each value of t_k to provide insight into how the state of the quantum register evolves during the execution of the circuit.

We can also easily estimate the instantaneous fidelity $F(t_k)$ using the definitions in Equations (3.12) and (3.31). We start with the following observation:

$$\begin{aligned} \langle \phi(t_k) | &\approx \langle \psi(t_k) | = \langle 00 | X^{(2)} H^{(2)} \mathcal{U}^\dagger(0, t_k) \\ &\approx \langle 00 | X^{(2)} H^{(2)} \prod_{i=k}^1 U^\dagger(t_i), \end{aligned} \quad (3.32)$$

Thus, the overall unitary operator for the truncated circuit is:

$$\tilde{U}(t_k) = X^{(2)} H^{(2)} \prod_{i=k}^1 U^\dagger(t_i), \quad (3.33)$$

Table 3.1 Calibration data for *ibmq_belem*, *casablanca*, *ibmq_lima*, and *ibmq_manila*. The relaxation and dephasing times, T_1 and T_2 , are provided for qubits 0 and 1 of each system throughout the simulation. The implementation times for the CNOT gate and the decomposition of the short-time propagators are denoted as τ_{CNOT} and τ_U , respectively.

IBMQ system	T_1 [μs]		T_2 [μs]		τ_{CNOT} [ns]	τ_U [ns]
<i>ibmq_belem</i>	102.6	70.4	127.3	104.5	810.7	≈2500
<i>ibmq_casablanca</i>	111.7	130.1	40.7	102.2	760.9	≈2400
<i>ibmq_lima</i>	101.6	113.0	180.0	106.9	305.8	≈1000
<i>ibmq_manila</i>	136.0	244.2	112.8	46.7	277.3	≈900

This is enough to derive an expression for $F(t_k)$ as follows:

$$\begin{aligned}
F(t_k) &= |\langle \phi(t_k) | \psi(t_k) \rangle| \approx |\langle 00 | \tilde{U}(t_k) | \psi(t_k) \rangle| \\
&= |\langle 00 | \tilde{\psi}(t_k) \rangle| \\
&= |\tilde{c}_{00}|.
\end{aligned} \tag{3.34}$$

Thus, each instantaneous fidelity $F(t_k)$ is equivalent to $|\tilde{c}_{00}|$, which is the square root of the probability of measuring $|\psi_k\rangle$ in the $|00\rangle$ state, which can be estimated from repeated measurements of $|\psi_k\rangle$ in the default **Z** basis.

This process is repeated to obtain estimations for $\langle H_T \rangle$ and $F(t)$ for each IBMQ system in question. In Fig. 3.6, we compare the results we obtained from the IBMQ systems with those produced by the same process on an ideal quantum processor simulation using Qiskit’s Aer simulator, which is not affected by environmental interactions.

The ideal output, depicted by a dotted line, reveals that the gate error included in our calculations for $F(t)$ (from Equation 3.10) does not considerably interfere with the ability to prepare a target state in two-qubit systems. However, experiments on *ibmq_belem*, *ibmq_casablanca*, *ibmq_lima*, and *ibmq_manila*, represented by color markers, diverge from the ideal evolution, achieving target state fidelities of 60%, 75%, 80%, and 72%, respectively. This loss of fidelity during evolution leads to significant discrepancies between the properties extracted from the reached state and those of the ground state of the target Hamiltonian, as illustrated by the evolution of $\langle H_T \rangle(t) = \langle \psi(t) | H_T | \psi(t) \rangle$ in Fig. 3.6 **b**). The mean values for this expectation value obtained from the IBMQ system range from -1.25 to -0.25 , while the ideal simulation attains the target value of $E_T = -2.328$.

Interestingly, the fidelity loss rate remains approximately constant throughout the evolution. This observation is unexpected since the interpolation functions defining $H(t)$ (defined in Equation (3.7)) cause the evolution to be “slowest” at its initial and final points (see Fig. 3.1), where minimal fidelity losses were anticipated. The constant fidelity loss rate implies that this effect is predominantly influenced by either the gate error accumulated from the elementary gates realizing the short-time propagator (with the most significant contributions arising from CNOT gate errors)

or the coherence loss during their combined implementation time.

These findings highlight the primary challenge of preparing eigenstates with NISQ devices using adiabatic evolution. The implementation time of the quantum circuit that expresses the adiabatic evolution of Equation (3.12) in terms of elementary gates is comparable to the decoherence times of current quantum devices. In this study, the implementation times of adiabatic evolution on *ibmq_belem*, *ibmq_casablanca*, *ibmq_lima*, and *ibmq_manila* are approximately 52.5 μ s, 50.4 μ s, 21 μ s, and 18.9 μ s, respectively, which are similar to the relaxation and dephasing decoherence times (T_1 and T_2) of these systems, as listed in Table 3.1.

3.4.2 Customized Gates Emulated with Classical Devices

As an alternative to implementing adiabatic evolution through elementary gates, this section explores the efficacy of using pulse sequences that can produce short-time propagators in a single step, as explained in Section 3.3. This approach seeks to reduce the depth of the corresponding circuit and, as a result, decrease the time it takes to execute on the device.

To investigate this approach, a two-transmon processor performing adiabatic evolution was emulated with classical computing techniques. Specifically, the Quantum Toolbox in Python (QuTiP) [59, 58] was used to solve the Lindblad master equation. This equation is defined as follows:

$$\begin{aligned} \dot{\rho} = & -\frac{i}{\hbar} [H_{\text{QPU}}, \rho] \\ & + \frac{1}{T_1} \sum_{i=1}^2 \left(a_i \rho a_i^\dagger - \frac{1}{2} \{ a_i a_i^\dagger, \rho \} \right) \\ & + \frac{1}{T_2} \sum_{i=1}^2 \left(a_i^\dagger a_i \rho a_i a_i^\dagger - \frac{1}{2} \{ a_i a_i^\dagger a_i^\dagger a_i, \rho \} \right), \end{aligned} \quad (3.35)$$

where ρ represents the density of the two-transmon system. In this equation, H_{QPU} denotes the two-transmon Hamiltonian introduced in Equation (3.24), with control pulses optimized to achieve the short-time propagators described in Equation (3.12). The master equation accounts for the decoherence mechanisms of relaxation and dephasing, which are represented by the decoherence times T_1 and T_2 . The model utilizes parameter values obtained from calibration data from IBMQ

devices that implemented the adiabatic evolution, as seen in Table 3.1. This enables the emulation of system interactions with their surrounding environment.

We investigated this implementation of quantum adiabatic evolution by simulating various scenarios. First, we emulated the adiabatic evolution by implementing each short-time propagator in Equation (3.12) with a single custom gate of pulse length $\tau = 120$ ns. This approach yielded target state fidelities of approximately 95%, a significant enhancement compared to IBMQ system runs, as depicted in Figures 3.8 a), c), e), & g) . Although the pulse length of $\tau = 120$ ns minimizes gate infidelities below 10^{-4} , the resulting pulse amplitudes may exceed a threshold, rendering them unsuitable for implementation on quantum devices, represented arbitrarily as the horizontal dotted lines at $|\epsilon_{I,Q}^i| < \alpha/20$, for example.

We also emulated the algorithm implementation using custom gates of length $\tau = 400$ ns, which is comparable to the implementation times of the CNOT gates in IBMQ systems. Despite this longer pulse length, our results still surpassed those of IBMQ system runs, achieving target state fidelities of approximately 90%. This can be explained by the fact that the elementary gate decomposition explored in Section 3.3.1 requires two CNOT gates per two-qubit unitary evolution, leading to longer overall circuit execution times.

In order to facilitate comparison with IBMQ results, we also simulated the adiabatic evolution using custom gates of length approximately equal to the implementation time of a short-time propagator on IBMQ hardware, denoted τ_U . The specific values for τ_U are provided in Table 3.1. These simulations resulted in gates with fidelity ranging from 65% to 85%, which aligned closely with the IBMQ results. This close match suggests that the total execution time of the circuit may be the most important factor in accurately preparing quantum states on NISQ devices.

For an even more direct comparison, we emulated the output of the IBMQ digital quantum simulation by realizing each elementary gate in the IBMQ quantum circuit with custom control pulses. In this final set of simulations, we combined each pair of simultaneous U3 gates in the short-time propagators into two-qubit $U3^{(2)} = U3_1 U3_2$ gates, as illustrated in Figure 3.7. Then, the CNOT and $U3^{(2)}$ gates were implemented using control pulses, the cumulative durations of which

approximated the execution time of short-time propagators on IBMQ systems, denoted as τ_U .

In Figure 3.8, a comparison is made between the output of our emulation and the results of the digital quantum simulations performed on IBMQ systems.

The emulated digital quantum simulation, achieved by implementing the adiabatic evolution through the CNOT and U3⁽²⁾ gates (denoted by green triangles), aligns closely with the IBMQ results (represented by blue crosses). The emulated outputs, obtained by realizing the short-time propagators in Eq. (3.12) via single two-qubit gates with control pulses of lengths $\tau = \tau_U$, $\tau = 400$ ns, and $\tau = 120$ ns (depicted as orange diamonds, squares, and circles, respectively), highlight the enhancements achievable through the incorporation of custom gates within quantum algorithms. The value derived from the shortest classical emulation, $\langle H_T \rangle (T) = -2.2$, was much closer to the exact result of $E_T = -2.328$ than the energies extracted from IBMQ systems (mean values ranging from -1.25 to -0.25). Even though this demonstration was conducted on a basic problem involving only two qubits, the enhancement seen over the simple gate composition method indicates that the pursuit of optimal control for custom gates in quantum algorithms is promising for NISQ devices.

3.4.3 Discussion of Error Sources

Various error sources impact the performance of this adiabatic evolution method on IBMQ hardware. Our model already considers systematic gate infidelities, stochastic dissipative processes during circuit execution, stochastic measurement errors during readout, and statistical noise. To assess these errors' effects on adiabatic evolution, we compare Qiskit's Aer simulator-based simulations with runs on IBMQ systems. We use Qiskit's Aer simulator to determine the state $H^{(2)} |00\rangle = (|00\rangle + |01\rangle + |10\rangle + |11\rangle) / 2$ via the `AerSimulator.from_backend` built-in method. This method develops a classical simulator with an approximate noise model for any IBMQ device that includes gate errors, readout errors, and dissipative processes. Figure 3.9 illustrates the average deviation of the measured probabilities from the ideal value $1/4$, as the number of shots N increases. As N increases, the statistical error component diminishes and the average error levels off to a device-specific plateau, which we associate with the measurement error for each IBMQ system.

These simulation outcomes indicate that measurement and statistical noise are not significant error contributions, as they only cause deviations from the ideal fidelities and expectation values of about $\delta F \approx 0.01$ and $\delta \langle H_T \rangle \approx 0.1$ for IBMQ runs with $N = 2500$ shots, significantly smaller than those presented in Figure 3.8.

Next, we repeated the adiabatic evolution simulations using Qiskit’s Aer simulator with the “*statevector*” method. This comes with the option to record per-shot amplitudes, although doing so prevents readout error simulation. We ran 100,000 executions of each circuit to determine the instantaneous fidelity and expectation value of H_T for each time value t_k . Using the recorded amplitudes, we calculated the energy for each simulated shot, resulting in the distribution of possible energy measurements. We repeated a similar procedure for fidelity. To isolate the effects of dissipative processes from other error sources, we formed the density matrix for each circuit by averaging the shots as follows:

$$\rho = \frac{1}{N_{\text{shots}}} \sum_{i \in \text{shots}} |\psi_i\rangle \langle \psi_i|, \quad (3.36)$$

where $|\psi_i\rangle$ is the final state of the simulated circuit. We then perform the singular value decomposition on the density matrix to identify the most probable state, which is the state with the highest singular value, and then calculate the fidelity and energy based on these amplitudes. If only gate errors and readout errors are present, one should observe a singular value very close to one. However, dissipative processes generate mixed states, causing the highest singular value to be significantly lower than its ideal value, decreasing as the circuit depth increases. In Figure 3.10, we show the resultant distributions at each step for both **a)** the fidelity and **b)** the expectation value of H_T as blue violin plots, illustrating the mean and 1σ quantiles. The green lines represent the expected result from an ideal evolution, whereas the orange dashed lines depict the outcomes using the principal singular vector of the density matrices. The discrepancy between the dashed orange and solid blue lines is mainly due to dissipative noise processes. In contrast, the proximity of the dashed orange and solid green lines indicates that other errors are relatively insignificant. The blue markers provide an estimate of the lower bound on the total uncertainty anticipated from simulations on IBMQ systems.

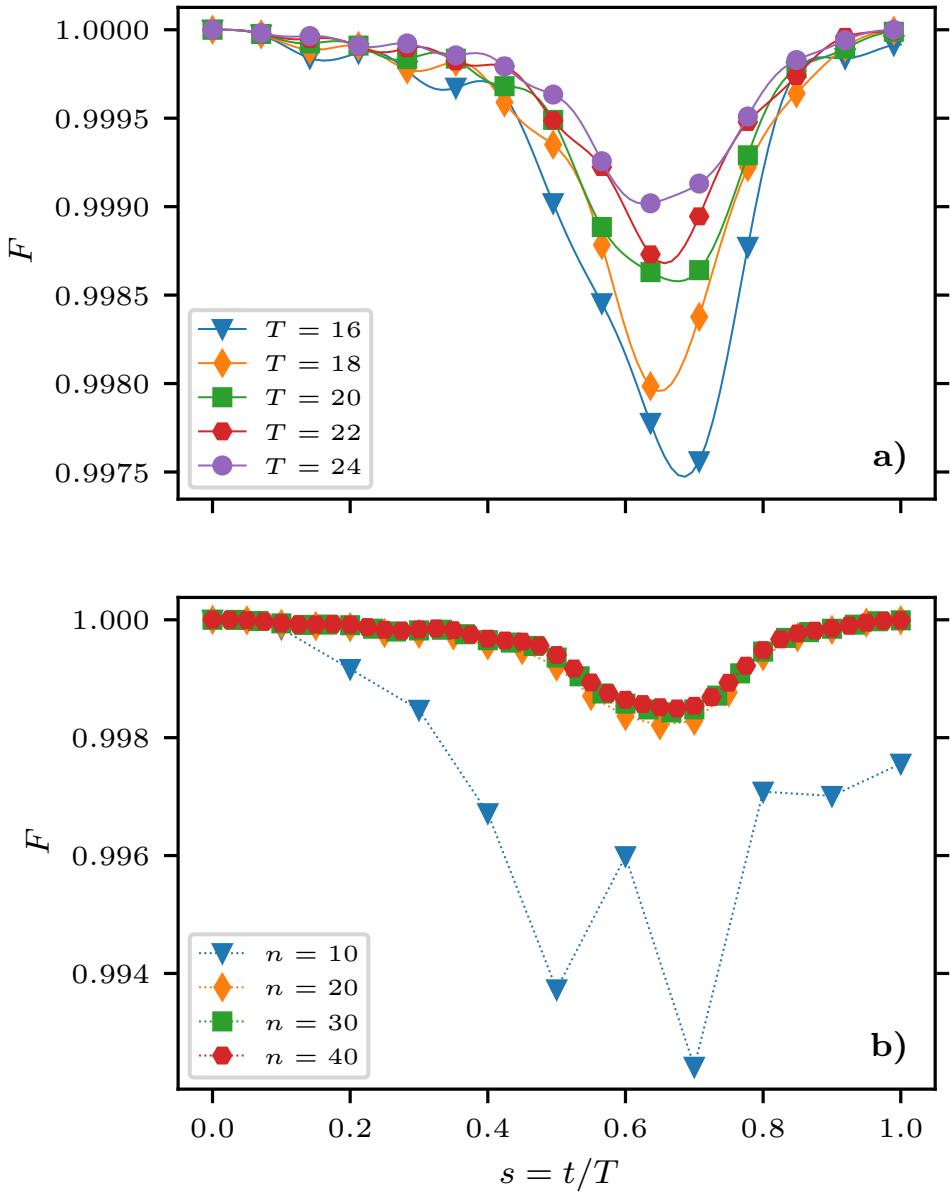
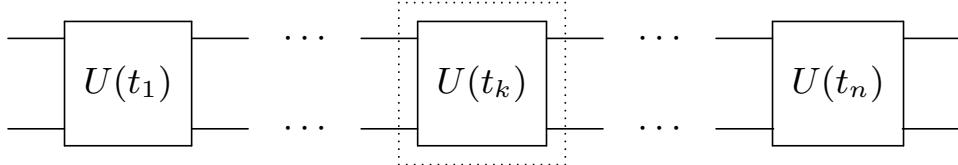


Figure 3.1 Evolution of the fidelity between the device state and the ground state of $H(t)$ for various values of **a)** the evolution time T and **b)** the number of time steps n as a function of the parameter $s \equiv t/T$. The adiabatic evolution implemented on quantum devices utilizes $T = 20$ and $n = 20$, as this combination results in high fidelity while minimizing the number of gates required for device implementation.

a)



b)

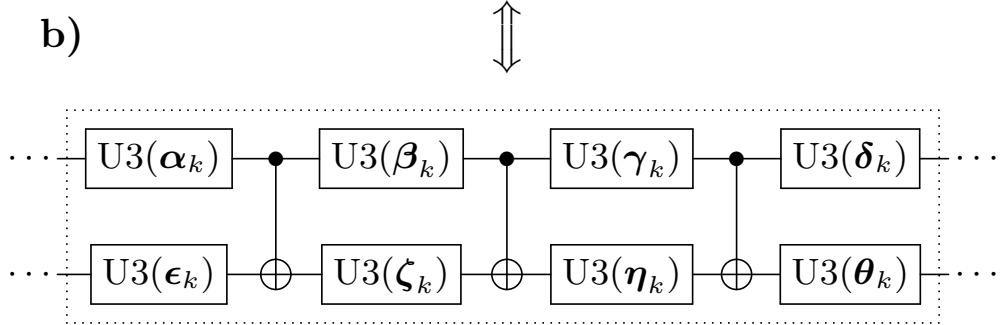
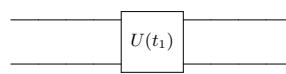
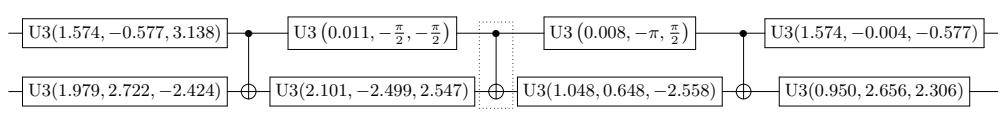


Figure 3.2 Decomposition of the k -th unitary transformation in the sequence of Equation (3.12) [circuit a)] into CNOT and U3 gates [circuit b)]. Each U3 gate depends on three Euler angles. The quantum circuit resulting from this decomposition can be directly implemented on IBMQ systems.

a)



b)



c)

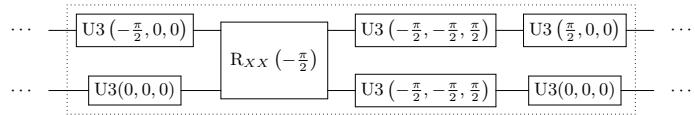


Figure 3.3 Decomposition of the first short-time propagator in Eq. (3.12) [circuit a)] into CNOT and U3 gates [circuit b)]. Circuit c) shows how the same type of decomposition can be used to decompose the CNOT gate into elementary gates.

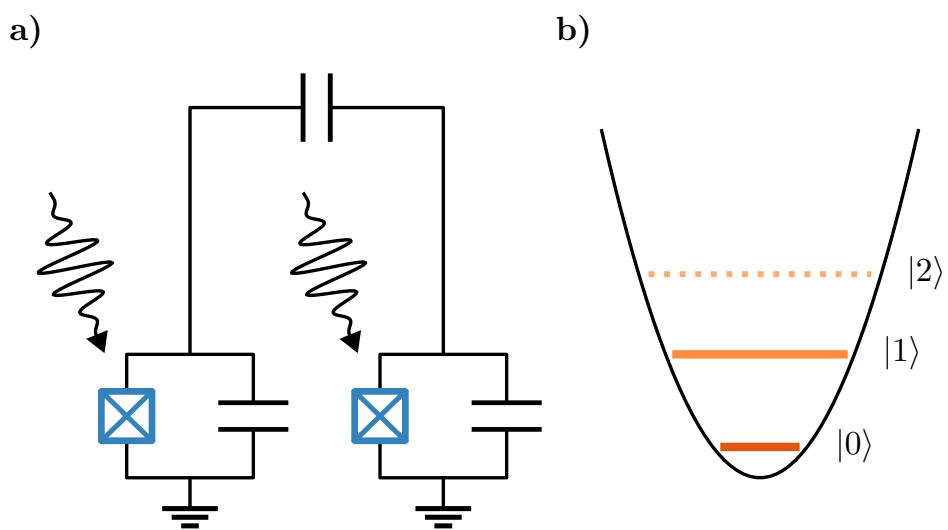


Figure 3.4 Schematic depiction of **a)** the model two-qubit processor, consisting of two capacitively coupled superconducting transmons controlled with microwave pulses, and **b)** the energy spectrum of a superconducting transmon in which the lowest two levels, shown as solid lines, are the computational qubit.

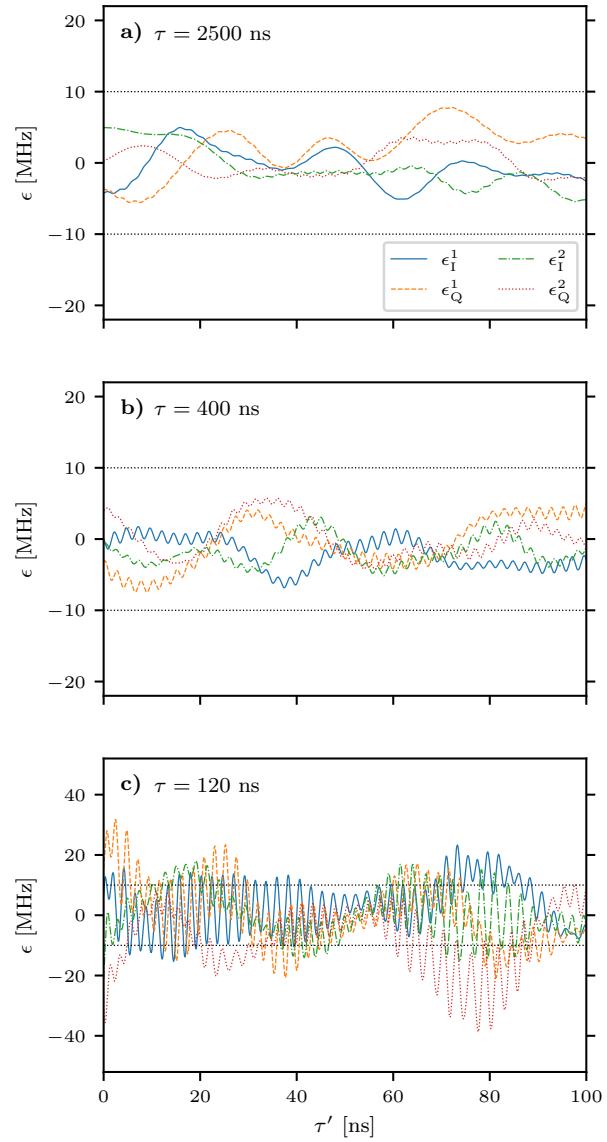


Figure 3.5 First 100 ns of control pulses with lengths **a)** $\tau = 2500$ ns, **b)** $\tau = 400$ ns, and **c)** $\tau = 120$ ns, realizing the first unitary in the sequence of Equation (3.12). The correlation between the amplitude of the control pulse with its length establishes a lower bound for the implementation time of two-qubit gates on current quantum devices.

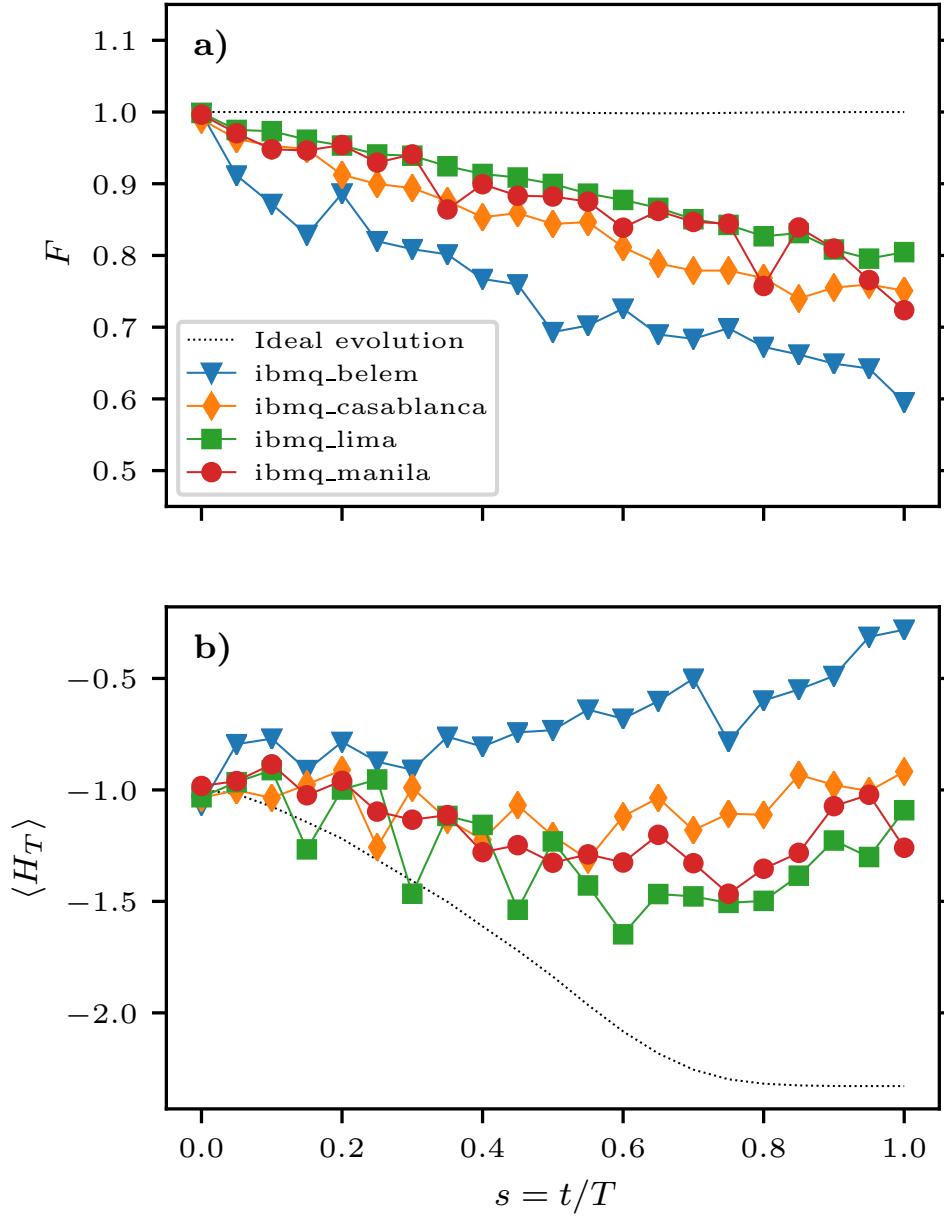
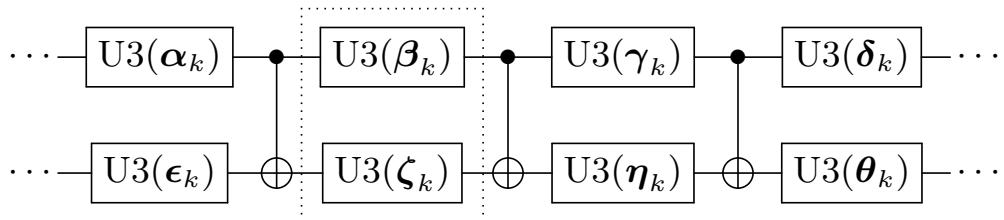


Figure 3.6 Adiabatic evolution of **a)** instantaneous fidelity $F(t)$ and **b)** expectation value $\langle \psi(t) | H_T | \psi(t) \rangle$, obtained through the application of the unitary sequence from Equation (3.12) decomposed into elementary gates on four IBM quantum processors: *ibmq_belem* (blue triangles), *ibmq_casablanca* (orange diamonds), *ibmq_lima* (green squares), and *ibmq_manila* (red circles). The significant discrepancies between the experimental results and the ideal quantum processor simulations (dotted lines), conducted using Qiskit's Aer simulator, can be attributed to decoherence due to long implementation times.

a)



b)

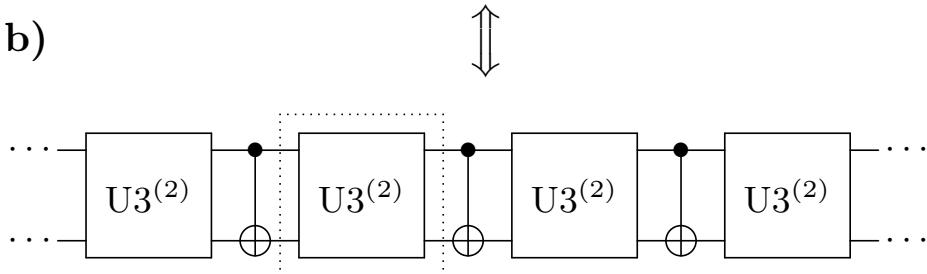


Figure 3.7 Combination of simultaneous U3 gates [circuit a)] into two-qubit $U^{(2)}$ gates [circuit b)]. The CNOT and $U3^{(2)}$ gates in circuit b) are executed with control pulses, the total durations of which closely correspond to the implementation time of circuit a) on IBMQ systems.

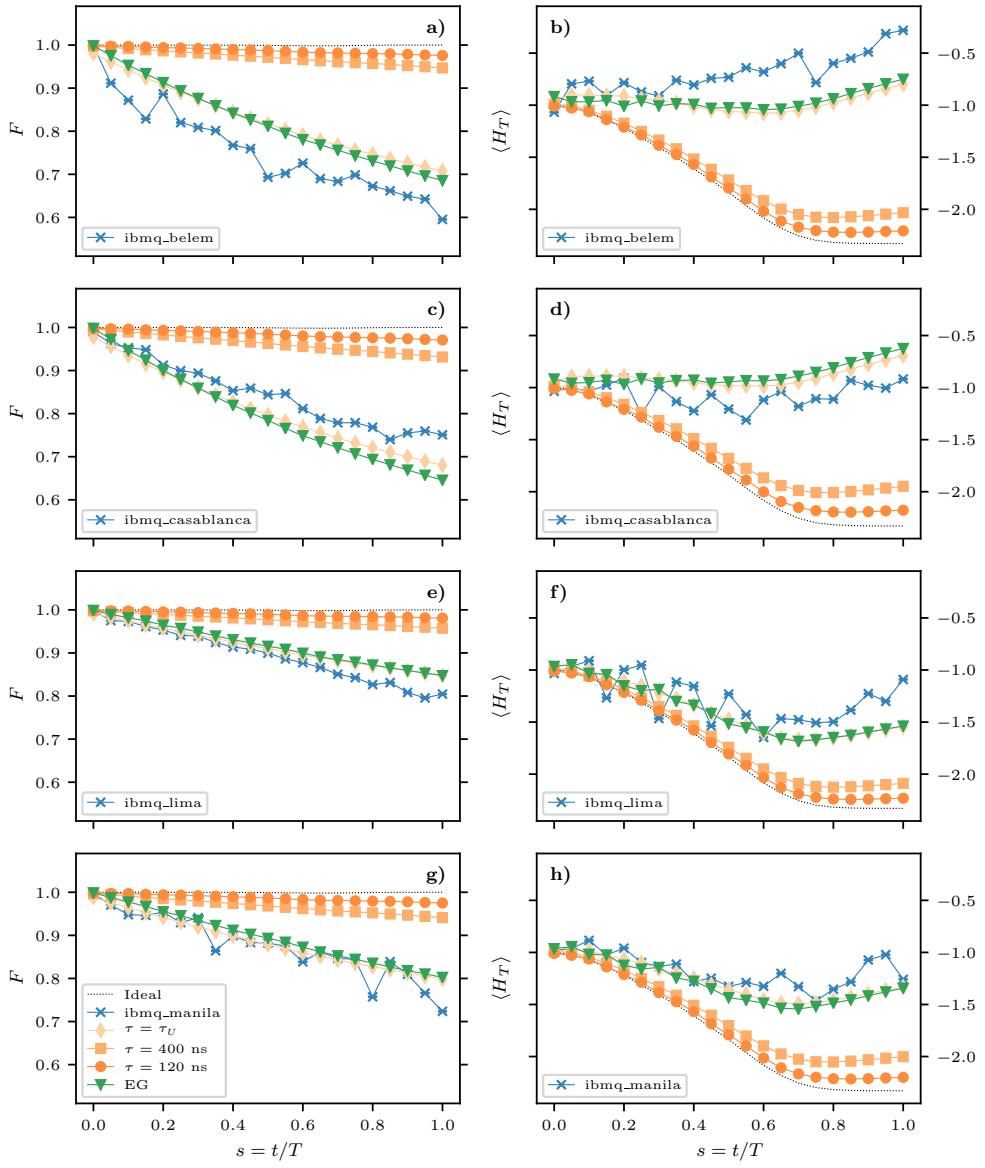


Figure 3.8 A comparison of the emulated outputs of the adiabatic evolution of Eq. (3.12) and the results obtained from *ibmq_belem*, *ibmq_casablanca*, *ibmq_lima*, and *ibmq_manila* systems. Panels **a), c), e) and g)** display the progression of the instantaneous fidelity F . Classical emulations implementing the adiabatic evolution through either a circuit of elementary gates or custom gates of length τ_U (represented by green triangles and orange diamonds, respectively) achieve target state fidelities comparable to those attained by IBMQ systems (indicated by blue crosses). Emulated outputs utilizing shorter gates (denoted by orange squares and circles) reached approximately 95% fidelity, facilitating enhanced extraction of the target state's spectroscopic information, as demonstrated by the evolution of $\langle H_T \rangle$ in panels **b), d), f) and h)**.

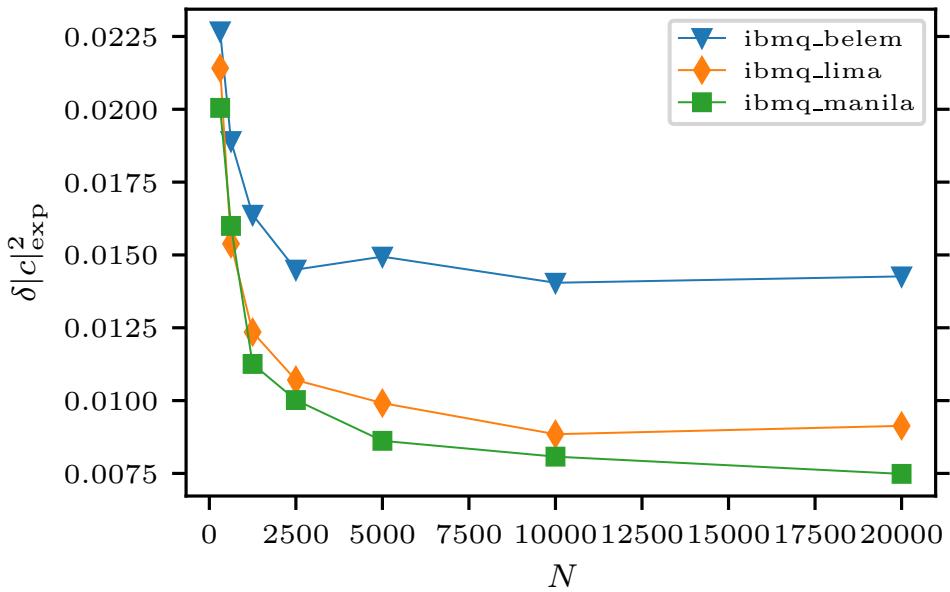


Figure 3.9 Mean error in measured occupation probabilities as a function of the number of shots N . A higher number of shots reduces statistical noise and provides an estimation of the measurement error of the quantum computer. The error in occupation probabilities from these simulations is capped at $\delta|c|^2_{\text{exp}} = 0.016$, which constitutes only a minor portion of the error seen in calculated fidelities and expectation values.

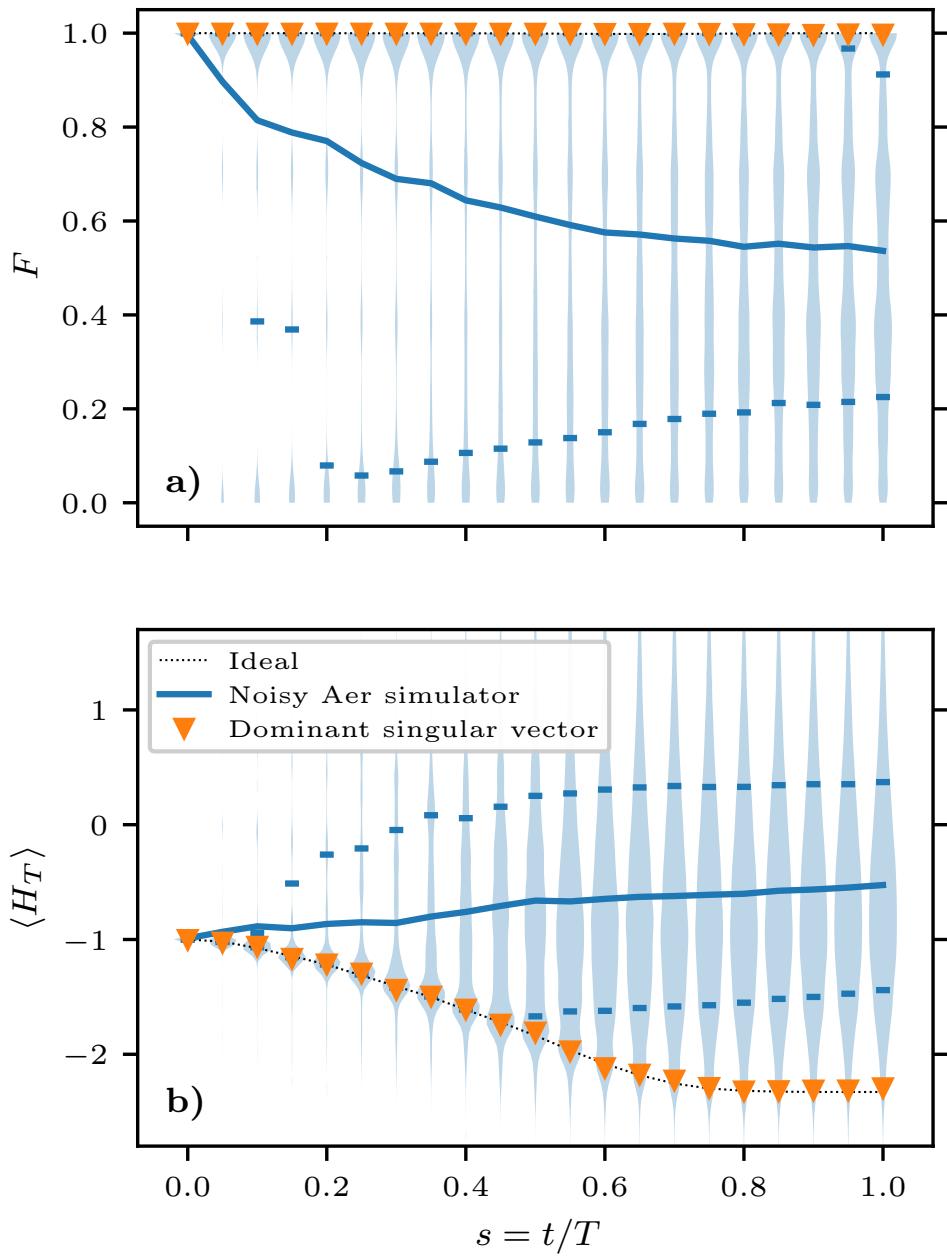


Figure 3.10 Evolution of **a)** fidelity and **b)** expectation value $\langle H_T \rangle$ using Qiskit's Aer simulator with a noise model based on *ibmq_belem*. Solid blue lines represent the expected results when incorporating all error sources considered by Aer, while the blue shaded areas show the probable values at each time interval. Dotted black lines indicate the ideal evolution in a noiseless system. Orange triangles come from singular value decomposition filtering of the data supporting the blue distributions. The closeness between the dotted lines and the orange triangles strongly implies that dissipative processes are the primary source of noise in IBMQ simulations, even at the early stages of the evolution.

CHAPTER 4

RODEO ALGORITHM ANALYSIS AND DEMONSTRATION

Quantum adiabatic evolution, while theoretically an accurate method for preparing eigenstates of quantum Hamiltonians, poses computational challenges when applied to large systems. In this chapter, we discuss the rodeo algorithm as a viable alternative. The Rodeo algorithm (RA) uses stochastic interference to suppress unwanted eigenvectors, offering an exponential speedup over quantum phase estimation and adiabatic evolution [29].

In Section 4.1, we present a detailed analysis of the rodeo algorithm, and in Section 4.2, we discuss a demonstration of its application on quantum hardware by calculating the eigenvalues of a stochastic one-qubit Hamiltonian using IBM's Casablanca device.

4.1 Rodeo algorithm

Prior to the development of the rodeo algorithm, there had been recent advancements in Hamiltonian evolution on quantum computers using tools such as Lie-Trotter-Suzuki formulas[110, 106] and linear combinations of unitary matrices [27]. Hardware limitations presented a significant challenge for all known quantum state preparation approaches. This algorithm was developed in an attempt to mitigate those challenges and enable the preparation of eigenstates on NISQ-era devices.

The rodeo algorithm is an iterative method that prepares a target eigenvector by "shaking off" all other states like a rodeo horse. It is a dissipative process inspired by the projected cooling algorithm [71, 48] that can be applied recursively and converges to the desired eigenstate exponentially with the number of repetitions. It can theoretically prepare any target eigenstate of any quantum Hamiltonian given an initial state that has a sufficiently high overlap with that target state. It appears similar to iterative QPE [65] and fixed-time energy band filtering [42, 74]; however, these methods cannot efficiently prepare individual eigenstates of a general quantum Hamiltonian, making the rodeo algorithm more suitable for quantum simulation applications.

4.1.0.1 Algorithm Description

We designate the Hamiltonian of interest as the "object Hamiltonian", H_{obj} , and the part of the quantum register on which it acts as the "object system". We then choose an energy interval

$[E - \epsilon, E + \epsilon]$. The goal of the algorithm is to prepare this object system in an eigenstate of H_{obj} with an eigenvalue within that energy interval.

The object system starts in an initial state denoted $|\psi_I\rangle$ and evolves into a final state $|\psi_F\rangle$. Auxiliary qubits, called ancilla qubits, allow indirect manipulation of the object system through entangling gate operations and measurements. This set of ancilla qubits is informally termed the 'rodeo arena.' When mid-circuit measurements are allowed, as is the case for one-way quantum computers [95] and quantum computers that support dynamic circuits [31] such as the superconducting quantum computers available on IBMQ, a single ancilla qubit can be reused throughout the recursive process, which allows the rodeo arena to merge into a single qubit. Since mid-circuit measurements are becoming more widely available, subsequent analysis will assume access to them.

One circuit of the RA is implemented as follows:

1. **Choose parameters:** Set E as a guess for the eigenvalue of an eigenstate of H_{obj} and choose a random non-zero value for t .
2. **Initialization:** The ancilla qubit is prepared in the $|0\rangle$ state, then a Hadamard gate is applied to place it in a superposition of the computational basis. The object system can be in any state, labeled $|\psi_I\rangle$. For best results, this state should be chosen to have non-negligible overlap with the eigenstate of H_{obj} closest to E .
3. **Controlled time evolution:** Use the ancilla qubit to control the time evolution operation with Hamiltonian H_{obj} and time parameter t on the object system.
4. **Phase Shift Gate:** Apply a phase rotation gate $P(Et)$ on the ancilla qubit. This imparts the phase e^{iEt} to the $|0\rangle$ component while leaving the $|1\rangle$ component unchanged.
5. **Rotate and Measure:** Apply the Hadamard gate to the ancilla qubit once more, then measure only the ancilla qubit.

The quantum circuit for this algorithm with one ancilla qubit is shown in Figure 4.1.

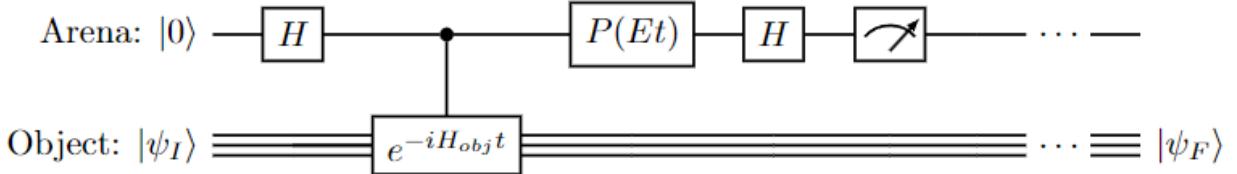


Figure 4.1 One cycle of the Rodeo algorithm. Additional copies of this circuit can be appended to increase its efficacy. If mid-circuit measurements are allowed, only one ancilla is needed for the arena, otherwise additional ancilla qubits may be used, with one ancilla qubit participating in each cycle.

This choice of gates is intuitive because of the similarity between the time evolution operator $e^{-iH_{obj}t_n}$ and the phase rotation e^{iEt_n} . As we show in Section 4.1.1 their combined effect on the quantum register will result in interference in the wave function of the object system, exponentially suppressing its relative overlap with each eigenstate by a factor proportional to the difference between its eigenvalue and E . The suppression factor for each eigenstate will also unpredictably depend on t_n , but the effect of this choice can be diminished by selecting different values for each t_n .

4.1.1 Mathematical Analysis of RA

By representing gate operations in matrix form, we can exactly predict the state vector output of the RA circuit. The ancilla qubit is initialized in the $|0\rangle$ state and the object system is initialized in the $|\psi_I\rangle$ state, so the initial state can be written as $|0\rangle \otimes |\psi_I\rangle$. Writing the gate operations as matrix operations on this initial state shows the effect of one cycle of the RA on this state.

change 1 to 0 in one cycle derivation

$$\text{RA}(|0\rangle \otimes |\psi_I\rangle) = \begin{bmatrix} \frac{I}{\sqrt{2}} & \frac{I}{\sqrt{2}} \\ \frac{I}{\sqrt{2}} & \frac{-I}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I e^{iEt} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & e^{-iH_{obj}t} \end{bmatrix} \begin{bmatrix} \frac{I}{\sqrt{2}} & \frac{I}{\sqrt{2}} \\ \frac{I}{\sqrt{2}} & \frac{-I}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} |\psi_I\rangle \\ 0 \end{bmatrix} \quad (4.1)$$

$$= \begin{bmatrix} \left[\frac{I}{2} + \frac{1}{2} e^{-i(H_{obj}-E)t} \right] |\psi_I\rangle \\ \left[\frac{I}{2} - \frac{1}{2} e^{-i(H_{obj}-E)t} \right] |\psi_I\rangle \end{bmatrix} \quad (4.2)$$

$$= |\psi_F\rangle \quad (4.3)$$

Here, I represents the identity operator with the same dimension as the object system. We will represent these eigenvectors of H_{obj} as $|E_k\rangle$ with the corresponding eigenvalues E_k . Since these eigenvectors form a basis for the object system, $|\psi_I\rangle$ can be written as a linear combination of these eigenstates as follows:

$$|\psi_I\rangle = \sum_k c_k E_k |E_k\rangle \quad (4.4)$$

H_{obj} commutes with all gates in the circuit, so the total effect of the circuit can be understood in terms of its effect on each of these components of $|\psi_I\rangle$. The probability of measuring the ancilla qubit in the $|0\rangle$ state can then be extracted from the first entry in the final statevector in Equation 4.3 as follows:

$$P(|0\rangle) = \langle \psi_I | \left[\frac{I}{2} + \frac{1}{2} e^{-i(H_{\text{obj}}-E)t} \right]^\dagger \left[\frac{I}{2} + \frac{1}{2} e^{-i(H_{\text{obj}}-E)t} \right] |\psi_I\rangle \quad (4.5)$$

$$= \sum_k c_k^2 \left| \frac{1}{2} + \frac{1}{2} e^{-i(E_k-E)t} \right|^2 \quad (4.6)$$

$$= \sum_k c_k^2 \cos^2 \left[(E_k - E) \frac{t}{2} \right] \quad (4.7)$$

When the RA is repeated for N cycles with different values of the time parameter t_n , the probability of measuring the ancilla qubit in the $|0\rangle$ state each time is the product of these probabilities.

$$P_N = \prod_{n=1}^N \sum_k c_k^2 \cos^2 \left((E_k - E) \frac{t_n}{2} \right) \quad (4.8)$$

We call this probability the "success probability" because when this measurement occurs, it corresponds to a high probability that the final state of the object system is in an eigenstate of H_{obj} with eigenvalue E . If t_n are sampled from a uniform distribution, as N becomes large, the spectral weight of any $E_k \neq E$ is suppressed by a factor of $1/4^N$, due to the fact that the geometric mean of the \cos^2 function is $\frac{1}{4}$. If t_n are sampled from a Gaussian distribution where the root mean square value is σ , then when σ scales as $O(1/\epsilon)$, eigenstates that fall outside the interval $[E - \epsilon, E + \epsilon]$ are exponentially suppressed. Thus, we can intuitively say that altering σ changes the magnification of the energy sensor of the RA

In the case where E is close to an eigenvalue E_{obj} (associated with the eigenvector $|E_{\text{obj}}\rangle$) of H_{obj} , for large N , the probability of consecutive measurement of the ancilla in the $|0\rangle$ state N times converges to the following:

$$P_N(E) = |\langle \psi_I | E_{\text{obj}} \rangle|^2 \prod_{n=1}^N \cos^2 \left((E_{\text{obj}} - E) \frac{t_n}{2} \right) \quad (4.9)$$

This is a simplified version of Equation 4.8, where all terms in the sum are excluded except the one involving $|E_j\rangle$, which was the least suppressed, and the coefficient for that term is replaced by the overlap $p = |\langle \psi_I | E_{\text{obj}} \rangle|^2$ of the initial state of the object system. As E approaches the perfect guess of $E = E_{\text{obj}}$, this expression simplifies even further to $P_N(E_{\text{obj}}) = p$. With the spectral weights of other eigenvectors outside $[E - \epsilon, E + \epsilon]$ reduced by δ , the computational effort for the rodeo algorithm scales as $N\sigma/p = O[|\log \delta|/(p\epsilon)]$. Furthermore, when t_n are sampled from the Gaussian distribution with mean value 0 and root mean square σ (that is, $t_n \sim \mathcal{N}(\mu, \sigma^2)$), the product in Equation 4.9 can be averaged over the time values t_n to derive a simple expression for the success probability as:

$$P_N(E) = \left[\frac{1 + e^{-(E_{\text{obj}} - E)^2 \sigma^2/2}}{2} \right]^N. \quad (4.10)$$

When a good guess for the parameter E is not available, a simple search algorithm can fine-tune it to approximate the energy eigenvalue E_{obj} within a given confidence interval. Achieving precision ϵ requires $O(\log \epsilon)$ energy scans, where each scan reduces the energy range by a factor K . This can be accomplished by performing each scan at multiple evenly spaced E values with a specified number of rodeo cycles and increasing σ by a factor of K each time. The total time evolution scales as $O(1/\epsilon)$, and high-fidelity energy scans add a factor $(\log \epsilon)^2 p$. Thus, the computational complexity to identify E_{obj} with error ϵ scales as $O[(\log \epsilon)^2/(p\epsilon)]$, approaching the theoretical limit of $O(1/\epsilon)$ set by the Heisenberg uncertainty principle. In comparison, phase estimation has a computational complexity of $O[1/(p\epsilon)]$ plus $O[(\log \epsilon)^2]$ for the quantum Fourier transform[83]. Iterative phase estimation, which does not require the Fourier transform, is limited to finding the

energies of pure eigenstates and requires $O(1/\epsilon^2)$ measurements to calculate the expectation value of such a state for a Hamiltonian due to statistical errors.

In addition to computing eigenvalues, the RA is also effective as an eigenstate preparation technique. When a good guess for E_{obj} is known and the initial state $|\psi_I\rangle$ has non-zero overlap with the corresponding eigenstate $|E_{\text{obj}}\rangle$, the success probability P_N grows rapidly with the number of cycles. Whenever a success occurs (that is, the ancilla qubit was measured in the $|0\rangle$ state every time), the final state of the object system $|\psi_F\rangle$ has significant overlap with $|E_{\text{obj}}\rangle$. This can be explained by the partial collapse of the object system wave function that occurs when the entangled ancilla qubits are measured. The configuration of the entanglement induced by the RA ensures that a measurement of $|0\rangle$ corresponds to the wave function collapsing toward the target state. Since no measurement operations are necessary on the object system qubits in the RA, these can then be fed forward into another algorithm, such as a quantum simulation.

To prepare an eigenstate $|E_j\rangle$ with a residual orthogonal component Δ , the computational cost is $O(\log \Delta/p)$. We maintain σ large and constant to filter the desired eigenstate, which requires $N = O(\log \Delta)$ iterations of the rodeo algorithm, and $1/p$ measurements per iteration. E must be centered on the peak of E_j , which introduces a constant factor to the computational cost. For phase estimation, the cost is $O[1/(p\Delta)]$, and for adiabatic evolution, it is $O(1/\Delta)$, adjusted by p based on the adiabatic path. Thus, for eigenstate preparation, the rodeo algorithm is exponentially more efficient as Δ approaches zero. Useful estimates for the magnitude of the residual orthogonal component Δ can be written as a function of the number of rodeo cycles N as follows:

$$F_A \equiv \sqrt{2^{-N}(1-p)/[p + 2^{-N}(1-p)]},$$

$$F_G \equiv \sqrt{4^{-N}(1-p)/[p + 4^{-N}(1-p)]}. \quad (4.11)$$

When N is much smaller than the number of eigenstates that have non-negligible overlap with the initial state, F_A is sufficiently accurate. This has a spectral suppression factor of $1/2$ for undesired eigenstates, equating to the arithmetic mean of \cos^2 . On the other hand, F_G is more accurate when N is greater than the number of eigenstates that have non-negligible overlap, with a suppression factor of $1/4$, equivalent to the geometric mean of \cos^2 .

RA performance largely depends on the overlap between the initial state and the target eigenstate, represented by the overlap probability p . Since for most systems of interest it is not trivial to prepare an initial state with a high value of p , it may often be helpful to use a preconditioning technique to improve the quality of the initial state and increase the effectiveness of the RA. One method of accomplishing this would be to combine the RA with a variational quantum algorithm; this idea is explored in great detail in Chapter 5.

4.2 Demonstration of the Rodeo Algorithm

In this section, we show two examples to demonstrate the application of the RA. In Section 4.2.1, we introduce a simple example using the Heisenberg model and emulate the RA circuit on a classical computer. Then, in Section 4.2.2, we use the RA to compute the eigenvalues of a random single-qubit Hamiltonian on the IBM Casablanca quantum computing device through IBMQ. For this single-qubit Hamiltonian, we also fully demonstrate the sequential scanning method discussed in Section 4.1.1 to compute the full energy spectrum of the Hamiltonian. We also invoke the Hellman-Feynman theorem to compute the expectation values of an arbitrary single-qubit observable.

4.2.1 Heisenberg Model Classical Simulation Example

In this section, we consider the spin- $\frac{1}{2}$ Heisenberg model in a uniform magnetic field with 10 interacting sites forming a closed loop in a one-dimensional chain [53]. This Hamiltonian can be written in the following way:

$$H_{\text{obj}} = J \sum_{\langle j, k \rangle} \vec{\sigma}_j \cdot \vec{\sigma}_k + h \sum_j \sigma_j^z, \quad (4.12)$$

Where J is the exchange coupling coefficient between the sites, $\vec{\sigma}_j$ are the Pauli matrices on site j , $\langle j, k \rangle$ is shorthand for nearest-neighbor pairs of sites, and h is the coupling to a uniform magnetic field in the z direction. For this example, we will consider the antiferromagnetic case, which requires setting the values $J = 1$ and $h = 3$. The initial state is chosen to be the alternating tensor product state:

$$|\psi_I\rangle = |0101010101\rangle. \quad (4.13)$$

This choice of initial state is motivated by its high degree of symmetry, which allows us to predict that it will have a nonzero overlap with a relatively small number of energy eigenstates. The energy eigenstates of H_{obj} will be labeled $|E_j\rangle$. The initial-state spectral function $S(E)$, which describes the energy distribution of $|\psi_I\rangle$ in terms of these eigenstates, is defined as:

$$S(E) = \begin{cases} |\langle E_j | \psi_I \rangle|^2 & \text{if } E = E_j, \\ 0 & \text{otherwise.} \end{cases}$$

Here, $|\langle E_j | \psi_I \rangle|^2$ is the probability that the initial state $|\psi_I\rangle$ overlaps with the energy eigenstate $|E_j\rangle$. In cases of degenerate eigenvalues, the total contribution is the sum of contributions from all degenerate states. Figure 4.2 shows the initial-state spectral function using the rodeo algorithm for the Heisenberg spin chain with $N = 3$ (thin blue line), $N = 6$ (thick green line) and $N = 9$ (medium red line) cycles. We used 20 sets of random t_n values sampled from a Gaussian distribution with $\sigma = 5$ to reduce stochastic noise and create a constant background distinguishable from the spectral signal. The exact initial-state spectral function is shown with black open circles. Since both σ and N are not very large, relatively few gates would be needed to implement this circuit on a real device, which is important on NISQ devices.

In addition to computing the initial-state spectral function, we can generate any energy eigenstate that overlaps the initial state. Table 4.1 shows the overlap of $|\psi_F\rangle$ with the energy eigenvector $|E_j\rangle$ after N cycles of the rodeo algorithm. Gaussian random values are used for t_n with $\sigma = 5$ and $E = E_j$. The table shows that all such energy eigenvectors can be prepared accurately with only a few rodeo cycles.

In Figure 4.3, we show the logarithm of the error of the wave function using the RA to prepare the energy eigenstate $|E_j\rangle$ with $E_j = -18.1$. We plot $\log \Delta$ against the total propagation time T for $\sigma = 1$ to compare with $\log F_A$ and $\log F_G$ from the error estimates in Equation (4.11). For small T , $\log F_A$ is a good estimate; for large T , $\log F_G$ is better. We also compare results using classical

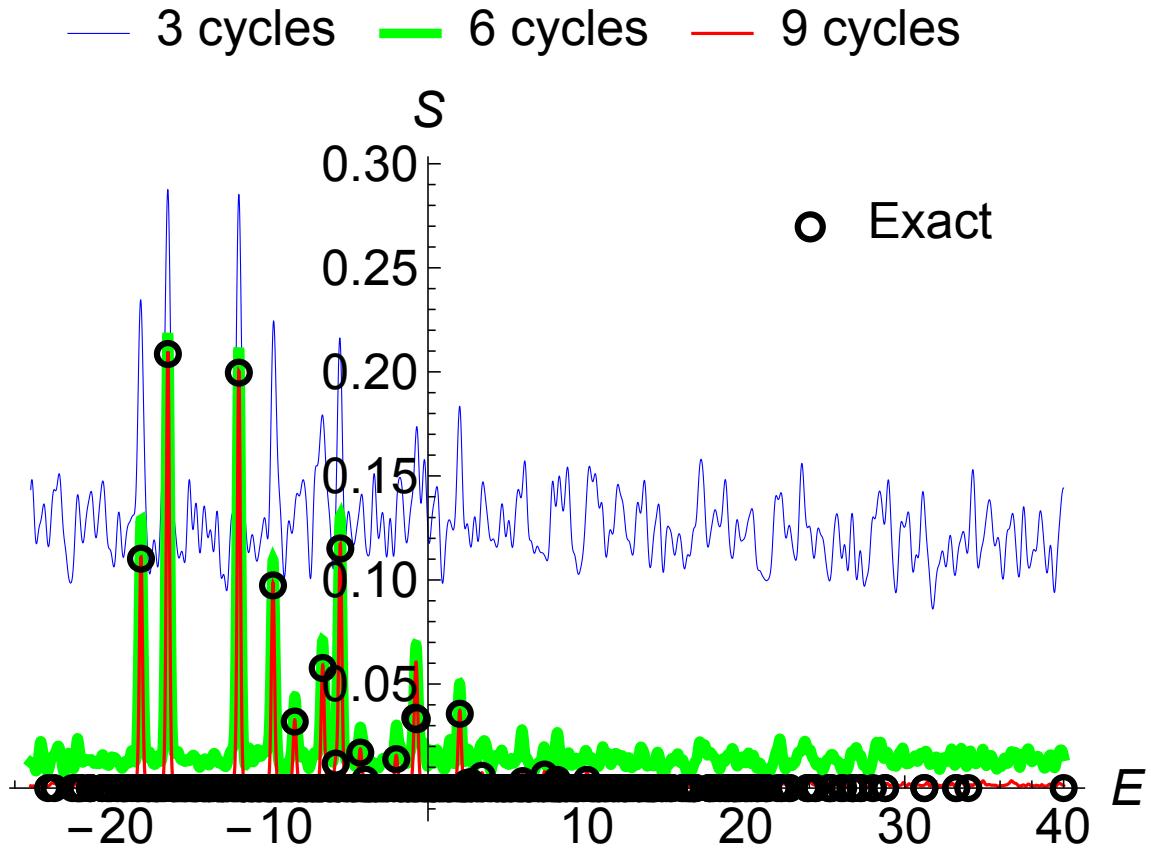


Figure 4.2 (color online) **Initial-state spectral function for the Heisenberg model.** Using the RA, we depict the initial-state spectral function for the Heisenberg spin chain with 3 (thin blue line), 6 (thick green line), and 9 (medium red line) cycles. The average was taken over 20 sets of Gaussian random values for t_n with $\sigma = 5$. For reference, the exact initial-state spectral function is presented with black open circles.

simulations of QPE and adiabatic evolution starting from the same initial state for the same total propagation times. For adiabatic evolution, the initial Hamiltonian is $H_I = \sum_{j=1}^{10} (-1)^j \sigma_j^z$ with an interpolating function $H(t) = \cos^2[\pi t/(2T)]H_I + \sin^2[\pi t/(2T)]H_{\text{obj}}$. Phase estimation and adiabatic evolution perform similarly, but the rodeo algorithm is exponentially faster.

4.2.2 Applications to Single-Qubit Hamiltonians on Quantum Hardware

We consider a quantum register with two qubits: the primary "object" qubit and the ancillary "arena" qubit. A single-qubit Hamiltonian is represented in its general form as:

$$H_{\text{obj}} = c_I I + c_X X + c_Y Y + c_Z Z, \quad (4.14)$$

Table 4.1 Overlap of $|\psi_F\rangle$ with energy eigenvector $|E_j\rangle$ after N cycles of the rodeo algorithm using Gaussian random values for t_n with $\sigma = 5$ and $E = E_j$.

E_j	$N = 0$	$N = 3$	$N = 6$	$N = 9$
-18.1	0.110	0.746	0.939	0.997
-16.4	0.209	0.841	0.993	1.000
-11.9	0.200	0.629	0.889	0.999
-9.76	0.0974	0.488	0.903	0.999
-8.38	0.0320	0.467	0.832	0.993
-6.63	0.0577	0.309	0.818	0.996
-5.81	0.0118	0.179	0.637	0.817
-5.52	0.115	0.456	0.766	0.997
-4.26	0.0171	0.144	0.696	0.995
-3.95	0.00401	0.0430	0.343	0.952
-2.00	0.0139	0.158	0.593	0.942
-0.802	0.0338	0.216	0.545	0.594
-0.704	0.0331	0.286	0.540	0.585
2.00	0.0357	0.371	0.925	0.994
2.42	0.00235	0.0122	0.0874	0.521
2.68	0.00291	0.0845	0.639	0.929
3.39	0.00592	0.0360	0.754	0.943
5.96	0.00336	0.0951	0.559	0.981
7.33	0.00650	0.184	0.792	0.978
8.13	0.00393	0.0832	0.665	0.841
8.24	0.00105	0.0275	0.142	0.289
10.0	0.00397	0.0128	0.295	0.902

where I is the identity operator, X, Y, Z are Pauli operators and c_I, c_X, c_Y, c_Z are real coefficients.

The object qubit is initialized in state $|\psi_I\rangle$, and the ancillary qubit is initialized in state $|0\rangle$. We perform N cycles of the RA as defined in Section 4.1 with the energy parameter E . IBMQ’s mid-circuit measurements permit reuse of the ancillary qubit so that only two qubits total are needed. The time parameter t_n for each cycle is randomly sampled from a Gaussian distribution with a mean of zero and a standard deviation of σ . From Equation 4.10, it can be inferred that the success probability will decrease exponentially as the difference between E and the nearest eigenstate of H_{obj} increases.

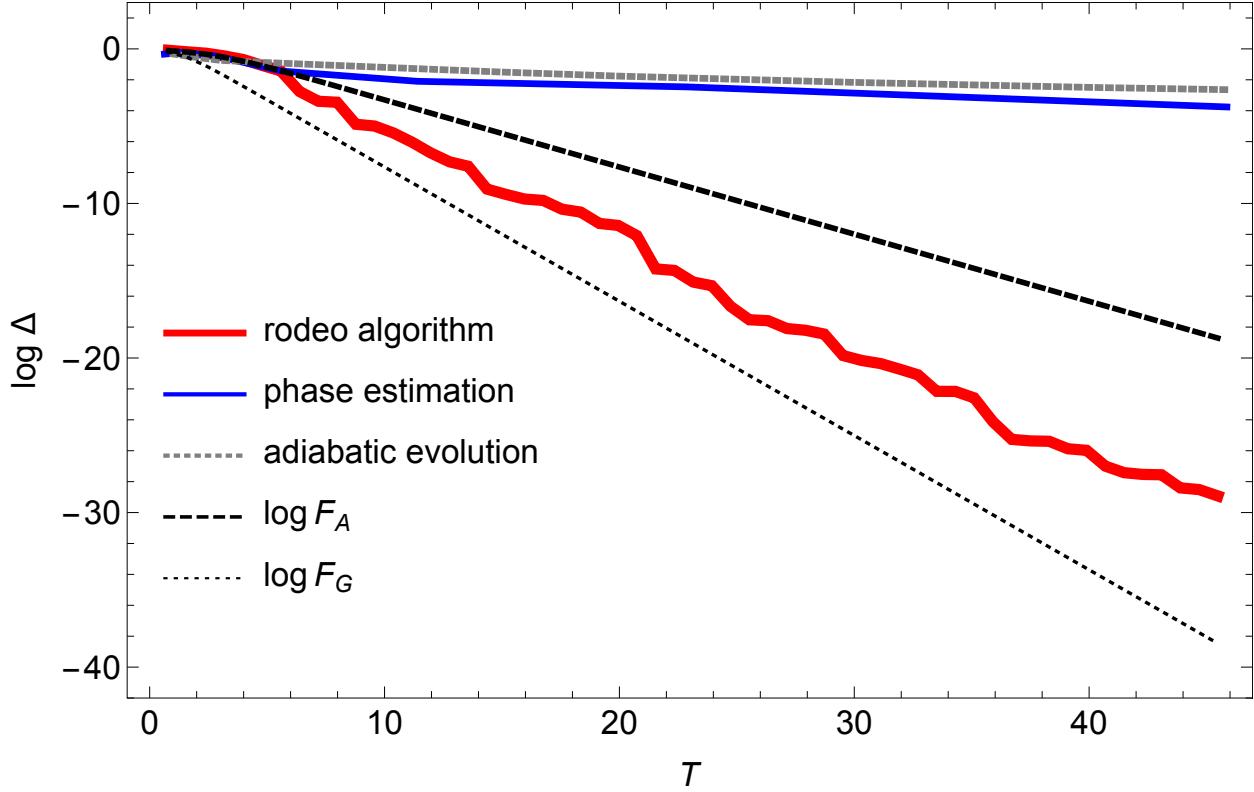


Figure 4.3 (color online) **Logarithm of the wave function error versus the total propagation time for the Heisenberg model.** We plot $\log \Delta$, versus the total propagation time, T , for the Heisenberg model. We show results for the rodeo algorithm, phase estimation, and adiabatic evolution. We also show the asymptotic estimates $\log F_A$ and $\log F_G$.

4.2.2.1 Gate Implementation For Single Qubit System

The description of the quantum circuit for the RA in Section 4.2.2 includes a controlled time-evolution gate in each cycle. Although such a gate operation could be done on a quantum device using optimal control as in Chapter 3, in general such an operation is not available, and it must be decomposed into elementary quantum gates in order to be implemented.

The time evolution operator can be written from the single-qubit object Hamiltonian H_{obj} as follows:

$$U(t) = e^{-iH_{\text{obj}}t} = e^{-ic_I t} e^{-\frac{i\theta}{2}\hat{n}\cdot\vec{\sigma}} \equiv e^{-ic_I t} R_{\hat{n}}(\theta), \quad (4.15)$$

where $R_{\hat{n}}(\theta)$ represents a rotation matrix around the unit vector \hat{n} in three dimensions by an angle θ . This matrix can parameterize any matrix in the fundamental representation of the $SU(2)$

group and thus can be used as a general representation of any single qubit gate operation. It can be written in terms of the coefficients from Equation 4.14 as follows:

$$R_{\hat{n}}(\theta) = e^{-\frac{i\theta}{2}\hat{n}\cdot\vec{\sigma}}$$

$$= \begin{bmatrix} \cos(\frac{\theta}{2}) - i \sin(\frac{\theta}{2})n_Z & -i \sin(\frac{\theta}{2})(n_X - in_Y) \\ -i \sin(\frac{\theta}{2})(n_X + in_Y) & \cos(\frac{\theta}{2}) + i \sin(\frac{\theta}{2})n_Z \end{bmatrix}, \quad (4.16)$$

where

$$\theta = 2t\sqrt{c_X^2 + c_Y^2 + c_Z^2}, \quad (4.17)$$

and

$$\hat{n} = \frac{1}{\sqrt{c_X^2 + c_Y^2 + c_Z^2}} \begin{bmatrix} c_X \\ c_Y \\ c_Z \end{bmatrix} = \begin{bmatrix} n_X \\ n_Y \\ n_Z \end{bmatrix}. \quad (4.18)$$

A generic single-qubit quantum operation U is conventionally parameterized with three Euler angles γ, β, δ as follows:

$$U(\gamma, \beta, \delta) = \begin{bmatrix} \cos(\frac{\gamma}{2}) & -e^{i\delta} \sin(\frac{\gamma}{2}) \\ e^{i\beta} \sin(\frac{\gamma}{2}) & e^{i(\delta+\beta)} \cos(\frac{\gamma}{2}) \end{bmatrix}. \quad (4.19)$$

Applying the $Z - Y$ decomposition for a single qubit [83], we can rewrite Equation 4.19 as:

$$U(\gamma, \beta, \delta) = e^{i\frac{\delta+\beta}{2}} \begin{bmatrix} e^{-i\frac{\delta+\beta}{2}} \cos(\frac{\gamma}{2}) & -e^{i\frac{\delta-\beta}{2}} \sin(\frac{\gamma}{2}) \\ e^{-i\frac{\delta-\beta}{2}} \sin(\frac{\gamma}{2}) & e^{i\frac{\delta+\beta}{2}} \cos(\frac{\gamma}{2}) \end{bmatrix}$$

$$= e^{i\frac{\delta+\beta}{2}} R_Z(\beta) R_Y(\gamma) R_Z(\delta)$$

$$\equiv e^{i\frac{\delta+\beta}{2}} R_{\hat{n}}(\theta). \quad (4.20)$$

Equating the upper-left and lower-left entries of the matrices $R_{\hat{n}}(\theta)$ and $R_Z(\beta)R_Y(\gamma)R_Z(\delta)$ yields the following constraints:

$$\cos\left(\frac{\delta+\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) = \cos\left(\frac{\theta}{2}\right), \quad (4.21)$$

$$-\sin\left(\frac{\delta+\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) = -n_Z \sin\left(\frac{\theta}{2}\right), \quad (4.22)$$

$$\cos\left(\frac{\delta-\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) = n_Y \sin\left(\frac{\theta}{2}\right), \quad (4.23)$$

$$-\sin\left(\frac{\delta-\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) = -n_X \sin\left(\frac{\theta}{2}\right). \quad (4.24)$$

The global phase $e^{i\frac{\delta+\beta}{2}}$ in the gate U can be replaced with the overall phase controlled by the term $c_I I$, generating two additional terms ξ in the argument of the phase shift gate. Solving for the parameters δ , β , γ , and ξ yields the following:

$$\delta = \tan^{-1} \left[n_Z \tan\left(\frac{\theta}{2}\right) \right] + \tan^{-1} \left(\frac{n_X}{n_Y} \right), \quad (4.25)$$

$$\beta = \tan^{-1} \left[n_Z \tan\left(\frac{\theta}{2}\right) \right] - \tan^{-1} \left(\frac{n_X}{n_Y} \right), \quad (4.26)$$

$$\gamma = 2 \cos^{-1} \left[\frac{\cos(\theta/2)}{\cos\left(\frac{\delta+\beta}{2}\right)} \right], \quad (4.27)$$

$$\xi = -c_I t - \frac{\delta + \beta}{2}. \quad (4.28)$$

The parameters δ , β , and γ can be used as arguments in the elementary controlled- U gate operation in order to implement the controlled time evolution in the quantum circuit.

4.2.2.2 Determining the Energy Spectrum

To find the eigenvalues of H_{obj} , we implement RA while scanning the target energy E from E_{min} to E_{max} , a range that is estimated from the operator norm of H_{obj} . The eigenvalues appear as peaks in the success probability distribution. We used multiple scans, each with higher resolution by adjusting the width parameter σ , inversely proportional to the sharpness of the resolution of the energy. Three scans are used in this example; the second and third scans are centered around the peaks of their previous scans, as illustrated in Figure 4.4.

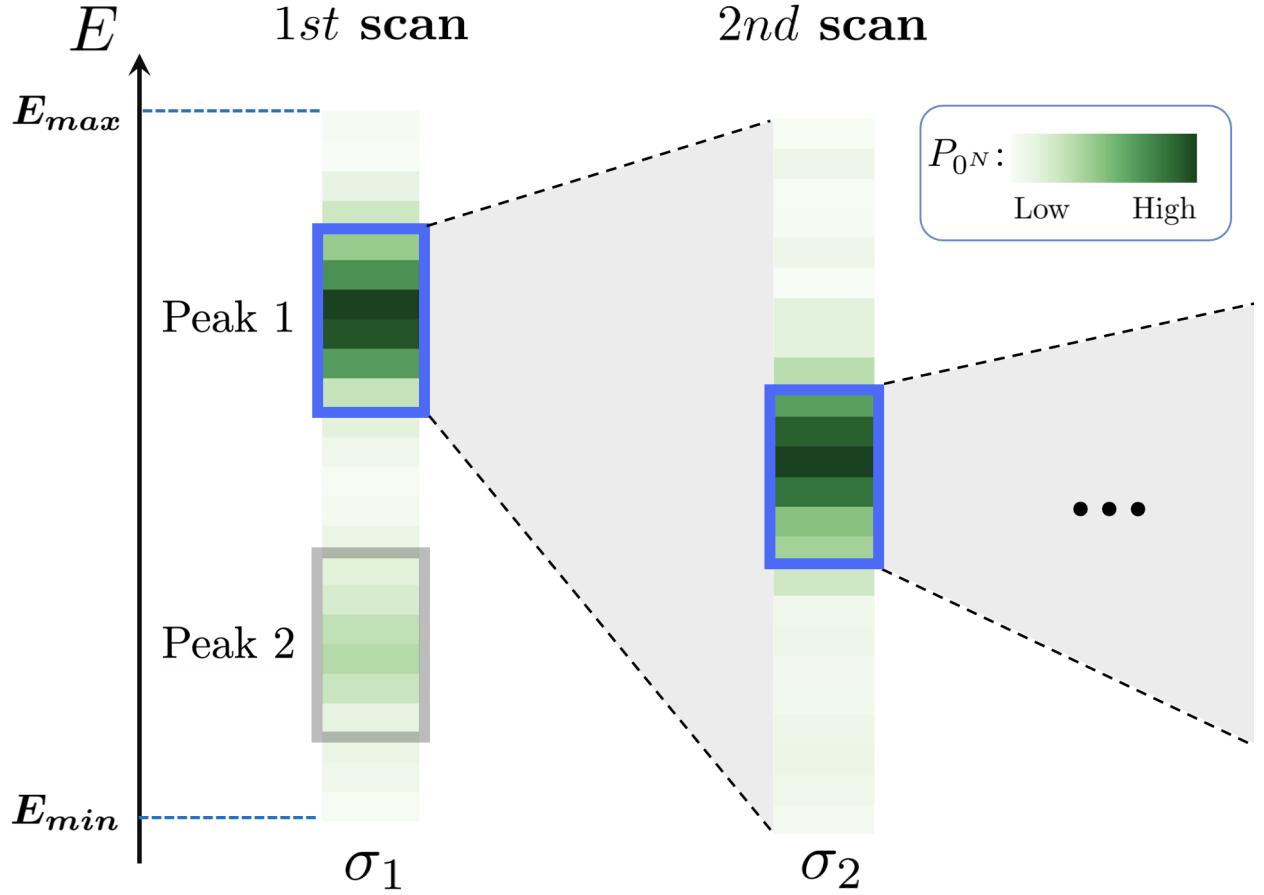


Figure 4.4 Sequential scans of the energy. Each bin represents a distinct RA circuit for target energy E and width parameter σ . The color and shading indicates the success probability $P_{0^N}(E)$. Centered around each of the peaks from the first scan, a second scan is performed using with a large value of σ and better energy resolution. This is then repeated for the third scan.

For the purposes of the following analysis, we consider a one-parameter family of one-qubit Hamiltonians:

$$H_{\text{obj}}(\phi) = H^{(0)} + \phi H^{(1)}, \quad (4.29)$$

Written in terms of two single-qubit Hamiltonians:

$$H^{(0)} = -0.08496I - 0.89134X + 0.26536Y + 0.57205Z, \quad (4.30)$$

and

$$H^{(1)} = -0.84537I + 0.00673X - 0.29354Y + 0.18477Z. \quad (4.31)$$

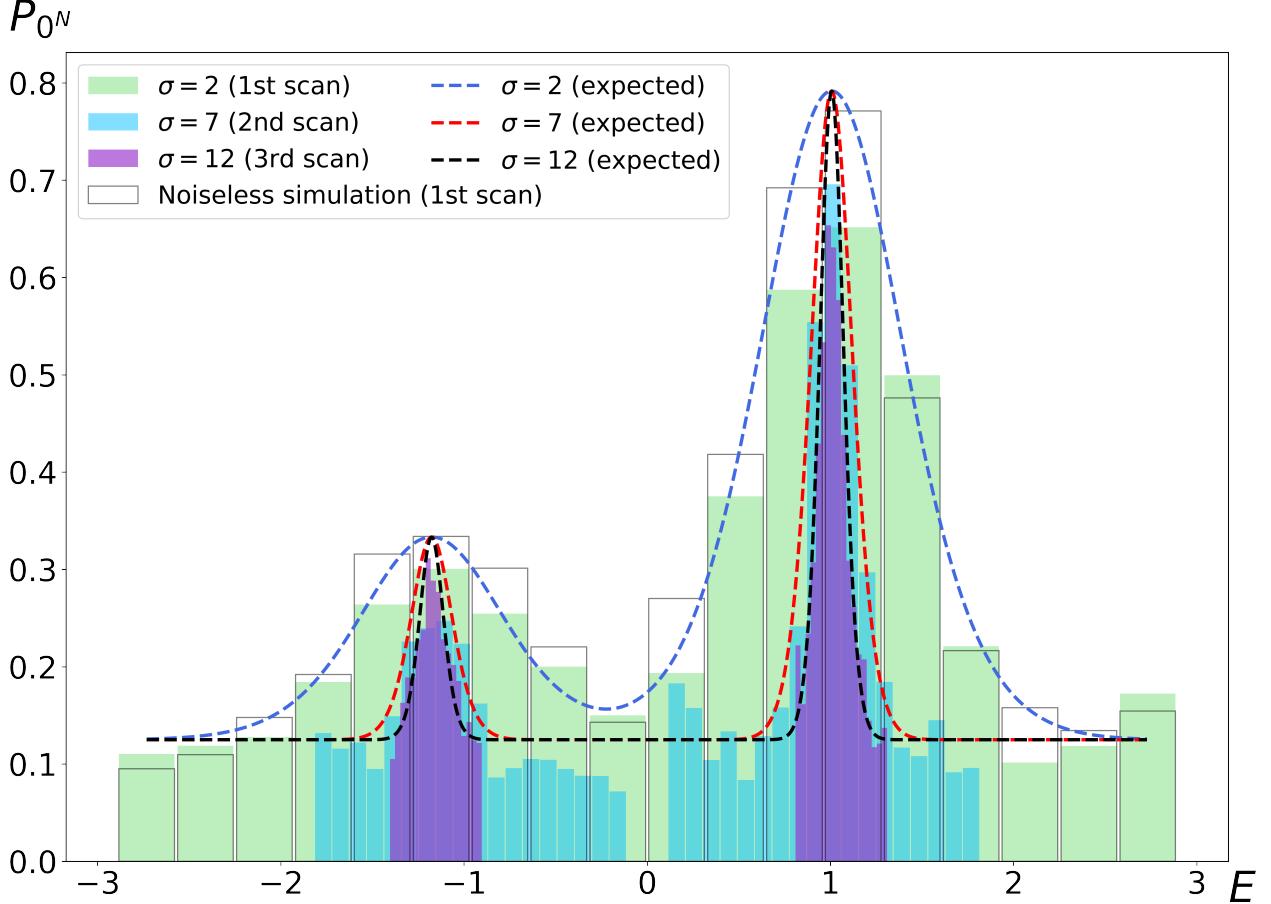


Figure 4.5 **Energy scans for $H_{\text{obj}}(0)$** . The results were obtained from experiments on IBMQ’s Casablanca device. The dashed lines show predicted outcomes from classical success probability calculations. The initial scan also includes noiseless quantum device simulation results.

The coefficients in both Hamiltonians were randomly sampled from a uniform distribution over the interval $[-1, 1]$. We also set the number of cycles $N = 3$ for all quantum circuits in this section. This choice minimizes error due to decoherence by keeping the number of gates low, and the results from Section 4.2.1 suggest that this should be enough to see the effect of RA. The results for $H_{\text{obj}}(0)$ are illustrated in Figure 4.5. The initial state is set to $|0\rangle$, and we perform three separate energy scans with σ values set to 2, 7, and 12 to show its inverse effect on the magnification of the energy sensor. The experimental results were obtained using the IBMQ Casablanca device, focusing on two interconnected qubits characterized by minimal real-time error rates. The dashed lines represent the analytic results as derived from classical success probability calculations. Additionally, for the initial scan, we include results from a noiseless quantum device simulator.

Although the heights of the peaks are lower, their locations remain accurate in the presence of noise on the quantum device. Peak positions are estimated by fitting Gaussian functions to the success probability data in the vicinity of each peak. When the difference between E and the target eigenvalue is greater than $1/\sigma$, the probability of measuring $|0\rangle$ is approximately $1/2$ per cycle. With the number of cycles $N = 3$, the probability of success is $(1/2)^3 = 0.125$, which is illustrated by the background value in Figure 4.5.

4.2.2.3 Applying the Hellmann-Feynman Theorem

Using the eigenvalues of $H_{\text{obj}}(\phi)$ for small ϕ , we can apply the Hellmann-Feynman theorem to find the expectation value of $H^{(1)}$ for the eigenstates of $H^{(0)}$ [37]. This theorem essentially represents the first-order perturbation theory for energy. If $E_n(\phi)$ are the energy eigenvalues of $H_{\text{obj}}(\phi)$ and $|\psi_n(\phi)\rangle$ are the corresponding eigenstates, then:

$$\frac{dE_n(\phi)}{d\phi} = \langle \psi_n(\phi) | H^{(1)} | \psi_n(\phi) \rangle. \quad (4.32)$$

At $\phi = 0$, we obtain the expectation values of $H^{(1)}$ for the eigenstates of $H_{\text{obj}}(0) = H^{(0)}$. In Figure 4.6, the energy eigenvalues of $H_{\text{obj}}(\phi)$ are plotted. The upper panel shows the higher energy eigenvalue E_1 , and the lower panel shows the lower energy eigenvalue E_2 . As ϕ varies, a scan with $\sigma = 12$ suffices to find each of these eigenvalues. We perform 2500 measurements for 25 random sets of t_n for E_1 and 5000 measurements for 50 random sets of t_n for E_2 , each set containing 3 numbers due to the chosen value $N = 3$. RA results (filled circles) are plotted with a quadratic fit (solid line) and three-standard-deviation error bands (shaded). The exact results (filled squares) and their quadratic fit (dashed line) are also shown. The error bars on the RA data represent errors of one standard deviation from the mean. The smaller error bars for E_1 compared to E_2 are due to a larger overlap of the initial state with the corresponding eigenstate.

Using quadratic fitting of the RA data points, we determine the energy eigenvalues of $H^{(0)}$ and the expectation values of $H^{(1)}$ for the eigenstates of $H^{(0)}$. The results are in Table 4.2. The error bars represent the one-sigma uncertainties from statistical noise and Gaussian peak fitting. The exact results are provided for comparison. The relative error in determining the energies of $H^{(0)}$ is 0.08%, which falls within our error estimates even without the application of error mitigation

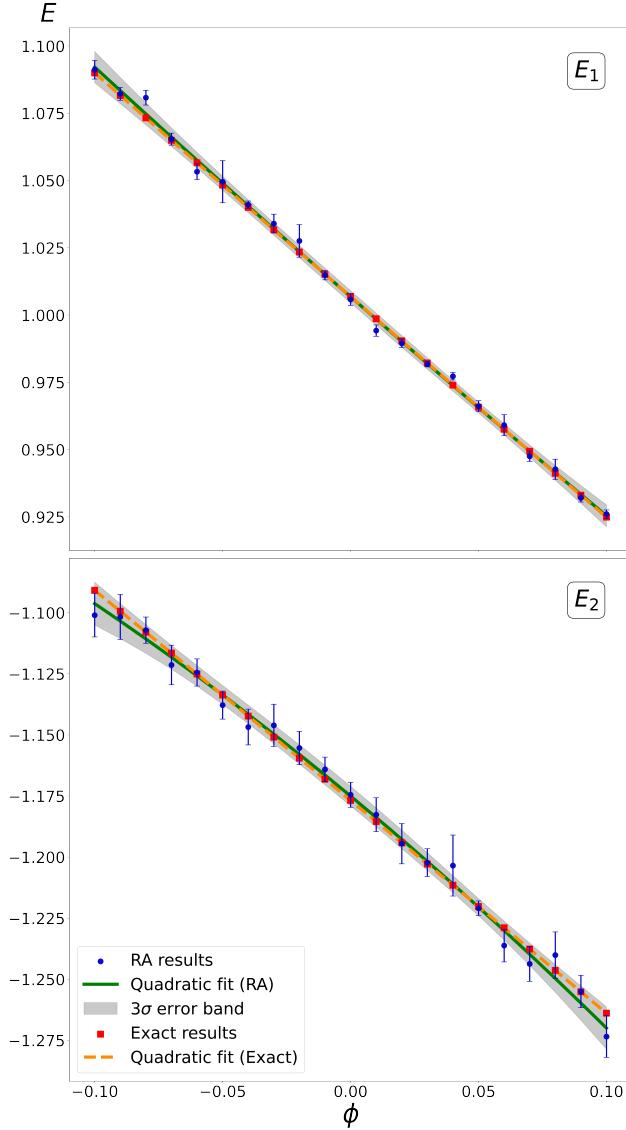


Figure 4.6 Eigenvalues E_1 (top) and E_2 (bottom) vs. ϕ . The plot shows RA data (circles), RA quadratic fit (solid line) with 3σ error bands (shaded), exact values (squares), and exact quadratic fit (dashed line).

techniques. The application of the Hellmann-Feynman theorem results in larger error bars due to the error being proportional to the derivative of the energy ; however, the uncertainty in the expectation value of $H^{(1)}$ remains relatively small. The relative error for the eigenvalues of $H^{(1)}$ is 0. 7%, which is well in agreement with our predicted error estimates.

	$ \psi_1(0)\rangle$	exact	$ \psi_2(0)\rangle$	exact
$\langle H^{(0)} \rangle$	1.00681(66)	1.00690	-1.1750(12)	-1.1768
$\langle H^{(1)} \rangle$	-0.8338(89)	-0.8254	-0.868(14)	-0.8653

Table 4.2 Results obtained using RA for the energy eigenvalues of $H^{(0)}$ and the expectation values of $H^{(1)}$ corresponding to the eigenstates of $H^{(0)}$. For reference, exact values are also provided.

4.2.2.4 Testing Eigenstate Preparation

To assess the outcomes of the RA as a method to prepare eigenstates, we used the IBMQ Casablanca device to prepare eigenstates $|\psi_1(0)\rangle$ and $|\psi_2(0)\rangle$ and measure the expectation values of $H^{(0)}$ and $H^{(1)}$. This calculation can be thought of as an upper limit on the accuracy of the variational quantum eigensolver [88]. Table 4.3 shows the results without using any error mitigation techniques, displaying expectation values of $X, Y, Z, H^{(0)}$, and $H^{(1)}$. The error bars represent statistical errors from 10 trials of 5000 measurements for each Pauli operator and eigenstate. The relative error for $H^{(0)}$ is 5% while for $H^{(1)}$ it is 0.6%, with the smaller deviation for $H^{(1)}$ attributed to the smaller coefficients of the Pauli matrices. Both deviations significantly exceed statistical error estimates, indicating systematic errors likely due to measurement bias.

	$ \psi_1(0)\rangle$	exact	$ \psi_2(0)\rangle$	exact
$\langle X \rangle$	-0.7455(44)	-0.8164	0.8055(22)	0.8164
$\langle Y \rangle$	0.2750(36)	0.2430	-0.2196(25)	-0.2430
$\langle Z \rangle$	0.5356(46)	0.5239	-0.4632(21)	-0.5239
$\langle H^{(0)} \rangle$	0.9589(48)	1.0069	-1.1262(24)	-1.1768
$\langle H^{(1)} \rangle$	-0.8321(14)	-0.8254	-0.86109(84)	-0.8653

Table 4.3 Results for eigenstate preparation without measurement error mitigation.

As expected, the results without error mitigation on a noisy device were sub-optimal. Thus, we reanalyzed the data while including measurement error mitigation techniques. Before the 10 trials, we collected data for the confusion matrix using the same method as in Section 3.3.1.1, showing the probability of measuring $|0\rangle$ or $|1\rangle$ for pure states. We adjusted our measurement statistics by multiplying by the inverse confusion matrix. The results in Table 4.4 show a significant reduction in errors. Using all results for $|\psi_1\rangle$ and $|\psi_2\rangle$, the relative errors for $H^{(0)}$ and $H^{(1)}$ are 0.2% and 0.5%, respectively. Residual errors in Pauli operator expectations suggest remaining systematic

errors, which shows that further error reduction is limited to techniques beyond increasing the measurement statistics.

	$ \psi_1(0)\rangle$	exact	$ \psi_2(0)\rangle$	exact
$\langle X \rangle$	-0.8119(46)	-0.8164	0.8152(27)	0.8164
$\langle Y \rangle$	0.2569(83)	0.2430	-0.2596(79)	-0.2430
$\langle Z \rangle$	0.5297(80)	0.5239	-0.5151(89)	-0.5239
$\langle H^{(0)} \rangle$	1.0100(65)	1.0069	-1.1751(60)	-1.1768
$\langle H^{(1)} \rangle$	-0.8283(28)	-0.8254	-0.8589(29)	-0.8653

Table 4.4 Results for eigenstate preparation with measurement error mitigation.

4.2.3 Discussion

Our analysis shows that the RA for the one-qubit Hamiltonian $H^{(0)}$ achieves a relative error of 0.08% for energy eigenvalues, outperforming the expectation values of the eigenvector prepared directly from $H^{(0)}$ after applying measurement error mitigation techniques. Despite the Hamiltonian only operating on one qubit, six two-qubit CNOT gates are used, so total errors, which include gate error, measurement error, and qubit decoherence, exceed 0.08%. This shows that the RA's resilience and design allow it to achieve precise energy results even without error mitigation and in the presence of substantial noise. Even with noise reducing the spectral weight of the target eigenstate, the signal remains discernible from the background with sufficient statistics.

For single-qubit benchmarks, RA with precise gate calibration can further reduce relative errors. The computational cost to determine energy eigenvalues with relative error ϵ scales as $O[(\log \epsilon)^2 / (p\epsilon)]$ [28], with p being the squared overlap of the initial state and the target eigenvector, contrasted with the scaling $O(1/\epsilon^2)$ for the expectations of the eigenvector prepared directly. This estimate $O(1/\epsilon^2)$ also serves as a lower bound for variational quantum eigensolvers, not including the additional cost of variational search.

The calculations of the Hellmann-Feynman theorem of $H^{(1)}$ using the RA show a relative error of 0.7%, an order of magnitude larger, due to the need for numerical derivatives of $E_n(\phi)$. However, the theorem accurately computes observables on a quantum device. To maintain the relative error ϵ in the operator expectation values, the energies of $H_{\text{obj}}(\phi)$ need to be computed with an error

tolerance of $O(\epsilon^2)$ for ϕ values around $O(\epsilon)$, resulting in a computational complexity scaling of $O[(\log \epsilon)^2 / (p\epsilon^2)]$.

4.2.3.1 Future Work

This Section demonstrated RA’s performance in both classical simulations for the Heisenberg model Hamiltonian and physical devices executions for a general one-qubit Hamiltonian. An interesting next step would be to test RA on Hamiltonians of more than one qubit. This would introduce systematic errors due to the Suzuki-Trotter decomposition needed for the time evolution operator [110, 106, 26]. However, an effective Hamiltonian can be defined that reproduces the Trotterized time evolution. Based on the results in Section 4.2.2, the eigenvalues of such an effective Hamiltonian can likely be determined with similar accuracy as in the one-qubit case, given a signal strong enough to rise above the random background.

Although the energy-scanning method has been successful in detecting the energy spectrum, it may encounter difficulties when applied to larger systems that require significantly more scans. One of the main scalability challenges of the RA is ensuring that the initial state has a sufficiently large overlap with the target eigenstate, which may be essential for increasing the success probability enough to discern the energy peaks from the background noise. Achieving this condition becomes especially difficult for larger systems of interest. Therefore, a possible next step, thoroughly investigated in Chapter 5, is to combine RA with a variational technique such as QAOA, using classical optimization to help prepare an initial state that ensures a sufficiently high success probability for the RA.

CHAPTER 5

VARIATIONAL RODEO ALGORITHM

This chapter expands on the Rodeo algorithm (RA) studied and evaluated in Chapter 4, by converting it into a variational technique, which we call the Variational Rodeo Algorithm (VRA). The objective is to address the main shortcoming of the RA, specifically that its success rate is contingent upon the overlap between the initial state and the target eigenstate, which is hard to control for many systems of interest. We test the algorithm by performing classical optimizations using emulated quantum circuits and also compare it with QAOA for the task of ground-state preparation. Our results, combined with those of Chapter 4, suggest that this method should be effective in preparing high-fidelity eigenstates on quantum computers.

5.1 Algorithm

Like other variational methods, the VRA uses a two-part method that alternates between executing a parameterized quantum circuit and classically optimizing the circuit parameters. The quantum circuit of VRA has two parts: a parameterized section followed by the RA circuit. Any parameterized circuit can be used for the first part, but in this chapter, we consider only the QAOA circuit because of its widespread recognition and because its resemblance to adiabatic evolution may give it an advantage in preparing ground-state eigenvectors.

The Hamiltonian of interest is labeled as the "object Hamiltonian", H_{obj} . Just like in the RA, a quantum register is divided into an object system, on which H_{obj} is allowed to act, and a set of ancilla qubits called the 'rodeo arena.' If mid-circuit measurements are allowed, only one ancilla qubit is needed for the rodeo arena. First, the QAOA algorithm with mixer Hamiltonian H_{mix} and problem Hamiltonian H_{obj} is applied to the object system (see Section 2.2.2.3); then the RA with the chosen energy parameter E is applied immediately thereafter. This circuit is executed repeatedly to determine the probability that each ancilla-qubit measurement is in the $|1\rangle$ state, which is the same as the "success probability" in Section 4.1.1. This probability is at a maximum when the final state of the object system after RA, $|\psi_F\rangle$, is an eigenstate of H_{obj} . Thus, to prepare eigenstates, VRA aims to maximize this probability, using one minus the probability as the cost function for classical

optimization.

5.1.1 Comparison to other Methods

Traditionally, the cost function for the classical optimization part of QAOA is the expectation value of the energy of the state stored in the quantum register with respect to H_{obj} , which can be measured from repeated executions of the QAOA circuit without requiring full tomography. Due to the variational principle, the energy expectation value of any state provides an upper bound on the energy of the ground state, and the state with the minimum energy from a sample of states can be used as an approximation of the true ground state[45]. In many but not all cases, this state approaches the true ground state as its energy with respect to H_{obj} tends to its minimum value. Although it often provides a good approximation for small systems (and is, in fact, universally capable of ground states of specific types of Hamiltonians[78]), QAOA was designed as a heuristic approach and excels at providing approximations for ground state energies rather than preparing eigenstates with high fidelity[80]. Its limitations become apparent when QAOA is applied to problems where the Hilbert space of H_{obj} is large and the number of parameters in the circuit is low[4, 87, 105]. Due to its inspiration from adiabatic evolution, QAOA can still be useful to converge to eigenstates for many problems due to the adiabatic theorem [15].

For simplicity, the mixer Hamiltonian H_{mix} in QAOA is often chosen so that its time evolution operation can be implemented with $O(n)$ quantum gates, but the time evolution of the problem Hamiltonian H_{obj} may be much more complicated, often requiring Trotter approximations and costly 2-qubit gate decompositions[117]. Thus, in order to be feasible to compute on NISQ-era devices, QAOA circuits for reasonably-sized problem Hamiltonians are often truncated to a small depth with fewer optimization parameters. Increasing this depth may lead to more accurate results, but would also quickly undermine the effectiveness of the algorithm on physical devices due to decoherence.

Rather than using QAOA alone to estimate the ground state energy, in this novel approach we use its output state as the initial state for the RA. This choice is based on the observation that the states produced by QAOA have significantly higher ground state overlap than randomly generated

states, even for large systems[87]. This makes it a good candidate as a precursor algorithm for RA, which should be capable of transforming its output state into one with even higher ground state overlap. Taking this idea one step further, the RA circuit can be appended to the QAOA circuit during optimization so that the cost function depends not on the energy but on the success probability of the RA, which is also easy to estimate from circuit executions.

As we reason in Section 5.2 based on the conclusions of Section 4.1.1, the success probability of the RA with the energy parameter E is directly proportional to the overlap of the initial state with the particular problem Hamiltonian eigenstate with the closest eigenvalue to E . In contrast, the relationship between the energy of a quantum state and its ground-state overlap is often unclear. Our results in Section 5.3 suggest that the state with the minimum energy reachable by a QAOA circuit can differ significantly from the state with the highest overlap. As problem size increases or QAOA steps decrease, the ground-state overlap of the minimum energy state may be no higher than that of a random state in the Hilbert space. This implies that the RA success probability should provide a more useful cost function for the use case of eigenstate preparation. By avoiding energy minimization, it is also straightforward to prepare excited states with VRA.

5.2 Mathematical Comparison of Cost Functions

In this section, we explore the effectiveness of three theoretical cost functions for classical optimization in parameterized quantum circuits. First, we investigate energy minimization, the most common approach used in VQE and QAOA methods. We then examine maximizing the overlap with the ground state, which, while ideal in theory, is impractical as it assumes prior knowledge of the system’s ground state, an objective typically sought by such algorithms. Lastly, we evaluate the cost function proposed in our method, which maximizes the success probability of the RA. This method requires only an approximate guess for the ground state energy and, as we demonstrate, converges to the maximization of the ground state overlap in the limit where the guessed energy E approaches the true ground state energy E_0 , and the number of RA cycles and the standard deviation σ of the random time parameters become large.

For this section, we will consider a general Hamiltonian H with eigenstates and eigenvalues

$|E_n\rangle$ and E_n where $n = \{0, 1, \dots, N - 1\}$ and $E_0 \leq E_1 \leq \dots \leq E_{N-1}$. Each cost function will be analyzed in the context of the objective of calculating the ground state energy E_0 . This can be done in the context of a standard gradient descent algorithm by examining the direction of the gradient vector with respect to the optimization parameters to understand the path that an optimization would take.

5.2.1 Energy Minimization

To analyze the energy minimization cost function, we start by decomposing the general state $|\psi\rangle$ into energy eigenstates $|E_n\rangle$:

$$|\psi\rangle = \sum_{n=0}^{N-1} c_n |E_n\rangle. \quad (5.1)$$

The energy expectation value for the normalized state is given by:

$$E_\psi = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} = \frac{\sum_{n=0}^{N-1} |c_n|^2 E_n}{\sum_{n=0}^{N-1} |c_n|^2}. \quad (5.2)$$

The complex phase of each c_n does not affect the energy expectation value. Thus, we can express c_n as $|c_n|e^{i\theta_n}$ and keep each $e^{i\theta_n}$ fixed for $n = 0, \dots, N - 1$. We now consider the squared absolute values $|c_n|^2$ as independent variables. The state $|\psi\rangle$ can be parameterized in terms of the vector $\{|c_0|^2, \dots, |c_{N-1}|^2\}$. Taking the partial derivative of E_ψ with respect to $|c_n|^2$ yields the following.

$$\frac{\partial E_\psi}{\partial |c_n|^2} = \frac{\sum_{m=0}^{N-1} |c_m|^2 (E_n - E_m)}{\left[\sum_{m=0}^{N-1} |c_m|^2\right]^2}. \quad (5.3)$$

A simple rescaling of the overall normalization of ψ does not change the energy expectation value, evident from:

$$\sum_{n=0}^{N-1} |c_n|^2 \frac{\partial E_\psi}{\partial |c_n|^2} = \frac{\sum_{n=0}^{N-1} \sum_{m=0}^{N-1} |c_n|^2 |c_m|^2 (E_n - E_m)}{\left[\sum_{m=0}^{N-1} |c_m|^2\right]^2} = 0. \quad (5.4)$$

Given this rescaling invariance, we normalize $|\psi\rangle$ such that:

$$\sum_{m=0}^{N-1} |c_m|^2 = 1. \quad (5.5)$$

Therefore,

$$\frac{\partial E_\psi}{\partial |c_n|^2} = E_n - E_\psi. \quad (5.6)$$

Following the steepest descent path involves moving in the direction of the infinitesimal vector $\{\Delta|c_0|^2, \dots, \Delta|c_{N-1}|^2\}$, where:

$$\Delta|c_n|^2 = -(E_n - E_\psi), \quad (5.7)$$

for $n = 0, \dots, N-1$. This approach shows that $|c_n|^2$ decreases for $E_n > E_\psi$ and increases for $E_n < E_\psi$. We note that this optimization path may be inefficient in a large vector space.

5.2.2 Ground State Overlap Maximization

To maximize overlap with the ground state $|E_0\rangle$, we define:

$$P_\psi^0 = \frac{\langle \psi | E_0 \rangle \langle E_0 | \psi \rangle}{\langle \psi | \psi \rangle} = \frac{|c_0|^2}{\sum_{m=0}^{N-1} |c_m|^2}. \quad (5.8)$$

Taking the partial derivative of P_ψ^0 with respect to $|c_n|^2$ gives:

$$\frac{\partial P_\psi^0}{\partial |c_n|^2} = \frac{\delta_{n,0} \sum_{m=0}^{N-1} |c_m|^2 - |c_0|^2}{\left[\sum_{m=0}^{N-1} |c_m|^2 \right]^2}. \quad (5.9)$$

Similarly as in Equation 5.4, a rescaling of the overall normalization of ψ does not change the overlap:

$$\sum_{n=0}^{N-1} |c_n|^2 \frac{\partial P_\psi^0}{\partial |c_n|^2} = \frac{\sum_{n=0}^{N-1} |c_n|^2 \delta_{n,0} \sum_{m=0}^{N-1} |c_m|^2 - \sum_{n=0}^{N-1} |c_n|^2 |c_0|^2}{\left[\sum_{m=0}^{N-1} |c_m|^2 \right]^2} = 0. \quad (5.10)$$

The state $|\psi\rangle$ can be normalized as in Equation 5.11.

$$\sum_{m=0}^{N-1} |c_m|^2 = 1, \quad (5.11)$$

Following the steepest ascent path involves moving in the direction of the infinitesimal vector $\{\Delta|c_0|^2, \dots, \Delta|c_{N-1}|^2\}$:

$$\Delta|c_n|^2 = \delta_{n,0} - |c_0|^2 = \delta_{n,0} - P_\psi^0, \quad (5.12)$$

for $n = 0, \dots, N-1$. When P_ψ^0 is small, the movement is primarily towards $|c_0|^2$ corresponding to the eigenvector of the ground state, facilitating an efficient search in a large vector space due to the low rank of the cost function.

5.2.3 Rodeo Algorithm Success Probability Maximization

To maximize the success probability of the rodeo algorithm for M cycles, with target energy E and standard deviation of time values σ , we first define the success probability P_ψ as follows:

$$P_\psi = \frac{\sum_{n=0}^{N-1} \left[\frac{1}{2} + \frac{1}{2} e^{-(E_n - E)^2 \sigma^2 / 2} \right]^M \langle \psi | E_n \rangle \langle E_n | \psi \rangle}{\langle \psi | \psi \rangle} \quad (5.13)$$

$$= \frac{\sum_{n=0}^{N-1} \left[\frac{1}{2} + \frac{1}{2} e^{-(E_n - E)^2 \sigma^2 / 2} \right]^M |c_n|^2}{\sum_{m=0}^{N-1} |c_m|^2}. \quad (5.14)$$

Taking the partial derivative of P_ψ with respect to $|c_n|^2$ gives:

$$\frac{\partial P_\psi}{\partial |c_n|^2} = \frac{\sum_{m=0}^{N-1} |c_m|^2 \left\{ \left[\frac{1}{2} + \frac{1}{2} e^{-(E_n - E)^2 \sigma^2 / 2} \right]^M - \left[1 + e^{-(E_m - E)^2 \sigma^2 / 2} \right]^M \right\}}{\left[\sum_{m=0}^{N-1} |c_m|^2 \right]^2}. \quad (5.15)$$

Since normalization of ψ does not change the success probability of RA, it follows that:

$$\sum_{n=0}^{N-1} |c_n|^2 \frac{\partial P_\psi}{\partial |c_n|^2} \quad (5.16)$$

$$= \frac{\sum_{n=0, m=0}^{N-1} |c_n|^2 |c_m|^2 \left\{ \left[\frac{1}{2} + \frac{1}{2} e^{-(E_n - E)^2 \sigma^2 / 2} \right]^M - \left[\frac{1}{2} + \frac{1}{2} e^{-(E_m - E)^2 \sigma^2 / 2} \right]^M \right\}}{\left[\sum_{m=0}^{N-1} |c_m|^2 \right]^2} \quad (5.17)$$

$$= 0. \quad (5.18)$$

Following the steepest ascent path, the direction is given by the infinitesimal vector with N components ($n = 0, 1, \dots, N - 1$) defined as follows:

$$\Delta|c_n|^2 = \left[\frac{1}{2} + \frac{1}{2} e^{-(E_n - E)^2 \sigma^2 / 2} \right]^M - P_\psi, \quad (5.19)$$

for $n = 0, \dots, N - 1$. When P_ψ is small, the movement is primarily towards the eigenvectors where $\left[\frac{1}{2} + \frac{1}{2} e^{-(E_n - E)^2 \sigma^2 / 2} \right]^M$ is significant. Increasing M narrows the energy window around E , leading to better search efficiency in a large vector space due to the effective low rank of the rodeo algorithm success probability. As M increases, the gradient tends to increase the magnitude of the component with the smallest value of $|E_n - E|$ and decrease the magnitude of all other components.

5.3 Simulation and Results

In this section, we investigate the convergence of VRA compared to other variational quantum algorithms by performing classical optimization based on the output of classically simulated quantum circuits. For each of the following methods, we consider a QAOA circuit with n qubits as the parameterized circuit for variational optimization. The objective Hamiltonian H_{obj} is defined by creating a random Hermitian matrix with dimensionality $2^n \times 2^n$. The Hermitian property is enforced by first generating a random triangular matrix of complex values, then summing it with its own conjugate transpose. For the mixer Hamiltonian H_{mix} , we use the so-called x -mixer defined as follows:

$$H_{\text{mix}} = \sum_{j=1}^n \mathbf{X}_j, \quad (5.20)$$

where \mathbf{X}_j represents an \mathbf{X} gate on the qubit at index j . The QAOA circuit is initialized by putting the quantum register in the ground state of H_{mix} , denoted $|\psi_0\rangle$. The circuit then consists of time evolution gates acting on the quantum register alternating over H_{obj} and H_{mix} with adjustable time parameters for each. The circuit diagram for this QAOA circuit is shown in figure 5.1. The number of time evolution gates, and therefore parameters, in the circuit is adjusted based on the number of qubits and the desired fidelity of state preparation.

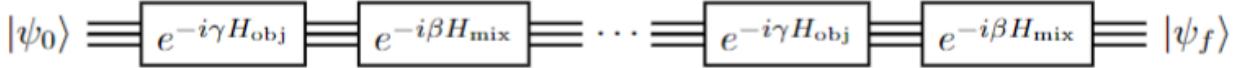


Figure 5.1 A circuit diagram of a QAOA circuit with objective Hamiltonian H_f , mixer Hamiltonian H_i , and tunable parameters γ and β . Each time evolution gate has its own adjustable parameter, typically restricted to be in the range $(0, 2\pi)$.

In each of the following demonstrations, the exact final state $|\psi_f\rangle$ of the QAOA circuit is found classically using matrix-vector multiplications so that no circuit executions are necessary. From this state vector, the energy can be calculated as $\langle\psi_f|H_{\text{obj}}|\psi_f\rangle$, and the modulus squared overlap with any particular eigenstate $|E_k\rangle$ of H_{obj} can be calculated as $|\langle E_k|\psi_f\rangle|^2$. If $|\psi_f\rangle$ is used as the initial state in a Rodeo algorithm circuit with N cycles and time values sampled from a Gaussian distribution with mean 0 and standard deviation σ , then the probability of success for the Rodeo algorithm is given by P_N in Equation 5.13, which we use instead of circuit executions or simulations.

The optimal parameters for each QAOA circuit are estimated using the classical BFGS optimization algorithm (see Section 2.2.3.3). The optimization is initialized with random parameters in the range $(0, 2\pi)$. To avoid local minima or barren plateaus, we repeat the optimization with different random parameters and select the one with the best value of the cost function as the best estimate of the global optimum. We choose this method because it is simple to implement and each individual optimization run can provide insight into the optimization landscape. Better estimates of global optima can also be obtained through different optimization strategies such as simulated annealing [112] and basin hopping [115] or QAOA-specific strategies that adjust the depth of the QAOA circuit[86] or use machine learning to guess better starting parameters[121].

In Section 5.3.1, we demonstrate the ability of VRA to prepare excited states of a 6 and 10 qubit system by adjusting the energy parameter E of the Rodeo algorithm. In Section 5.3.2, we compare VRA with traditional QAOA as well as a two-step optimization approach alternating between both cost functions to prepare the ground state of a 6-qubit system.

5.3.1 Simulation Method 1

First, we demonstrate the ability of VRA to prepare eigenstates other than the ground state, a feat that is not possible with techniques that use energy minimization. Since VRA aims to maximize the success probability of the Rodeo algorithm, the cost function to minimize is defined as $1 - P_N$. We used repeated BFGS optimizations with random initial parameters to estimate the optimal set of parameters, repeating the whole process for different values of the RA energy parameter E . The number of cycles is chosen as $N = 4$, and the standard deviation of the time values in the definition of P_N given in Equation 5.13 is set to be constant at $\sigma = 3$. We repeat these steps for two random Hamiltonians on 6 and 10 qubits, and we consider 12 parameters and 16 parameters for their QAOA circuits, respectively.

The relative overlap of the output state $|\psi_f\rangle$ of the QAOA circuit with each eigenstate of its corresponding Hamiltonian is visualized with the shaded rectangles in the graphs in Figure 5.2. In each case, $|\psi_f\rangle$ was dominated by its component with the eigenvector $|E_k\rangle$ with E_k closest to E . Many of the shaded squares are therefore too light to see, so for each value of E used in the VRA cost functions, a red dot is placed at the y coordinate corresponding to the eigenstate that had the highest overlap, which was the one with the eigenvalue closest to E in every case. These simulations suggest that VRA may be an effective method for the preparation of excited states on quantum computers.

5.3.2 Simulation Method 2

In this method, we compare VRA with pure QAOA as methods to prepare a ground-state eigenvector of a random 6-qubit Hamiltonian H_{obj} . Before performing optimizations, we directly solve for the eigenvectors $|E_k\rangle$ of H_{obj} and use the ground-state energy E_0 as the target energy parameter in Equation 5.13 for P_N along with a σ value of 10. We use the traditional VQE cost function (minimizing the energy of the output state $|\psi_f\rangle$) and the cost function $1 - P_N$ for VRA as before. The parameterized circuits for both cost functions were QAOA circuits with 20 parameters. As before, we used multiple BFGS optimizations to get better estimates of the global minima. In practice, one could discard the results of all but one optimization that ended with the best final

$|\langle \psi_f | E_k \rangle|^2$ vs. E full spectrum analysis for VRA

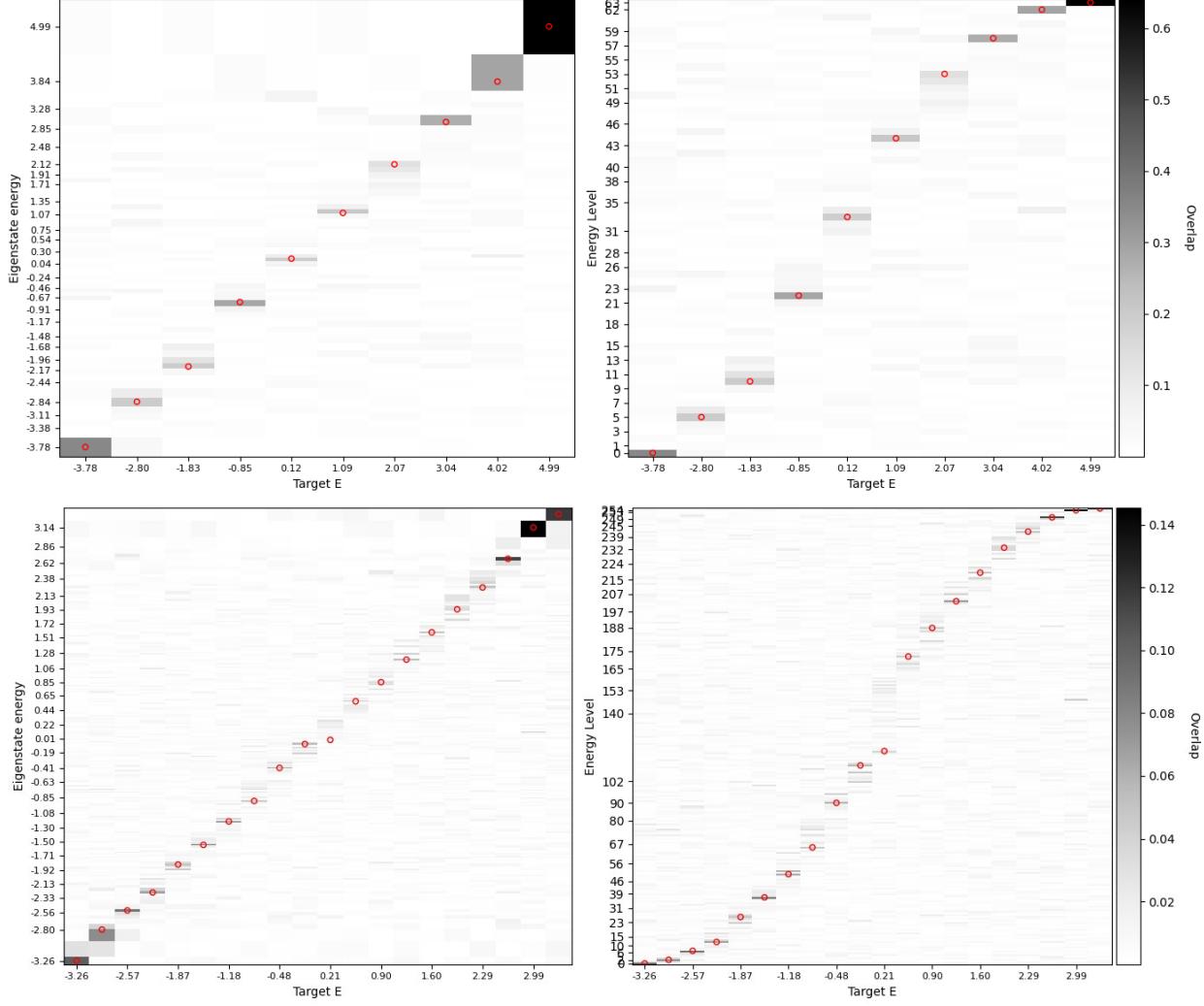


Figure 5.2 An analysis of the final states $|\psi_f\rangle$ of VRA simulations on random 6-qubit (top row) and 10-qubit (bottom row) systems, repeated for different values of E , increasing linearly along each x -axis. The y -axes represent the exact eigenstates ordered by their eigenvalues scaled by energy (left column) and level number (right column). The overlap of the final state of the QAOA circuit with each eigenvector is represented by the relative darkness of each shaded rectangle. Note that many rectangles are too light to be seen due to extremely low overlap. For each VRA optimization, a red circle is placed at the eigenstate with the highest overlap

value of the cost function, but here we present all 10 optimizations side-by-side to better illustrate the optimization landscapes. On the x axis is the iteration number of the BFGS optimization, and on the y axes there are several values of interest for $|\psi_f\rangle$ with the circuit parameters in that iteration. Specifically, we tracked energy $\langle\psi_f|H_{\text{obj}}|\psi_f\rangle$, ground state overlap squared $|\langle E_0|\psi_f\rangle|^2$, and RA success probability P_N . The results of each optimization are represented as the different colored lines in the plots in Figure 5.3.

These results show that the VRA cost function is comparable to energy minimization in reaching states with low energies but superior in preparing states with high overlap with the ground state. The difference is noticeable here primarily because 20 QAOA parameters are not sufficient to allow the QAOA circuit to prepare the ground state with perfect fidelity, so the energy minimization technique tends to settle in a mixture of low-lying energy states, while the VRA method reduces the overlap with all states other than the ground state roughly equally. If the depth of the QAOA circuit is increased, the variational principle takes effect and the state produced by energy minimization would theoretically converge to the ground state. However, QAOA is an NISQ-era algorithm intended to be used with a relatively small number of parameters to compensate for the fact that current quantum devices decohere rather quickly, so these results are closer to what should be typical when applying QAOA or VQE to any reasonably sized system. We note that the plots produced by VRA for ground state overlap and P_N are similar because, in this example, the RA parameters $E = E_0$ and $\sigma = 10$ make it so that the energy filter of the RA is smaller than the energy gap to the first excited state. The best optimizations for both cost functions resulted in final states with a RA success probability of about 85%, which makes it clear that the addition of RA to the QAOA circuit after optimization aids in the preparation of the ground state with high fidelity.

To further show the difference between these two cost functions, we combine them in a two-step optimization approach in which the final parameters of a 100 step QAOA optimization with energy minimization are used as initial parameters in a 100 step VRA optimization. This type of approach may be more practical because the VRA circuit depth is higher than that of the QAOA circuit alone. Therefore, it may save time to use energy minimization for the bulk of the optimization and then

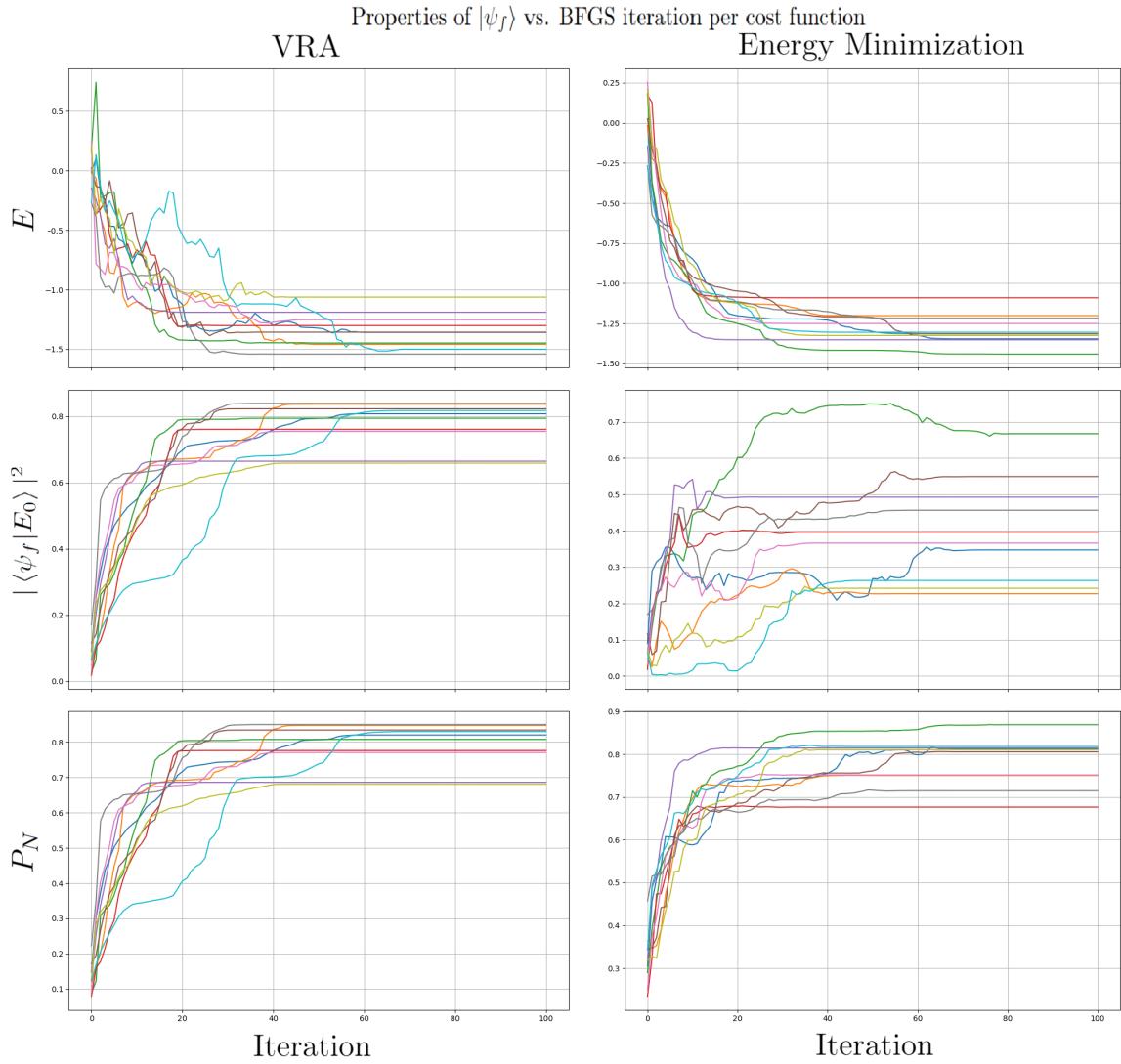


Figure 5.3 A comparison of the VRA (left) and VQE (right) cost functions for BFGS optimizations of a 20 parameter 6 qubit QAOA circuit. The shared x axes represent the BFGS iteration number from 1 to 100. The y axes in each row represent properties of the output state $|\psi_f\rangle$ of the QAOA circuit, namely energy (top), ground state overlap (middle), and RA success probability (bottom). Note the different scales on the y -axes. 10 random sets of 20 parameters each were generated and used as the initial circuit parameters in separate optimizations for both methods, with each set represented by a different color on the plots.

append the Rodeo algorithm for a few steps of fine-tuning. This approach effectively finds the closest local maximum of the RA success probability in the vicinity of a local energy minimum.

We use a slightly different example from before by choosing a different random 6 qubit Hamiltonian and reducing the number of QAOA parameters to 12. Figure 5.4 shows how the output state $|\psi_f\rangle$ produced by the QAOA circuit evolves throughout the optimizations. For circuits with few parameters like this one where the variational principle is not in effect, these two cost functions do not necessarily share the same local minima. Thus, changing the cost function from minimizing energy to maximizing the RA success probability results in an immediate correction to the output state of the QAOA circuit, giving it higher energy but also a higher RA success probability. This change also results in a larger overlap with the ground state because of its correlation with the probability of success of RA, which can be clearly seen by the similarity of the middle and bottom plots in the figure. In this case, with only 12 parameters, adding VRA to the optimization resulted in a much better overlap with the ground state, increasing the success probability of the best optimizations from roughly 20% to almost 40%, with more drastic improvements in the cases where VQE resulted in much lower ground state fidelity. For larger problems as the optimization landscape gets more complicated, we would expect VQE results to have a lower ground state overlap more often, so using VRA for fine-tuning may be critical for state preparation with any reasonable fidelity.

5.4 Future Work

This section will describe future avenues of research to improve and test this novel VRA technique, as well as some intermediate results that we have obtained in that direction.

5.4.1 Comparison of Optimization Landscapes

The results in Section 5.3.2 suggest that the cost functions in QAOA and VRA have different local minima and barren plateaus. A better understanding of these differences can provide insight into when it is worthwhile to use VRA over QAOA. Barren plateaus, which are subspaces in the optimization landscape where the gradient vanishes, typically arise in quantum optimization problems whenever there are more than a few qubits or parameters in the circuit[77]. This makes them inherently difficult to visualize in two- or three-dimensional plots. One possible method to

Properties of $|\psi_f\rangle$ vs. two-step BFGS iteration

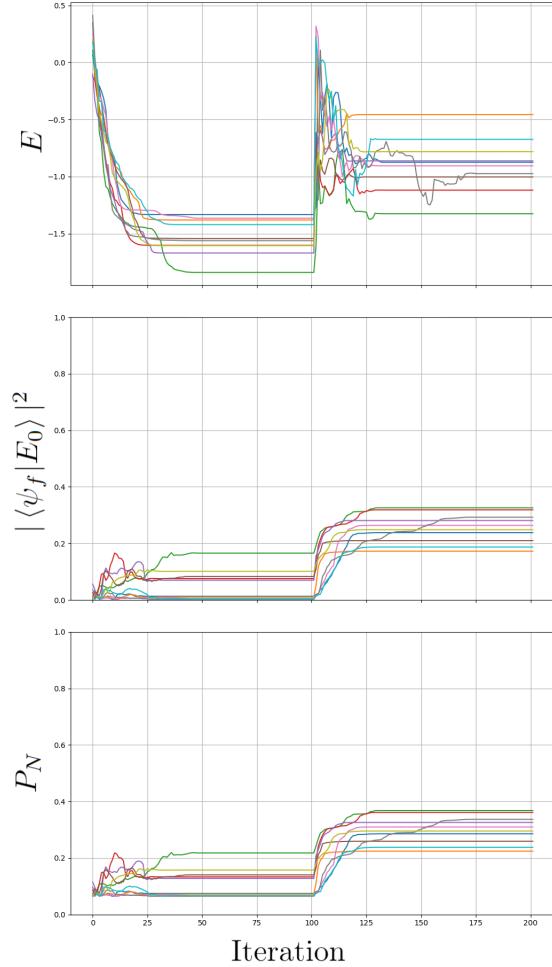


Figure 5.4 Results for a two-step BFGS optimization of a 12 parameter 6 qubit QAOA circuit with 100 steps of energy minimization followed by 100 steps of VRA. The axes are the same as in Figure 5.3 but with 200 total iterations now on the x axis. 10 random sets of 12 parameters each were generated and used as the initial circuit parameters in separate optimizations, with each set represented by a different color on the plots.

test the density of local minima and barren plateaus is to perform many optimization attempts at different starting sets of parameters and see how frequently they get "stuck" at a parameter state with zero gradient. However, this would not allow for differentiation between local minima and barren plateaus. It is also possible to examine the behavior of the cost functions when only a small number of parameters are allowed to vary, which may lead to insights into the optimization landscapes.

5.4.2 Application to Interesting Problems

This chapter mainly focused on solving for eigenstates of random Hamiltonians in order to avoid the extra complexity of identifying and justifying the use of a Hamiltonian with implications for solving real problems of interest. However, many such problems exist that may show the practical use cases of VRA. More research in this area may also point towards certain problems where VRA may theoretically outperform classical methods on supercomputers. Many interesting Hamiltonians are available for benchmarking purposes in online databases. For example, the HamLib library[101] contains many interesting Hamiltonians in binary-variable optimization, condensed matter physics, and quantum chemistry problems, which are already mapped into qubit representations. Some of these problems may benefit from VRA's enhanced fidelity in preparing eigenstates. It may be especially interesting to apply the VRA to nuclear structure and reaction Hamiltonians, where better solutions may lead to advances in areas such as nuclear medicine and energy. These Hamiltonians typically have more terms than similar Hamiltonians in quantum chemistry and therefore require larger circuit depth to replicate time evolution operations, which is why more focus is currently placed on applying NISQ-era algorithms like QAOA to quantum chemistry applications. However, there may be theoretically interesting small nuclear systems that are good candidates for more accurate approximate solutions on quantum computers.

CHAPTER 6

SUMMARY

6.1 Adiabatic Evolution with Optimal Control

In Chapter 3, a robust method was presented to execute arbitrary sequences of unitary transformations, focusing on the adiabatic evolution of a two-spin system using Near-term Intermediate Scale Quantum (NISQ) devices. A two-qubit processor model, comprised of two capacitively coupled superconducting transmons, was employed to emulate adiabatic evolution with custom two-qubit gates. The output was then compared with the digital quantum simulations performed on IBMQ systems.

High-fidelity custom gates for short-time propagators could be realized with control pulses of varying duration, from 2500 to 120 ns. The lower limit was determined by the validity of the model employed and the specifications of the standard waveform generators. When the implementation time of short-time propagators was similar, the state fidelities of the emulated output and IBMQ were comparable, suggesting that the two-qubit processor model offered a realistic representation of quantum hardware. As the duration of control pulses and the implementation time of adiabatic evolution decreased, the loss of coherence due to the interaction between the quantum computer and its environment was significantly reduced. The fidelity achieved with custom gates exceeded that obtained through elementary gates, reaching fidelity of up to 95% for control pulses with a length of $\tau = 120$ ns.

6.2 Rodeo Algorithm

In Chapter 4, we discussed the Rodeo Algorithm for generating quantum eigenstates and analyzing spectral characteristics. In summary, the Rodeo Algorithm uses a tunable energy filter and stochastic methods to prepare eigenstates of a quantum Hamiltonian or observable with exponential efficiency and short gate depth, surpassing the speeds of phase estimation and adiabatic evolution. The latter part of the chapter showcased the RA's performance through a simulation of a basic Heisenberg model and an implementation of a general one-qubit Hamiltonian on an actual device. The findings highlighted the potential of the RA in preparing eigenstates with high accuracy, which

subsequently inspired the creation of the Variational Rodeo Algorithm discussed in the following chapter.

6.3 Variational Rodeo Algorithm

In Chapter 5, we introduce the Variational Rodeo Algorithm (VRA) as a variational extension of the Rodeo Algorithm (RA) to improve eigenstate preparation by overcoming the RA’s dependency on the overlap between the initial and target eigenstates. We evaluated VRA using classical optimizations of emulated quantum circuits and compared its performance with the Quantum Approximate Optimization Algorithm (QAOA) for ground-state preparation. Our findings indicate that VRA effectively prepares high-fidelity eigenstates on quantum computers.

Unlike QAOA, which relies on energy minimization and often struggles with high-fidelity eigenstate preparation, the VRA maximizes the success probability of the RA, offering a more effective approach, particularly for larger systems or those with fewer optimization parameters. Simulations demonstrated the ability of VRA to readily prepare excited states by adjusting the RA energy parameter. Using QAOA’s output as the initial state for the RA improved ground-state overlap, even when QAOA alone struggled with fidelity.

Overall, VRA presents a promising method for quantum eigenstate preparation, overcoming limitations in other algorithms like QAOA. Future research will focus on refining VRA’s optimization landscapes and applying it to quantum chemistry and nuclear physics problems to demonstrate its potential in solving complex quantum systems.

BIBLIOGRAPHY

- [1] Gert Aarts and Ion-Olimpiu Stamatescu. Stochastic quantization at finite chemical potential. *Journal of High Energy Physics*, 2008(09):018–018, September 2008.
- [2] Nobel Prize Outreach AB. The nobel prize in physics 2022, 2024. Accessed: 2024-07-14.
- [3] Dorit Aharonov. A simple proof that toffoli and hadamard are quantum universal, 2003.
- [4] V. Akshay, H. Philathong, M.E.S. Morales, and J.D. Biamonte. Reachability deficits in quantum approximate optimization. *Physical Review Letters*, 124(9), March 2020.
- [5] T. Albash and D. A. Lidar. Adiabatic quantum computation. *Rev. Mod. Phys.*, 90:015002, Jan 2018.
- [6] A. Anand, P. Schleich, S. Alperin-Lea, P. W. K. Jensen, S. Sim, M. Díaz-Tinoco, J. S. Kottmann, M. Degroote, A. F. Izmaylov, and A. Aspuru-Guzik. A quantum computing view on unitary coupled cluster theory. *Chemical Society Reviews*, 51:1659–1684, 2022.
- [7] Md Sajid Anis et al. Qiskit: An open-source framework for quantum computing, 2021.
- [8] Aristotle. *Physics*. Internet Classics Archive, 350 B.C. Translated by R.P. Hardie and R.K. Gaye.
- [9] H. Badem, A. Baştürk, A. Çalışkan, and M. E. Yüksel. A new hybrid optimization method combining artificial bee colony and limited-memory bfgs algorithms for efficient numerical optimization. *Applied Soft Computing*, 70:826–844, 2018.
- [10] Z. Bai and J. W. Silverstein. Spectral analysis of large dimensional random matrices. *Springer Series in Statistics*, 2010.
- [11] Leonardo Banchi and Gavin E. Crooks. Measuring analytic gradients of general quantum evolution with the stochastic parameter shift rule. *Quantum*, 5:386, January 2021.
- [12] Barenco, A. and Bennett, C. H. and Cleve, R. and DiVincenzo, D. P. and Margolus, N. and Shor, P. and Sleator, T. and Smolin, J. A. and Weinfurter, H. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457, Nov 1995.
- [13] Rodney J Baxter. One-dimensional anisotropic heisenberg chain. *Annals of Physics*, 70(2):323–337, 1972.
- [14] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. Zibrov, M. Endres, M. Greiner, V. Vuletić, and M. Lukin. Probing many-body dynamics on a 51-atom quantum simulator. *Nature*, 551:579–584, 2017.

- [15] Lennart Binkowski, Gereon Koßmann, Timo Ziegler, and René Schwonnek. Elementary proof of qaoa convergence. *New Journal of Physics*, 26(7):073001, July 2024.
- [16] R. Blatt, H. Häffner, C. F. Roos, C. Becher, and F. Schmidt-Kaler. Ion trap quantum computing with ca+ ions. *Quantum Information Processing*, 3:61–73, 2004.
- [17] S. Boixo, E. Knill, and R. D. Somma. Eigenpath traversal by phase randomization. *Quantum Inf. Comput.*, 9(9&10):833–855, 2009.
- [18] S. Boixo and R. D. Somma. Necessary condition for the quantum adiabatic approximation. *Phys. Rev. A*, 81:032308, Mar 2010.
- [19] M. Born and V. Fock. Beweis des Adiabatensatzes. *Zeitschrift für Physik*, 51:165, March 1928.
- [20] Joel M Bowman, Kurt Christoffel, and Frank Tobin. Application of scf-si theory to vibrational motion in polyatomic molecules. *Journal of Physical Chemistry*, 83(8):905–912, 1979.
- [21] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation, 2002.
- [22] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16:1190–1208, 1995.
- [23] Yudong Cao, Jonathan Romero, Jonathan P Olson, Matthias Degroote, Peter D Johnson, Mária Kieferová, Ian D Kivlichan, Tim Menke, Borja Peropadre, Nicolas PD Sawaya, et al. Quantum chemistry in the age of quantum computing. *Chemical reviews*, 119(19):10856–10915, 2019.
- [24] Joseph Carlson, David J. Dean, Morten Hjorth-Jensen, David Kaplan, John Preskill, Kenneth Roche, Martin J. Savage, and Matthias Troyer. Quantum computing for theoretical nuclear physics, a white paper prepared for the u.s. department of energy, office of science, office of nuclear physics, 1 2018.
- [25] Hai-Ping Cheng, Erik Deumens, James K. Freericks, Chenglong Li, and Beverly A. Sanders. Application of quantum computing to biochemical systems: A look to the future. *Frontiers in Chemistry*, 8, 2020.
- [26] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Physical Review X*, 11(1), February 2021.
- [27] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Inf. Comput.*, 12(11-12):901–924, 2012.
- [28] Kenneth Choi, Dean Lee, Joey Bonitati, Zhengrong Qian, and Jacob Watkins. Rodeo

algorithm for quantum computing. *Phys. Rev. Lett.*, 127(4):040505, 2021.

- [29] Kenneth Choi, Dean Lee, Joey Bonitati, Zhengrong Qian, and Jacob Watkins. Rodeo algorithm for quantum computing. *Physical Review Letters*, 127(4), July 2021.
- [30] J. Cox, C. Gatringer, K. Holland, B. Scarlet, and U.-J. Wiese. Meron-cluster solution of fermion and other sign problems. *Nuclear Physics B - Proceedings Supplements*, 83–84:777–791, April 2000.
- [31] A.D. Córcoles, Maika Takita, Ken Inoue, Scott Lekuch, Zlatko K. Minev, Jerry M. Chow, and Jay M. Gambetta. Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits. *Physical Review Letters*, 127(10), August 2021.
- [32] K.A. Dibner and C.E. Snow. *Science Literacy: Concepts, Contexts, and Consequences*. National Academies Press, 2016.
- [33] Jens Eisert, Martin Wilkens, and Maciej Lewenstein. Quantum games and quantum strategies. *Physical Review Letters*, 83(15):3077–3080, October 1999.
- [34] E. Epelbaum, H.-W. Hammer, and Ulf-G. Meißner. Modern theory of nuclear forces. *Reviews of Modern Physics*, 81(4):1773–1825, December 2009.
- [35] Jonny Evans. Example: $\text{Su}(2)$ to $\text{so}(3)$, 2020. Licensed under CC-BY-SA 4.0.
- [36] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution, 2000.
- [37] R. P. Feynman. Forces in molecules. *Phys. Rev.*, 56:340–343, Aug 1939.
- [38] R. P. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21:467, 1982.
- [39] E. Fradkin. *Field theories of condensed matter physics*. Cambridge University Press, 2013.
- [40] D. Frame, R. He, I. C. F. Ipsen, D. Lee, D. Lee, and E. Rrapaj. Eigenvector continuation with subspace learning. *Physical Review Letters*, 121, 2018.
- [41] F. Fucito, E. Marinari, G. Parisi, and C. Rebbi. A Proposal for Monte Carlo Simulations of Fermionic Systems. *Nucl. Phys. B*, 180:369, 1981.
- [42] Yimin Ge, Jordi Tura, and J. Ignacio Cirac. Faster ground state preparation and high-precision ground energy estimation with fewer qubits, 2018.
- [43] H. A. Gersch and G. C. Knollman. Quantum cell model for bosons. *Phys. Rev.*, 129:959–967, Jan 1963.

- [44] C. Granade, C. Ferrie, and S. Flammia. Practical adaptive quantum tomography. *New Journal of Physics*, 19:113017, 2017.
- [45] D.J. Griffiths. *Introduction to Quantum Mechanics*. Cambridge University Press, 2017.
- [46] Lov K. Grover. A fast quantum mechanical algorithm for database search, 1996.
- [47] Gian Giacomo Guerreschi and Mikhail Smelyanskiy. Practical optimization for hybrid quantum-classical algorithms, 2017.
- [48] Erik J. Gustafson. Projective cooling for the transverse Ising model. *Phys. Rev. D*, 101(7):071504(R), 5, 2020.
- [49] Stuart Hadfield, Zhihui Wang, Bryan O’Gorman, Eleanor G. Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, February 2019.
- [50] G Hagen, T Papenbrock, M Hjorth-Jensen, and D J Dean. Coupled-cluster computations of atomic nuclei. *Reports on Progress in Physics*, 77(9):096302, September 2014.
- [51] William W. Hager. *Applied Numerical Linear Algebra*. SIAM, 2021.
- [52] B.C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Graduate Texts in Mathematics. Springer, 2003.
- [53] W. Heisenberg. Zur theorie des ferromagnetismus. *Zeitschrift Fr Physik*, 49:619–636, 1928.
- [54] Eric T. Holland, Kyle A. Wendt, Konstantinos Kravvaris, Xian Wu, W. Erich Ormand, Jonathan L DuBois, Sofia Quaglioni, and Francesco Pederiva. Optimal control for the quantum simulation of nuclear dynamics. *Phys. Rev. A*, 101:062307, Jun 2020.
- [55] David Hume. *An Enquiry Concerning Human Understanding*. Project Gutenberg, 2006. eBook #9662, Project Gutenberg.
- [56] IBM Quantum. <https://quantum-computing.ibm.com>, 2021.
- [57] M. A. H. Ibrahim, M. Mamat, and W. J. Leong. The hybrid bfgs-cg method in solving unconstrained optimization problems. *Abstract and Applied Analysis*, 2014:1–6, 2014.
- [58] J. R. Johansson, P. D. Nation, and F. Nori. QuTiP: An open-source Python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.*, 183(8):1760, 2012.
- [59] J. R. Johansson, P. D. Nation, and F. Nori. QuTiP 2: A Python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.*, 184(4):1234, 2013.

- [60] Tyler Jones, Kaiah Steven, Xavier Poncini, Matthew Rose, and Arkady Fedorov. Approximations in transmon simulation. *arXiv e-prints*, February 2021.
- [61] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, September 2017.
- [62] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [63] S. Kirkpatrick, C. D. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [64] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem, 1995.
- [65] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem, 1995.
- [66] A.Y. Kitaev, A. Shen, and M.N. Vyalyi. *Classical and Quantum Computation*. Graduate studies in mathematics. American Mathematical Society, 2002.
- [67] Jens Koch, Terri M. Yu, Jay Gambetta, A. A. Houck, D. I. Schuster, J. Majer, Alexandre Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Phys. Rev. A*, 76:042319, Oct 2007.
- [68] A. F. Kockum and F. Nori. Quantum bits with josephson junctions. *Fundamentals and Frontiers of the Josephson Effect*, pages 703–741, 2019.
- [69] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver. A quantum engineer’s guide to superconducting qubits. *Appl. Phys. Rev.*, 6(2):021318, 2019.
- [70] Kurt Langfeld and Biagio Lucini. Form the density-of-states method to finite density quantum field theory, 2016.
- [71] Dean Lee, Joey Bonitati, Gabriel Given, Caleb Hicks, Ning Li, Bing-Nan Lu, Abudit Rai, Avik Sarkar, and Jacob Watkins. Projected cooling algorithm for quantum computation. *Physics Letters B*, 807:135536, 2020.
- [72] Dean Lee, Buğra Borasoy, and Thomas Schaefer. Nuclear lattice simulations with chiral effective field theory. *Physical Review C*, 70(1), July 2004.
- [73] S. Lloyd. Universal Quantum Simulators. *Science*, 273(5278):1073, 1996.
- [74] Sirui Lu, Mari Carmen Bañuls, and J. Ignacio Cirac. Algorithms for quantum simulation at finite energies. *PRX Quantum*, 2(2), May 2021.
- [75] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch. Variational quantum algorithms

for nonlinear problems. *Physical Review A*, 101, 2020.

- [76] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18:023023, 2016.
- [77] 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), November 2018.
- [78] M. E. S. Morales, J. D. Biamonte, and Z. Zimborás. On the universality of the quantum approximate optimization algorithm. *Quantum Information Processing*, 19(9), August 2020.
- [79] F. Motzoi, J. M. Gambetta, P. Rebentrost, and F. K. Wilhelm. Simple pulses for elimination of leakage in weakly nonlinear qubits. *Phys. Rev. Lett.*, 103:110501, Sep 2009.
- [80] Charles Moussa, Henri Calandra, and Vedran Dunjko. To quantum or not to quantum: towards algorithm selection in near-term quantum optimization. *Quantum Science and Technology*, 5(4):044009, October 2020.
- [81] N. Khaneja and T. Reiss and C. Kehlet and T. Schulte-Herbrüggen and S. J. Glaser. Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *J. Magn. Reson.*, 172(2):296, 2005.
- [82] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, 01 1965.
- [83] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- [84] NIST. 2022 codata value: Newtonian constant of gravitation. The NIST Reference on Constants, Units, and Uncertainty, May 2024. <https://physics.nist.gov/cgi-bin/cuu/Value?bg>.
- [85] A. Nogga, P. Navrátil, B. R. Barrett, and J. P. Vary. Spectra and binding energy predictions of chiral interactions for
$$\text{li} \text{mprescripts} \text{mathvariant="normal"}$$
$$\text{mrow} \text{mn} 7 \text{mrow} \text{mmultiscripts} \text{math}$$
. *Physical Review C*, 73(6), June 2006.
- [86] Y. Pan, Y. Tong, and Y. Yang. Automatic depth optimization for a quantum approximate optimization algorithm. *Physical Review A*, 105, 2022.
- [87] Elijah Pelofske, Andreas Bärtschi, and Stephan Eidenbenz. *Quantum Annealing vs. QAOA: 127 Qubit Higher-Order Ising Problems on NISQ Computers*, page 240–258. Springer

Nature Switzerland, 2023.

- [88] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O’Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1), July 2014.
- [89] S. C. Pieper and R. B. Wiringa. Quantum monte carlo calculations of light nuclei. *Annual Review of Nuclear and Particle Science*, 51:53–90, 2001.
- [90] Poulin, D. and Qarry, A. and Somma, R. and Verstraete, F. Quantum Simulation of Time-Dependent Hamiltonians and the Convenient Illusion of Hilbert Space. *Phys. Rev. Lett.*, 106:170501, Apr 2011.
- [91] W. Poundstone. *Prisoner’s Dilemma*. Doubleday, 1992.
- [92] J. Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.
- [93] A. Pérez-Obiol, A. M. Romero, J. Menéndez, A. Rios, A. García-Sáez, and B. Juliá-Díaz. Nuclear shell-model simulation in digital quantum computers. *Scientific Reports*, 13(1), July 2023.
- [94] Zhengrong Qian, Jacob Watkins, Gabriel Given, Joey Bonitati, Kenneth Choi, and Dean Lee. Demonstration of the rodeo algorithm on a quantum computer, 2024.
- [95] Robert Raussendorf, Daniel E. Browne, and Hans J. Briegel. Measurement-based quantum computation on cluster states. *Physical Review A*, 68(2), August 2003.
- [96] A. Roggero, C. Gu, A. Baroni, and T. Papenbrock. Preparation of excited states for nuclear dynamics on a quantum computer. *Phys. Rev. C*, 102:064624, Dec 2020.
- [97] Y. Saad. *Iterative methods for sparse linear systems*. Philadelphia, PA, 2003.
- [98] M. Sadiku, M. Tembely, and S. Musa. quantum computing: a primer. *International Journal of Advanced Research in Computer Science and Software Engineering*, 7:129, 2017.
- [99] Raffaele Santagati, Alan Aspuru-Guzik, Ryan Babbush, Matthias Degroote, Leticia González, Elica Kyoseva, Nikolaj Moll, Markus Oppel, Robert M. Parrish, Nicholas C. Rubin, Michael Streif, Christofer S. Tautermann, Horst Weiss, Nathan Wiebe, and Clemens Utschig-Utschig. Drug design on quantum computers. *Nature Physics*, 20(4):549–557, March 2024.
- [100] A. J. Sarty et al. Measurement of the reaction He-3 (gamma, pp) n and its relation to three-body forces. *Phys. Rev. C*, 47:459–467, 1993.
- [101] Nicolas PD Sawaya, Daniel Marti-Dafcik, Yang Ho, Daniel P Tabor, David E Bernal Neira,

Alicia B Magann, Shavindra Premaratne, Pradeep Dubey, Anne Matsuura, Nathan Bishop, Wibe A de Jong, Simon Benjamin, Ojas D Parekh, Norm Tubman, Katherine Klymko, and Daan Camps. Hamlib: A library of hamiltonians for benchmarking quantum algorithms and hardware, 2024.

- [102] R. Scalettar, G. G. Batrouni, A. P. Kampf, and G. T. Zimányi. Simultaneous diagonal and off-diagonal order in the bose-hubbard hamiltonian. *Physical Review B*, 51:8467–8480, 1995.
- [103] Vivek V. Shende, Igor L. Markov, and Stephen S. Bullock. Minimal universal two-qubit controlled-not-based circuits. *Phys. Rev. A*, 69:062321, Jun 2004.
- [104] Smite-Meister. Image from wikipedia commons. <https://commons.wikimedia.org/w/index.php?curid=5829358>, 2024. CC BY-SA 3.0.
- [105] Michael Streif and Martin Leib. Forbidden subspaces for level-1 quantum approximate optimization algorithm and instantaneous quantum polynomial circuits. *Physical Review A*, 102(4), October 2020.
- [106] Masuo Suzuki. Generalized Trotter’s formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Comm. Math. Phys.*, 51(2):183–190, 1976.
- [107] Yohichi Suzuki, Shumpei Uno, Rudy Raymond, Tomoki Tanaka, Tamiya Onodera, and Naoki Yamamoto. Amplitude estimation without phase estimation. *Quantum Information Processing*, 19(2), January 2020.
- [108] A. Szabo and N.S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Books on Chemistry. Dover Publications, 1996.
- [109] Andreas Trabesinger. Quantum simulation. *Nature Phys.*, 8(4):263, 2012.
- [110] H. F. Trotter. On the product of semi-groups of operators. *Proc. Amer. Math. Soc.*, 10:545–551, 1959.
- [111] Farrokh Vatan and Colin Williams. Optimal quantum circuits for general two-qubit gates. *Phys. Rev. A*, 69:032315, Mar 2004.
- [112] C. Venkateswaran, M. Ramachandran, R. Kurinjimalar, P. Vidhya, and G. Mathivanan. Application of simulated annealing in various field. *Materials and Its Characterization*, 1:01–08, 2022.
- [113] Ruben Verresen. Everything is a quantum ising model, 2023.
- [114] G. Vidal and C. M. Dawson. Universal quantum circuit for two-qubit transformations with

three controlled-not gates. *Phys. Rev. A*, 69:010301, Jan 2004.

- [115] David J. Wales and Jonathan P. K. Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.
- [116] C. P. Williams. Quantum gates. *Texts in Computer Science*, pages 51–122, 2011.
- [117] Dennis Willsch, Madita Willsch, Fengping Jin, Kristel Michielsen, and Hans De Raedt. Gpu-accelerated simulations of quantum annealing and the quantum approximate optimization algorithm. *Computer Physics Communications*, 278:108411, September 2022.
- [118] Xian Wu, S. L. Tomarken, N. Anders Petersson, L. A. Martinez, Yaniv J. Rosen, and Jonathan L. DuBois. High-fidelity software-defined quantum logic on a superconducting qudit. *Phys. Rev. Lett.*, 125:170502, Oct 2020.
- [119] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt. quantum annealing for industry applications: introduction and review. *Reports on Progress in Physics*, 85:104001, 2022.
- [120] Saadia Zahidi. The global risks report 2024, 2024. ISBN: 978-2-940631-64-3.
- [121] R. Zhao, T. Cheng, R. Wang, X. Fan, and H. Ma. Artificial intelligence warm-start approach: optimizing the generalization capability of qaoa in complex energy landscapes. *New Journal of Physics*, 26:053016, 2024.