Early Fault-Tolerant Quantum Algorithms for Open Quantum Systems

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
by Research
in
Computational Natural Sciences

by

Kushagra Garg
2019113020
kushagra.garg@research.iiit.ac.in

Center for Quantum Science and Technology (CQST)
&
Center for Computational Natural Sciences and Bioinformatics (CCNSB)
, International Institute of Information Technology (IIIT)
Hyderabad - 500 032, INDIA
September 2025

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Early Fault-Tolerant Quantum Algorithms" by Kushagra Garg, has been carried out under my supervision and has not been submitted elsewhere for a degree.

_____ _____

Date Co-Adviser: Prof. Shantanav Chakraborty

_____ _____

Date Co-Adviser: Prof. Subhadip Mitra

To my family.

Abstract

Quantum computing stands at the threshold of practical applications, although large-scale, fault-tolerant quantum computers remain distant. Indeed, despite practical quantum computers having been perpetually "ten years away" for the past ten years, incremental yet substantial advances continue to fuel optimism. Remarkable progress over the past decade has led to increasingly robust, larger-scale, and better-connected quantum hardware. Notably, recent years have witnessed several demonstrations of quantum computing devices whose physical error rates lie below critical thresholds required by certain error-correction schemes, signaling a significant transitional phase in quantum computing technology. Nevertheless, fully fault-tolerant quantum computers capable of executing arbitrarily large quantum circuits with an unrestricted number of qubits are still far off. Instead, near-term quantum hardware will be limited to simpler, shallow-depth circuits operating on a restricted number of qubits. Such early fault-tolerant quantum devices cannot accommodate advanced quantum subroutines like amplitude amplification, block-encoding, or Quantum Singular Value Transformation (QSVT), which inherently rely on deep quantum circuits, multi-controlled gates, and extensive ancillary qubit resources. Consequently, it becomes imperative to rethink quantum algorithm design, focusing specifically on algorithms that operate within these restricted computational frameworks.

Motivated by this challenge, this thesis develops quantum algorithms explicitly tailored for early fault-tolerant quantum computing devices. We introduce a randomized quantum algorithm for efficiently simulating quantum collision models which provide a versatile theoretical framework for modeling open quantum system dynamics. Quantum collision models describe physical processes in which the system undergoes successive interactions with a sequence of environmental degrees of freedom. Our algorithm employs a novel approach that composes multiple controlled short-time evolutions of simple Hamiltonians, interspersed with operations that trace out the environmental registers. Crucially, our method requires neither specialized quantum subroutines such as block-encodings nor extensive ancillary qubit resources, thus aligning closely with the limitations of near-term quantum devices.

Moreover, by exploiting the fundamental correspondence between Lindbladian evolution and Markovian collision models, we present an end-to-end quantum algorithm for simulating Lindbladian dynamics tailored specifically for near-term quantum hardware. For a quantum system consisting of n qubits, we provide a detailed analysis of the circuit depth required to estimate expectation values of observables with respect to the state evolved under open quantum dynamics for evolution time t. Our algorithm performs this simulation using at most $n+2$ qubits, achieving a circuit depth that scales polynomially with the number of qubits, desired precision, and evolution time. Additionally, we offer a detailed comparison of our algorithm with existing approaches, highlighting advantages in terms of required resources and suitability for near-term quantum hardware. We further generalize our framework beyond the memoryless scenario, developing an efficient method to simulate memory-retaining collision models, where successive environmental interactions give rise to non-Markovian dynamics. This extension allows us to model more intricate and realistic open-system behaviors that inherently exhibit memory effects.

By providing novel quantum algorithms that robustly simulate both Markovian and non-Markovian quantum dynamics within early fault-tolerant quantum computing platforms, this thesis significantly expands the scope of feasible quantum simulations. Collectively, these advances offer a deeper understanding of the computational capabilities and inherent limitations of imminent quantum technologies, laying an essential foundation for further theoretical and practical developments in quantum computing.

Acknowledgment

I want to start by thanking Prof. Shantanav Chakrabarty, who introduced me to the field of quantum algorithms. I want to thank him for providing guidance and inspiration during the entire period of my Master's program. Whether it was to talk about research or just to get some perspective on life, he was always available and willing to help. I've learned a lot from those conversations. I'm also really grateful to Prof. Subhadip Mitra for providing me the freedom to explore, steady support, and space to figure things out on my own, and most importantly for believing in me.

I especially want to thank Zeeshan Ahmed, one of my greatest friends, for all the unforgettable memories. He is the best person to work with. I have always looked forward to insightful discussions with him on almost anything. I want to thank him for listening to my ideas and taking part in realizing them. Without him, this work would not have been possible. I also want to thank all my lab-mates and friends—Arjo, Shreyas, Rutvij, Anvay, Shrinidhi, Aaryaman, Shreeyas, Sehreen, and Soumyabrata. I also want to thank my seniors—Aditya, Anurudh, and Mehati—who provided me guidance at multiple stages throughout my degree. Finally, I want to thank Gunjan for all the love and support.

Contents

List of Theorems

List of Algorithms

List of Figures

Chapter 1

Introduction

At the dawn of the 20$^{\text{th}}$ century, David Hilbert and Wilhelm Ackermann posed a foundational question in mathematical logic: *Is there a general algorithm that can determine whether an arbitrary mathematical statement is provable?* This question, which came to be known as the *Entscheidungsproblem*, was far more than technical, it was also philosophical. It asked whether the act of mathematical discovery itself could be reduced to a mechanical procedure. It remained open until 1936, when Alan Turing [1] and Alonzo Church [2], working independently, demonstrated that no such universal algorithm can exist.

To address this problem, it was first necessary to formalize the notion of an algorithm. Turing introduced an abstract model of computation, now known as the Turing machine. The Turing machine is a theoretical device which manipulates symbols on an infinite tape according to a finite set of deterministic rules. He argued convincingly that this model captures the limits of all physically possible computation, thereby formalizing the concept of all possible algorithmic processes. Using a diagonalization argument, Turing showed that certain problems, such as the Halting Problem, are undecidable, i.e., there exists no Turing machine that can solve them for all inputs. In parallel, Alonzo Church developed a formalism known as the lambda calculus, a mathematical language based on function abstraction and application. Church demonstrated that the equivalence of arbitrary lambda expressions is undecidable, thus arriving at the same negative resolution to the Entscheidungsproblem. Together, these works provided the first rigorous definitions of *computability* and formed the basis of what is now known as the Church-Turing thesis: any function that can be computed by an effective procedure can be computed by a Turing machine or, equivalently, by lambda calculus.

These models of computation are now widely accepted as the fundamental framework for defining algorithmic processes. However, while they delineate the boundary of what is computable, they do not address the efficiency of computation. This leads to the question: Are all physically realizable models of computation equally efficient? The Extended Church-Turing Hypothesis posits that any "reasonable" computational model can be efficiently simulated on a probabilistic Turing machine. More precisely, it asserts that there exists a polynomial-time reduction from any arbitrary physical computation model to a probabilistic Turing machine.

The advent of quantum computing has raised serious challenges to this belief, suggesting that certain quantum processes may outperform classical computation in both speed and expressiveness. The fundamental quantum properties of matter such as superposition, entanglement, and interference may be beneficial for certain computational tasks, potentially providing a more powerful model of computation than classical physics allows. This line of thought was first formalized in 1985 by David Deutsch [3], who introduced the quantum Turing machine model and proposed what is now known as the Deutsch algorithm. It was one of the earliest examples demonstrating that a quantum computer can solve certain problems more efficiently than any classical counterpart. This inspired a sequence of early algorithms such as the Deutsch-Jozsa algorithm [4] and Simon's algorithm [5]. Although these algorithms addressed highly contrived problems with limited practical relevance, they offered the first rigorous evidence of a separation between classical and quantum computational complexity. Examples such as Simon's problem demonstrate exponential separations between quantum and classical query complexity, while recent breakthroughs like $\text{MIP}^* = \text{RE}$ [6] suggest that quantum entanglement can expand the fundamental scope of computation itself. However, such expressiveness separations typically arise in oracle or interactive models rather than standard computational models.

A key turning point came with Grover's algorithm [7], which provided a quadratic speedup for the unstructured search problem. Although its asymptotic advantage is smaller, Grover's algorithm addressed a broad class of practically relevant problems. Shortly thereafter, Shor's algorithm [8] for integer factorization revealed a dramatic exponential speedup over the best known classical algorithms. Given the central role of factorization in cryptographic primitives, Shor's result amplified the perceived practical impact of quantum computing and

firmly established the field's significance. In addition to factoring and search, quantum algorithms have shown significant promise in simulating quantum physical systems, a task inherently challenging for classical computers. A variety of quantum algorithms, including those based on quantum phase estimation and quantum walks, provide substantial speedups for simulating Hamiltonian dynamics and estimating properties of quantum states. Moreover, algorithms such as the Harrow-Hassidim-Lloyd (HHL) method [9] offer exponential improvements for solving large linear systems under certain conditions, a fundamental problem appearing frequently in scientific and engineering applications.

Perhaps the most compelling practical advantage of quantum computing lies in quantum simulation itself. Classical computers face a fundamental barrier when simulating quantum systems due to the exponential growth of the Hilbert space with system size. A system of n qubits is described by a vector in a 2^n-dimensional complex vector space, making direct simulation on a classical device rapidly infeasible as n increases. This observation was central to Richard Feynman's [10] provocative assertion: *"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical."* The idea that quantum systems are best simulated by other quantum systems laid the conceptual groundwork for quantum computing.

Together, these developments indicate that quantum computers are capable of solving certain problems exponentially or at least quadratically faster than classical machines, motivating the ongoing exploration of new quantum algorithms and complexity-theoretic separations. The pursuit is not only about computational advantage, but is also about much deeper philosophical inquiry about the nature of computation and information. If the computation is a fundamental aspect of physical reality then understanding quantum computation can amount to understanding the universe itself.

Quantum algorithms have clearly demonstrated potential advantages in theory, translating these theoretical gains into practical results remains extremely challenging. Quantum computers function by precisely preparing, controlling, and maintaining quantum states, making them inherently sensitive to noise and imperfections. Consequently, current quantum hardware severely limits both the complexity of circuits and the number of qubits that can be reliably manipulated. Quantum algorithms such as Grover's search and Shor's factorization typically involve intricate quantum operations, including multi-controlled logic gates and substantial ancillary space, necessitating fully fault-tolerant quantum circuits with robust quantum error correction. Achieving fault tolerance generally involves encoding logical qubits into many physical qubits using quantum error-correcting codes (e.g., surface codes or concatenated codes). Such encoding schemes significantly increase the physical resources required, as each logical operation involves numerous physical gate operations and error-correction steps. These overheads place fault-tolerant quantum computing firmly beyond the capabilities of current and near-term quantum devices. Thus, there exists a crucial intermediate computational regime between today's noisy quantum devices and idealized fault-tolerant quantum computers. This regime often referred to as *early fault-tolerant quantum computing* (EFTQC).

Early fault-tolerant quantum computing seeks to bridge this practical gap by designing algorithms that can operate effectively under more realistic assumptions: fewer available logical qubits, limited quantum coherence times, simpler logical gates, and significantly reduced circuit depth. EFTQC algorithms are specifically engineered to minimize resource requirements, achieving meaningful computational tasks with modest quantum hardware capabilities. This thesis contributes towards developing novel early fault-tolerant quantum algorithms that enable practically feasible quantum simulations.

Among the variety of problems suited for quantum computing, simulating open quantum systems holds particular significance. Every physical system interacts, to some degree, with its surrounding environment. These interactions can introduce dissipation, noise, and decoherence, and they play a central role in quantum chemistry, condensed matter physics, thermodynamics, and quantum information processing. Understanding open-system dynamics is therefore not only of fundamental scientific interest, but also of practical importance for emerging quantum technologies. The standard mathematical framework for modeling such dynamics is provided by the Lindblad master equation. In this description, the evolution of a density operator is captured by completely positive, tracepreserving (CPTP) maps that generalize unitary dynamics to include dissipative effects. The Lindblad equation thus serves as the quantum analogue of the classical master equation for Markov processes, and it provides a unifying way to treat relaxation, thermalization, and dephasing within a single framework. However, while conceptually elegant, simulating Lindbladian evolution on a classical computer is notoriously difficult. The exponential growth of the underlying Hilbert space quickly makes exact calculations intractable, and even approximate classical methods, such as tensor network techniques [11] or Monte Carlo sampling [12], encounter severe limitations in regimes of high entanglement or strong correlations.

Quantum computers provide a natural platform for addressing this challenge. Yet, existing quantum algorithms for open-system simulation often demand resources beyond the reach of near-term fault-tolerant devices. Traditional approaches based on block-encodings, quantum singular value transformation, or Hamiltonian dilations require deep circuits, multiple layers of error correction, and large numbers of ancilla qubits [13, 14, 15].

These demands place them firmly in the domain of large-scale fault-tolerant quantum computing, leaving a gap between theoretical algorithms and what can realistically be run on the first generations of error-corrected machines.

To make progress in the early fault-tolerant regime, it is essential to look for models of open-system dynamics that are both physically meaningful and operationally simple. One such framework is provided by the *collision model* [16, 17, 18, 19, 20]. In this picture, the environment is represented not as an inaccessible bath, but as a sequence of ancillary systems that interact with the system of interest one at a time. Each interaction, or "collision," is followed by discarding the ancilla, producing an effective irreversible evolution on the system. By repeating such collisions in sequence, one recovers a discrete approximation to continuous Lindbladian dynamics. Despite their conceptual simplicity, collision models can describe a remarkably broad range of physical phenomena. Applications span quantum thermodynamics—including quantum batteries [21, 22], Landauer's principle [23], and thermalization processes [24, 25, 26, 27, 28, 29, 30]—as well as quantum optics [31, 32, 33, 34, 35, 36, 37], continuous measurements [38], and quantum metrology [39, 40].

The appeal of collision models in the early fault-tolerant setting is twofold. First, they are modular: each collision involves only a small subsystem of the environment, making the overall simulation more manageable. Second, they are resource-friendly: the required quantum circuits are shallow, the number of ancillas used at each step is minimal, and the structure naturally accommodates error-corrected qubits with limited coherence times. These features make collision models a promising candidate for practical open-system simulation on early fault-tolerant quantum computers. They not only provide a direct connection to the physics of dissipation and decoherence, but also align with the stringent constraints of near-term fault-tolerant hardware. Simulation of collision model needs a protocol for repeated Hamiltonian simulation. The interaction between the system and an environment subsystem is simply the time evolution of the corresponding total Hamiltonian (sum of the system, sub-environment, and interaction Hamiltonian), and repeated interactions boil down to implementing a composition of Hamiltonian evolutions. Thus, these models provide an avenue to simulate various open quantum systems dynamics using only Hamiltonian simulation algorithms (which have seen remarkable progress) as core subroutines. Indeed, this includes both state-of-the-art Hamiltonian simulation algorithms with near-optimal complexities [41, 42, 43, 44] as well as methods suitable for near-term implementation, such as qDRIFT and (low-order) Trotter methods, that are simple and provide better performance in practice [45, 46, 47, 48].

This thesis builds upon the theoretical foundations established by Pocrnic et al. [49], who provided explicit bounds for approximating Lindbladian dynamics through Markovian collision models and analyzed the query complexity using qubitization [44] and higher-order Trotter methods [48]. We develop end-to-end algorithm for simulating open quantum systems using collision models with minimal ancilla qubits and without access to specialized oracles such as block encodings. Our approach encompasses both established methods—including Trotter formulas [48] and qDRIFT [47]—and novel techniques specifically designed for resource-constrained quantum computers. Central to our contribution is the incorporation of a randomized Hamiltonian simulation technique based on the single-ancilla Linear Combination of Unitaries (SA-LCU) method [50, 51, 52]. This approach represents a significant departure from traditional LCU implementations, requiring only a single ancilla qubit while maintaining polynomial scaling in precision. By integrating SA-LCU with collision models, we demonstrate that meaningful open quantum system simulations can be performed on the first generations of fault-tolerant quantum computers, bridging the gap between theoretical quantum algorithms and practical realization.

1.1 Notation

In this section we collect the main notations that will be used throughout the thesis. These conventions will allow us to describe quantum states, operators, and algorithms in a consistent way, and to state our results in terms of precise resource bounds.

Quantum states and operators. A pure quantum state on n qubits is represented by a unit vector

$$|\psi\rangle \in \mathbb{C}^{2^n}.$$

Mixed states are described by *density operators*, which are positive semidefinite matrices $\rho \in \mathbb{C}^{2^n \times 2^n}$ with unit trace, $\rho \geq 0$ and $\mathrm{Tr}(\rho) = 1$. The evolution of quantum states is described by *quantum channels*, i.e., completely positive and trace-preserving (CPTP) maps $\mathcal{E} : \mathbb{C}^{2^n \times 2^n} \to \mathbb{C}^{2^n \times 2^n}$.

Pauli matrices. We use the standard Pauli matrices

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The n-qubit Pauli group $\mathcal{P}_n$ consists of all n-fold tensor products of these matrices, possibly multiplied by ± 1 or $\pm i$.

Norms. For a vector $v \in \mathbb{C}^d$, we denote by $\|v\|$ its Euclidean norm. For a matrix $A \in \mathbb{C}^{d\times d}$, the *spectral norm* (or operator norm) is

$$\|A\| = \sup_{\|v\|=1} \|Av\|,$$

which equals the largest singular value of A. The *trace norm* (or Schatten 1-norm) is

$$\|A\|_1 = \mathrm{Tr}\left(\sqrt{A^\dagger A}\right),$$

equal to the sum of singular values of A. We define the *induced 1-norm* of a linear map $\mathcal{E}$ acting on density matrices as:

$$\|\mathcal{E}\|_{1\to 1} = \max_{\rho} \frac{\|\mathcal{E}(\rho)\|_1}{\|\rho\|_1},$$

where the maximum is taken over all possible density matrices ρ.

Asymptotic notation. Throughout this thesis, we analyze the complexity of quantum algorithms using standard asymptotic notation. We write $O(f(n))$ for quantities that grow at most as fast as $f(n)$ up to constant factors, while $\Omega(f(n))$ indicates growth at least as fast as $f(n)$. When both upper and lower bounds match, we use $\Theta(f(n))$ to denote tight asymptotic behavior.

A particularly useful notation in quantum algorithm analysis is $\widetilde{O}(f(n))$, which suppresses polylogarithmic factors:

$$\widetilde{O}(f(n)) = O(f(n) \cdot \mathrm{polylog}(f(n))).$$

We follow the standard convention that "efficient" quantum algorithms are those with time or gate complexity polynomial in the input size.

1.2 Overview

This thesis develops practical quantum algorithms for simulating open quantum systems on early fault-tolerant quantum computers. The work bridges fundamental theory in open quantum systems with algorithmic techniques designed for near-term quantum hardware constraints. The thesis is organized into six chapters, each building upon the foundations established in earlier sections.

Chapter 2: Quantum Computing Foundations. We establish the mathematical and computational background required throughout the thesis. The chapter begins with a philosophical exploration of the transition from classical determinism to quantum uncertainty, using the double-slit experiment to motivate the need for new mathematical frameworks. We then formalize quantum mechanics through its postulates, introducing state vectors, observables, and measurement. Specializing to finite-dimensional systems, we develop the qubit formalism, Pauli operators, and the Bloch sphere representation. The chapter covers essential quantum gates, universal gate sets, and the Solovay-Kitaev theorem. We conclude by establishing the circuit model of quantum computation and demonstrate key primitives like the Hadamard test for expectation value estimation.

Chapter 3: Open Quantum Systems. This chapter provides the theoretical foundation for understanding quantum systems that interact with their environment. We begin with the formalism of mixed states and density matrices, explaining how classical probability enters quantum mechanics when dealing with subsystems of larger entangled systems. We develop the theory of quantum channels as completely positive, trace-preserving (CPTP) maps, presenting the Kraus representation and its physical interpretation. The chapter then focuses on Markovian dynamics, deriving the Gorini-Kossakowski-Sudarshan-Lindblad (GKSL) master equation from first principles and explaining the physical assumptions (Born, Markov, and secular approximations) underlying it. We introduce the collision model framework as an alternative microscopic description of open-system dynamics, where the environment is modeled as a sequence of ancillary systems that interact with the system of interest. A key result is the derivation showing how Lindbladian dynamics emerges from collision models in the appropriate continuum limit, providing a bridge between discrete and continuous descriptions. The chapter concludes with error analysis for finite collision models and an introduction to non-Markovian extensions.

Chapter 4: Quantum Algorithms. We survey the landscape of quantum algorithms with particular emphasis on their adaptation to early fault-tolerant quantum computing. The chapter begins by reviewing paradigmatic fault-tolerant primitives including Linear Combinations of Unitaries (LCU), block-encoding, amplitude amplification, and quantum singular value transformation. We analyze their resource requirements and

identify the bottlenecks that make them challenging for near-term implementation. The discussion then turns to the overhead of quantum error correction, explaining how logical qubit encoding and magic state distillation create multiplicative resource penalties. We examine Hamiltonian simulation as a central application, comparing different approaches: product formulas (Trotter-Suzuki methods and qDRIFT), qubitization, and linear combination of unitaries (LCU). A major focus is the single-ancilla LCU algorithm, which trades asymptotic optimality for practical implementability by using only one ancilla qubit and avoiding complex control structures. We provide detailed resource analysis comparing single-ancilla LCU with qubitization, highlighting the former's advantages in the early fault-tolerant regime. The chapter concludes with a survey of recent literature on early fault-tolerant algorithms.

Chapter 5: Simulating Markovian Collision Models. This chapter presents our main algorithmic contribution: a complete framework for simulating Markovian open quantum systems using collision models on early fault-tolerant quantum computers. We formalize the notion of a Markovian K-collision map, where a system undergoes K sequential interactions with independent environmental subsystems. The core algorithm employs the single-ancilla LCU technique to implement each collision while maintaining overall efficiency. We prove correctness of the randomized algorithm and derive complexity bounds showing how the number of repetitions scales with the desired precision ε. The chapter includes detailed comparisons with alternative approaches such as Trotter formulas and qDRIFT, highlighting trade-offs between circuit depth, ancilla requirements, and sampling overhead. As a concrete application, we show how to simulate Lindbladian dynamics by implementing collision models with thermal environments, providing explicit parameter-dependent resource bounds. Numerical benchmarks on the transverse-field Ising model with amplitude damping demonstrate the practical utility of our approach and validate the theoretical predictions.

Chapter 6: Simulating Non-Markovian Quantum Collision Models. We extend the collision model framework to capture non-Markovian dynamics, where memory effects arise from correlations between the system and environment. The key innovation is introducing interactions between consecutive environmental subsystems via CPTP channels, allowing information to propagate through the environment and return to the system at later times. We focus on partial swap interactions as a concrete mechanism for controlling memory strength through a tunable parameter. The algorithmic approach adapts the single-ancilla LCU method to handle the additional intra-environment coupling while preserving complete positivity at each step. We formalize the non-Markovian K-collision map and present a randomized algorithm for estimating expectation values with provable accuracy guarantees. We then provide a numerical demonstrations on a ten-qubit transverse-field Ising chain with partial swap interactions showcase the algorithm's ability to capture non-trivial memory phenomena while remaining compatible with early fault-tolerant hardware constraints.

Chapter 7: Discussion and Outlook. The final chapter synthesizes the main contributions of the thesis and discusses their broader implications for quantum algorithm design and open quantum system simulation. We summarize the key theoretical and algorithmic advances, highlighting how the collision model framework provides a natural bridge between fundamental physics and practical quantum computing. The discussion covers the advantages and limitations of our approach compared to existing methods, identifying regimes where single-ancilla techniques are most beneficial. We outline several promising directions for future research, including extensions to more complex environmental structures, applications to quantum chemistry and condensed matter physics, and potential optimizations for specific quantum hardware architectures. The chapter concludes with reflections on the broader trajectory of early fault-tolerant quantum computing and the role of resource-efficient algorithms in realizing practical quantum advantage.

Throughout the thesis, we maintain a focus on practical implementability while preserving theoretical rigor. The collision model framework serves as a unifying theme, providing both physical insight into open-system dynamics and a natural structure for algorithm design. The single-ancilla LCU technique emerges as a versatile primitive that enables efficient simulation while respecting the constraints of early fault-tolerant quantum hardware. Together, these contributions advance our understanding of how to perform meaningful quantum simulations in the challenging intermediate regime between noisy intermediate-scale quantum devices and fully fault-tolerant quantum computers.

Chapter 2

Foundations of Quantum Computing

"The boundary between physics and metaphysics is like the boundary between sanity and insanity—all the interesting stuff happens at the boundary."

— John Wheeler

Chapter Summary

This chapter establishes the mathematical and computational background used throughout the thesis. We begin with a conceptual motivation by contrasting classical physics with the inherent probabilistic structure of the quantum mechanics. We then formalize the theory through standard postulates of quantum mechanics.

Specializing to finite-dimensional systems relevant for computation, we introduce the qubit, its geometric Bloch-sphere representation, and the Pauli operators. We explain how measurement statistics and expectation values arise. Building from these primitives, we review commonly used quantum gates, including entangling gates such as CNOT and Toffoli. We then motivate universality by adding a non-Clifford gate and recall the Solovay–Kitaev theorem to justify approximation by finite gate sets.

Finally, we adopt the circuit model of quantum computation as the language for algorithms used later in the thesis. We illustrate it with two canonical examples: preparing entanglement from product states and the Hadamard test for estimating expectation values of unitaries. Together, these preliminaries set the toolbox for the thesis.

2.1 The Collapse of Certainty

Consider a game of billiards, if we strike the cue ball with a known velocity and in a specific direction, we can use Newtonian mechanics to calculate exactly where each ball will come to rest. Every bounce, every collision, every final position is determined by the initial conditions and the immutable laws of motion. In a similar way, if you can register the position and momentum of every particle in the universe, one can predict the future by absolute certainty. This was the bold premise behind Laplace's demon. Classical mechanics rendered the universe completely deterministic. A vast clockwork mechanism governed by precise mathematical laws.

We can consider any physical system as an extension of this billiards game: planets orbiting stars, complex weather patterns, the neurons firing in our brains. In this worldview, the entire universe operates like a complex billiards table. Thus, the only barriers to predicting the future were practical ones. That is, one merely needs precise initial data and sufficient computational power. Within this paradigm, one can always compute the precise state of the universe at any future moment.

This deterministic picture, however, began to crack at the turn of the 20th century. Experiments on matter and radiation revealed behaviors that classical mechanics and electrodynamics could not explain. Planck, in addressing blackbody radiation, found that assuming continuous energy led to divergences. Einstein showed in the photoelectric effect that light could eject electrons only in whole quanta, not in arbitrary amounts of energy, defying the wave-only picture. Bohr's model of the atom further demonstrated that electrons cannot orbit the nucleus at arbitrary energies, but only in sharply quantized levels. These discoveries pointed to a profound break with classical physics. At the atomic scale, energy and motion are not continuous billiards-like variables, but inherently quantized. But it was the full development of quantum mechanics that delivered the final blow to the classical picture.

Werner Heisenberg was trying to build a consistent mathematical framework for atomic physics. To rigorously formulate the quantum world he developed matrix mechanics. In this approach, physical quantities

such as position and momentum are described not by numbers but by linear operators, often represented as matrices, called *observables*. He discovered that the position and momentum observables do not commute. This directly leads to the famous uncertainty relation. Unlike classical particles with well-defined positions and momenta, quantum particles existed in a strange state of uncertainty, described not by definite values but by probability distributions. The uncertainty principle fundamentally challenges classical determinism, introducing inherent randomness and probability into physical laws. Laplace's demon required knowing exact position and momentum of every particle simultaeosly. Heisenberg proved this to be impossible.

But what does this quantum uncertainty actually look like in practice? How do we witness the collapse of classical intuition? A striking demonstration comes from a simple experiment.

Imagine shooting marbles at a wall with two parallel slits cut into it. A detector screen is placed behind the wall to record where the marbles land. This behavior is predictable, each marble travels through one slit or the other, creating two distinct bands on the detector screen directly behind each opening. If we block one slit, marbles can only go through the remaining opening. If both slits are open, we simply get the sum of what we would see from each slit individually. This is exactly what our billiards-table universe would predict. But when we repeat this experiment with quantum particles such as electrons or photons, something weird happens. Instead of two distinct bands, an interference pattern emerges on the detector screen. Multiple alternating bright and dark stripes appear, exactly like the pattern created when water waves pass through two openings and interfere with each other. Interference patterns are the signature of waves, not particles. When two waves meet, they can add constructively or destructively. It seemed like the electrons are actually waves, not particles. Perhaps our notion of particle-like electrons was simply wrong.

The truly shocking result comes when we send electrons through the apparatus one at a time. Fire a single electron, wait for it to hit the screen, record its position, then fire another. The same interference pattern emerges. Each individual electron must be interfering with itself. Somehow, a single particle is simultaneously going through both slits and interfering with itself. When we place detectors at each slit to determine which path each electron actually takes. We get an even more bizarre result: the moment we tried to observe which slit the electron used, the interference pattern disappeared. The electrons behaved like classical particles again, going through one slit or the other, producing the expected two-band pattern. The act of measurement itself changes the nature of reality.

This experiment reveals that we need entirely new mathematical tools. Classical physics has no language for an object that simultaneously explores multiple paths. We need a mathematical framework that can handle superposition (being in multiple states at once), probability amplitudes (mathematical objects that determine interference), and the measurement process itself (which seems to collapse multiple possibilities into a single reality).

2.1.1 The New Mathematics

To make sense of these phenomena, we need to use a new mathematical framework. We will describe this framework here as the postulates of quantum mechanics. These postulates provide the rules for describing states, dynamics and measurements in this quantum world.

Postulate I: The State Space

Every isolated quantum system is associated with a complex vector space called the Hilbert space $\mathcal{H}$. The system's state is represented by a unit vector $|\psi\rangle \in \mathcal{H}$, called a *state vector*. For example, for a particle on a line, the state is described by a wavefunction $\psi(x) \in \mathbb{C}$, with

$$P(x) = |\psi(x)|^2, \tag{2.1.1}$$

giving the probability density for finding the particle at position x. The normalization condition

$$\int_{-\infty}^{\infty} |\psi(x)|^2 dx = 1 \tag{2.1.2}$$

ensures that the particle is found somewhere.

Postulate II: Observables

Every measurable physical quantity, such as position, momentum, or energy, is represented by a Hermitian operator A in the state space $\mathcal{H}$. The possible outcomes of a measurement are precisely the eigenvalues of this operator. The mathematical structure of Hermitian operators ensures that these eigenvalues are always real.

Postulate III: Measurement

Suppose a system is in the state $|\psi\rangle$, and we measure an observable A with spectral decomposition

$$A = \sum_a a P_a,$$

where P_a is the projection onto the eigenspace associated with eigenvalue a. Then:

- The probability of obtaining outcome a is

$$\Pr(a) = \|P_a|\psi\rangle\|^2. \tag{2.1.3}$$

- Immediately after the measurement, if the outcome was a, the system's state collapses to

$$|\psi'\rangle = \frac{P_a|\psi\rangle}{\|P_a|\psi\rangle\|}. \tag{2.1.4}$$

Thus measurement is inherently probabilistic, and the act of measuring alters the system's state.

Postulate IV: Time Evolution

The state of an isolated quantum system evolves smoothly and deterministically according to the Schrödinger equation

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle, \tag{2.1.5}$$

where H is the Hamiltonian of the system. The solution can be written as

$$|\psi(t)\rangle = e^{-iHt/\hbar}|\psi(0)\rangle, \tag{2.1.6}$$

so that time evolution is described by a unitary operator.

2.2 Discrete Quantum Systems

So far, we have described quantum systems such as particles on a line, where observables like position and momentum take values from a continuum. However, not all quantum systems are continuous. Some have only a finite number of possible states. The simplest nontrivial example is a *two-level system*, which can exist in one of two distinct configurations.

Classically, this would correspond to a system with two possible outcomes, such as a light switch being either "on" or "off." Quantum mechanically, the story is richer. Although there are only two measurement outcomes, the state can be a superposition of the two possibilities.

Instead of a wavefunction defined across continuous space, the state of a two-level system lives in a two-dimensional complex vector space. We choose an orthonormal basis

$$\psi_1 = \begin{pmatrix}1\\0\end{pmatrix}, \qquad \psi_2 = \begin{pmatrix}0\\1\end{pmatrix}. \tag{2.2.1}$$

Then, any state can be written as a linear combination of these two vectors:

$$|\psi\rangle = \alpha\psi_1 + \beta\psi_2 = \begin{pmatrix}\alpha\\\beta\end{pmatrix}, \tag{2.2.2}$$

where $\alpha, \beta \in \mathbb{C}$ are probability amplitudes with the normalization condition

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2.2.3}$$

ensuring that probabilities sum to one.

Working with column vectors quickly becomes complicated as systems grow larger. Therefore we adopt a more compact *Dirac notation*:

- $|\psi\rangle$ (a "ket") represents a column vector,
- $\langle\psi|$ (a "bra") is its conjugate transpose,

- $\langle\phi|\psi\rangle$ (a "bracket") is the inner product between two states.

In this notation our two basis states are written as

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \quad \langle 0| = \begin{pmatrix}1 & 0\end{pmatrix}, \tag{2.2.4}$$

$$|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}, \quad \langle 1| = \begin{pmatrix}0 & 1\end{pmatrix}. \tag{2.2.5}$$

They satisfy the orthonormality relations

$$\langle 0|0\rangle = \langle 1|1\rangle = 1, \qquad \langle 0|1\rangle = \langle 1|0\rangle = 0. \tag{2.2.6}$$

2.2.1 The Qubit

This two-level system is the building block of quantum computation. Classical computation is based on bits, which can be either 0 or 1. Quantum computation generalizes this concept with the *qubit*. A qubit's computational basis states are $|0\rangle$ and $|1\rangle$, and a general qubit state is

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2.2.7}$$

with $|\alpha|^2 + |\beta|^2 = 1$.

Unlike a classical bit, which must be in exactly one of its two states, a qubit can exist in a superposition of both. When measured in the computational basis, the outcomes are probabilistic:

- 0 occurs with probability $|\alpha|^2$,
- 1 occurs with probability $|\beta|^2$.

For example, the state

$$|\psi\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{2.2.8}$$

produces either 0 or 1 with equal probability. Before measurement, however, the qubit is not partly 0 and partly 1, but rather a coherent superposition that can interfere with other quantum states.

A single qubit lives in a two-dimensional Hilbert space, but thanks to normalization and the irrelevance of global phase, its state can be described using just two real parameters. This can be describe by a powerful geometric picture called the *Bloch sphere* as shown in Fig. 2.1.

Any normalized qubit state can be written as

$$|\psi\rangle = \cos\left(\tfrac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\tfrac{\theta}{2}\right)|1\rangle, \tag{2.2.9}$$

where $0 \le \theta \le \pi$ and $0 \le \phi < 2\pi$. The angle θ determines the latitude of the point on the sphere, while ϕ specifies the longitude.

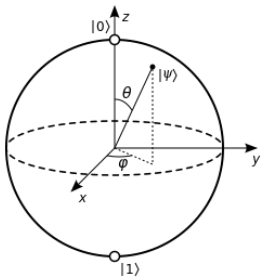


Figure 2.1: The Bloch sphere representation. In this picture, $|0\rangle$ corresponds to the north pole, and $|1\rangle$ to the south pole. States like $(|0\rangle + |1\rangle)/\sqrt{2}$ and $(|0\rangle - |1\rangle)/\sqrt{2}$ lie on the equator at opposite points. Global phases (multiplying $|\psi\rangle$ by $e^{i\gamma}$) do not change the point on the sphere, reflecting their physical irrelevance.

As we will see later, the Bloch sphere provides an intuitive way to visualize single-qubit operations:

- Pauli gates correspond to 180° rotations about the x, y, or z axes.
- The Hadamard gate maps the poles to points on the equator, creating superpositions.
- Rotation gates $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$ correspond to genuine rotations of the Bloch vector about the respective axes.

2.2.2 Observables and Measurements

In discrete quantum systems, observables are represented by Hermitian matrices. For qubits, the most fundamental observables are the *Pauli operators*:

$$\sigma_x = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{2.2.10}$$

$$\sigma_y = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \tag{2.2.11}$$

$$\sigma_z = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2.2.12}$$

together with the identity operator

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.2.13}$$

These operators form a complete basis for all 2×2 Hermitian matrices, which means that any single-qubit observable can be expressed as a linear combination of them. Beyond their mathematical role, they have direct physical interpretations: X, Y, and Z correspond to spin measurements along the x, y, and z axes, respectively, when the qubits are realized as a spin-1/2 particle.

Among these, the Pauli-Z operator plays a central role because it represents measurement in the *computational basis*. Its eigenvalues are

$$+1 \quad \text{with eigenvector } |0\rangle, \qquad -1 \quad \text{with eigenvector } |1\rangle.$$

Thus, measuring a qubit in the Z basis is equivalent to asking whether it is in the state $|0\rangle$ or $|1\rangle$.

For a general qubit state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

the expectation value of a Z-measurement is

$$\langle Z\rangle = \langle\psi|Z|\psi\rangle = |\alpha|^2 \cdot (+1) + |\beta|^2 \cdot (-1) = |\alpha|^2 - |\beta|^2. \tag{2.2.14}$$

This quantity captures the average outcome of many repeated measurements on identically prepared states: it is positive if the system is more likely to be found in $|0\rangle$, negative if it is more likely to be in $|1\rangle$, and zero for an equal superposition.

In this way, the Pauli operators provide both a compact mathematical language and a direct physical interpretation of qubit measurements. They will also serve as the basic building blocks of quantum gates and Hamiltonians in later sections.

2.2.3 Evolution of Discrete Quantum Systems

For a system of n qubits, the state is represented by a vector $|\psi(t)\rangle$ in a 2^n-dimensional Hilbert space and the dynamics are governed by a Hamiltonian H, a $2^n \times 2^n$ Hermitian matrix, through the time-dependent Schrödinger equation:

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle. \tag{2.2.15}$$

If H is time-independent, the solution takes the form

$$|\psi(t)\rangle = e^{-iHt/\hbar}|\psi(0)\rangle, \tag{2.2.16}$$

where $|\psi(0)\rangle$ is the initial state. The operator

$$U(t) = e^{-iHt/\hbar} \tag{2.2.17}$$

is the *time-evolution operator*. Since H is Hermitian, $U(t)$ is always unitary, ensuring reversibility and conservation of probability.

This principle underlies quantum computation. A *quantum gate* is simply a unitary operator acting on one or more qubits:

$$|\psi_{\text{final}}\rangle = U|\psi_{\text{initial}}\rangle. \tag{2.2.18}$$

Quantum algorithms are constructed by composing such gates into sequences that transform an input state into a desired output state. While an arbitrary unitary on n qubits cannot be implemented directly, it can be

decomposed into simpler one and two qubit gates. This parallels classical digital circuits, where complex logic is built from simple gates such as AND, OR, and NOT.

A key concept is *universality*: a small set of gates is universal if any unitary operation can be approximated to arbitrary accuracy using only gates from that set. Hence, studying a handful of elementary gates is sufficient to understand how any quantum algorithm can be realized in practice.

We begin with the most common single-qubit and multi-qubi gates and then how these gates form the universal sets.

Single-Qubit Gates. Single-qubit gates act on individual two-level systems and provide the basic tools for creating and manipulating superpositions.

Pauli Gates. The Pauli operators are among the most fundamental gates:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.2.19}$$

The X gate performs a bit flip ($X|0\rangle = |1\rangle$), the Z gate applies a phase flip ($Z|1\rangle = -|1\rangle$), and the Y gate combines both operations.

Hadamard Gate. The Hadamard gate creates superpositions:

$$H = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{2.2.20}$$

mapping $|0\rangle \mapsto (|0\rangle + |1\rangle)/\sqrt{2}$ and $|1\rangle \mapsto (|0\rangle - |1\rangle)/\sqrt{2}$.

Phase and T Gates. The phase gate S and the T (or $\pi/8$) gate apply controlled phase rotations:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}. \tag{2.2.21}$$

The S gate corresponds to a $\pi/2$ rotation about the Z axis, while the T gate corresponds to a $\pi/4$ rotation. The T gate plays a crucial role in universal quantum computation.

Rotation Gates. More generally, qubits can be rotated continuously about any axis of the Bloch sphere. For example:

$$R_x(\theta) = e^{-i\theta X/2} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}, \tag{2.2.22}$$

$$R_z(\phi) = e^{-i\phi Z/2} = \begin{pmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{pmatrix}. \tag{2.2.23}$$

Multi-Qubit Gates. Single-qubit gates cannot generate entanglement, which is essential for quantum advantage. Multi-qubit gates introduce correlations between qubits.

CNOT Gate. The CNOT gate acts on two qubits: a *control* and a *target*. It flips the target qubit if and only if the control qubit is in state $|1\rangle$. In matrix form:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.2.24}$$

Its action on the computational basis states is:

$$|00\rangle \mapsto |00\rangle, \quad |01\rangle \mapsto |01\rangle, \quad |10\rangle \mapsto |11\rangle, \quad |11\rangle \mapsto |10\rangle.$$

In other words, the first qubit (control) remains unchanged, while the second (target) flips when the control is $|1\rangle$. This entangling gate is essential in quantum computing: combined with single-qubit operations. It circuit

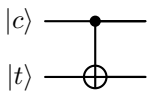


Figure 2.2: CNOT gate circuit representation.

representation is shown in 2.2.

Toffoli Gate (CCNOT). The Toffoli gate generalizes CNOT to three qubits: two controls and one target. It flips the target qubit if and only if both controls are in state $|1\rangle$. Its matrix is 8×8, with most entries identical to the identity matrix, except for the last two basis states:

$$|110\rangle \mapsto |111\rangle, \quad |111\rangle \mapsto |110\rangle.$$

All other basis states remain unchanged.

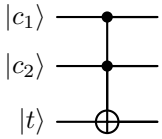


Figure 2.3: Toffoli gate circuit representation.

The Toffoli gate frequently appears in quantum algorithms that implement arithmetic, modular exponentiation, and error-correction routines. Although this can be relea

Clifford Gates and Universal Gate Sets

The set $\{H, S, \text{CNOT}\}$, where H is the Hadamard gate, S is the phase gate, and CNOT is the controlled-NOT gates is called the *Clifford group*. The circuits composed solely of them can be efficiently simulated on a classical computer, a result known as the *Gottesman–Knill theorem* [53, 54]. Despite this efficient simulability, Clifford gates are far from trivial. They are used in quantum error-correcting codes, stabilizer states. However, Clifford operations alone are not universal for quantum computation as they cannot generate arbitrary unitary transformations.

To achieve universality, one must add at least one non-Clifford gate. The most common choice is the T gate (or $\pi/8$ gate), which introduces an additional $\pi/4$ phase:

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

The extended set $\{H, S, \text{CNOT}, T\}$ forms a *universal gate set*, meaning that any unitary transformation on any number of qubits can be approximated to arbitrary precision using only these gates.

Repeated combinations of these gates can approximate any desired unitary to within any given error ϵ. A key theoretical result guarantees this:

Solovay–Kitaev Theorem [55]. This theorem states that if a finite gate set is universal, then any target unitary can be approximated to accuracy ϵ using only $\mathcal{O}(\log(1/\epsilon))$ gates.

Taken together, these results show that all of quantum computing rests on a surprisingly small foundation. Notably using H, S, T, and CNOT, one can, in principle, implement any quantum algorithm.

2.2.4 The Circuit Model of Quantum Computation

The most widely used framework for describing quantum algorithms is the *quantum circuit model*. In this representation information is carried by wires, processed by gates, and finally read out by measurements. The wires represent qubits, and the gates correspond to unitary operations.

- **Qubits.** Each wire in a quantum circuit carries a qubit, which may be in a superposition of $|0\rangle$ and $|1\rangle$. An n-qubit register is represented by a 2^n-dimensional state vector.

- **Quantum Gates.** Gates are unitary transformations acting on one or more qubits. They evolve the state according to $|\psi\rangle \mapsto U|\psi\rangle$. As discussed earlier, universal computation can be achieved using a small set of gates, such as $\{H, S, T, \text{CNOT}\}$.
- **Circuit Structure.** A quantum circuit is a sequence of gates arranged in layers. The ordering of gates specifies the order of operations.
- **Measurement.** At the end of the circuit, selected qubits are measured, typically in the computational basis $\{|0\rangle, |1\rangle\}$. Measurement collapses a qubit into a definite classical outcome, which can be used to extract information.

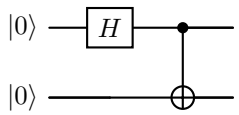
To illustrate the circuit model, let's look at some examples.

Creating Entanglement

Starting from $|00\rangle$, we apply a Hadamard gate to the first qubit, followed by a CNOT with the first qubit as control and the second as target:

$$|00\rangle \xrightarrow{H\otimes I} \tfrac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \xrightarrow{\text{CNOT}} \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

The resulting entangled state cannot be written as a product of single-qubit states, demonstrating the unique power of quantum circuits.



This simple circuit generates one of the four maximally entangled Bell states. Such states are essential resources in quantum communication, teleportation, and many algorithms.

The Hadamard Test

A bit more sophisticated example is the *Hadamard test*, which allows us to estimate the expectation value $\langle\psi|U|\psi\rangle$ of a unitary U on a state $|\psi\rangle$. This is especially useful in quantum algorithms for tasks such as amplitude estimation or phase estimation.

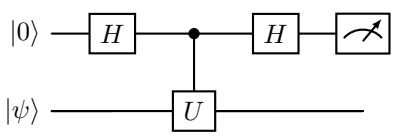
The circuit works as follows:

- Start with an ancilla qubit in $|0\rangle$ and the system in $|\psi\rangle$.
- Apply a Hadamard to the ancilla.
- Perform a controlled-U with the ancilla as control and $|\psi\rangle$ as target.
- Apply another Hadamard to the ancilla and measure it.

The probability of obtaining outcome 0 in the ancilla is

$$P(0) = \tfrac{1}{2}\left(1 + \text{Re}\langle\psi|U|\psi\rangle\right).$$

Thus, repeated runs of the test allow us to estimate the real part of the expectation value. With a slight modification (adding a phase gate S before the second Hadamard), one can also estimate the imaginary part.



This illustrates how the circuit model can encode quantum operations.

Chapter 3

Open quantum systems

"In physics, you don't have to go around making trouble for yourself. Nature does it for you."

— Frank Wilczek

Chapter Summary

Quantum systems are never perfectly isolated. Real quantum systems exchange energy and information with surrounding degrees of freedom. These systems are called to be "Open". This chapter discribes a framework for studying reduced dynamics of open quantum systems that will be used throughout the thesis. We begin with the basic language of mixed states and subsystems, explain how general physical evolutions are captured by quantum channels, and then specialize to Markovian master equations in Gorini–Kossakowski–Sudarshan–Lindblad (GKSL) form. We complement this continuum picture with a alternative model based on repeated interactions, and we prove a precise correspondence between the two in an appropriate limit.

For readers seeking a broader perspective, Lidar's lecture notes offer an accessible, modern introduction to the theory of open quantum systems, and much of the initial discription is inspired from it [56]. On the collision-model side, the review by Ciccarello et al. provides a comprehensive account of repeated-interaction dynamics and its extensions [57]. Recent work has also clarified how repeated interactions can serve as a bridge to algorithm design by relating collision models to implementable Hamiltonian-simulation primitives [58]. The chapter is structed in the following manner. First, we set up the basic formalism needed for open dynamics. We introduce density operators, explain partial trace and reduced states, and recall unitary evolution and measurement.

Then we discuss quantum channels: completely positive trace-preserving (CPTP) maps that encode the most general physical evolutions on a system when the environment is ignored. We present the Kraus representation. We develop Markovian master equations. We introduce the GKSL generator, summarize the physical assumptions behind it (Born, Markov, and secular/rotating-wave), and record its key properties, including complete positivity, trace preservation, contractivity, and typical steady-state behavior.

Finally, we turn to quantum collision models. We motivate collision models as a microscopic model of environment coupling and as a constructive route to simulation. We then derive Lindbladian dynamics from discrete collisions. We also quantify finite-step errors. The last part of the chapter briefly outlines mechanisms for non-Markovian behavior within collision models.

3.1 Open quantum systems

Many analyses of quantum systems begin with the idealization of perfect isolation. In practice, no system is perfectly closed; it exchanges energy, particles, and information with an environment comprising many uncontrolled degrees of freedom. Because the environment is large and only partially characterized, tracking the joint state is generally intractable. We therefore seek equations for the reduced state that capture the environment's effect on the system. Several phenomena shape the reduced dynamics through interactions with the environment. Quantum superpositions lose coherence when system degrees of freedom become correlated with environmental modes, a process known as decoherence. The system also dissipates energy into its surroundings, often relaxing toward a thermal state. Furthermore, practical quantum technologies use external drives, measurement, and feedback that couple the system to engineered reservoirs and detectors. Understanding the induced noise channels is essential for high-fidelity control, metrology, and error mitigation.

Because the environment is high dimensional and only partially characterized, we use effective equations for the system that capture its average influence. This is the setting of open quantum systems that we focus on in this thesis.

3.1.1 Mixed states, Density matrix and POVM

So far we have considered the state of a quantum system that can always be described by a single wavefunction. This picture is adequate for closed systems, but it breaks down once we focus on a subsystem of a larger entangled system. A simple example makes this clear.

Consider two qubits prepared in the maximally entangled Bell state

$$|\Psi\rangle = \tfrac{1}{\sqrt{2}}\big(\,|00\rangle + |11\rangle\,\big).$$

If we have access to both qubits, then $|\Psi\rangle$ is a perfectly valid description of the system. But suppose we are only interested in the first qubit and do not have access to the second. What state should we assign to the first qubit alone?

No single pure state captures the correct statistics. For instance, if we measure in the computational basis, the first qubit is equally likely to be $|0\rangle$ or $|1\rangle$. If we measure in the Hadamard basis, the outcomes are also uniformly random. Thus the reduced description must incorporate classical randomness in the outcomes, not just quantum superposition. In other words, the first qubit is not in a pure state, but in a *mixed state*, which can be thought of as a probabilistic mixture of pure states.

This illustrates why classical probabilities enter the theory of open quantum systems. When part of a larger entangled system is inaccessible the effective description of the subsystem necessarily includes statistical uncertainty. To formalize this, we introduce the *density matrix*.

A density matrix is a positive semidefinite operator ρ of unit trace that encodes the statistics of all measurements on a quantum system. Pure states $|\psi\rangle$ are represented by rank-one projectors,

$$\rho = |\psi\rangle\langle\psi|\,,$$

while classical mixtures $\{p_i, |\psi_i\rangle\}$ are represented by

$$\rho = \sum_i p_i\ |\psi_i\rangle\langle\psi_i|\,.$$

In the Bell-state example, removeing the second qubit yields

$$\rho_1 = \tfrac{1}{2}\big(\,|0\rangle\langle 0| + |1\rangle\langle 1|\,\big),$$

the maximally mixed state of a single qubit. This operator compactly encodes the fact that all measurement outcomes are uniformly random, regardless of basis.

The density matrix formalism provides a unified language that simultaneously captures quantum coherence and classical randomness. It becomes indispensable in open quantum systems, where interaction with an environment generically drives subsystems into mixed states.

The example of an entangled pair hinted at an important operation: discarding part of a larger system. Formally, this is accomplished by the *partial trace*. Suppose the joint state of a system S and environment E is described by ρ_{SE}. If we are only interested in the system, we construct its reduced state by tracing out the environment:

$$\rho_S = \mathrm{Tr}_E\left[\rho_{SE}\right]. \tag{3.1.1}$$

This procedure guarantees that ρ_S reproduces all statistics of measurements performed solely on S, regardless of the inaccessible environment E.

In the Bell state example, $\rho_{SE} = |\Psi\rangle\langle\Psi|$ with $|\Psi\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$. Tracing out one qubit yields $\rho_S = \mathcal{I}/2$, the maximally mixed state. This illustrates how entanglement with an unobserved environment naturally generates mixed states.

To complete the formalism, we must understand how to describe measurements on mixed states. In the pure-state picture, projective measurements suffice. The outcomes correspond to orthogonal projectors $\{P_i\}$ with probabilities $\langle\psi|P_i|\psi\rangle$. However, in realistic scenarios, measurements can be more general. The framework of *positive operator-valued measures* (POVMs) captures this generality.

A POVM is a collection of positive semidefinite operators $\{M_i\}$ that sum to the identity,

$$M_i \geq 0, \qquad \sum_i M_i = I. \tag{3.1.2}$$

When measuring a state ρ, the probability of outcome i is given by

$$p(i) = \mathrm{Tr}[M_i \rho] \,. \tag{3.1.3}$$

Projective measurements are a special case where the M_i are orthogonal projectors. For example, consider a qubit with density matrix ρ. Measuring in the computational basis corresponds to the POVM $\{M_0 = |0\rangle\langle 0| \,, M_1 = |1\rangle\langle 1|\}$. On the other hand, a noisy detector that sometimes fails to distinguish outcomes might be modeled by $M_0 = 0.9\,|0\rangle\langle 0| + 0.1\,|1\rangle\langle 1|$ and $M_1 = I - M_0$. Such a measurement is no longer projective but still yields well-defined probabilities through Eq. (3.1.3).

3.1.2 Unitary evolution of mixed states

With this foundation in place, we now turn to the question of how mixed states evolve in time. For pure states, the evolution is given by the Schrödinger equation

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H\,|\psi(t)\rangle \ , \tag{3.1.4}$$

where H is the Hamiltonian of the system. The formal solution is

$$|\psi(t)\rangle = U(t)\,|\psi(0)\rangle \,, \quad U(t) = e^{-iHt/\hbar}.$$

If the system begins in a mixed state $\rho(0)$, its time evolution must be consistent with two requirements. First, probabilities should evolve linearly so that ensembles of states remain ensembles. Second, the rule should reduce to Schrödinger evolution when $\rho(0)$ is a pure state. Under the assumption that the system is closed and evolves unitarily, these requirements uniquely determine the following dynamical law, known as the von Neumann equation:

$$i\hbar \frac{d}{dt} \rho(t) = [H, \rho(t)] \,. \tag{3.1.5}$$

Here $[H, \rho] = H\rho - \rho H$ is the commutator. The formal solution mirrors the pure-state case,

$$\rho(t) = U(t)\rho(0)U^\dagger(t).$$

The eigenvalues of $\rho(t)$, which represent the probabilities in the mixture, are preserved in time. Thus unitary evolution cannot create or destroy classical randomness; it only rotates the basis in which coherence is expressed. For example, the maximally mixed state $\rho = I/2$ remains invariant under any unitary evolution, reflecting its complete symmetry. It is important to emphasize that Eq. (3.1.5) only governs the dynamics of isolated quantum systems. Once a system interacts with an environment, the dynamics need not be unitary and a more general framework is required.

3.1.3 General quantum dynamics: CPTP maps and Kraus operators

The von Neumann equation provides the correct dynamical law for closed systems evolving unitarily. However, once a system interacts with an environment, its evolution need not be unitary. Tracing out the environment can introduce decoherence, dissipation, or noise. To describe these processes consistently, we require a more general notion of dynamics than Eq. (3.1.5).

Completely positive trace-preserving maps.

The most general physical evolution of a density matrix ρ is given by a linear map

$$\rho \mapsto \mathcal{E}(\rho),$$

where $\mathcal{E}$ satisfies two key conditions:

1. **Trace-preserving**: Probabilities must add up to one, so

$$\mathrm{Tr}[\mathcal{E}(\rho)] = \mathrm{Tr}[\rho] = 1. \tag{3.1.6}$$

2. **Complete positivity**: Not only must $\mathcal{E}$ map every valid density matrix to another positive semidefinite operator, but it must also do so when the system is part of a larger entangled state. Formally, $\mathcal{E} \otimes I$ must map positive operators to positive operators for any ancillary system. This ensures that entangled states remain physical after the map acts.

Maps that satisfy both properties are called *completely positive trace-preserving* (CPTP) maps. They form the most general class of quantum operations consistent with the postulates of quantum mechanics.

Kraus representation.

A remarkable theorem states that any CPTP map can be expressed in the *Kraus form*

$$\mathcal{E}(\rho) = \sum_k K_k \rho K_k^\dagger, \tag{3.1.7}$$

where the operators $\{K_k\}$, called *Kraus operators*, satisfy the completeness relation

$$\sum_k K_k^\dagger K_k = I. \tag{3.1.8}$$

This condition guarantees that $\mathcal{E}$ preserves the trace of ρ.

The Kraus representation provides an intuitive picture of noisy evolution: the system undergoes one of several possible transformations K_k, each occurring with probability $\mathrm{Tr}[K_k \rho K_k^\dagger]$. The final state is the ensemble average of these outcomes.

Example: Qubit dephasing. Consider a qubit subject to dephasing noise, which destroys off-diagonal coherences but leaves populations intact. The corresponding CPTP map is

$$\mathcal{E}(\rho) = (1-p)\rho + pZ\rho Z,$$

where Z is the Pauli-Z operator and p is the dephasing probability. This can be written in Kraus form using

$$K_0 = \sqrt{1-p}\, I, \qquad K_1 = \sqrt{p}\, Z.$$

One checks that $K_0^\dagger K_0 + K_1^\dagger K_1 = I$, so the map is trace-preserving.

The Kraus representation thus generalizes unitary evolution to the setting of open systems. In the next section, we will generalize further to continuous-time dynamics, where CPTP maps are generated by master equations such as the Lindblad equation.

3.2 Lindbladian dynamics

The Kraus representation shows that any physical evolution over a finite time interval can be described by a completely positive trace-preserving (CPTP) map. But in physical settings, it is natural to model the dynamics as a continuous-time dynamics. That is, we would like to describe the system at any intermediate time t, not just before and after some discrete channel has acted. This leads us to the theory of quantum master equations, and in particular to the Gorini–Kossakowski–Sudarshan–Lindblad (GKSL) form [59, 60].

Suppose $\{\mathcal{E}_t : t \geq 0\}$ is a one-parameter family of CPTP maps describing the evolution of a system. To be physically consistent, these maps should satisfy the semigroup property

$$\mathcal{E}_{t+s} = \mathcal{E}_t \circ \mathcal{E}_s, \qquad \mathcal{E}_0 = I, \tag{3.2.1}$$

which expresses the idea that evolving for time s and then time t is the same as evolving for time $t+s$. Such families are called *quantum dynamical semigroups*.

A fundamental result states that the generator $\mathcal{L}$ of any quantum dynamical semigroup that is CPTP must take the Lindblad form:

$$\frac{d}{dt}\rho(t) = \mathcal{L}(\rho(t)) = -\frac{i}{\hbar}[H, \rho(t)] + \sum_j \Big(L_j \rho(t) L_j^\dagger - \tfrac{1}{2}\{L_j^\dagger L_j, \rho(t)\} \Big). \tag{3.2.2}$$

Here:

- The first term $-\frac{i}{\hbar}[H, \rho]$ describes the unitary part of the evolution, generated by an effective Hamiltonian H.
- The operators $\{L_j\}$ are called *Lindblad operators* or *jump operators*, and they encode the dissipative processes induced by the environment.
- The anticommutator $\{A, B\} = AB + BA$ ensures that the dissipative terms preserve Hermiticity and trace.

Equation (3.2.2) is known as the GKSL equation [59, 60], or simply the Lindblad master equation.

The Lindblad form is derived under several physical approximations:

1. **Born approximation:** The system–environment coupling is weak, so that the environment remains approximately in its stationary state. This justifies factorizing the joint state as $\rho_{SE}(t) \approx \rho_S(t) \otimes \rho_E$.
2. **Markov approximation:** The environment has no memory of its past interactions, and the dynamics is local in time.
3. **Secular or rotating-wave approximation:** Fast oscillating terms in the interaction picture are averaged out, leaving only resonant contributions. This guarantees complete positivity of the generator.

Together, these approximations place us in the regime of *Markovian weakly coupled open quantum systems*, where Eq. (3.2.2) is valid.

Dynamics governed by Eq. (3.2.2) has several important properties:

- **Complete positivity and trace preservation:** By construction, the map $e^{t\mathcal{L}}$ is CPTP for all $t \geq 0$.
- **Contractivity (trace norm):** If $\Phi_t = e^{t\mathcal{L}}$, then for all density operators ρ, σ,

$$\|\Phi_t(\rho - \sigma)\|_1 \leq \|\rho - \sigma\|_1 . \tag{3.2.3}$$

  Equivalently, the induced $1 \to 1$ norm is nonincreasing and bounded by one:

$$\|\Phi_t\|_{1\to 1} \leq 1, \qquad \|\Phi_{t+s}\|_{1\to 1} \leq \|\Phi_t\|_{1\to 1} \quad (s, t \geq 0). \tag{3.2.4}$$

3.2.1 Examples

Dephasing noise. A qubit subject to pure dephasing can be modeled with Lindblad operator $L = \sqrt{\gamma}\, Z$, where Z is the Pauli-Z operator. The master equation is

$$\frac{d}{dt}\rho = \gamma\big(Z\rho Z - \rho\big).$$

This leaves populations invariant but exponentially suppresses off-diagonal terms.

Amplitude damping. Energy relaxation from $|1\rangle$ to $|0\rangle$ can be modeled with $L = \sqrt{\gamma}\, |0\rangle\langle 1|$. The resulting dynamics drives the system to the ground state $|0\rangle$.

The Lindblad equation thus provides a unified framework for modeling both coherent and dissipative processes in quantum systems.

3.3 Collision Model Framework

Collision models offer a complementary perspective on open quantum system dynamics that is both conceptually simple and practically useful. Instead of describing the environment as a large, continuous reservoir that generates a smooth Lindbladian evolution, collision models represent the system–environment interaction as a sequence of discrete interactions with well-defined environmental subsystems. Each of these interactions, or "collisions," is governed by a unitary evolution on the joint Hilbert space. This framework is especially appealing because it reduces the dynamics to unitary evolution, making it easier to simulate.

In a typical collision model, the environment is decomposed into a collection of mutually independent subsystems, often modeled as qubits or finite-dimensional ancillas. Each ancilla is prepared in a prescribed initial state and interacts with the system only once, after which it is discarded. The system Hamiltonian H_S and the ancilla Hamiltonian H_{E_n} govern their free evolution, while an interaction Hamiltonian H_{I_n} couples them during the n-th interaction. The total unitary for that step is

$$U_n = \exp(-i\tau H_n), \qquad H_n = H_S + H_{E_n} + H_{I_n}, \tag{3.3.1}$$

where τ is the collision duration.

This description captures the idea that the system interacts locally and briefly with a fresh piece of the environment, rather than with an entire infinite bath at once.

The collision model evolution proceeds through the following steps:

1. **Initialization**: The system starts in state $\rho_S(0)$ and each environmental subsystem E_n is prepared in state ρ_{E_n}.

2. **Sequential Collisions**: The system interacts with environmental subsystems one at a time through unitary operations:
$$\rho_{SE_n}(t_n) = U_n \left(\rho_S(t_{n-1}) \otimes \rho_{E_n}\right) U_n^\dagger \tag{3.3.2}$$

3. **Partial Trace**: After each collision, the environmental subsystem is traced out, updating the system state:
$$\rho_S(t_n) = \mathrm{Tr}_{E_n}[\rho_{SE_n}(t_n)] \tag{3.3.3}$$

4. **Repetition**: This process repeats for all environmental subsystems, with the final system state serving as input for the next round of collisions.

Collision models make explicit the link between microscopic interactions and effective system dynamics. By appropriately choosing the ancilla states and interaction Hamiltonians, one can recover familiar noise channels such as dephasing, amplitude damping, or thermalization. In the limit of short collision time τ and weak interactions, the repeated-collision dynamics converges to a continuous Lindbladian evolution of the GKSL form. Thus, collision models provide both an intuitive picture and a constructive route to derive master equations.

From a computational viewpoint, collision models are attractive because they can be directly simulated with finite resources. The environment is represented by a stream of simple ancillas, each interacting unitarily with the system. This makes them valuable not only as a conceptual tool but also as a platform for algorithmic and experimental studies of open quantum dynamics.

Furthermore, collision models provide a natural framework for studying non-Markovian effects, as the sequence of collisions can be engineered to capture memory effects and other dynamical features that are challenging to describe in the Lindblad framework.

3.4 Lindbladian Dynamics from Collision Models

One of the most remarkable aspects of the collision model framework is its ability to recover Lindbladian dynamics in a suitable limiting regime. While the discrete description emphasizes the sequential nature of system–environment interactions, the continuous Lindblad master equation emerges when collisions become frequent. This connection provides a constructive route to simulate Markovian open-system dynamics.

To see this correspondence, consider a system S colliding sequentially with a stream of identically prepared ancillas E_n. Each collision lasts a short time τ and is governed by a Hamiltonian of the form

$$H_n = H_S + H_{E_n} + \lambda H_{I_n}, \tag{3.4.1}$$

where H_S acts only on the system, H_{E_n} acts only on the n-th ancilla, and H_{I_n} couples the two. The joint unitary for a single collision is

$$U_n = \exp(-i\tau H_n). \tag{3.4.2}$$

The reduced system dynamics is then described by a map

$$\Phi_n(\rho_S) = \mathrm{Tr}_{E_n}\left[U_n \left(\rho_S \otimes \rho_{E_n}\right) U_n^\dagger\right], \tag{3.4.3}$$

where ρ_{E_n} is the initial state of the ancilla. Iterating this map produces the system's discrete-time trajectory.

To analyze the continuous-time limit, expand the unitary to second order in τ:

$$U_n \approx \mathbb{I} - i\tau(H_S + H_{E_n} + \lambda H_{I_n}) - \tfrac{\tau^2}{2}(H_S + H_{E_n} + \lambda H_{I_n})^2. \tag{3.4.4}$$

This leads to an expansion of the map

$$\Phi_n(\rho_S) = \rho_S - i\tau[H_S, \rho_S] - \frac{\tau^2\lambda^2}{2}\,\mathrm{Tr}_{E_n}([H_{I_n}, [H_{I_n}, \rho_S \otimes \rho_{E_n}]]) + O(\tau^2). \tag{3.4.5}$$

Here we have used two simplifications:

- $\mathrm{Tr}_{E_n}([H_{E_n}, \rho_S \otimes \rho_{E_n}]) = 0$, since H_{E_n} acts only on the ancilla.
- Cross terms of order $\lambda\tau$ vanish under the assumption $\mathrm{Tr}_{E_n}([H_{I_n}, \rho_S \otimes \rho_{E_n}]) = 0$.

To obtain a nontrivial dissipative limit, the coupling is scaled as $\lambda^2\tau = 1$. Taking $\tau \to 0$ then yields the differential equation

$$\frac{d}{dt}\rho_S = -i[H_S, \rho_S] + \mathrm{Tr}_{E_n}\left(H_{I_n}\left(\rho_S \otimes \rho_{E_n}\right) H_{I_n} - \tfrac{1}{2}\{H_{I_n}^2, \rho_S \otimes \rho_{E_n}\}\right). \tag{3.4.6}$$

This is precisely of the Gorini–Kossakowski–Sudarshan–Lindblad form. The first term corresponds to coherent evolution under the system Hamiltonian H_S, while the second term describes dissipative contributions generated by repeated collisions with ancillas.

For concreteness, if the interaction Hamiltonian takes the form

$$H_{I_n} = V \otimes a^\dagger + V^\dagger \otimes a, \tag{3.4.7}$$

with ancillas prepared in thermal states at inverse temperature β, the dissipative part reduces to

$$\sum_{j=1}^{2}\left(L_j \rho_S L_j^\dagger - \tfrac{1}{2}\{L_j^\dagger L_j, \rho_S\}\right), \tag{3.4.8}$$

where the Lindblad operators are

$$L_1 = \sqrt{z}\, V, \qquad L_2 = \sqrt{z-1}\, V^\dagger, \tag{3.4.9}$$

with coefficients z determined by the partition function. Thus, in the scaling limit of frequent, strong collisions, the discrete collision model converges to Lindbladian dynamics.

For a collision model with m environment subsystems, each interacting with the system sequentially, we must renormalize the system Hamiltonian as $H_S \to H_S/m$ to prevent the system from moving ahead from the interaction. The resulting Lindblad equation becomes:

$$\mathcal{L}(\rho_S) = -i[H_S, \rho_S] - \frac{1}{2}\sum_{k=1}^{m} \mathrm{Tr}_{E_k}[H_{I_k}, [H_{I_k}, \rho_S \otimes \rho_{E_k}]] \tag{3.4.10}$$

For the error analysis that follows, it is convenient to define the effective superoperator that governs each collision in this multi-environment setting:

$$\tilde{\mathcal{H}}_n(\rho) = -i\left[\frac{1}{m}H_0 + H_{E_n} + \frac{1}{\sqrt{\tau}}H_{I_n}, \rho\right] \tag{3.4.11}$$

This superoperator already incorporates both the scaling condition $\lambda^2\tau \to 1$ and the renormalization by m required for multiple environment subsystems. The derived correspondence is exact in the limit $\tau \to 0$ with $\lambda^2\tau = 1$. For finite collision times, the collision model approximates Lindbladian dynamics with controllable error, making it practically useful for both theoretical analysis and numerical simulation of open quantum systems. This fundamental connection between discrete collision models and continuous Lindbladian dynamics forms the theoretical foundation for the quantum algorithms we develop in subsequent chapters.

3.5 Error Analysis for Finite Collisions

The exact equivalence between continuous Lindbladian dynamics and discrete collision models holds only in the limit of infinitesimal collision time and infinitely many collisions. Quantum algorithms necessarily operate with a finite time step and a finite number of collisions. How far is the discrete collision evolution from the target Lindbladian at a fixed physical time t?

Let $\mathcal{L}$ denote the Lindblad generator on the system system. The continuous evolution is

$$\rho_S(t) = e^{t\mathcal{L}}[\rho_S(0)]\,.$$

In a finite collision model we partition t into ν rounds of duration $\tau = t/\nu$. In each round the system interacts sequentially with m fresh environment subsystems through reduced collision maps $\tilde{\Phi}_1(\tau), \ldots, \tilde{\Phi}_m(\tau)$. We collect one round of collisions into the channel

$$\mathcal{M}_m(\tau) \;:=\; \bigcirc_{n=1}^{m} \tilde{\Phi}_n(\tau)\,, \tag{3.5.1}$$

and we write the *ν-step collision evolution* as

$$\mathcal{M}_{m,\nu}(t) \;:=\; \big(\mathcal{M}_m(t/\nu)\big)^{\circ\nu}\,. \tag{3.5.2}$$

We measure the worst–case error in induced trace norm,

$$\epsilon(\nu, t, m) \;:=\; \left\| e^{t\mathcal{L}} - \mathcal{M}_{m,\nu}(t) \right\|_{1\to 1}, \tag{3.5.3}$$

At small step size τ, both $e^{\tau\mathcal{L}}$ and $\mathcal{M}_m(\tau)$ admit first–order expansions of the form $I + \tau\mathcal{L} + O(\tau^2)$. The global error after ν steps is then governed by the single–step second–order remainder, amplified at most linearly in ν by channel contractivity. The second–order remainder contains two contributions. The first comes from truncating the continuous semigroup $e^{\tau\mathcal{L}}$, and scales like $\tau^2\|\mathcal{L}\|_{1\to1}^2$. The second comes from the finite–time collision round, and scales like $\tau^2\sum_{n=1}^m \gamma_n^4$, where

$$\gamma_n \;:=\; \max\left\{\|H_S\|,\; \|H_{E_n}\|,\; \|H_{I_n}\|\right\}$$

is a characteristic energy scale for the n-th collision. Combining these ingredients yields the $O(1/\nu)$ decay of the total error for fixed t.

Theorem 3.5.1 (Error bounds for discrete step collision models)**.** *Let $\tilde{\Phi}_n(\tau)$ be the* collision maps *of one collision round and let $\mathcal{L}$ be the target Lindbladian specified by the weak–coupling collision–model construction. Assume the small–step conditions*

$$\frac{t}{\nu}\,\|\mathcal{L}\|_{1\to1} \le c_0 \qquad \textit{and} \qquad \frac{t}{\nu}\,\gamma_n \le c_1 \quad \textit{for all } n\,,$$

for absolute constants c_0, c_1. Then for all $t > 0$ and integers $m, \nu \ge 1$,

$$\left\| e^{t\mathcal{L}} - \mathcal{M}_{m,\nu}(t) \right\|_{1\to1} \;\in\; O\!\left(\frac{t^2}{\nu}\left(\|\mathcal{L}\|_{1\to1}^2 + m\,\max_n \gamma_n^4\right)\right). \tag{3.5.4}$$

Proof. Set $\tau = t/\nu$. Completely positive trace–preserving maps are contractions in the induced $1\to1$ norm, hence for any CPTP A, B one has

$$\left\|A^{\circ\nu} - B^{\circ\nu}\right\|_{1\to1} \;\le\; \nu\,\|A - B\|_{1\to1}\,. \tag{3.5.5}$$

With $A = e^{\tau\mathcal{L}}$ and $B = \mathcal{M}_m(\tau)$, this gives

$$\left\|e^{t\mathcal{L}} - \mathcal{M}_{m,\nu}(t)\right\|_{1\to1} \;\le\; \nu\left\|e^{\tau\mathcal{L}} - \mathcal{M}_m(\tau)\right\|_{1\to1}. \tag{3.5.6}$$

Duhamel's formula yields

$$\left\|e^{\tau\mathcal{L}} - (I + \tau\mathcal{L})\right\|_{1\to1} \;\le\; \frac{\tau^2}{2}\,e^{\tau\|\mathcal{L}\|_{1\to1}}\,\|\mathcal{L}\|_{1\to1}^2 \;\in\; O\big(\tau^2\,\|\mathcal{L}\|_{1\to1}^2\big), \tag{3.5.7}$$

under $\tau\|\mathcal{L}\|_{1\to1} \le c_0$.

Each reduced interaction map admits a short–time expansion

$$\tilde{\Phi}_n(\tau) \;=\; I + \tau\,\mathcal{L}_n \;+\; E_n(\tau)\,, \qquad \|E_n(\tau)\|_{1\to1} \;\in\; O\big(\tau^2\,\gamma_n^4\big). \tag{3.5.8}$$

Here $\mathcal{L}_n$ is the first–order generator extracted from the system–environment unitary of the n-th collision, and the $O(\tau^2\gamma_n^4)$ bound follows from a second–order cumulant estimate for short–time Hamiltonian evolutions, using that commutators contribute at most quadratically in the local energy scales. The RI construction guarantees the *first–order consistency*

$$\sum_{n=1}^{m} \mathcal{L}_n \;=\; \mathcal{L}\,. \tag{3.5.9}$$

Multiplying the m factors in (3.5.1) and using (3.5.9) one finds

$$\mathcal{M}_m(\tau) \;=\; I + \tau\mathcal{L} \;+\; R_{\mathrm{coll}}(\tau)\,, \tag{3.5.10}$$

where $R_{\mathrm{coll}}(\tau)$ consists of two types of second–order contributions: the local remainders $E_n(\tau)$, and the cross terms produced by noncommutativity when multiplying $\prod_n(I+\tau\mathcal{L}_n)$. Using submultiplicativity and $\|\mathcal{L}_n\|_{1\to1} \in O(\gamma_n^2)$, one obtains

$$\left\|R_{\mathrm{coll}}(\tau)\right\|_{1\to1} \;\in\; O\Big(\tau^2\sum_{n=1}^{m}\gamma_n^4\Big) \;\subseteq\; O\big(\tau^2\,m\,\max_n\gamma_n^4\big). \tag{3.5.11}$$

Combining (3.5.7)–(3.5.11),

$$\left\|e^{\tau\mathcal{L}} - \mathcal{M}_m(\tau)\right\|_{1\to1} \;\le\; O\big(\tau^2\,\|\mathcal{L}\|_{1\to1}^2\big) \;+\; O\big(\tau^2\,m\,\max_n\gamma_n^4\big). \tag{3.5.12}$$

Insert (3.5.12) into (3.5.6) and use $\tau = t/\nu$ to obtain (3.5.4). □

Equation (3.5.4) shows $\epsilon(\nu, t, m) = O(1/\nu)$ at fixed t. Doubling the number of rounds halves the error. This scaling matches the intuition that both dynamics agree at first order in τ and differ only at second order per round. Given a tolerance $\varepsilon > 0$, the bound (3.5.4) implies a sufficient number of rounds

$$\nu \in \Theta\left(\frac{t^2}{\varepsilon}\left(\|\mathcal{L}\|_{1\to 1}^2 + m \max_n \gamma_n^4\right)\right).$$

This translates directly into circuit depth for implementations that realize each collision map using some Hamiltonian simulation primitive.

3.6 Non-Markovian Collision Models

Many physical environments possess dynamics on timescales that are comparable to the system dynamics. In such situations the standard semigroup picture, where future evolution depends only on the present state, is in correct. It is often singnified by *information backflow*. If $\rho_1(t)$ and $\rho_2(t)$ are two system states evolving under the same open dynamics, the trace distance $D(\rho_1(t), \rho_2(t)) = \frac{1}{2}\|\rho_1(t) - \rho_2(t)\|_1$ quantifies their distinguishability. Markovian evolutions contract this distance monotonically. An increase of the trace distance at some time indicates that information previously lost to the environment returns to the system, which is a hallmark of non-Markovian behavior [61, 62]. Formally, one widely used characterization connects Markovianity to *CP-divisibility*: a dynamics with dynamical maps $\{\Phi_{t,0}\}_{t\geq 0}$ is Markovian if each intermediate propagator $\Phi_{t,s} = \Phi_{t,0}\Phi_{s,0}^{-1}$ is completely positive for all $t \geq s$. Violation of CP-divisibility witnesses non-Markovianity [63].

In the basic collision model, the system system interacts once, in sequence, with fresh and mutually uncorrelated ancillas which yields a CP-divisible dynamics. To introduce memory, one must relax one or more of the basic hypotheses. Some of the non-Markovian extensions are: *(i)* adding ancilla–ancilla collisions, *(ii)* preparing the ancillas in initially correlated states, and *(iii)* letting the system collide multiple times with a given ancilla. Each mechanism transfers correlations across time and breaks CP-divisibility of the reduced system dynamics.

Adding ancilla–ancilla collisions: Allow ancillas to interact with each other. Physically, this models a bath where environmental subunits interact and propagate correlations forward. As a result, when system meets the next ancilla, that ancilla already carries information about system from previous steps. The key consequence is that the overall step-n dynamical map on system is not CP-divisible, hence the dynamics is non-Markovian. In a suitable continuous-time limit, the reduced dynamics obeys an integro-differential *memory-kernel* master equation. For a broad class of kernels, one can guarantee that the resulting time-continuous dynamics remains CPTP.

Initially correlated ancillas: If the ancilla chain is prepared in a correlated state before any interaction with the system, the first collision already entangles the system with the whole bath. This immediately destroys CP-divisibility of the reduced dynamics and can produce pronounced information backflow. A prototypical construction mixes several Markovian trajectories, producing a non-Markovian overall map that cannot be described by a single Lindblad equation. These models have been analyzed theoretically and realized on quantum processors, and they have been used to probe the relationship between coarse-graining time, correlation time, and CP-divisibility.

Multiple collisions with the same ancilla: Allow the system to interact again with a previously visited ancilla after a finite delay. In this bi-local or delayed-feedback architecture, the state of the returning ancilla already encodes records of system from the earlier step. When the second encounter occurs, the intermediate propagator $\Phi_{t,s}$ is no longer CP because system and the returning ancilla are correlated, so CP-divisibility breaks.

All the above extensions preserve a crucial property at the level of a *single* discrete step: the joint unitary on system plus the relevant ancillas, followed by tracing out ancillas, yields a CPTP map on system. The composition of such steps over a fixed time horizon remains CPTP. What fails is CP-divisibility of the *intermediate* evolutions, because those would require complete positivity after conditioning on a state of the environment that is correlated with the system. This reconciles the presence of information backflow with the fundamental requirement that the overall dynamical map $\Phi_{t,0}$ is physical.

Non-Markovian collision models bridge the gap between the microscopic models and reduced open dynamics. They are flexible enough to embed structured reservoirs, finite-memory channels, and feedback, while staying

close to the implementable primitives. In the rest of the thesis we will focus on the collision models with ancilla-ancilla interactions to model non-Markovian dynamics.

Chapter 4

Quantum algorithms

"In theory, there is no difference between theory and practice. In practice, there is."

— Yogi Berra

Chapter Summary

This chapter explores the landscape of quantum algorithms with particular emphasis on their practical implementation in the early fault-tolerant era. We begin by tracing the conceptual origins of quantum algorithms, from Deutsch's first proof-of-principle demonstration to the paradigmatic breakthroughs of Shor and Grover that established quantum computing's potential for exponential and quadratic speedups. The chapter then examines the powerful but resource-intensive primitives that underlie modern quantum algorithm design, including Linear Combination of Unitaries (LCU), block-encoding, amplitude amplification, and Quantum Singular Value Transformation (QSVT). While these frameworks achieve asymptotic optimality, their implementation demands deep circuits, many ancilla qubits, and extensive use of non-Clifford gates, making them challenging for near-term fault-tolerant devices.

We analyze the overhead of quantum error correction, explaining how logical qubit encoding and magic state distillation create multiplicative resource penalties that can render theoretically efficient algorithms practically infeasible. This motivates our focus on early fault-tolerant algorithms that prioritize implementability over asymptotic optimality. We examine Hamiltonian simulation as a central application, comparing product formulas (which are ancilla-free but have polynomial scaling in precision) with qubitization (which is asymptotically optimal but resource-intensive). A major contribution is the detailed analysis of the single-ancilla LCU algorithm, which uses only one ancilla qubit while achieving logarithmic dependence on precision. This approach trades quadratic scaling in evolution time and state preparation with estimating expectation values for dramatic reductions in qubit overhead and circuit complexity, making it well-suited for the early fault-tolerant regime.

The chapter concludes with a survey of recent developments in early fault-tolerant quantum computing, highlighting how the field is evolving toward practical algorithms that can extract meaningful quantum advantage with limited resources.

4.1 From Physics to Algorithms

At its core, every computation is a physical process. A classical computer is a carefully engineered physical device that evolves in time according to the laws of classical physics. When we run an algorithm on silicon, we are effectively performing an experiment in classical mechanics and electromagnetism. This perspective makes it natural to ask: if the fundamental rules of physics are not classical but quantum mechanical, should there also exist new kinds of computation? In other words, does quantum physics have computational significance?

This question guided the earliest explorations of quantum algorithms. In 1985, David Deutsch formalized the notion of a universal quantum computer [64], motivated by the idea that the Church–Turing principle should extend to the quantum world. He proposed the first nontrivial quantum algorithm, showing that quantum mechanics can provide an advantage even for simple decision problems. The problem he studied was: given a Boolean function $f : \{0,1\} \to \{0,1\}$ determine if it is constant or balanced. Classically, two evaluations are necessary, but Deutsch's quantum algorithm achieves the task with a single query to a quantum oracle by exploiting superposition and interference. Although the practical impact of this example is not much, it served as the first proof of principle that quantum processes could be harnessed for faster computation.

This idea was extended by Deutsch and Jozsa in 1992 [65] to functions $f : \{0,1\}^n \to \{0,1\}$ under the promise that f is either constant or balanced. Their algorithm solves the problem with a single query, compared to the $\Omega(2^{n-1})$ queries needed classically in the worst case. This algorithm provided the first example of an *exponential separation* between quantum and classical computing.

A breakthrough came shortly thereafter with Simon's algorithm [66]. Simon studied the problem of determining a hidden XOR mask $s \in \{0,1\}^n$ given access to a black-box function $f : \{0,1\}^n \to \{0,1\}^n$ with the promise that $f(x) = f(y)$ if and only if $x \oplus y = s$. Classically, solving this requires exponential queries, but Simon's quantum algorithm recovers s with only polynomially many queries by exploiting the quantum Fourier transform. This result was the first to demonstrate an exponential speedup for a problem with an algebraic structure, and it provided the conceptual blueprint for Shor's factoring algorithm.

The true turning point came with Shor's discovery in 1994 [67]. Shor's algorithm demonstrated that quantum mechanics could efficiently solve problems of great number-theoretic and cryptographic importance, such as factoring and discrete logarithms. This breakthrough transformed quantum computing into a field with possibly profound practical implications.

Grover introduced his quantum search algorithm in 1996 [68]. Grover's algorithm provides a quadratic speedup for unstructured search, requiring $O(\sqrt{N})$ queries compared to the $\Theta(N)$ classically. Though less dramatic than Shor's exponential speedup, Grover's method applies broadly and has since become a versatile primitive in quantum algorithm design.

In summary, the idea that *computation is a physical process* lies at the heart of quantum algorithms. By treating quantum mechanics as the governing law of computation, we can uncover new algorithmic paradigms unavailable in the classical world. The earliest algorithms, though simple, paved the way toward systematic frameworks that now form the foundation of the field. In the sections that follow, we will study important quantum algorithm primitives to understand the modern algorithmic techniques.

4.2 Paradigmatic Fault-Tolerant Quantum Primitives

Many of modern quantum algorithm design is built on a set of general primitives that can be combined and repurposed across many applications. These primitives such as the Quantum Fourier Transform, Quantum Phase Estimation, Block-Encoding, Amplitude Amplification, Linear Combination of Unitaries, and Quantum Singular Value Transformation provide the structural building blocks for Hamiltonian simulation, ground state preparation, optimization and other quantum algorithms. They represent the paradigmatic tools of the fault-tolerant era: powerful abstractions that achieve asymptotic optimality. But these are often at the expense of significant circuit depth, large ancilla requirements, and heavy use of costly non-Clifford gates. In what follows, we describe some of these frameworks, highlighting both their conceptual role in quantum algorithms and the practical resource demands that determine their feasibility on real hardware.

4.2.1 Linear Combination of Unitaries

In many quantum applications one must implement a non-unitary operator. For example, H^{-1} arises in quantum linear solvers, and e^{Ht} appears as the imaginary-time propagator for ground state preparation, where H is a Hermitian operator. Quantum hardware, however, can only execute unitary gates. The Linear Combination of Unitaries (LCU) framework addresses this challenge by expressing a target operator A as a weighted sum of unitaries, and then using quantum interference to approximately implement A within a larger unitary circuit. This makes LCU one of the most powerful primitives in quantum algorithm design.

Any linear operator can be expressed as a linear combination of unitaries. This suggests that if we can implement a generic LCU construction, then a wide class of non-unitary operators can be approximated within any desired accuracy.

Suppose A is an operator that admits an LCU decomposition

$$\tilde{A} = \sum_{j=1}^{M} \alpha_j U_j, \qquad \alpha_j > 0,\ U_j^\dagger U_j = \mathbb{I}, \tag{4.2.1}$$

such that $\|\tilde{A} - A\| \leq \epsilon$. Here α_j are positive real coefficients, and U_j are unitary operators. If such a decomposition exists, then the LCU framework allows us to approximately implement A on a quantum computer.

The LCU algorithm. The standard LCU algorithm [69] uses a *prepare–select–unprepare* procedure:

1. Start with the system register in $|\psi\rangle$ and an $s = \lceil \log M \rceil$-qubit ancilla initialized to $|0^s\rangle$.

2. Apply the *prepare* operation P to the ancilla:

$$P|0^s\rangle = \sum_{j=1}^{M} \sqrt{\frac{\alpha_j}{\alpha}}|j\rangle, \tag{4.2.2}$$

where $\alpha = \sum_j \alpha_j$.

3. Apply the *select* operation

$$S = \sum_{j=1}^{M} |j\rangle\langle j| \otimes U_j, \tag{4.2.3}$$

which applies U_j to the system whenever the ancilla is in state $|j\rangle$.

4. Apply $P^\dagger$ to uncompute the ancilla.

The resulting transformation is

$$(P^\dagger \otimes I)S(P \otimes I)|0^s\rangle|\psi\rangle = \frac{1}{\alpha}\Big(\sum_{j=1}^{M} \alpha_j U_j\Big)|\psi\rangle \otimes |0^s\rangle + |\Phi^\perp\rangle, \tag{4.2.4}$$

where $|\Phi^\perp\rangle$ has the ancilla orthogonal to $|0^s\rangle$. On post-selecting the ancilla onto $|0^s\rangle$, the system register evolves as $\tilde{A}|\psi\rangle/\alpha$.

The schematic structure of the LCU method is shown in Figure 4.1. The ancilla is first put into a superposition encoding the weights α_j, the appropriate U_j is selected on the system register, and the ancilla is then uncomputed.

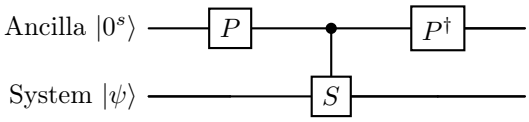


Figure 4.1: Linear Combination of Unitaries with a s-qubit ancilla register and a system register. The prepare map P encodes weights via $P|0^s\rangle = \sum_{j=1}^{M} \sqrt{\alpha_j/\alpha}\,|j\rangle$ with $\alpha = \sum_j \alpha_j$. The multiplexed operator $\mathrm{S}(U) = \sum_{j=1}^{M} |j\rangle\langle j| \otimes U_j$ acts on the system, after which the ancilla is uncomputed by $P^\dagger$. Postselecting the ancilla in $|0^s\rangle$ effects $\big(\sum_j \alpha_j U_j/\alpha\big)|\psi\rangle$ on the system.

Resource analysis. While elegant, the standard LCU algorithm is resource-intensive:

- **Ancillas.** The scheme requires atleast $s = \lceil \log M \rceil$ ancillas to index the unitaries.
- **Prepare/Unprepare.** Implementing P and $P^\dagger$ may require decomposition via quantum Shannon decomposition, which can incur large depth.
- **Select operation.** The multi-controlled U_j gates are complex and often require large T-gate counts once compiled.
- **Success probability.** The scheme succeeds with probability

$$p = \frac{\sum_j \alpha_j^2}{\alpha^2}, \tag{4.2.5}$$

which in the worst case can be as low as $1/M$. To boost the probability to near one, amplitude amplification must be used, further increasing circuit depth and ancilla requirements.

4.2.2 Block-Encoding

Block-encoding has become a central input model in many quantum algorithms [44, 70]. The key idea is to embed a non-unitary operator A into a larger unitary by placing A/α (up to error) in a sub-block. This allows one to apply polynomial transformations to A via purely unitary dynamics, unlocking powerful techniques like quantum singular value transformation (QSVT). Block-encoding thus underlies many algorithms in Hamiltonian simulation, quantum linear algebra, and quantum optimization.

Many quantum algorithms manipulate a linear operator A that is not inherently unitary. To interface with quantum hardware, which executes unitaries, we require a representation of A inside a unitary framework. The Block-encoding provides exactly this

A unitary U acting on $s+n$ qubits is an (α, s, ϵ)-*block-encoding* of an n-qubit operator A if

$$\left\| A \;-\; \alpha\big(\langle 0^{\otimes s}| \otimes I\big)\, U\, \big(|0^{\otimes s}\rangle \otimes I\big) \right\| \;\leq\; \epsilon. \tag{4.2.6}$$

Here $\alpha > 0$ is a normalization factor, s is the number of ancilla qubits, and ϵ is the permitted error. Equivalently, preparing ancillas in $|0^s\rangle$, applying U, and projecting those ancillas back onto $|0^s\rangle$ effects A/α (up to error) on the system register.

One often pictures U in block-matrix form as

$$U \;\approx\; \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}, \tag{4.2.7}$$

so that the upper-left block contains the (rescaled) operator A.

Block-encoding is used in many algorithmic frameworks such as:

- *Hamiltonian simulation:* Given a block-encoding of H, one can apply QSVT to simulate e^{-iHt} [44].
- *Quantum linear system solving:* Many modern solvers first block-encode the system matrix A, then implement A^{-1} via QSVT [70, 71].

In practice, block-encodings are frequently constructed from LCU decompositions. Suppose

$$A = \sum_{j=1}^{M} a_j U_j, \qquad \sum_j |a_j| \leq \alpha,$$

where each U_j is unitary. One can (i) prepare an ancilla state encoding the amplitudes a_j, (ii) perform a multiplexed controlled-U_j, and then (iii) uncompute the ancilla. The resulting unitary is an $(\alpha, s, 0)$-block-encoding of A, where $s = \lceil \log M \rceil$. This construction mirrors LCU but without the post-selection, as we require deterministic access to A for downstream operations.

Resource costs. The major costs mirror those of LCU:

- **Ancilla qubits.** One needs $s = \lceil \log M \rceil$ qubits to index the terms plus workspace for amplitude preparation.
- **Circuit depth.** The prepare and unprepare of the ancilla can be very deep, especially for nonuniform coefficients.
- **Non-Clifford gate overhead.** Arbitrary amplitude rotations and multiplexed controls translate into many T-gates after compilation.

In summary, block-encoding is a clean and unifying primitive that underlies many quantum algorithmic frameworks. But like LCU, its realization is resource-intensive, making it a critical target for optimization in early fault-tolerant architectures.

4.2.3 Amplitude Amplification

Another central primitive in the quantum algorithmic toolkit is *amplitude amplification*. It generalizes Grover's search algorithm [7] and provides a systematic way to boost the success probability of a quantum procedure. Whereas Grover's algorithm is tailored to unstructured search, amplitude amplification abstracts the idea of repeated reflections into a general framework that can amplify the amplitude of any marked subspace [72]. This

makes it a versatile building block for algorithms ranging from unstructured search to quantum Monte Carlo methods.

Suppose we have a quantum algorithm $\mathcal{A}$ that, starting from the all-zero state, prepares a superposition of the form

$$\mathcal{A}|0^n\rangle = \sqrt{p}\,|\text{good}\rangle + \sqrt{1-p}\,|\text{bad}\rangle, \tag{4.2.8}$$

where $|\text{good}\rangle$ denotes states encoding successful outcomes, and p is the probability of success upon measurement. If p is small, many repetitions of $\mathcal{A}$ are needed to observe a good outcome with high probability. Amplitude amplification allows us to increase the amplitude on $|\text{good}\rangle$ quadratically faster than classical repetition: $O(1/\sqrt{p})$ repetitions instead of $O(1/p)$.

Amplitude amplification works by repeatedly applying a Grover-type iterate composed of two reflections:

- Reflection about the "good" subspace, denoted S_{good}.
- Reflection about the starting state $\mathcal{A}|0^n\rangle$, denoted S_ψ.

The Grover iterate is

$$Q = -S_\psi S_{\text{good}}. \tag{4.2.9}$$

Each application of Q performs a rotation in the two-dimensional space spanned by $|\text{good}\rangle$ and $|\text{bad}\rangle$, gradually increasing the amplitude of $|\text{good}\rangle$. After $O(1/\sqrt{p})$ applications, the state is close to $|\text{good}\rangle$, and measuring yields a successful outcome with high probability.

Amplitude amplification is a generic subroutine that underlies many algorithms:

- *Quantum Monte Carlo*: amplifies the amplitude of samples encoding expected values, leading to quadratic speedups over classical sampling [73].
- *Amplitude estimation*: combines amplitude amplification with phase estimation to obtain precise estimates of probabilities [72].

To implement amplitude amplification, we require two reflections:

1. S_{good} can be realized if we have an oracle that marks the good states by a phase flip. For example, if the good states are those where an indicator function $f(x) = 1$, then S_{good} applies a -1 phase to $|x\rangle$ whenever $f(x) = 1$.
2. S_ψ can be implemented using $\mathcal{A}$ and its inverse: $S_\psi = \mathcal{A}S_0\mathcal{A}^\dagger$, where S_0 flips the phase of the all-zero state $|0^n\rangle$.

Each Grover iterate thus requires two calls to $\mathcal{A}$, plus one application of the marking oracle.

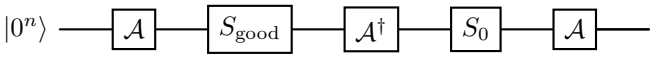


Figure 4.2: Schematic circuit for one Grover iterate $Q = -S_\psi S_{\text{good}}$. The operator $\mathcal{A}$ prepares the state, S_{good} flips the marked states, and S_ψ is implemented by $\mathcal{A}S_0\mathcal{A}^\dagger$.

A useful variant is *oblivious amplitude amplification* (OAA) [42]. Unlike the standard version, which requires knowledge of the "good" subspace and reflections about $\mathcal{A}|0^n\rangle$, OAA works in settings where an operator is embedded as a block of a larger unitary (e.g., via LCU or block-encoding). Here, one does not need to measure or postselect the ancilla qubits. The amplification acts "obliviously" to their state. OAA is therefore crucial for converting the probabilistic outputs of LCU constructions into deterministic block-encodings, and it forms a key ingredient in Hamiltonian simulation algorithms.

Resource analysis. The resource requirements of amplitude amplification depend on the cost of $\mathcal{A}$ and the oracle:

- **Circuit depth.** Each Grover iterate requires two applications of $\mathcal{A}$ and one oracle query. To boost success probability from p to constant, $O(1/\sqrt{p})$ iterations are needed. Thus, depth scales as $O(\text{Cost}(\mathcal{A})/\sqrt{p})$.
- **Qubit count.** No extra ancillas are required beyond what $\mathcal{A}$ and the oracle use. However, implementing S_ψ requires both $\mathcal{A}$ and $\mathcal{A}^\dagger$, which may increase the ancilla count.

- **Gate decomposition.** If $\mathcal{A}$ or the oracle uses non-Clifford gates, each Grover iterate inherits their cost. Since the algorithm requires many iterations, T-gate counts can become large in fault-tolerant settings.

Amplitude amplification illustrates both the power and the cost of quantum primitives. It offers a quadratic speedup over classical sampling, and it underpins important algorithms like amplitude estimation and Hamiltonian simulation. Yet, its implementation requires repeated uses of both $\mathcal{A}$ and its inverse, and relies heavily on oracles and non-Clifford gates. Even for minor asymptotic seperation from classical algorithms, it usually poses significant challenges in implementation.

4.2.4 Quantum Singular Value Transformation and Signal Processing

Quantum Singular Value Transformation (QSVT) is one of the most general and powerful frameworks in quantum algorithms. It enables the implementation of polynomial transformations of the singular values of block-encoded matrices, and it underlies many of the best known asymptotic algorithms for Hamiltonian simulation, linear system solving, and quantum machine learning. QSVT is built upon a core primitive known as Quantum Signal Processing (QSP), introduced by Low and Chuang [74], which provides a method for implementing polynomial transformations of eigenvalues of single-qubit rotations. Together, QSP and QSVT unify many algorithmic ideas into a single, flexible framework [70].

Suppose we have a block-encoding of a matrix A with singular value decomposition $A = W\Sigma V^\dagger$. In many algorithms, we wish to apply a polynomial function f to the singular values σ_i of A. For example:

- In Hamiltonian simulation, we need to approximate e^{-iHt} as a polynomial in H [44].
- In quantum linear system solving, we wish to approximate $1/x$ on the nonzero singular values of A [75].
- In quantum machine learning, applying functions like the sign or step function to singular values enables classification tasks [76].

QSVT provides a systematic way to implement such transformations in a fault-tolerant setting.

At the heart of QSVT is QSP, which applies polynomial transformations to the eigenvalues of a single-qubit rotation. Given a sequence of phases $\Phi = (\phi_1, \ldots, \phi_m)$, QSP constructs a unitary

$$U_\Phi(x) = e^{i\phi_1 Z} R_x(\theta) e^{i\phi_2 Z} R_x(\theta) \cdots e^{i\phi_m Z}, \tag{4.2.10}$$

where $R_x(\theta)$ is a rotation whose angle depends on x. By carefully choosing the phases ϕ_j, one can ensure that the $(1,1)$ entry of $U_\Phi(x)$ is a polynomial $P(x)$ of bounded degree [74]. Thus, QSP provides a way to implement any bounded polynomial transformation of a variable encoded in a rotation.

QSVT generalizes QSP to matrices. If U is an $(\alpha, s, 0)$-block-encoding of a matrix A, then using QSP as a subroutine one can implement a polynomial transformation P of the singular values of A, producing a new block-encoding of $P(A/\alpha)$ [70]. This means that, given access to a block-encoding of A, we can construct block-encodings of functions such as $\exp(-iAt)$, $\mathrm{sign}(A)$, or A^{-1}. This versatility makes QSVT a universal tool for designing algorithms.

A QSVT circuit alternates between applications of the block-encoding U, its inverse $U^\dagger$, and single-qubit phase rotations determined by the QSP phase sequence Φ. Each use of U is controlled by ancillas, and the length of the phase sequence determines the degree of the polynomial. Thus, the cost of implementing $P(A)$ grows with the degree of the polynomial approximation.

A schematic illustration of QSVT is shown in Figure 4.3. The system register encodes A via its block-encoding U, while an ancilla qubit controls the application of U and $U^\dagger$. Between these applications, phase rotations $R_z(\phi_j)$ are applied to the ancilla qubit, following the QSP sequence.

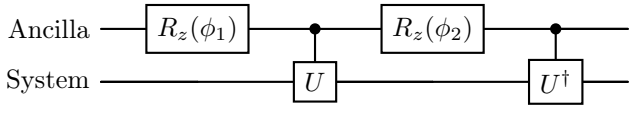


Figure 4.3: Schematic QSVT circuit with two steps. The ancilla qubit undergoes phase rotations while controlling applications of the block-encoding U and $U^\dagger$. By choosing the phases appropriately, one implements a polynomial transformation of the singular values of A.

Resource analysis. While QSVT is asymptotically optimal for many problems, it comes with heavy resource requirements:

- **Ancillas.** QSVT requires at least one ancilla to control the block-encoding, in addition to the s ancillas needed to implement the block-encoding itself.
- **Circuit depth.** Each step requires an application of U or $U^\dagger$ plus a phase rotation. To implement a degree-d polynomial, $O(d)$ such steps are needed. Thus, depth scales linearly with the degree of the polynomial approximation.
- **Gate synthesis.** The phase rotations $R_z(\phi_j)$ may involve arbitrary angles, which require long T-gate sequences. The block-encoding U itself may be costly to implement, and QSVT requires many calls to it.

QSP and QSVT represent some of the most powerful primitives in the quantum algorithm designer's toolkit. They unify a wide range of problems under the same framework of polynomial transformations, and they achieve asymptotically optimal scaling for Hamiltonian simulation and linear system solving. However, their implementation requires deep circuits, many uses of block-encodings, and heavy non-Clifford overhead. This makes them unlikely to be realized in the early generations of error-corrected quantum devices.

4.2.5 Quantum Fourier Transform

The Quantum Fourier Transform (QFT) is one of the most fundamental primitives in quantum computation. It is the quantum analogue of the classical discrete Fourier transform and appears as a core subroutine in many algorithms, most famously in Shor's factoring algorithm [8] and in quantum phase estimation [77].

The Fourier transform is a central tool in classical algorithms, used to extract periodicity and frequency information. In the quantum setting, periodicity plays an equally important role. For example, Shor's algorithm reduces factoring to period finding, and QPE relies on interference patterns sharpened by a Fourier transform. The QFT provides the mechanism by which periodicity encoded in the amplitudes of a quantum state can be efficiently extracted

On n qubits, the QFT is the unitary operator defined by

$$\mathrm{QFT}_N : |x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i x y/N} |y\rangle, \tag{4.2.11}$$

where $N = 2^n$ and $x \in \{0, \ldots, N-1\}$. This transformation is the discrete Fourier transform applied to the amplitude vector of the state.

The QFT can be implemented exactly using a sequence of Hadamard gates and controlled phase rotations. On input $|x_{n-1}x_{n-2}\ldots x_0\rangle$, the QFT works by applying:

- A Hadamard on the most significant qubit.
- Controlled rotations $R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$ on lower qubits to encode the binary expansion of x in the phases.
- This pattern repeats for each qubit, producing the Fourier basis state.

Finally, the order of qubits is reversed (bit-reversal) to obtain the standard Fourier ordering [78].

The structure of an $n = 3$ QFT circuit is shown in Figure 4.4. Each qubit undergoes a Hadamard, followed by controlled rotations from qubits of lower significance.

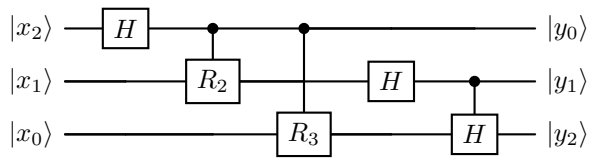


Figure 4.4: Quantum circuit for the 3-qubit QFT. Hadamards create superpositions, and controlled rotations R_k introduce the Fourier phases.

Resource analysis. The QFT is elegant but resource-heavy in practice:

- **Circuit depth.** The exact QFT on n qubits requires $O(n^2)$ two-qubit controlled rotations, yielding quadratic depth.
- **Gate synthesis.** Controlled-R_k gates involve rotations by angles as small as $2\pi/2^n$. Approximating these with Clifford+T gates requires long sequences, leading to large T-counts in a fault-tolerant setting.

The QFT is thus theoretically efficient and central to several quantum algorithms, but its exact implementation involves many small-angle rotations that are difficult to synthesize fault-tolerantly.

4.3 The Overhead of Quantum Error Correction

All of the resource estimates we have discussed so far assume that we can perform logical quantum gates with negligible noise. In reality, physical qubits are noisy, they decohere, gates suffer from control errors, and measurements are not precise. Typical error rates in state-of-the-art devices are in the range of 10^{-3} to 10^{-4} per gate, which is far too high to execute the deep circuits required by algorithms such as phase estimation, Hamiltonian simulation, or linear system solving. Without error correction, errors accumulate over thousands or millions of operations and completely overwhelm the computation. The quantum threshold theorem guarantees that scalable quantum computation is possible provided error rates are below a threshold and sufficient redundancy is used [79, 80].

Quantum error correction encodes one logical qubit into many physical qubits, in such a way that local errors can be detected and corrected. The most widely studied scheme is the *surface code*, which arranges physical qubits on a two-dimensional lattice and performs repeated stabilizer measurements to detect errors [81]. The surface code has a high threshold (around 1%), making it attractive for near-future devices. However, the cost of protection is large. To achieve a logical error rate p_L, one requires a code distance d such that $p_L \sim (p/p_{\text{th}})^{(d+1)/2}$, where p is the physical error rate and p_{th} the threshold. The number of physical qubits required scales as $O(d^2)$ per logical qubit, so that hundreds or thousands of physical qubits may be needed to reliably encode a single logical qubit. Alternative codes, such as concatenated codes or low-density parity-check (LDPC) codes, offer different tradeoffs in threshold, locality, and decoding complexity. Nevertheless, all known schemes introduce large overheads.

Not all logical gates are equally expensive. Clifford operations such as CNOT, Hadamard, and phase (S) gates can be implemented relatively easily in many codes, often transversally. By contrast, non-Clifford gates such as the T gate cannot generally be realized transversally. Instead, they must be implemented through *magic state distillation*, in which noisy resource states are purified using large stabilizer circuits [82]. Magic state distillation consumes vast numbers of additional physical qubits and logical gate operations. For many large-scale algorithms, especially those requiring long sequences of controlled rotations (e.g., in QSVT, QFT), the cost of T gates and their distillation dominates the overall resource estimate [83, 84].

Error correction also introduces temporal overhead. Each logical gate requires repeated rounds of stabilizer measurement and decoding to detect and correct errors. The cycle time for surface code error correction is set by measurement and classical decoding speeds.

These overheads have dramatic consequences for algorithm design. Resource estimates that appear reasonable in the logical model can become prohibitive once error correction is included. For example, even algorithms with a few hundred logical qubits and reasonable depth may demand millions of physical qubits and years of runtime when mapped to a fault-tolerant architecture. As a result, a central challenge in quantum algorithms is to minimize qubit counts, circuit depth, and especially the number of non-Clifford gates. This has led to the development of *early fault-tolerant algorithms*: approaches that are designed specifically to reduce error correction overhead, often by trading asymptotic efficiency for shallow depth or Clifford dominance.

4.4 Hamiltonian Simulation in the Fault-Tolerant Setting

We will use the Hamiltonian simulation problem to study how the primitives we have discussed can be used to implement quantum algorithms in the fault-tolerant setting. Simulating the dynamics generated by a Hamiltonian is one of the canonical tasks of quantum computing. Formally, given a Hamiltonian H, we aim to implement the time evolution operator e^{-iHt} on a quantum computer. This problem is ubiquitous as it underlies quantum chemistry, condensed matter physics, and other quantum algorithms as a subroutine. Over the years, many Hamiltonian simulation methods have been developed, but the *qubitization* method introduced by Low and Chuang [85] achieves asymptotically optimal scaling in all parameters.

The starting point of qubitization is the observation that if we have an $(\alpha, s, 0)$ block-encoding U of a Hamiltonian H, then one can embed the spectrum of H/α into the eigenvalues of a certain unitary walk operator. Explicitly, define the operator

$$W = (2|0^s\rangle\langle 0^s| \otimes I - I)(U), \tag{4.4.1}$$

which acts as a *quantum walk* whose eigenphases are related to the eigenvalues of H. By carefully interleaving applications of U and $U^\dagger$ with single-qubit rotations (the QSP sequence), one can implement polynomial transformations of the singular values of H/α. In particular, choosing the polynomial approximation to e^{-iHt} allows the exact simulation of time evolution up to precision ϵ.

Thus, the qubitization framework reduces Hamiltonian simulation to two primitive tasks:

1. Implementing a block-encoding of H (usually via LCU).
2. Applying Quantum Singular Value Transformation (QSVT) to approximate e^{-iHt}.

Both of these tasks we have already discussed, and both carry heavy costs.

At a high level, the qubitization circuit alternates between controlled applications of the block-encoding U (and its inverse $U^\dagger$) and single-qubit phase rotations on an ancilla, dictated by the QSP phase sequence. This is exactly the structure of QSVT, specialized to the exponential function. The full simulation requires a sequence of length proportional to the degree of the polynomial approximation, which scales as $O(\alpha t + \log(1/\epsilon))$.

4.4.1 Resource analysis

Although qubitization is optimal asymptotically, its implementation is extremely resource-intensive in the fault-tolerant setting:

1. Block-encoding overhead. To block-encode a general Hamiltonian $H = \sum_j a_j U_j$ with $\sum_j |a_j| \leq \alpha$, one requires:

- $s = \lceil \log M \rceil$ ancilla qubits to index terms.
- A prepare/unprepare pair for the amplitude distribution $\{a_j/\alpha\}$, which can be as costly as arbitrary state preparation.
- A multiplexed SELECT(U) operation applying U_j conditioned on the ancilla, which requires multi-controlled operations and large T-gate counts.

2. QSVT depth. To approximate e^{-iHt} to precision ϵ, QSVT requires a polynomial of degree $O(\alpha t + \log(1/\epsilon))$. Each step uses one call to U and one to $U^\dagger$, plus a single-qubit phase rotation.

3. Non-Clifford cost. The phase rotations $R_z(\phi_j)$ generally have arbitrary angles. Compiling these into Clifford+T gates requires long sequences, producing huge T-counts. Since T gates require magic state distillation in the surface code.

4. Qubit count. In addition to the s ancillas for block-encoding and one QSP ancilla, thousands of qubits may be needed to run parallel magic state factories for T-gates. Large-scale Hamiltonian simulation estimates often demand millions of physical qubits once error correction is included [83, 84].

Qubitization represents the optimal Hamiltonian simulation algorithm in the asymptotic sense, with linear scaling in αt and logarithmic scaling in precision. Yet, when viewed through the lens of fault tolerance, its costs are daunting: deep circuits, vast numbers of non-Clifford gates, and massive qubit counts for error correction and distillation. This makes large-scale Hamiltonian simulation one of the most resource-demanding targets for early quantum computers, and it motivates the search for alternative, low-depth primitives that may be realizable in the early fault-tolerant era.

4.5 Early Fault-Tolerant Algorithms

The algorithms discussed so far, such as qubitization and quantum singular value transformation, achieve optimal asymptotic scaling for Hamiltonian simulation, linear system solving, and related problems. However, these methods rely heavily on primitives like block-encoding, amplitude amplification, and long sequences of non-Clifford gates.

To bridge the gap between asymptotically optimal algorithms and practical realizability, we will explore Early Fault-Tolerant Algorithms. These are algorithms designed explicitly for the first era of fault-tolerant devices, when logical qubit counts will be limited, and circuit depth will be tightly constrained. Trying to keep the asymptotic efficiency, we will try on reducing overhead by keeping the number of ancilla qubits small, minimizing T-gate counts, and shortening circuit depth.

In what follows, we will survey some of the most promising approaches to early fault-tolerant algorithm design. We begin with Hamiltonian simulation via product formulas, a conceptually simple but resource-light approach. We then turn to *single-ancilla LCU algorithms*, which form the central primitive of this thesis and provide a powerful alternative to block-encoding-based methods. Finally, we highlight some of the recent proposals in the literature, such as iterative phase estimation, Hadamard-test-based Fourier transforms, and lightweight variants of QSVT, which further illustrate the diversity of techniques being explored for the early fault-tolerant regime.

4.5.1 Product Formulas

One of the earliest and most conceptually simple methods for Hamiltonian simulation is based on product formulas, sometimes called Trotter–Suzuki decompositions. In this setting, the Hamiltonian is assumed to be given as a linear combination of finitely many local Hermitian terms, often Pauli strings:

$$H = \sum_{j=1}^{M} h_j H_j, \qquad \|H_j\| \leq 1, \tag{4.5.1}$$

where the h_j are real coefficients and each H_j acts nontrivially on a constant number of qubits. This input model reflects the way Hamiltonians arise in quantum chemistry and condensed matter, where M may be large but each H_j is simple to implement.

Trotter–Suzuki product formulas. The idea of product formulas, introduced by Lloyd [45], is to approximate the full evolution e^{-iHt} by interleaving short evolutions under each term H_j. The first-order Trotter formula is

$$e^{-iHt} \approx \left(\prod_{j=1}^{M} e^{-ih_j H_j t/r} \right)^r, \tag{4.5.2}$$

where r is the number of Trotter steps. The error per step scales as $O\big((\Lambda t/r)^2\big)$, where $\Lambda = \sum_j |h_j|$, so to achieve accuracy ϵ one needs

$$r = O\left(\frac{(\Lambda t)^2}{\epsilon} \right). \tag{4.5.3}$$

Higher-order symmetric formulas, due to Suzuki [86], improve the scaling by canceling more error terms. The second-order formula,

$$S_2(t/r) = \prod_{j=1}^{M} e^{-ih_j H_j t/(2r)} \prod_{j=M}^{1} e^{-ih_j H_j t/(2r)}, \tag{4.5.4}$$

achieves error $O\big((\Lambda t/r)^3\big)$, requiring only

$$r = O\left(\frac{(\Lambda t)^{3/2}}{\epsilon^{1/2}} \right) \tag{4.5.5}$$

steps to reach accuracy ϵ. In general, a p-th order Suzuki formula has error scaling $O\big((\Lambda t/r)^{p+1}\big)$, giving

$$r = O\left(\frac{(\Lambda t)^{1+1/p}}{\epsilon^{1/p}} \right). \tag{4.5.6}$$

The cost per step grows with p, however, so in practice second- and fourth-order formulas often strike the best balance [87].

Randomized product formulas: qDRIFT. An alternative approach is to randomize the simulation. The qDRIFT algorithm of Campbell [47] simulates time t by applying N short unitaries, each chosen by randomly sampling a term H_j with probability $|h_j|/\Lambda$. Each unitary has the form

$$e^{-i\,\mathrm{sgn}(h_j)H_j\,\Lambda t/N}. \tag{4.5.7}$$

The expected error scales as

$$O\left(\frac{\Lambda^2 t^2}{N}\right), \tag{4.5.8}$$

so to achieve accuracy ϵ one requires

$$N = O\left(\frac{\Lambda^2 t^2}{\epsilon}\right). \tag{4.5.9}$$

The key advantage of qDRIFT is that its depth does not depend on M. Even if the Hamiltonian has millions of terms, the number of gates is set only by Λt and the target precision. This makes qDRIFT especially appealing in quantum chemistry applications with very large Hamiltonians.

Circuit depth and resource counts. The resource costs of product-formula methods can be summarized as follows:

- **Circuit depth.** A first-order Trotter formula requires $O(M(\Lambda t)^2/\epsilon)$ exponentials in total, since each of the $r = O((\Lambda t)^2/\epsilon)$ steps applies M exponentials. The second-order formula doubles the number of exponentials per step but requires fewer steps, yielding depth $O(M(\Lambda t)^{3/2}/\epsilon^{1/2})$. qDRIFT, by contrast, requires $O(\Lambda^2 t^2/\epsilon)$ exponentials in total, independent of M.
- **Gate decomposition.** Each $e^{-ih_j H_j \tau}$ is typically a Pauli rotation. Implementing it requires a short Clifford circuit plus one R_z rotation by angle $2h_j\tau$. Approximating this rotation to accuracy δ requires $O(\log(1/\delta))$ T gates.
- **Ancillas.** No extra ancilla qubits are required beyond those encoding the system. This is a major advantage compared to LCU or block-encoding methods.

Product formulas are not asymptotically optimal, but they are simple and ancilla-free. For local Hamiltonians, refined analyses that take commutator structure into account show that product formulas can achieve nearly optimal scaling [88, 89]. They remain among the most practical candidates for Hamiltonian simulation in the early fault-tolerant regime. The qDRIFT method further expands the toolkit by trading precision scaling for simplicity and M-independence, reducing circuit depth in settings where Hamiltonians are very large.

4.6 Single-Ancilla LCU Algorithm

The single-ancilla LCU algorithm is a variant of the standard LCU algorithm that uses a single ancilla qubit to implement a non-unitary operator A on a quantum system. The algorithm is based on the idea of using classical randomness to sample from the LCU decomposition of A and estimate expectation values through repeated measurements.

Consider a quantum system in an initial state ρ_0 and suppose we wish to estimate the expectation value of an observable O with respect to the state ρ prepared by applying an operator A. The quantity of interest is $\mathrm{Tr}[OA\rho_0 A^\dagger]$. Since A need not be an unitary operator, $A\rho_0 A^\dagger$ is not necessarily a valid density matrix. To make it a valid density matrix we need to normalize it:

$$\rho = \frac{A\rho_0 A^\dagger}{\mathrm{Tr}[A\rho_0 A^\dagger]} \tag{4.6.1}$$

Let $A = \sum_{j=1}^{M} \alpha_j U_j$ with $\alpha_j \geq 0$ and unitary U_j. We wish to estimate

$$\langle O\rangle = \frac{\mathrm{Tr}[OA\rho_0 A^\dagger]}{\mathrm{Tr}[A\rho_0 A^\dagger]} \tag{4.6.2}$$

for an initial state ρ_0 and observable O. Expanding gives

$$\langle O\rangle = \frac{\sum_{j,k} \alpha_j \alpha_k \,\mathrm{Tr}[OU_j\rho_0 U_k^\dagger]}{\sum_{j,k} \alpha_j \alpha_k \,\mathrm{Tr}[U_j\rho_0 U_k^\dagger]}. \tag{4.6.3}$$

Note that the quantities $\mathrm{Tr}[OU_j\rho_0U_k^\dagger]$ and $\mathrm{Tr}[U_j\rho_0U_k^\dagger]$ are just applying the unitaries U_j and U_k on the left and right of the initial state ρ_0 respectively and then tracing out with respect to the observable O or the identity operator $\mathbb{I}$. As we will see, this can be easily implemented using a simple quantum circuit with only a single ancilla qubit.

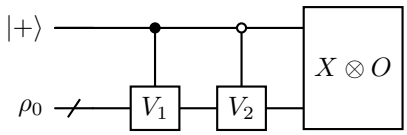


Figure 4.5: Quantum circuit for the single-ancilla LCU algorithm. The ancilla qubit is initialized in the mixed state $|+\rangle\langle+| = \frac{1}{2}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|)$, while the system register (represented as an n-qubit bundle) contains the initial state ρ_0. Two sampled unitaries from the LCU decomposition are applied: V_1 is controlled on the ancilla being in state $|1\rangle$, and V_2 is anti-controlled on the ancilla being in state $|0\rangle$. Finally, a joint POVM measurement of $X \otimes O$ is performed using a single measurement operator, where X denotes Pauli-X measurement on the ancilla and O is the observable of interest measured on the system. This simple circuit, repeated with different sampled unitaries V_1 and V_2, provides an unbiased estimator of the expectation value $\langle O\rangle$ while requiring only a single ancilla qubit and no multi-qubit controlled operations.

The algorithm samples pairs of unitaries (U_1, U_2) from the distribution $\Pr[U_j] = \alpha_j/\alpha$. For each sample, we prepare the ancilla in $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and apply controlled-U_1 and anti-controlled-$U_2^\dagger$. This creates the joint state

$$\sigma = \frac{1}{2}\big(|0\rangle\langle 0| \otimes U_2\rho_0U_2^\dagger + |0\rangle\langle 1| \otimes U_2\rho_0U_1^\dagger + |1\rangle\langle 0| \otimes U_1\rho_0U_2^\dagger + |1\rangle\langle 1| \otimes U_1\rho_0U_1^\dagger\big). \tag{4.6.4}$$

Measuring $X \otimes O$ yields the random variable $\mu_t = \frac{\alpha^2}{2}\mathrm{Tr}[O(U_1\rho_0U_2^\dagger + U_2\rho_0U_1^\dagger)]$, whose expectation over (U_1, U_2) gives

$$\mathbb{E}[\mu_t] = \mathrm{Tr}[OA\rho_0A^\dagger]. \tag{4.6.5}$$

Similarly, measuring $X \otimes I$ recovers the normalization $\mathrm{Tr}[A\rho_0A^\dagger]$. The ratio of these estimators provides the desired expectation value.

Algorithm 1 Single-Ancilla LCU Estimator. This algorithm estimates the expectation value $\mathrm{Tr}[OA\rho A^\dagger]$ with arbitratry precision ϵ and confidence $1-\delta$ where $A = \sum_{j=1}^M \alpha_j U_j$ is a linear combination of unitaries with real and positive coefficients.

Input: Initial quantum state ρ_0, observable O, operator A with LCU decomposition $\{(\alpha_j, U_j)\}_{j=1}^M$, number of repetitions T.

Output: Estimator $\hat{\mu}$ such that $|\hat{\mu} - \mathrm{Tr}[OA\rho A^\dagger]| \le \epsilon$ with probability $\ge 1-\delta$.

1: Compute $\alpha = \sum_{j=1}^M \alpha_j$.
2: Prepare the probability ensemble $\mathcal{D}_j = \left\{U_j, \frac{\alpha_j}{\alpha}\right\}$.
3: Set number of samples $T = \mathcal{O}(\|\alpha\|_1^4\|O\|^2\ln(1/\delta)/\epsilon^2)$
4: **for** $t = 1$ to T **do**
5: Independently sample the Unitaries U_1 and U_2 from the probability ensemble $\mathcal{D}_j$.
6: Initialize ancilla qubit in $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
7: Apply controlled unitary: $U_1^{(c)} = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes U_1$
8: Apply anti-controlled unitary: $U_2^{(a)} = |0\rangle\langle 0| \otimes U_2^\dagger + |1\rangle\langle 1| \otimes \mathbb{I}$
9: Perform POVM measurement of $X \otimes O$ on ancilla-system composite
10: Record the outcome μ_t
11: **end for**
12: Compute sample averages: $\mu = \frac{1}{T}\sum_{t=1}^T \mu_t$
13: **return** μ

Lemma 4.6.1 (Estimator Accuracy)**.** *Let $A = \sum_j \alpha_j U_j$ with $\alpha_j \ge 0$ and $\alpha = \sum_j \alpha_j$. If $\|O\| \le 1$, then for any $\epsilon, \delta \in (0,1)$, Algorithm 1 with*

$$T = \Theta\left(\frac{\alpha^4\ln(1/\delta)}{\epsilon^2}\right)$$

produces $\hat{\mu}$ such that

$$\Pr\big[\,|\hat{\mu} - \mathrm{Tr}[OA\rho_0 A^\dagger]| \le \epsilon\,\big] \ge 1-\delta.$$

Proof. The proof follows the analysis of the "single–ancilla LCU" estimator in [52].

Let U_1, U_2 be the two unitaries sampled in one run according to $\Pr[U_j] = \alpha_j/\alpha$. Conditioned on the choices (U_1, U_2) and the initial state ρ_0, the circuit prepares the state

$$\sigma \;=\; \frac{1}{2}\Big(|0\rangle\langle 0| \otimes U_2\rho_0 U_2^\dagger + |0\rangle\langle 1| \otimes U_2\rho_0 U_1^\dagger + |1\rangle\langle 0| \otimes U_1\rho_0 U_2^\dagger + |1\rangle\langle 1| \otimes U_1\rho_0 U_1^\dagger\Big).$$

Measuring $X \otimes O$ on σ therefore outputs the random variable

$$\mu_t \;=\; \mathrm{Tr}[\,(X\otimes O)\sigma] \;=\; \frac{1}{2}\mathrm{Tr}\big[O(U_1\rho_0 U_2^\dagger + U_2\rho_0 U_1^\dagger)\big].$$

Taking expectation over the independent choices U_1, U_2 yields

$$\mathbb{E}[\mu_t] = \frac{1}{\alpha^2}\,\mathrm{Tr}\big[OA\rho_0 A^\dagger\big],$$

where we used $\mathbb{E}[U_j] = \sum_j (\alpha_j/\alpha)U_j = A/\alpha$. Multiplying the circuit outcome by α^2 (absorbed into the definition of μ_t above) makes the estimator μ_t
unbiased for the desired quantity $\mathrm{Tr}[OA\rho_0 A^\dagger]$.

Each outcome satisfies the operator norm bound $|\mu_t| \le \alpha^2\,\|O\|$. Applying Hoeffding's inequality to the i.i.d. samples $\{\mu_t\}_{t=1}^T$ gives

$$\Pr\Big[\big|\mu - \mathbb{E}[\mu_t]\big| > \epsilon\Big] \;\le\; 2\exp\!\left(-\frac{2T\epsilon^2}{\alpha^4\|O\|^2}\right).$$

Choosing $T = \Theta\big(\alpha^4\|O\|^2\ln(1/\delta)/\epsilon^2\big)$ ensures the right–hand side is at most δ, establishing the claimed success probability.

Each coherent run uses one control qubit and two (controlled) applications of a unitary drawn from $\{U_j\}$, as is immediate from the circuit description, no additional ancillae or multi–controlled gates are required. □

This algorithm can be used inidividually to estimate both the numerator and the denominator (normalization term) with arbitrary precision. By choosing the appropriate precision for each estimator, we can control the overall error in the ratio estimator and compute the overall error.

Lemma 4.6.2 (Ratio estimator stability)**.** *Let $\mu = \mathrm{Tr}[OA\rho_0 A^\dagger]$ and $\nu = \mathrm{Tr}[A\rho_0 A^\dagger]$ with estimates $\hat{\mu}$ and $\hat{\nu}$. If*

$$|\hat{\mu}-\mu| \le \tfrac{\varepsilon}{3}\nu, \qquad |\hat{\nu}-\nu| \le \min\Big\{\tfrac{\nu}{2}, \tfrac{\varepsilon\nu}{6\|O\|}\Big\},$$

then $|\hat{\mu}/\hat{\nu} - \mu/\nu| \le \varepsilon$.

Proof. Using the triangle inequality,

$$\left|\frac{\hat{\mu}}{\hat{\nu}} - \frac{\mu}{\nu}\right| \le \frac{|\hat{\mu}-\mu|}{\nu} + \frac{|\mu|}{|\hat{\nu}|}\cdot\frac{|\hat{\nu}-\nu|}{\nu} \le \frac{\varepsilon}{3} + 2\|O\|\cdot\frac{\varepsilon}{6\|O\|} = \frac{2\varepsilon}{3} < \varepsilon,$$

where we used $|\mu| \le \|O\|\nu$. □

Together, Lemmas 4.6.1 and 4.6.2 enable us to estimate the required expectation value to arbitrary precision by appropriately setting the parameters for the numerator and the normalization. We will use this framework in the next subsection for Hamiltonian simulation. In that setting the evolution operator $U(t) = e^{-iHt}$ is unitary, so the normalization factor equals $Z = \mathrm{Tr}[U(t)\rho_0 U(t)^\dagger] = \mathrm{Tr}[\rho_0] = 1$, and no separate normalization estimate is needed.

4.6.1 Hamiltonian Simulation using Single-Ancilla LCU

Having established the single-ancilla LCU framework for general expectation value estimation, we now turn to its application: Hamiltonian simulation. The goal is to estimate expectation values of the form $\langle O\rangle = \mathrm{Tr}[Oe^{-i\tau H}\rho_0 e^{i\tau H}]$.

The key insight is to decompose the unitary evolution as a linear combination of simpler unitaries that can be efficiently implemented. For a Hamiltonian $H = \sum_{\ell=1}^L p_\ell P_\ell$ expressed as a convex combination of Pauli strings, we construct an LCU approximation $\widetilde{U} = \sum_j \alpha_j W_j$ by truncating the Taylor series of $e^{-i\tau H}$. This

approach, developed in [50, 51, 52], yields an approximation with $\|\widetilde{U} - e^{-i\tau H}\| \leq \varepsilon$ and manageable coefficient norm $\|\alpha\|_1 = O(e^{\tau^2/r})$ for an appropriate segmentation parameter r.

Each unitary W_j in the decomposition consists of r repetitions of a block containing $q = O(\log(r)/\log\log(r/\varepsilon))$ Clifford operations and a single Pauli rotation. This structure makes the unitaries well-suited for the near-term implementations.

Lemma 4.6.3 (LCU decomposition of time-evolution operator)**.** *Let $H = \sum_{\ell=1}^{L} p_\ell P_\ell$ be a Hamiltonian expressed as a convex combination of Pauli strings, with $p_\ell \geq 0$ and $\sum_\ell p_\ell = 1$. For any $\tau > 0$ and error tolerance $\varepsilon \in (0,1)$ there exists an LCU decomposition*

$$S = \sum_m \alpha_m W_m,$$

such that

$$\left\|e^{-i\tau H} - S\right\| \;\leq\; \varepsilon, \qquad \text{and} \qquad |\alpha| := \sum_m |\alpha_m| \;\leq\; e^{\tau^2/r}, \tag{4.6.6}$$

for some integer $r \geq \tau$ and polynomial truncation order $q = \mathcal{O}\Big(\frac{\log(r/\varepsilon)}{\log\log(r/\varepsilon)}\Big)$. Moreover, each unitary W_m is a product of r blocks, where each block takes the form

$$U_j = (-i)^k\, P_{\ell_1} \cdots P_{\ell_k}\, e^{i\theta_k P_m},$$

with $0 \leq k \leq q$ even, indices $\ell_1, \ldots, \ell_k, m \in \{1, \ldots, L\}$, and

$$\theta_k = \arccos\left(\left[1 + \left(\tfrac{\tau/r}{k+1}\right)^2\right]^{-1/2}\right).$$

Proof. We begin by writing the time-evolution operator as

$$e^{-i\tau H} \;=\; (S_r)^r, \qquad S_r := e^{-i(\tau/r)H}.$$

That is, we divide the total evolution into r short-time steps of size τ/r.

Expand S_r in a Taylor series and truncate at order q:

$$\widetilde{S}_r = \sum_{k=0}^{q} \frac{(-i\,\tau H/r)^k}{k!}.$$

It is known [42] that choosing

$$q = \mathcal{O}\left(\frac{\log(r/\varepsilon)}{\log\log(r/\varepsilon)}\right) \tag{4.6.7}$$

ensures $\|S_r - \widetilde{S}_r\| \leq \varepsilon/r$. Thus, $(\widetilde{S}_r)^r$ approximates $(S_r)^r = e^{-i\tau H}$ within error ε.

To rewrite $\widetilde{S}_r$ as an LCU, we use the identity (cf. [50]):

$$\sum_{k=0}^{q} \frac{(-iX)^k}{k!} \;=\; \sum_{\substack{k=0 \\ k\,\text{even}}}^{q} \frac{(-iX)^k}{k!}\Big(I - \frac{iX}{k+1}\Big), \tag{4.6.8}$$

valid for any operator X. Here we set $X = (\tau/r)H$.

Substituting $H = \sum_{\ell=1}^{L} p_\ell P_\ell$ and expanding gives

$$\widetilde{S}_r = \sum_{\substack{k=0 \\ k\,\text{even}}}^{q} \frac{(-i\tau/r)^k}{k!} \sum_{\ell_1,\ldots,\ell_k=1}^{L} p_{\ell_1} \cdots p_{\ell_k} P_{\ell_1} \cdots P_{\ell_k} \sum_{m=1}^{L} p_m \Big(I - \tfrac{i\tau}{r(k+1)} P_m\Big). \tag{4.6.9}$$

Note that

$$e^{i\theta_k P_m} = \frac{1}{\sqrt{1 + \left(\frac{\tau}{r(k+1)}\right)^2}}\Big(I - \tfrac{i\tau}{r(k+1)} P_m\Big),$$

with

$$\theta_k = \arccos\left(\left[1 + \left(\tfrac{\tau}{r(k+1)}\right)^2\right]^{-1/2}\right).$$

This allows us to factor each term as a product of Pauli strings and a single Pauli rotation:

$$U_j = (-i)^k \, P_{\ell_1} \cdots P_{\ell_k} e^{i\theta_k P_m},$$

with coefficient

$$c_j = \frac{(\tau/r)^k}{k!} \sqrt{1 + \left(\tfrac{\tau}{r(k+1)}\right)^2} \, p_{\ell_1} \cdots p_{\ell_k} p_m.$$

Thus

$$\widetilde{S}_r = \sum_{j \in M} c_j U_j,$$

where j encodes $(k, \ell_1, \ldots, \ell_k, m)$ and k is even.

Summing over coefficients,

$$\sum_{j \in M} |c_j| \leq \sum_{\substack{k=0 \\ k \text{ even}}}^{\infty} \frac{(\tau/r)^k}{k!} \sqrt{1 + \left(\tfrac{\tau}{r(k+1)}\right)^2} \tag{4.6.10}$$

$$\leq \sum_{k=0}^{\infty} \frac{(\tau/r)^{2k}}{k!} \tag{4.6.11}$$

$$= e^{\tau^2/r^2}. \tag{4.6.12}$$

Therefore, for the r-fold product,

$$|\alpha| = \sum_m |\alpha_m| \;\leq\; \Big(\sum_{j \in M} |c_j|\Big)^r \;\leq\; e^{\tau^2/r}.$$

Finally, we bound the approximation error:

$$\|(S_r)^r - \widetilde{S}_r^r\| \leq \sum_{s=0}^{r-1} \|S_r\|^{r-1-s} \, \|S_r - \widetilde{S}_r\| \, \|\widetilde{S}_r\|^s \tag{4.6.13}$$

$$\leq r \cdot (\varepsilon/r) \cdot e^{(r-1)\tau^2/r^2} \tag{4.6.14}$$

$$\leq \varepsilon \, e^{\tau^2/r - 1}. \tag{4.6.15}$$

Thus, choosing $r \geq \tau^2$ ensures $\|e^{-i\tau H} - S\| \leq \varepsilon$, completing the proof. □

We have constructed an explicit LCU $S = \sum_m \alpha_m W_m$ that ε-approximates $e^{-i\tau H}$, with total coefficient weight $|\alpha| \leq e^{\tau^2/r}$; in particular, taking $r \geq \tau^2$ yields $|\alpha| \leq e = \mathcal{O}(1)$. We can use this decomposition with the single-ancilla LCU estimator 1 to simulate time evolution.

Theorem 4.6.4. *Let $H = \sum_{\ell=1}^{L} p_\ell P_\ell$ be a Hamiltonian, such that $\sum_{\ell=1}^{L} p_\ell = 1$, ρ an initial state, O an observable, and $\tau > 0$. There exists a randomized procedure based on the single-ancilla LCU estimator that outputs μ such that*

$$\left|\mu - \mathrm{Tr}\left[O \, e^{-i\tau H} \rho \, e^{i\tau H}\right]\right| \;\leq\; \varepsilon \tag{4.6.16}$$

with probability at least $1 - \delta$, using

$$T = \mathcal{O}\left(\frac{\|O\|^2 \, \log(1/\delta)}{\varepsilon^2}\right) \tag{4.6.17}$$

runs of circuits, each of maximum depth

$$D_{\max} \;=\; \mathcal{O}\left(\tau^2 \, \frac{\log\big(\|O\|/\varepsilon\big)}{\log\log\big(\|O\|/\varepsilon\big)}\right). \tag{4.6.18}$$

Proof. Let $U := e^{-i\tau H}$. By Lemma 4.6.3, for any $\varepsilon_h > 0$ there exists an LCU representation

$$S = \sum_m \alpha_m W_m$$

such that

$$\|U - S\| \;\leq\; \varepsilon_h. \tag{4.6.19}$$

Applying the single-ancilla LCU estimator with

$$T = \mathcal{O}\left(\frac{\|O\|^2 \log(1/\delta)}{\varepsilon_s^2}\right)$$

samples produces an estimate μ satisfying

$$\left|\mu - \mathrm{Tr}[O\, S\rho S^\dagger]\right| \;\le\; \varepsilon_s \tag{4.6.20}$$

with probability at least $1-\delta$. Here we used Lemma 4.6.1, together with the fact that by Lemma 4.6.3 we can take $r = \Theta(\tau^2)$ so that $|\alpha| \le e = \mathcal{O}(1)$. This suppresses the constant α^4 factor in the sample complexity.

We bound the difference:

$$\begin{aligned}
\left|\mathrm{Tr}[O\, S\rho S^\dagger] - \mathrm{Tr}[O\, U\rho U^\dagger]\right| &\le \|O\| \left\|S\rho S^\dagger - U\rho U^\dagger\right\|_1 && (4.6.21)\\
&\le \|O\| \Big(\left\|(S-U)\rho S^\dagger\right\|_1 + \left\|U\rho(S^\dagger - U^\dagger)\right\|_1 \Big) && (4.6.22)\\
&\le \|O\| \Big(\|S-U\|\,\|S^\dagger\| + \|U\|\,\|S-U\| \Big). && (4.6.23)
\end{aligned}$$

Using $\|X\rho Y\|_1 \le \|X\|\,\|\rho\|_1\,\|Y\| = \|X\|\,\|Y\|$, $\|U\| = 1$, and $\|S^\dagger\| \le \|S-U\| + \|U\| \le 1 + \varepsilon_h$, we obtain

$$\left|\mathrm{Tr}[O\, S\rho S^\dagger] - \mathrm{Tr}[O\, U\rho U^\dagger]\right| \;\le\; 3\,\|O\|\,\varepsilon_h. \tag{4.6.24}$$

Combining (4.6.20) with the above bound, and applying the triangle inequality,

$$\left|\mu - \mathrm{Tr}[O\, U\rho U^\dagger]\right| \;\le\; \varepsilon_s + 3\,\|O\|\,\varepsilon_h.$$

Choosing $\varepsilon_s = \varepsilon/2$ and $\varepsilon_h = \varepsilon/(6\,\|O\|)$ ensures the right-hand side is $\le \varepsilon$, with success probability at least $1-\delta$.

Each W_m is a product of r blocks, where each block consists of q Pauli unitaries and one Pauli rotation (Lemma 4.6.3). Thus, the maximum depth per circuit run is

$$D_{\max} = \mathcal{O}(rq).$$

With $r = \Theta(\tau^2)$ and $q = \Theta(\log(r/\varepsilon_h)/\log\log(r/\varepsilon_h))$, this simplifies to

$$D_{\max} = \mathcal{O}\left(\tau^2\, \frac{\log(\|O\|/\varepsilon)}{\log\log(\|O\|/\varepsilon)}\right),$$

as claimed. □

This provides an end-to-end guarantee for estimating time-evolved expectation values with an error ε and confidence $1-\delta$ using the single-ancilla LCU method. It achieves near-optimal sample complexity $T = \mathcal{O}(\|O\|^2 \log(1/\delta)/\varepsilon^2)$ and per-run circuit depth $D_{\max} = \mathcal{O}\big(\tau^2\, \log(\|O\|/\varepsilon)/\log\log(\|O\|/\varepsilon)\big)$, while using only a single ancilla qubit.

4.6.2 Resource Analysis and Comparison with Qubitization

The single-ancilla LCU algorithm achieves Hamiltonian simulation with guarantees on accuracy and circuit depth, while requiring minimal qubit overhead. A key advantage is its logarithmic dependence on the target accuracy ε. The circuit depth scales as $O(\log(1/\varepsilon))$, which is exponentially better than the polynomial scaling $O(1/\varepsilon)$ of product formulas.

We now summarize its complete resource profile and compare it with both product formulas and the qubitization method of [85], which is the asymptotically optimal approach.

The single-ancilla method requires *one additional qubit* beyond those needed to represent the system and observable. In contrast, qubitization requires $\lceil \log L \rceil$ ancilla qubits to encode indices of the terms in the Hamiltonian, plus additional ancillas to implement the PREPARE and SELECT unitaries that form the block-encoding. This difference is substantial in the early fault-tolerant regime, where every logical qubit carries the overhead of thousands of physical qubits.

By Theorem 4.6.4, the depth per circuit run in the single-ancilla approach is

$$D_{\max} = \mathcal{O}\left(\tau^2\, \frac{\log(\|O\|/\varepsilon)}{\log\log(\|O\|/\varepsilon)}\right). \tag{4.6.25}$$

The crucial feature is the logarithmic dependence on $1/\varepsilon$, which represents an exponential improvement over product formulas.

While the scaling is quadratic in evolution time τ, no amplitude amplification is needed, and each run involves only two controlled applications of Pauli unitaries and one Pauli rotation. For moderate evolution times and high-precision requirements, this logarithmic advantage can more than compensate for the quadratic time dependence.

In fault-tolerant architectures, the dominant cost is often the number of T gates. For the single-ancilla method, each run requires synthesizing only r rotations. This results in T-counts scaling as

$$\mathcal{O}(\tau^2)\,.$$

By contrast, qubitization requires repeated block-encoded unitaries, amplitude amplification, and long sequences of arbitrary-angle rotations, leading to T-counts of order

$$\mathcal{O}\big(\tau\,\mathrm{polylog}(1/\varepsilon)\big),$$

but with very large hidden constants due to magic state distillation.

Because the single-ancilla approach is Monte Carlo in nature, one must repeat the circuit

$$T = \mathcal{O}\left(\frac{\|O\|^2 \log(1/\delta)}{\varepsilon^2}\right) \tag{4.6.26}$$

times to estimate expectation values with precision ε Which is same as qubitization when used to estimate expectation values. Nevertheless, the runs are shallow and qubit-efficient, making this overhead tolerable in the early fault-tolerant era.

Qubitization is asymptotically optimal in both τ and ε, achieving near-linear scaling in τ and polylogarithmic dependence on $1/\varepsilon$. However, its reliance on block-encodings, multi-ancilla state preparation, and amplitude amplification renders it prohibitively costly on near-term error-corrected hardware. The single-ancilla LCU algorithm, by contrast, trades away asymptotic optimality in favor of simplicity as it requires only one ancilla qubit, uses far fewer T gates, and produces circuits of modest depth. This makes it a compelling candidate for *early fault-tolerant Hamiltonian simulation*, when logical qubits will be scarce and magic state factories expensive. This illustrates the broader philosophy of early fault-tolerant algorithms: practical implementability with limited qubits and modest depth is often more valuable than optimal asymptotic scaling.

4.7 Further Literature on Early Fault-Tolerance

Early fault-tolerant quantum computing (EFTQC) aims to extract useful speedups with *few logical qubits, modest T-counts, and shallow circuits.* Across primitives such as phase estimation, Hamiltonian simulation, or state preparation, a common design pattern emerges: replace heavy block-encodings, large ancilla registers, and amplitude amplification with sampling, classical post-processing, and simple circuits. Below we survey representative lines of work that embody this philosophy.

Phase Estimation: Traditional quantum phase estimation requires many ancilla qubits and deep quantum Fourier transforms. Early fault-tolerant variants take a different approach. It uses short-time evolutions with random sampling, then perform classical post-processing to extract phase information [50, 90]. This randomized approach dramatically reduces ancilla requirements and circuit depth. Instead of building complex coherent superpositions, the algorithm runs many simple circuits and uses classical statistics to estimate phases. Recent work shows this can be more practical than standard QPE on near-term fault-tolerant devices [91, 92].

Hamiltonian Simulation: Product formulas like Suzuki–Trotter and qDRIFT remain strong baselines in the EFT regime because they don't need any ancilla's and compile to simple Pauli rotations. Single-ancilla LCU methods, as developed in this thesis, provide another approach that uses minimal ancilla overhead.

Recent work has developed interpolation and extrapolation techniques to improve product formulas by using classical post-processing. These methods run Trotter circuits at multiple step sizes, then use classical polynomial interpolation or Richardson extrapolation to cancel low order error terms and achieve polylogarithmic dependence on accuracy ε. The quantum circuits remain simple product formulas, while still achieving near-optimal scaling of $O(T^{1+1/p}\mathrm{polylog}(1/\varepsilon))$ for p-th order formulas, providing exponential improvement in precision over standard Trotter methods.

QSVT: Quantum singular value transformation (QSVT) typically requires complex block encodings with many ancilla qubits. Recent work shows how to implement QSVT using only basic Hamiltonian simulation primitives like Trotter steps, achieving near-optimal complexity with minimal ancilla overhead [93]. This makes QSVT accessible for early fault-tolerant devices.

Error Correction: Beyond algorithmic improvements, early fault-tolerant quantum computing also benefits from tailored error correction schemes and compilation strategies. Akahoshi et al. [94] developed the STAR (Space-Time efficient Analog Rotation) architecture, which combines fault-tolerant Clifford gates with direct analog rotation gates to avoid expensive magic state distillation. Their approach uses parallel injection protocols and adaptive compilation techniques to reduce the overhead of implementing Trotter-based time evolution, achieving over $10\times$ speedup compared to naive serial compilation. Other work focuses on lightweight decoders, exploiting biased noise, and combining error correction with error mitigation techniques [95]. The goal is to minimize the number of physical qubits per logical qubit when only limited resources are available, complementing the algorithmic simplifications.

Surveys and benchmarks: Furthermore, several works propose *methodologies* and case studies to evaluate EFT viability, blending partial error correction with algorithm design choices. These include full-stack analyses of ground-state energy estimation, spectral CDF estimation, and modeling frameworks that tie algorithm knobs to logical noise rates, code distances, and factory provisioning [90, 96, 92]. Such studies are crucial for identifying the true EFT "sweet spots," where ancilla counts and T-factory loads dominate feasibility more than asymptotic query complexity.

Chapter 5

Simulating Markovian Collision Models

Chapter Summary

In this chapter we presents a complete, end-to-end algorithms for the simulation of open quantum systems, leveraging the collision model framework introduced in Chapter 3. The primary goal is to develop a computational scheme that is efficient and practical for early fault-tolerant quantum computers. To this end, we will employ the single-ancilla Linear Combination of Unitaries (LCU) technique as our core subroutine, which minimizes the resource requirements, particularly the number of ancilla qubits and the maximum circuit depth.

We begin by detailing the general algorithm for simulating any memoryless (Markovian) collision process, providing a formal definition of a K-collision map and design a single-ancilla, randomized algorithm based on the SA-LCU method that implements the sequence of collisions while avoiding exponential blow-up in composition. We prove correctness and derive complexity bounds. Then, we compare near-term Hamiltonian procedures highlighting trade-offs in circuit depth, ancilla qubits, and repetitions. As an application, we instantiate the framework to simulate Lindbladian dynamics via repeated collisions with thermal sub-environments, deriving circuit depth bounds and identifying regimes where SA-LCU or Trotter methods are preferable. We then demonstrate its practical utility with a numerical benchmark on the transverse-field Ising model with amplitude damping noise. The chapter thus provides a unified and practical toolkit for near-term simulation of Markovian open-system dynamics.

5.1 Markovian Collision Models

In this section, we develop a general algorithm for simulating Markovian collision models. The core idea is to decompose the total evolution into a sequence of discrete collisions, each of which can be efficiently implemented on a quantum computer. We will formalize the notion of a Markovian K-collision map and then present a randomized algorithm based on the SA-LCU method to simulate it.

5.1.1 The Markovian K-Collision Map

Let us consider a quantum system in an n-qubit Hilbert space $\mathcal{H}_S$ that is coupled to a quantum environment belonging to the space $\mathcal{H}_E$, formed out of the tensor product of m environment subsystems: $\mathcal{H}_E = \mathcal{H}_{E_1} \otimes \mathcal{H}_{E_1} \cdots \otimes \mathcal{H}_{E_m}$. We denote the system Hamiltonian by H_S. The environment is a discrete sum of environment subsystems, with the Hamiltonian of the sub-environment j denoted by H_{E_j}. Each sub-environment sequentially couples to the system through an interaction Hamiltonian H_{I_j}. Thus the Hamiltonian corresponding to the j-th collision is given by

$$H_j = H_S + H_{I_j} + H_{E_j}. \tag{5.1.1}$$

We assume that the total Hamiltonian corresponding to the j-th collision can be expressed as a Linear combination of Pauli operators, i.e.,

$$H_j = \sum_{i=1}^{L_j} h_{i,j} P_{i,j}, \tag{5.1.2}$$

where $P_{i,j}$ belongs to the n-qubit Pauli group $\mathcal{P}_n := \{\omega\, P : \omega \in \{\pm 1\},\ P \in \{I, \sigma^x, \sigma^y, \sigma^z\}^{\otimes n}\}$. We, without loss of generality, assume each $h_{i,j} \in \mathbb{R}^+$ by absoring the sign (-1) into $P_{i,j}$. Also, let $L = \max_j L_j$. Finally,

for $1 \leq j \leq K$, we denote the normalized Hamiltonians $\overline{H}_j = H_j/\beta_j$, where $\beta_j = \sum_{i=1}^{L_j} h_{i,j}$. This normalization is also without loss of generality, as we can always rescale the evoltion time by β_j.

Now, we are in a position to describe the collision model dynamics. We assume the system to be in state represented by ρ_S. The system evolves by repeated collisions. In each collision, the system interacts with the j-th sub-environment, initialized in the state ρ_{E_j}, for a duration of Δt. The dynamics induced by the j-th collision is then given by the time evolution operator

$$U_j = e^{-i\beta_j \overline{H}_j \Delta t} \tag{5.1.3}$$

Following this, the sub-environment is traced out. For each collision a new sub-environment is initialized in state which is independent and seperable from the previous sub-environment and the state of the system, making it a Markovian process. Formally, we define a Markovian collision map as follows:

Definition 5.1.1 (Markovian collision map)**.** *Let $j \in [1, K]$, ρ_{E_j} represent the initial state of the j-th sub-environment, and U_j be the unitary representing the interaction between the system and the j-th sub-environment. The j-th collision map Φ_j is defined as:*

$$\Phi_j[.] \equiv \mathrm{Tr}_{E_j}\left[U_j\left(. \otimes \rho_{E_j}\right) U_j^\dagger\right] \tag{5.1.4}$$

Here, Tr_{E_j} denotes the partial trace over the j-th sub-environment, resulting in a reduced state for the system.

For multiple collisions, this process is repeated. The resulting iterative sequence for K collisions can be described by composing K such collision maps. Formally, we define a Markovian K-collision map as follows:

Definition 5.1.2 (Markovian K-collision map)**.** *Let Φ_1, Φ_2, ..., Φ_K be the collision maps from Definition 5.1.1. The K-collision map $\mathcal{M}_K$ is the composition of these maps, defined as:*

$$\mathcal{M}_K[\cdot] \equiv \left[\bigcirc_{j=1}^{K} \Phi_j[\cdot]\right]. \tag{5.1.5}$$

Our goal is to develop a randomized quantum algorithm that estimates the expectation value of an observable O for a system (initialized in ρ_S) after K collisions, up to an additive precision ε. That is, the algorithm outputs μ such that

$$|\mu - \mathrm{Tr}\left[O\mathcal{M}_K\left[\rho_S\right]\right]| \leq \varepsilon. \tag{5.1.6}$$

Each collision involves a time evolution of the joint system sub-environment state for some time Δt and a tracing out. Thus, the Hamiltonian simulation is the key subroutine in simulating a Markovian K-collision map on a quantum computer. In what follows, we estimate the precision required of any Hamiltonian simulation procedure to estimate μ with ε-additive accuracy. Let us denote by $\widetilde{U}_j$, the circuit corresponding to the Hamiltonian simulation procedure implementing $e^{-i\Delta t \overline{H}_j}$ to some accuracy (to be determined later). Then, it is possible to define approximate versions of the collision maps from Definitions 5.1.1 and 5.1.2, respectively, where the exact time evolution operator U_j is now replaced by $\bar{U}_j$:

$$\widetilde{\Phi}_j[\cdot] \equiv \mathrm{Tr}_{E_j}\left[\widetilde{U}_j\left(\cdot \otimes \rho_{E_j}\right)\widetilde{U}_j^\dagger\right], \tag{5.1.7}$$

and subsequently, the approximate K-Collision map is defined as

$$\widetilde{\mathcal{M}}_K[\cdot] \equiv \left[\bigcirc_{j=1}^{K} \widetilde{\Phi}_j[\cdot]\right]. \tag{5.1.8}$$

We now prove that if the Hamiltonian simulation procedures in the construction of the approximate Markovian K-collision map are implemented with a precision of $\varepsilon/(3K\|O\|)$, then the expectation value of observable O with respect to the state of the system with respect to the map $\widetilde{M}_K$ is ε-close to the desired expectation value. We do so via the following lemma:

Lemma 5.1.3 (Bounds on the Markovian collision map)**.** *Let O be an observable and $\widetilde{\mathcal{M}}_K$ represent the approximate Markovian K-collision map in Eq. (5.1.8), where*

$$\max_{1\leq j\leq K} \left\|U_j - \widetilde{U}_j\right\| \leq \frac{\varepsilon}{3K\|O\|}. \tag{5.1.9}$$

Then, the expectation value of O over a state evolved under the approximate map satisfies,

$$\left|\mathrm{Tr}\left[O\mathcal{M}_K[\rho]\right] - \mathrm{Tr}\left[O\widetilde{\mathcal{M}}_K[\rho]\right]\right| \leq \varepsilon. \tag{5.1.10}$$

Proof. Consider the error between the operations performed on the state ρ under U_j and $\widetilde{U}_j$. More precisely, let $\left\|U_j - \widetilde{U}_j\right\| \leq \xi_j$, where we denote the maximum error in any of the Hamiltonian simulation procedures in the definition of the approximate K-collision map by $\xi_{\max}$, i.e., $\xi_{\max} = \max_{1\leq j\leq K} \xi_j$. Then, using Theorem A.1.2, we have

$$\left\|U_j \rho U_j^\dagger - \widetilde{U}_j \rho \widetilde{U}_j^\dagger\right\|_1 \leq 3\xi_j, \tag{5.1.11}$$

for any quantum state ρ. Since partial trace is a CPTP operation, it is contractive under the trace norm. Therefore, we obtain,

$$\left\|\mathrm{Tr}_{E_j}\left[U_j \rho U_j^\dagger\right] - \mathrm{Tr}_{E_j}\left[\widetilde{U}_j \rho \widetilde{U}_j^\dagger\right]\right\|_1 \leq 3\xi_j. \tag{5.1.12}$$

This inequality implies that the trace distance between the exact map $\Phi_j[\rho]$ and the approximate map $\widetilde{\Phi}_j[\rho]$ is bounded by $3\xi_j$:

$$\left\|\Phi_j[\rho] - \widetilde{\Phi}_j[\rho]\right\|_1 \leq 3\xi_j. \tag{5.1.13}$$

So,

$$\max_j \left\|\Phi_j[\rho] - \widetilde{\Phi}_j[\rho]\right\|_1 \leq 3\xi_{\max}. \tag{5.1.14}$$

Now, from Lemma A.1.3, we obtain,

$$\left\|\bigcirc_{j=1}^K \Phi_j[\rho] - \bigcirc_{j=1}^K \widetilde{\Phi}_j[\rho]\right\|_1 \leq 3K\xi_{\max}. \tag{5.1.15}$$

Thus, we have

$$\left\|\mathcal{M}_K[\rho] - \widetilde{\mathcal{M}}_K[\rho]\right\|_1 \leq 3K\xi_{\max}. \tag{5.1.16}$$

Finally, we apply Theorem A.1.2 once again to bound the difference in the expectation values of the observable O under $\mathcal{M}_K$ and $\widetilde{\mathcal{M}}_K$:

$$\left|\mathrm{Tr}\left[O\mathcal{M}_K[\rho]\right] - \mathrm{Tr}\left[O\widetilde{\mathcal{M}}_K[\rho]\right]\right| \leq 3K\|O\|\xi_{\max}. \tag{5.1.17}$$

Therefore, by choosing $\xi_{\max} = \varepsilon/(3K\|O\|)$, we can ensure that

$$\left|\mathrm{Tr}\left[O\mathcal{M}_K[\rho]\right] - \mathrm{Tr}\left[O\widetilde{\mathcal{M}}_K[\rho]\right]\right| \leq \varepsilon. \tag{5.1.18}$$

□

Accordingly, any Hamiltonian simulation algorithm with error at most $\varepsilon/(3K\|O\|)$ suffices to implement the Markovian K-collision map. We can compare the complexities of using different Hamiltonian simulation algorithms to implement a Markovian K-collision map on a quantum computer. As mentioned earlier, we will primarily focus on near-term Hamiltonian simulation techniques, i.e., ones that do not require multiple ancilla qubits. We will consider Trotter methods, qDRIFT, and also the SA-LCU method of Ref. [52]. We need to estimate the cost of composing K Hamiltonian simulation algorithms, each implementing $e^{-i\overline{H}\Delta t}$ to a precision $\varepsilon/(3K\|O\|)$, from Lemma 5.1.3. So, if a Hamiltonian simulation algorithm requires a circuit depth of

$$\tau\left(\Delta t,\ \varepsilon/(K\|O\|)\right),$$

to implement U_j, up to an additive precision of $\mathcal{O}(\varepsilon/(K\|O\|))$, the K-collision map can be implemented within a circuit depth of

$$\tau_d = \mathcal{O}\left(K\tau\left(\Delta t,\ \varepsilon/(K\|O\|)\right) + K\tau_{\rho_E}\right),$$

where τ_{ρ_E} is the maximum of the circuit depths of the unitaries preparing the sub-environments, i.e., if for $j \in [1, K]$, if τ_{E_j} is the circuit depth of preparing ρ_{E_j}, then $\tau_{\rho_E} = \max_{j\in[1,K]} \tau_{\rho_{E_j}}$. While any near-term Hamiltonian simulation procedure can be used to implement a Markovian K-collision map, the SA-LCU algorithm [52] demands some attention. This method, described in the next section, expresses each U_j as a linear combination of strings of Pauli operators (say $\widetilde{U}_j$) with the total weight of the coefficients $\alpha = \mathcal{O}(1)$. Now, composing individual Hamiltonian simulations for implementing the Markovian K-collision map can potentially lead to an exponential scaling α^K, detrimentally affecting the circuit depth. However, below, we show that it is possible to implement the Markovian K-collision map while bypassing this exponential scaling.

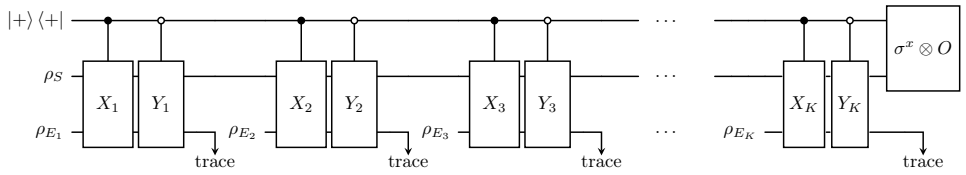


Figure 5.1: The quantum circuit for simulating a K-collision map using Hamiltonian simulation by SA-LCU. The algorithm applies controlled and anti-controlled sampled unitaries (X_j and Y_j, respectively) for the interaction between the system and each sub-environment, following which the latter is traced out. This sequence is repeated K times, corresponding to the K collisions. At the end of the process, the ancilla qubit and the system are measured. Notably, only a single environment register suffices, as it can be reused following the tracing out of the previous environment subsystem.

5.1.2 A Randomized Simulation Algorithm

To simulate the Markovian K-collision map, we employ the single-ancilla LCU algorithm. The key idea is to express each collision operator U_j as a linear combination of unitaries, $U_j = \sum_k \alpha_{j,k} W_{j,k}$, where the $W_{j,k}$ are Pauli strings. SA-LCU lets us implement U_j without deeply nested multi-qubit-controlled gates. We further adapt SA-LCU to apply all K interactions sequentially and defer the measurement to the end. A naive composition would suffer an exponential blow-up in the number of circuit runs; we avoid this by appropriately scaling the (Trotter) step size, keeping the overall sample complexity under control.

Each run of the algorithm involves running the quantum circuit shown in Fig. 5.1. The outcome of this circuit is a random variable that, in expectation value, estimates the desired quantity. The overall algorithm (outlined in Algorithm 2) involves running this circuit a total of T times and estimating the sample average of the outcomes.

We require three registers: the system register, the environment register, and a single qubit ancilla register. The system register is initialized in the state ρ_S; the environment registers in the state ρ_{E_1}, and the ancilla qubit in the state $|+\rangle$.

Let us first discuss the implementation of the approximate collision map $\widetilde{\Phi}_j$, corresponding to the j-th collision. We recall that the normalized Hamiltonian for the j-th collision $\overline{H}_j$ ($\|\overline{H}_j\| = 1$) can be expressed as a convex combination of strings of Pauli operators, i.e., $\overline{H}_j = \sum_{k=1}^{L} p_{jk} P_{jk}$. Then, we use the Lemma 4.6.3 to find the LCU decomposition of $U_j = e^{-i\beta_j \Delta t \overline{H}_j}$ such that for any $j \in [1, K]$,

$$\widetilde{U}_j = \sum_k \alpha_{jk} W_{jk}, \tag{5.1.19}$$

and

$$\left\| U_j - \widetilde{U}_j \right\| \leq \frac{\varepsilon}{6K\|O\|},$$

Note that each W_{ji} is a string of q Pauli operators and a single Pauli rotation, repeated r_j (to be decided later) times, where

$$q = \mathcal{O}\left(\frac{\log\left(r_j K \|O\|/\varepsilon\right)}{\log\log\left(r_j K \|O\|/\varepsilon\right)} \right), \tag{5.1.20}$$

and $\alpha^{(j)} = \sum_k \alpha_{jk} \leq e^{-\beta_j^2 \Delta t^2 / r_j^2}$.

Next, we draw two independent and identically distributed (i.i.d.) samples, X_j and Y_j, from the ensemble

$$\mathcal{D}_j = \left\{ W_{jk}, \frac{\alpha_{jk}}{\alpha^{(j)}} \right\}. \tag{5.1.21}$$

Note that $\mathbb{E}[X_j] = \mathbb{E}[Y_j] = \widetilde{U}_j / \alpha^{(j)}$. Then, we coherently apply the controlled version $X_j^{(c)}$ and the anti-controlled version $Y_j^{(a)}$ of these sampled unitaries:

$$X_j^{(c)} = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes X_j, \tag{5.1.22}$$

$$Y_j^{(a)} = |0\rangle\langle 0| \otimes Y_j + |1\rangle\langle 1| \otimes \mathbb{I}. \tag{5.1.23}$$

After applying $X_j^{(c)}$ and $Y_j^{(a)}$, we trace out the environment register, which can be reused for the next collision.

Algorithm 2 Algorithm to estimate the expectation value of an observable O with respect to a quantum state evolved under a K-collision map.

Input: Initial system state ρ_S; sub-environment states $\rho_{E_1}, \ldots, \rho_{E_K}$; observable O; unitaries $\widetilde{U}_1, \ldots, \widetilde{U}_K$ with LCU decompositions $\widetilde{U}_j = \sum_k \alpha_{jk} W_{jk}$ satisfying $\|\widetilde{U}_j - e^{-i\Delta t \beta_j \overline{H}_j}\| \leq \varepsilon'$ for all $j \in [K]$; desired additive error ε and failure probability δ.

Output: Estimator $\hat{\mu}$ such that $|\hat{\mu} - \mathrm{Tr}[O\,\mathcal{M}_K[\rho_S]]| \leq \varepsilon$ with probability $\geq 1-\delta$.

Compute $\alpha^{(j)} \leftarrow \sum_k \alpha_{jk}$ for each j and $\zeta \leftarrow \prod_{j=1}^K \alpha^{(j)}$.
Choose sample size T.
for $t = 1$ to T **do**
    Prepare $\rho_S \otimes |+\rangle$ on system-ancilla registers.
    for $j = 1$ to K **do**
        Prepare environment register in ρ_{E_j}.
        Independently sample unitaries X_j, Y_j from the ensemble $\mathcal{D}_j = \{(W_{jk}, \alpha_{jk}/\alpha^{(j)})\}$.
        Apply controlled unitary $X_j^{(c)} = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes X_j$.
        Apply anti-controlled unitary $Y_j^{(a)} = |0\rangle\langle 0| \otimes Y_j + |1\rangle\langle 1| \otimes \mathbb{I}$.
        Trace out the environment register.
    end for
    Measure $\sigma^x \otimes O$ obtaining outcome μ_t.
end for
$\hat{\mu} \leftarrow \frac{\zeta^2}{T} \sum_{t=1}^T \mu_t$.
return $\hat{\mu}$.

We repeat these steps for $j = 1$ to K to implement the approximate Markovian K-collision map, $\widetilde{\mathcal{M}}_K$. Then, we measure the observable $\sigma^x \otimes O$ in the ancilla and the system register. This corresponds to a single run of the algorithm. The outcome of the j-th run (for any $j \in [1, T]$) is a random variable μ_j whose expectation value is given as,

$$\mathbb{E}[\mu_j] = \mathrm{Tr}[O\ \mathcal{M}_K[\rho_S]]/\zeta^2,$$

where $\zeta = \prod_{j=1}^K \alpha^{(j)}$. Overall, by taking some T repetitions of this procedure, we collect random variables $\{\mu_j\}_{j=1}^T$, such that

$$\mu = \frac{\zeta^2}{T} \sum_{j=1}^T \mu_j,$$

approximates the desired expectation value within ε-additive accuracy with a success probability of at least $1-\delta$. We prove the validity of Algorithm 2 with the following Theorem.

Theorem 5.1.4. *Let $\varepsilon, \delta \in (0,1)$. Then, for $\varepsilon' = \varepsilon/(6K\|O\|)$, Algorithm 2 outputs μ with at least $1-\delta$ probability such that*

$$|\mu - \mathrm{Tr}[O\ \mathcal{M}_K[\rho_S]]| \leq \varepsilon,$$

using T repetitions of the circuit shown in Figure 5.1 where

$$T = \mathcal{O}\left(\frac{\|O\|^2 \log(1/\delta)}{\varepsilon^2}\right). \tag{5.1.24}$$

Each such coherent run has a circuit depth of

$$\tau_d = \mathcal{O}\left(\beta^2 K^2 \Delta t^2 \frac{\log(\beta K \|O\| \Delta t/\varepsilon)}{\log\log(\beta K \|O\| \Delta t/\varepsilon)} + K\tau_{\rho_E}\right). \tag{5.1.25}$$

Here, $\beta = \max_j \beta_j$ and $\tau_{\rho_E} = \max_j \tau_{\rho_{E_j}}$, where $\tau_{\rho_{E_j}}$ is the circuit depth of the unitary preparing the sub-environment in the state ρ_{E_j}.

Proof. Following Algorithm 2, we initialize the ancilla and system registers in the state

$$\rho_0 = |+\rangle\langle +| \otimes \rho_S \tag{5.1.26}$$

and the environment register in the state ρ_{E_1}. As shown in Fig. 5.1, we implement $X_1^{(c)}$ and $Y_1^{(a)}$ by sampling X_1 and Y_1 from $\mathcal{D}_1$, followed by tracing out of the environment E_1. The resulting state is

$$\rho_1 = \mathrm{Tr}_{E_1}\left[Y_1^{(a)} X_1^{(c)} (\rho_0 \otimes \rho_{E_1}) X_1^{(c)\dagger} Y_1^{(a)\dagger}\right] \tag{5.1.27}$$

For simplicity, let us define the map $\Phi_j^{(PQ)}$ as

$$\Phi_j^{(PQ)}[.] = \mathrm{Tr}_{E_j}\left[P\left(.\otimes\rho_{E_j}\right)Q^\dagger\right], \tag{5.1.28}$$

which represents applying the operator P from the left and $Q^\dagger$ from the right, followed by the tracing out of the environment register E_j. We can write ρ_1 in this notation as follows:

$$\begin{aligned}\rho_1 =&\frac{1}{2}\Big[|0\rangle\langle 0|\otimes\Phi_1^{(YY)}[\rho_S]+|0\rangle\langle 1|\otimes\Phi_1^{(YX)}[\rho_S]\\&+|1\rangle\langle 0|\otimes\Phi_1^{(XY)}[\rho_S]+|1\rangle\langle 1|\otimes\Phi_1^{(XX)}[\rho_S]\Big].\end{aligned}$$

Next, we prepare the environment register in the state ρ_{E_2}, and apply the next set of unitaries (obtained by sampling from $\mathcal{D}_2$), followed by the tracing out the environment E_2. It is easy to see that this sequence leave us with a composition of the maps, resulting in the state:

$$\begin{aligned}\rho_2 = \frac{1}{2}\Big[&|0\rangle\langle 0|\otimes\Phi_2^{(YY)}\Phi_1^{(YY)}[\rho_S]\\&+|0\rangle\langle 1|\otimes\Phi_2^{(YX)}\Phi_1^{(YX)}[\rho_S]\\&+|1\rangle\langle 0|\otimes\Phi_2^{(XY)}\Phi_1^{(XY)}[\rho_S]\\&+|1\rangle\langle 1|\otimes\Phi_2^{(XX)}\Phi_1^{(XX)}[\rho_S]\Big].\end{aligned} \tag{5.1.29}$$

Continuing in this way K times, we obtain by induction,

$$\begin{aligned}\rho_K = \frac{1}{2}\Big[&|0\rangle\langle 0|\otimes\bigcirc_{j=1}^K\Phi_j^{(YY)}[\rho_S]\\&+|0\rangle\langle 1|\otimes\bigcirc_{j=1}^K\Phi_j^{(YX)}[\rho_S]\\&+|1\rangle\langle 0|\otimes\bigcirc_{j=1}^K\Phi_j^{(XY)}[\rho_S]\\&+|1\rangle\langle 1|\otimes\bigcirc_{j=1}^K\Phi_j^{(XX)}[\rho_S]\Big].\end{aligned} \tag{5.1.30}$$

Finally, we measure the ancilla and system registers with the observable $\sigma^x\otimes O$. This constitutes one run of Algorithm 2.

Let us now look at the outcome of any such run. On measuring the ancilla in σ^x, the first and the last terms of Eq. (5.1.30) disappear, and so the output of the k-th run becomes,

$$\mu_k = \frac{1}{2}\mathrm{Tr}\left[O\left[\bigcirc_{j=1}^K\Phi_j^{(YX)}[\rho_S]+\bigcirc_{j=1}^K\Phi_j^{(XY)}[\rho_S]\right]\right]. \tag{5.1.31}$$

Then, by the linearity of expectation, we have

$$\mathbb{E}[\mu_k] = \frac{1}{\zeta^2}\mathrm{Tr}\left[\bigcirc_{j=1}^K\widetilde{\Phi}_j[\rho_S]\right] = \frac{1}{\zeta^2}\mathrm{Tr}\left[O\widetilde{\mathcal{M}}_K[\rho_S]\right], \tag{5.1.32}$$

where ζ is as defined in Algorithm 2. Thus, the outcome of each run is a random variable whose expectation value gives an estimate of the desired quantity (up to a multiplicative factor of $1/\zeta^2$).

We observe that the positive operator-valued measurement of the state at the end yields some eigenvalue of O in the range $[-\|O\|,\|O\|]$. So, outcome μ_k satisfies,

$$-\|O\|\zeta^2\le\zeta^2\mu_k\le\|O\|\zeta^2. \tag{5.1.33}$$

Thus, after T runs, we have a set of random variables $\{\mu_k\}_{k=1}^T$. By using Hoeffding's inequality, we can ensure

$$\mu = \frac{\zeta^2}{T}\sum_{k=1}^T\mu_k,$$

is close to its expectation value. Indeed,

$$\Pr\left[\left|\mu - \mathrm{Tr}\big[O\ \widetilde{\mathcal{M}}_K[\rho_S]\big]\right| \geq \varepsilon/2\right] \leq 2\exp\left[-\frac{T\varepsilon^2}{8\zeta^4\|O\|^2}\right].$$

Thus,

$$\left|\mu - \mathrm{Tr}\left[O\ \widetilde{\mathcal{M}}_K[\rho_S]\right]\right| \leq \varepsilon/2, \tag{5.1.34}$$

with at least $1-\delta$ probability for

$$T \geq \frac{8\|O\|^2\ln(2/\delta)\zeta^4}{\varepsilon^2}.$$

Now, for any $j \in [1, K]$, we get

$$\left\|U_j - \widetilde{U}_j\right\| \leq \varepsilon' = \frac{\varepsilon}{6K\|O\|}$$

from the statement of Lemma 4.6.3. Then, using Lemma 5.1.3 and the triangle inequality, we obtain

$$\begin{aligned}
&\left|\mu - \mathrm{Tr}\big[O\ \mathcal{M}_K[\rho_S]\big]\right| \\
&\leq \left|\mu - \mathrm{Tr}\big[O\ \widetilde{\mathcal{M}}_K[\rho_S]\big]\right| + \left|\mathrm{Tr}\big[O\ \widetilde{\mathcal{M}}_K[\rho_S]\big] - \mathrm{Tr}\big[O\ \mathcal{M}_K[\rho_S]\big]\right| \\
&\leq \varepsilon/2 + \varepsilon/2 = \varepsilon.
\end{aligned} \tag{5.1.35}$$

To estimate the circuit depth of Algorithm 2 and the number of classical repetitions T, we need to find ζ, which crucially depends on the choices of the parameters r_j. For the j-th collision, Lemma 4.6.3 shows how we can consider the LCU decomposition $\widetilde{U}_j$ approximating $U_j = e^{-i\Delta t\beta_j\overline{H}}$ with sufficient accuracy. If $\beta = \max_j \beta_j$ and $r = \min_j r_j$, then the sum of LCU coefficients $\alpha^{(j)} = \sum_k |\alpha_{jk}|$ satisfies,

$$\alpha^{(j)} \leq e^{(\beta_j\Delta t)^2/r_j} \leq e^{(\beta\Delta t)^2/r}, \tag{5.1.36}$$

with

$$\zeta \leq \left(e^{(\beta\Delta t)^2/r}\right)^K. \tag{5.1.37}$$

Thus we ensure that $\zeta = \mathcal{O}(1)$ by choosing $r = \mathcal{O}(\beta^2\Delta t^2 K)$. Consequently, the number of classical repetitions needed is

$$T = \mathcal{O}\left(\frac{\|O\|^2\log(1/\delta)}{\varepsilon^2}\right). \tag{5.1.38}$$

Now, for the circuit depth of each coherent run, the quantum circuit in Fig. 5.1 consists of $2K$ unitaries of the form $X_j^{(c)}$ and $Y_j^{(a)}$. Each of these unitaries comprises qr Pauli operators and r controlled single-qubit rotations, where q is the truncation parameter of the Taylor series. Thus, the overall circuit depth will be $\tau_d = \mathcal{O}(K(qr+r)) = \mathcal{O}(Kqr)$. For the above choice of r, Eq. (5.1.20) gives

$$q = \mathcal{O}\left(\frac{\log(\beta K\|O\|\Delta t/\varepsilon)}{\log\log(\beta K\|O\|\Delta t/\varepsilon)}\right),$$

which gives the overall circuit depth per coherent run as

$$\tau_d = \mathcal{O}\left(\beta^2K^2\Delta t^2\frac{\log(\beta K\|O\|\Delta t/\varepsilon)}{\log\log(\beta K\|O\|\Delta t/\varepsilon)} + K\tau_{\rho_E}\right). \tag{5.1.39}$$

Here, the additive term $K\tau_{\rho_E}$ appears because, in each run of the circuit, the sub-environment needs to be prepared K times, each time requiring a circuit depth of at most τ_{ρ_E}. This completes the proof. □

5.1.3 Complexity Comparison: Markovian K-Collision with Near-Term Hamiltonians

We now compare the complexity of other near-term Hamiltonian simulation algorithms to output an ε-additive accurate estimate of the expectation value $\mathrm{Tr}[O\,\mathcal{M}_K[\rho_S]]$. Primarily, we will compare the circuit depth, the number of ancilla qubits, and the number of classical repetitions needed. At the onset of the early fault-tolerant era, it is better to have multiple independent runs of a short-depth quantum circuit than a single run of a very deep quantum circuit. Thus, it is standard to separately analyze the cost of each run and the number of classical repetitions separately. The total complexity is, of course, the product of the circuit depth per coherent run and the total number of classical repetitions.

First, we observe that most Hamiltonian simulation algorithms can be incorporated into Algorithm 2. Steps 2a and 2b, which are essentially implementing the operator $U_j = e^{-i\Delta t \beta_j \overline{H}_j}$ in the Hamiltonian simulation by SA-LCU, can be replaced with other near-term techniques such as qDRIFT or Trotterization. In Step 3, a direct measurement of O on the system register would suffice for these two methods, as they do not require any ancilla registers. So, Lemma 5.1.3 can also be modified to incorporate different procedures: any Hamiltonian simulation procedure needs to be implemented with precision $\mathcal{O}(\varepsilon/(K\|O\|))$, in order to output an ε-accurate estimate of the expectation value of O. This circuit depth is essentially the cost of composing the underlying Hamiltonian simulation algorithm K times.

On the other hand, the expectation value of O can either be measured incoherently or coherently. The incoherent approach involves simply measuring O with respect to the prepared state, requiring $\mathcal{O}(\|O\|^2/\varepsilon^2)$ classical repetitions. It is also possible to use quantum amplitude estimation [97, 98] to estimate this quantity in cost scaling as $1/\varepsilon$ coherently. However, this requires access to a block encoding [99, 44] of the observable O, adding to the number of ancilla qubits required. More precisely, given an $(\alpha_O, a_O, 0)$ block-encoding of O, we can coherently estimate the desired expectation value using amplitude estimation for $\mathcal{O}\left(\alpha_O(\tau_d + \tau_O)/\varepsilon\right)$ cost. Here, τ_d is the circuit depth of the composition of K Hamiltonian simulation algorithms, and τ_O is the circuit depth of implementing the block encoding of O. So, along with the number of ancilla qubits, the circuit depth also increases substantially. Thus, amplitude estimation is not a technique that can be deployed in early fault-tolerant quantum computers. Consequently, we restrict ourselves to estimating the cost using the incoherent approach. Table 5.1 summarizes the complexities associated with different near-term Hamiltonian simulation methods.

Table 5.1: Comparison of the costs of estimating the expectation value of an observable O with respect to a quantum state that has undergone the K-collision map of Definition 5.1.2 using different near-term Hamiltonian simulation algorithms. The goal of the algorithm is to output the desired expectation value within an additive accuracy of ε with a constant success probability. Here, the ancilla qubits indicate the number of additional qubits (other than the system and environment qubits) required. We assume that for any of the K collisions, the total Hamiltonian is a linear combination of at most L strings of n-qubit Pauli operators, with the total weight of the coefficients upper bounded by β. Each collision corresponds to evolving according to the corresponding (total) Hamiltonian for a time Δt. Also, $\tau_{\rho_E} = \max_{j\in[1,K]} \tau_{\rho_{E_j}}$, where $\tau_{\rho_{E_j}}$ is the circuit depth of the unitary preparing the sub-environment E_j in the state ρ_{E_j}.

Algorithm	No. of ancilla qubits	Circuit depth per coherent run	Classical repetitions
1st-order Trotter	0	$\mathcal{O}\left(\frac{\beta^2 K^2 \|O\| L \Delta t^2}{\epsilon} + K\tau_{\rho_E}\right)$	$\mathcal{O}\left(\frac{\|O\|^2}{\epsilon^2}\right)$
qDRIFT	0	$\mathcal{O}\left(\frac{\beta^2 K^2 \|O\| \Delta t^2}{\varepsilon} + K\tau_{\rho_E}\right)$	$\mathcal{O}\left(\frac{\|O\|^2}{\varepsilon^2}\right)$
$2k$-order Trotter	0	$\mathcal{O}\left(L(K\beta\Delta t)^{1+\frac{1}{2k}}\left(\frac{\|O\|}{\varepsilon}\right)^{\frac{1}{2k}} + K\tau_{\rho_E}\right)$	$\mathcal{O}\left(\frac{\|O\|^2}{\varepsilon^2}\right)$
Single-Ancilla LCU	1	$\mathcal{O}\left(\beta^2 K^2 \Delta t^2 \frac{\log(\beta K \|\mathcal{O}\| \Delta t/\varepsilon)}{\log\log(\beta K \|O\| \Delta t/\varepsilon)} + K\tau_{\rho_E}\right)$	$\mathcal{O}\left(\frac{\|O\|^2}{\varepsilon^2}\right)$

Let us begin by considering the circuit depth of the First-order Trotter method [45, 48]. We note that no

ancilla qubit (other than the system and environment registers) is needed. The worst-case circuit depth of simulating the j-th collision map scales with the number of terms in the corresponding total Hamiltonian (L_j) as $\mathcal{O}(KL\beta_j^2\Delta t^2\|O\|/\varepsilon)$. Then, by composing K such collision maps and using the upper bounds β and L, we obtain the circuit depth for each coherent run as

$$\tau_d = \mathcal{O}\left(\frac{K^2L\beta^2\Delta t^2\|O\|}{\varepsilon} + K\tau_{\rho_E}\right).$$

In order to estimate the expectation value of observable O with a success probability of at least $1-\delta$, the number of independent runs required is $\mathcal{O}(\|O\|^2\log(1/\delta)/\varepsilon^2)$.

The randomized Hamiltonian simulation approach, qDRIFT [47], also requires no ancilla qubits. Moreover, the circuit depth does not depend on the number of terms in the Pauli decomposition of the underlying Hamiltonian. The overall circuit depth to simulate the Markovian K-collision map is given by

$$\tau_d = \left(\frac{K^2\beta^2\Delta t^2\|O\|}{\varepsilon} + K\tau_{\rho_E}\right).$$

Thus, compared to SA-LCU, both First-order Trotter and qDRIFT require an exponentially worse circuit depth in terms of $1/\varepsilon$.

We now move on to the complexity of implementing the Markovian K-collision map by using higher-order Trotter methods [48]. For any positive number k, the circuit depth of the $2k$-order Trotter method for implementing $e^{-i\Delta t\overline{H}_j}$ to within an accuracy of $\mathcal{O}(\varepsilon/K\|O\|)$ is

$$\mathcal{O}\left(5^{k-1}L_j\left(\beta_j\Delta t\right)^{1+\frac{1}{2k}}\cdot\left(\frac{K\|O\|}{\varepsilon}\right)^{\frac{1}{2k}}\right).$$

Then, composing the previously mentioned simulation procedure a total of K times requires a circuit depth of

$$\begin{aligned}\tau_d &= \mathcal{O}\left(5^{k-1}KL(\beta\Delta t)^{1+\frac{1}{2k}}\cdot\left(\frac{K\|O\|}{\varepsilon}\right)^{\frac{1}{2k}} + K\tau_{\rho_E}\right)\\ &= \mathcal{O}\left(5^{k-1}L(K\beta\Delta t)^{1+\frac{1}{2k}}\cdot\left(\frac{\|O\|}{\varepsilon}\right)^{\frac{1}{2k}} + K\tau_{\rho_E}\right).\end{aligned} \tag{5.1.40}$$

Note that the pre-factor $((\beta L)^{1+1/(2k)})$ in the complexity of Trotter-based methods scales with the norm of the sum of the nested commutators of the local Pauli terms in the description of the Hamiltonian. In certain cases, the pre-factor scaling is better than the worst-case bounds we consider here. We refer the readers to Ref. [48] for more details. Although higher-order Trotter methods do not require any ancilla qubits, the exponential scaling in the pre-factor makes it difficult to implement these methods for high k values. Typically, in practice, $k=1$ (the second-order method) and $k=2$ (the fourth-order method) are implemented.

Finally, the state-of-the-art Hamiltonian simulation method, qubitization, requires access to a block encoding of the underlying Hamiltonian [44]. In the case of simulating each collision $U_j = e^{-i\Delta t\beta_j\overline{H}_j}$, a block encoding to H_j is needed, which requires $\mathcal{O}(\log L_j)$ ancilla qubits. However, since we have to implement a composition of these individual collision maps, overall, we need $\mathcal{O}(\log L)$ ancilla qubits to implement each of these maps. Additionally, we would require some sophisticated controlled logic in executing this [49]. Although the circuit depth

$$\tau_d = \mathcal{O}\left(KL\beta\Delta t + K\log(K\|O\|/\varepsilon) + K\tau_{\rho_E}\right),$$

has a better dependence on K and Δt, as compared to near-term Hamiltonian simulation methods, implementing a K-collision map using qubitization is beyond the reach of early fault-tolerant quantum computers.

Overall, we have developed a general framework to simulate K memoryless collisions on a quantum computer using various near-term Hamiltonian simulation procedures. In the next section, we use such collisions to simulate Lindbladian dynamics.

5.2 Application: Simulating Lindbladian Dynamics

Having established a general framework for simulating Markovian collision models, we now turn our attention to a particularly important application: the simulation of Lindbladian dynamics. The Lindblad master equation, as we discussed in Chapter 3, is a cornerstone of open quantum system theory, describing a wide range of physical processes from spontaneous emission to dephasing.

In this section, we will demonstrate how the Markovian K-collision map can be tailored to approximate the continuous evolution described by the Lindblad equation. This connection provides a powerful and physically intuitive way to simulate Lindbladian dynamics on a quantum computer. We will first recall the correspondence between the collision model parameters and the terms in the Lindblad equation. Then, we will present a specialized version of our randomized simulation algorithm for this purpose and showcase its performance with a numerical benchmark.

5.2.1 From Collision Models to Lindblad Equations

The Lindblad master equation describes the time evolution of a quantum system undergoing dissipative dynamics in the presence of an environment. It assumes that the underlying system is weakly coupled to the environment at all times so that the Born-Markov and secular approximations hold. For a system with Hamiltonian H_S, the Lindblad master equation, describing the reduced state of the system ρ_S, is given by

$$\mathcal{L}[\rho_S] \equiv \frac{\partial \rho_S}{\partial t} = -i[H_S, \rho] + \sum_j \left(A_j \rho A_j^\dagger - \frac{1}{2}\{A_j^\dagger A_j, \rho\} \right). \tag{5.2.1}$$

Here, the evolution comprises two distinct parts: a unitary component governed by the system Hamiltonian H_S and a dissipative component described by the so-called quantum jump operators A_j, obtained from the interaction between the system and the environment. Note that for a d-dimensional system, $A_j \in \mathbb{C}^{d\times d}$, are not necessarily Hermitian. Simulating the Lindblad dynamics for time t on a quantum computer essentially means implementing the map $e^{\mathcal{L}t}$.

We can now analyze the complexity of simulating Lindblad dynamics using the Markovian K-collision map from Sec. 5.1.1. As discussed in the previous section, quantum collision models simulate open system dynamics by discretizing the continuous system-environment interaction into a sequence of brief collisions between the system and independent sub-environments. The correspondence between Lindblad dynamics and the collision model has been derived earlier. The appropriate choices of the time of each collision (Δt), the total number of collisions (K), and the interaction Hamiltonian H_{I_j} the Markovian K-collision map can approximate Lindblad dynamics, and (a slightly modified version of) Algorithm 2 can be used to efficiently estimate $\mathrm{Tr}[O\, e^{\mathcal{L}t}[\rho_S]]$ for any observable O.

As in Sec. 5.1, we consider an n-qubit system with Hamiltonian H_S, prepared initially in the quantum state ρ_S. The environment consists of m discrete, single qubit sub-environments prepared in some state ρ_{E_j}, for $j \in [1, m]$. As before, these sub-environments sequentially interact with the system over small but equal time intervals, Δt, driving its evolution. The system evolves under its free Hamiltonian H_S, while the j-th sub-environment evolves under its local Hamiltonian H_{E_j}. The interaction Hamiltonian H_{I_j} governs the interaction between the system and the j-th sub-environment.

While Lindbladian dynamics effectively couple the system to all m environmental subsystems simultaneously, the collision model applies these interactions sequentially. To prevent the free system evolution from accumulating m-fold over one full cycle of m collisions, we renormalize the system Hamiltonian as $H_S \to H_S/m$. Without this rescaling, across a block of m collisions (total duration $m\,\Delta t$) the system would undergo m times the intended evolution under H_S, whereas each sub-environment interacts only for time Δt. Furthermore, the Lindblad dynamics is derived from collision maps in a regime where the system-environment coupling parameter λ is tuned to satisfy $\lambda^2 \Delta t = 1$. As Δt is typically small, intuitively, this results in repeated momentary collisions. This ensures the coupling is strong enough to drive dissipative dynamics even within a short interaction time Δt. In the remainder of this section, we shall assume that the coupling constant λ is diverging, i.e., $\lambda \to 1/\sqrt{\Delta t}$. Another standard assumption, for the derivation of Lindblad dynamics from collision models, is that the state of each sub-environment or the interaction Hamiltonians are so chosen that $\forall j$, $\mathrm{Tr}_{E_j}[[H_{I_j}, \rho_S \otimes \rho_{E_j}]] = 0$. Equivalently, we require $\mathrm{Tr}_{E_j}\left[H_{I_j}(\mathbb{I}\otimes\rho_{E_j})\right] = 0$ for all j (vanishing first moment of the environment operators), so that no first-order coherent renormalization (Lamb shift) arises.

5.2.2 Algorithm for Lindbladian Evolution

We now describe the structure of the Hamiltonian we consider that satisfies the above constraints. We assume that the system Hamiltonian H_S (rescaled by m) is a linear combination of strings of Pauli operators given by

$$H_S = \sum_{j=1}^{L_S} \lambda_j P_j / m,$$

such that $\beta_S = \sum_j |\lambda_j|$. In our case, the environment is a discrete sum of m sub-environments, such that $H_{E_j} = \beta_{E_j}\sigma^z$ (equivalent to the number operator up to an energy shift). Each sub-environment is prepared in the single qubit thermal state (at some inverse temperature ω), i.e.,

$$\rho_{E_j} = \frac{|0\rangle\langle 0| + e^{-\omega}|1\rangle\langle 1|}{1+e^{-\omega}}. \tag{5.2.2}$$

This state can be prepared efficiently by first preparing the entangled pure state

$$|\psi\rangle = \frac{|00\rangle + e^{-\omega/2}|11\rangle}{\sqrt{1+e^{-\omega}}},$$

and then tracing out the second qubit. Henceforth, we will assume that preparing ρ_{E_j} is a constant-depth unitary procedure.

We consider that the interaction Hamiltonian corresponding to the j-th collision (between the system and j-th sub-environment) H_{I_j} can also be expressed as a linear combination of some L_{I_j} Pauli operators. For instance, we can express it in terms of the Lindbladian jump operators as

$$H_{I_j} = \left(A_j \otimes \sigma^+ + A_j^\dagger \otimes \sigma^-\right),$$

where $\sigma^\pm = (\sigma^x \pm i\sigma^y)/2$ are the raising and lowering operators respectively. We assume β_{I_j} denotes the total weight of the coefficients in the description of H_{I_j} and that this Hamiltonian has L_{I_j} terms. Note that in some cases, the individual jump operator A_j may itself be unitary. Overall, the total Hamiltonian corresponding to the j-th collision is given by:

$$\overline{H}_j = \frac{1}{\beta_j}\left(\frac{1}{m}H_S + H_{E_j} + \lambda H_{I_j}\right), \tag{5.2.3}$$

where $\beta_j \leq \beta_S/m + \lambda\beta_{I_j} + \beta_{E_j}$, and $\lambda = 1/\sqrt{\Delta t}$. Hence, $\overline{H}_j$ also can be expressed as a linear combination of $L_S + L_{I_j} + 1$ Pauli operators with total weight at most β_j.

Let us now discuss the choice of Δt and K for a K-collision map to be close (in induced 1-norm) to $e^{\mathcal{L}t}[.]$. The collisions between the system and the m sub-environments occur one by one in a fixed order (E_1, E_2, …, E_m). Now, to simulate the Lindblad dynamics for a total evolution time of t, this sequence is repeated ν times, with each collision occurring for a time interval $\Delta t = t/\nu$. Thus, in all, there are $K = m \times \nu$ collisions, making it an (m,ν)-collision map.

As before, the interaction of j-th sub-environment with the system is given by

$$U_j = e^{-i\beta_j \overline{H}_j \Delta t},$$

and, from Definition 5.1.1, the j-th collision map is

$$\Phi_j[.] = \mathrm{Tr}_{E_j}\left[U_j\left(.\otimes\rho_{E_j}\right)U_j^\dagger\right]. \tag{5.2.4}$$

The (m,ν)-collision map is defined as follows:

Definition 5.2.1 ((m,ν)-collision map)**.** *Let Φ_1 to Φ_m be the collision maps as defined in Definition 5.1.1. Then a (m,ν)-collision map, $\mathcal{M}_{m,\nu}$, is defined as the application of these maps composed ν times as follows:*

$$\mathcal{M}_{m,\nu}[.] \equiv \left(\bigcirc_{j=1}^{m}\Phi_j[.]\right)^{\circ\nu}. \tag{5.2.5}$$

The value of ν, i.e., the number of times the sequence of m collisions should be repeated so that a (m,ν)-collision map approximates $e^{\mathcal{L}t}$ up to an additive accuracy ε was found in Ref. [49]. For this, let us define

$$\Gamma = \frac{\|\mathcal{L}\|_{1\to1}^2}{m} + \left[\max_{\ell\in[1,m]}\left(\beta_S, \beta_{I_\ell}, \beta_{E_\ell}\right)\right]^4, \tag{5.2.6}$$

where $\|\mathcal{L}\|_{1\to1}$ is the induced 1-norm of $\mathcal{L}[.]$. Now, we restate the result of [49] here:

Lemma 5.2.2 (Corollary 2.1 of [49])**.** *For $\varepsilon \in (0,1)$, a (m,ν)-collision map as defined in Definition 5.2.1, with interaction time $\Delta t = t/\nu$,*

$$\nu \geq O\left(\frac{t^2 m}{\varepsilon}\Gamma\right), \tag{5.2.7}$$

and $\lambda \to \frac{1}{\sqrt{\Delta t}}$ satisfies

$$\left\|e^{\mathcal{L}t} - \mathcal{M}_{m,\nu}[.]\right\|_{1\to1} \leq \varepsilon. \tag{5.2.8}$$

Algorithm 3 Estimator for $\mathrm{Tr}[O\, e^{\mathcal{L}t}[\rho_S]]$ via the (m,ν)-collision map

Input: Initial system state ρ_S; m single-qubit sub-environment states (Eq. (5.2.2)); observable O; unitaries $\{\widetilde{U}_j\}_{j=1}^m$ with LCU decompositions $\widetilde{U}_j = \sum_k \alpha_{jk} W_{jk}$ such that $\|\widetilde{U}_j - e^{-i\Delta t \beta_j \overline{H}_j}\| \leq \varepsilon'$; repetitions ν; precision ε'.

Output: Estimate μ satisfying $|\mu - \mathrm{Tr}[O\, e^{\mathcal{L}t}[\rho_S]]| \leq \varepsilon$ with high probability.

1: Set $K \leftarrow m\nu$ and prepare ancilla in $|+\rangle$ together with the system in ρ_S.
2: **for** $j = 0$ to $K-1$ **do**
3: $\quad \ell \leftarrow (j \bmod m) + 1$.
4: $\quad$ Initialise environment register in ρ_{E_ℓ}.
5: $\quad$ Draw i.i.d. samples $X_j, Y_j \sim \mathcal{D}_\ell = \left\{(W_{\ell k},\, \alpha_{\ell k}/\alpha^{(\ell)})\right\}$ where $\alpha^{(\ell)} = \sum_k |\alpha_{\ell k}|$.
6: $\quad$ Apply controlled unitary $X_j^{(c)}$ and anti-controlled unitary $Y_j^{(a)}$.
7: $\quad$ Trace out the environment register.
8: **end for**
9: Measure $\sigma^x \otimes O$ on ancilla-system and record outcome μ_i.
10: Repeat the above procedure T times to obtain $\{\mu_i\}_{i=1}^T$.
11: Compute $\mu \leftarrow \dfrac{\zeta^2}{T} \displaystyle\sum_{i=1}^{T} \mu_i$ where $\zeta = \prod_{j=1}^K \alpha^{(j)}$.
12: **return** μ.

This demonstrates that the (m,ν)-collision map $\mathcal{M}_{m,\nu}[.]$ provides an accurate approximation of Lindbladian dynamics, provided ν is as stated in Lemma 5.2.2. We use Hamiltonian simulation by SA-LCU and Algorithm 3 to achieve this.

We intend to implement the circuit shown in Fig. 5.2. Each sequence of unitaries Q_i corresponds to m collisions between the system and each single-qubit sub-environment E_1 through E_m. As this is repeated ν times, there are overall $K = m\nu$ collisions. Since each block Q_i corresponds to collisions with the same set of sub-environments, we do not have K distinct collisions with distinct sub-environments, but rather ν blocks of m-collisions between the system and the m sub-environments. So, to implement the circuit in Fig. 5.2, we slightly modify Algorithm 2.

We rewrite the (m,ν)-collision map as

$$\mathcal{M}_{m,\nu}[.] = \bigcirc_{j=0}^{K-1} \Phi_{j(\mathrm{mod}\ m)+1}\,[.]\,, \tag{5.2.9}$$

where $K = m\nu$. The right-hand side of Eq. (5.2.9) is simply a $\Phi_j[.]$ map, composed K times, where the cyclic order of the ν repetitions is respected. Thus, the problem of simulating Lindbladian dynamics implies implementing a specific K-collision map. This change is reflected in Step 3 of Algorithm 3. For the j-th iteration, the $\Phi_\ell[.]$ map is implemented, where $\ell = j(\mathrm{mod}\ m) + 1$, i.e. it implements a collision between the system and the single qubit environment E_ℓ. Overall, the correctness of Algorithm 3 is similar to Theorem 5.1.4. Formally, we state the results via the following theorem:

Theorem 5.2.3. *Let us consider an observable O, an n-qubit system prepared in the initial state ρ_S, and m single-qubit sub-environments with each initialized in the single-qubit thermal state defined in Eq. (5.2.2). Let $\varepsilon, \delta \in (0,1)$ and $K = m\nu$, where*

$$\nu = \mathcal{O}\left(\frac{t^2 m \|O\| \Gamma}{\varepsilon}\right). \tag{5.2.10}$$

Then, for $\varepsilon' = \varepsilon/(12K\|O\|)$, Algorithm 3 outputs an estimate μ with a probability of at least $1-\delta$, such that:

$$\left|\mu - \mathrm{Tr}\left[O e^{\mathcal{L}t}[\rho_S]\right]\right| \leq \varepsilon, \tag{5.2.11}$$

using T runs of the circuit shown in Figure 5.2, where

$$T = \mathcal{O}\left(\frac{\|O\|^2 \log(1/\delta)}{\varepsilon^2}\right). \tag{5.2.12}$$

Moreover, the circuit depth of each run is

$$\tau_d = \widetilde{\mathcal{O}}\left(\frac{m^3 t^3 \|O\|}{\varepsilon} \Gamma \beta_{\max}^2\right), \tag{5.2.13}$$

where

$$\beta_{\max} = \max_{\ell\in[1,m]} \left(\beta_{I_\ell},\ \sqrt{\frac{t}{m^2\nu}}\beta_S,\ \beta_{E_\ell}\sqrt{\frac{t}{\nu}} \right). \tag{5.2.14}$$

Proof. First, from the proof of Theorem 5.1.4, we know that for precision $\varepsilon' = \varepsilon/(12\|O\|K)$ and ν as chosen in the statement of the Theorem, Algorithm 3 outputs μ such that,

$$|\mu - \mathrm{Tr}\left[O\ \mathcal{M}_{m,\nu}[\rho_S]\right]| \leq \frac{\varepsilon}{2},$$

with probability at least $(1-\delta)$. This requires

$$T = \mathcal{O}\left(\frac{\|O\|^2}{\varepsilon^2}\log(1/\delta)\right),$$

repetitions of the circuit in Fig. 5.2. Furthermore, from Theorem 3.5.1, we know that our choice of ν ensures

$$\left\|e^{t\mathcal{L}} - \mathcal{M}_{m,\nu}[.]\right\|_{1\to 1} \leq \frac{\varepsilon}{2\|O\|}.$$

For any valid initial state of the system, ρ_S, we have from the definition of induced-1 norm:

$$\left\|e^{t\mathcal{L}}[\rho_S] - \mathcal{M}_{m,\nu}[\rho_S]\right\|_1 \leq \frac{\varepsilon}{2\|O\|}. \tag{5.2.15}$$

Using the tracial version of Hölder's inequality (Lemma A.1.1) we obtain

$$\left|\mathrm{Tr}\left[Oe^{t\mathcal{L}}[\rho_S]\right] - \mathrm{Tr}\left[O\mathcal{M}_{m,\nu}[\rho_S]\right]\right| \leq \frac{\varepsilon}{2}. \tag{5.2.16}$$

Then, the triangle inequality gives us

$$\begin{aligned}
&\left|\mu - \mathrm{Tr}\left[Oe^{t\mathcal{L}}[\rho_S]\right]\right| \\
&\quad\leq |\mu - \mathrm{Tr}\left[O\mathcal{M}_{m,\nu}[\rho_S]\right]| \\
&\qquad + \left|\mathrm{Tr}\left[O\mathcal{M}_{m,\nu}[\rho_S]\right] - \mathrm{Tr}\left[Oe^{t\mathcal{L}}[\rho_S]\right]\right| \\
&\quad\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.
\end{aligned} \tag{5.2.17}$$

Now, Theorem 5.1.4 gives the circuit depth of a K-collision map as

$$\tau_d = \mathcal{O}\left(\beta^2K^2\Delta t^2\frac{\log(\beta K\|O\|\Delta t/\varepsilon)}{\log\log(\beta K\|O\|\Delta t/\varepsilon)} + K\tau_{\rho_E}\right).$$

In our case, $K = m\nu$, $\Delta t = t/\nu$, and $\tau_{\rho_E} = \mathcal{O}(1)$. Moreover, β depends on t and ε as

$$\begin{aligned}
\beta &= \max_{\ell\in[1,m]}(\beta_j) = \max_{\ell}\left(\sqrt{\frac{\nu}{t}}\beta_{I_\ell} + \frac{1}{m}\beta_S + \beta_{E_\ell}\right) \\
&\leq \sqrt{\frac{\nu}{t}} \times \mathcal{O}\left(\beta_{\max}\right),
\end{aligned} \tag{5.2.18}$$

where, in the last line, we have used the fact that $\beta_{E_\ell} = 1$ for any $\ell \in [1,m]$. Here, $\beta_{\max}$ is as defined in the statement of this Theorem. Substituting these parameters, we obtain

$$\begin{aligned}
\tau_d &= \mathcal{O}\left(\beta^2m^2t^2\frac{\log(\beta mt\|O\|/\varepsilon)}{\log\log(\beta mt\|O\|/\varepsilon)} + m\nu\right) \\
&= \mathcal{O}\left(\nu m^2t\beta_{\max}^2\frac{\log\left(\beta_{\max}m\sqrt{\nu t}\|O\|/\varepsilon\right)}{\log\log\left(\beta_{\max}m\sqrt{\nu t}\|O\|/\varepsilon\right)} + m\nu\right).
\end{aligned}$$

Finally, substituting $\nu = \mathcal{O}(t^2m\|O\|\Gamma/\varepsilon)$, we obtain

$$\tau_d = \mathcal{O}\left(\frac{m^3t^3\beta_{\max}^2\|O\|\Gamma}{\varepsilon}\frac{\log(\beta_{\max}mt\Gamma\|O\|/\varepsilon)}{\log\log(\beta_{\max}mt\Gamma\|O\|/\varepsilon)}\right) \tag{5.2.19}$$

$$= \widetilde{\mathcal{O}}\left(\frac{m^3t^3\|O\|}{\varepsilon}\beta_{\max}^2\Gamma\right). \tag{5.2.20}$$

This completes the proof.

□

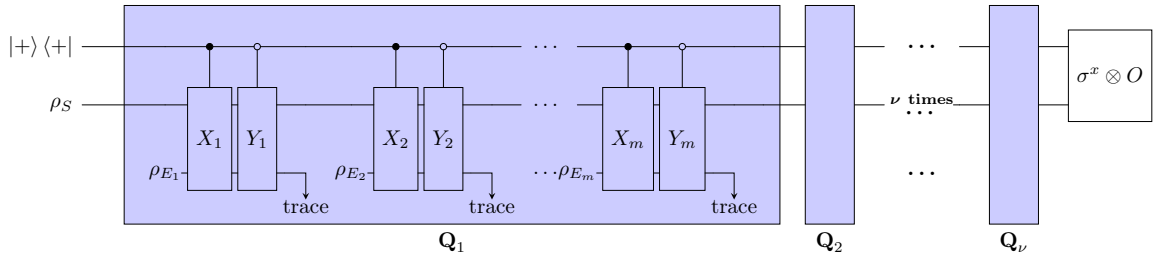


Figure 5.2: The circuit to estimate the expectation value of an observable O for a system evolved under Lindbladian dynamics. The ancilla qubit and the system is initialized in $|+\rangle\langle+|$ and ρ_S respectively. In each block, the unitaries X_j and Y_j are independently sampled and applied as controlled and anti-controlled operations conditioned on the ancilla qubit. After each interaction, the corresponding environment sub-system ρ_{E_j} is traced out, enforcing the Markovian condition. This process is repeated cyclically over m environments for ν iterations. Finally, the observable $\sigma^x \otimes O$ is measured to estimate the time-evolved expectation.

There are two primary sources of error in simulating Lindblad dynamics using quantum collision models: the first arises from approximating Lindbladian dynamics by collision models, and the second stems from the simulation of individual collision steps, which depends on the precision of the Hamiltonian simulation technique employed. While the latter can be mitigated by choosing Hamiltonian simulation algorithms with optimal precision dependence, the error coming from the inherent gap between the Lindblad map and the (m, ν)-collision map remains unaffected by the choice of Hamiltonian simulation. Indeed, Lindbladian dynamics can be approximated only if the system strongly couples with the sub-environments with a strength $\lambda = 1/\sqrt{\Delta t} = \sqrt{\nu/t} \propto t/\sqrt{\varepsilon}$ that grows stronger with the time we intend to simulate the dynamics. The norm of the j-th collision Hamiltonian is at most β_j, which also increases monotonically with t, affecting the circuit depth of all methods that simulate Lindblad dynamics using quantum collision models [19].

In contrast, the state-of-the-art methods (i.e., direct approaches) for simulating Lindbladian dynamics require a cost $\mathcal{O}(t \text{ polylog}(t/\varepsilon))$ [100, 13, 101, 102]. However, most of these methods require access to block encodings and use complicated, infeasible controlled operations for early fault-tolerant quantum computers. On the other hand, quantum collision models provide an easy-to-implement approach, not just for Lindbladian maps but also for other open systems dynamics. Moreover, Algorithm 3 provides a unified framework to compare the cost of implementing the (m, ν)-collision map using different near-term Hamiltonian simulation techniques. Consequently, in the next section, we compare the complexity of Algorithm 3 when other near-term Hamiltonian techniques are used to output an ε-additive estimate of $\text{Tr}[O\, e^{\mathcal{L}t}[\rho_S]]$.

5.2.3 Comparison with other near-term Hamiltonian simulation algorithms

We will borrow the circuit depths obtained in Sec. 5.1.3 (Table 5.1) for estimating $\text{Tr}[O\, e^{\mathcal{L}t}[\rho_S]]$ to ε-additive accuracy. In this case, the parameters are $K = m\nu$, $\Delta t = t/\nu$, and $\beta = \mathcal{O}(\beta_{\max}\sqrt{\nu/t})$, where $\nu = \mathcal{O}(t^2\|O\|m\Gamma/\varepsilon)$. From Theorem 5.2.3, we know that any Hamiltonian simulation procedure needs to be implemented with precision $\varepsilon' = \mathcal{O}(\varepsilon/K\|O\|)$. The circuit depth per coherent run, the total number of qubits needed, and the number of classical repetitions required are outlined in Table 5.2.

Let us first analyze the circuit depth for the first-order Trotter approach. In the worst case, it would depend on the maximum number of terms in $\overline{H}_\ell = H_S + H_{I_\ell} + H_{E_\ell}$ corresponding to the collisions. Let

$$L = L_S + \max_{\ell\in[1,m]} L_{I_\ell} + 1. \tag{5.2.21}$$

Then, for the first-order Trotter method, the appropriate substitution of the parameters yields

$$\tau_d = \mathcal{O}\left(\frac{Lm^3t^3\|O\|^2}{\varepsilon^2}\Gamma\beta_{\max}^2\right), \tag{5.2.22}$$

which indicates that the circuit depth is worse than the circuit depth of Hamiltonian simulation by SA-LCU [Eq. (5.2.13)]. This method is, however, qubit-efficient, requiring $n+1$ qubits overall.

The circuit depth of any procedure using qDRIFT to estimate the desired expectation value is given by

$$\tau_d = \mathcal{O}\left(\frac{m^3t^3\|O\|^2}{\varepsilon^2}\Gamma\beta_{\max}^2\right), \tag{5.2.23}$$

Table 5.2: Comparison of the complexities for simulating Lindblad dynamics via the quantum collision model using different near-term Hamiltonian simulation procedures. We consider an n-qubit system, prepared in ρ_S, with Hamiltonian H_S, expressed as a linear combination of L_S strings of Pauli operators, with total weight β_j. The environment is a discrete sum of m single-qubit sub-environments, each prepared in the (single-qubit) thermal state. The j-th collision corresponds to the interaction Hamiltonian H_{I_j}, which is also a linear combination of strings of Pauli operators of L_{I_j} terms with total weight β_{I_j}. We implement m collisions between the system and each sub-environment qubit, one by one, such that each block of m collisions is repeated a total of ν times. For any observable O, if $\nu = \mathcal{O}(t^2\|O\|m\Gamma/\varepsilon)$, our procedures output an estimate that is an ε-additive accurate estimate of $\mathrm{Tr}[O\, e^{\mathcal{L}t}[\rho_S]]$. Here, $L, \beta_{\max}$ and Γ are defined in Eq. (5.2.21), Eq. (5.2.14), and Eq. (5.2.6), respectively.

Algorithm	Total no. of. qubits	Circuit depth per coherent run	Classical repetitions
1st-order Trotter	$n+1$	$\mathcal{O}\left(\frac{Lm^3t^3\|O\|^2}{\varepsilon^2}\Gamma\beta_{\max}^2\right)$	$\mathcal{O}\left(\frac{\|O\|^2}{\epsilon^2}\right)$
qDRIFT	$n+1$	$\mathcal{O}\left(\frac{m^3t^3\|O\|^2}{\varepsilon^2}\Gamma\beta_{\max}^2\right)$	$\mathcal{O}\left(\frac{\|O\|^2}{\varepsilon^2}\right)$
2nd-order Trotter	$n+1$	$\mathcal{O}\left(L(mt)^{9/4}\left(\frac{\|O\|}{\varepsilon}\right)^{5/4}\Gamma\beta_{\max}^{3/2}\right)$	$\mathcal{O}\left(\frac{\|O\|^2}{\varepsilon^2}\right)$
Single-Ancilla LCU	$n+2$	$\widetilde{\mathcal{O}}\left(\frac{m^3t^3\|O\|}{\varepsilon}\Gamma\beta_{\max}^2\right)$	$\mathcal{O}\left(\frac{\|O\|^2}{\varepsilon^2}\right)$
$2k$-order Trotter $[k>2]$	$n+1$	$\widetilde{\mathcal{O}}\left(\frac{Lm^2t^2\|O\|}{\varepsilon}\Gamma\beta_{\max}\right)$	$\mathcal{O}\left(\frac{\|O\|^2}{\varepsilon^2}\right)$

wherein the advantage over first-order Trotter is in the absence of any dependence on L. However, this circuit depth is also worse than Eq. (5.2.13). The qDRIFT approach also requires $n+1$ qubits overall, which is one less than Hamiltonian simulation by SA-LCU.

For any $2k$-order Trotter method, we also incorporate the additive cost coming from the repeated preparation of the sub-environment register in the single qubit thermal state a total of $K = m\nu$ times (each such state can be prepared in $\mathcal{O}(1)$ circuit depth). Overall, we have

$$\tau_d = \mathcal{O}\Big(L(mt)^{\frac{3}{2}+\frac{3}{4k}} \left(\frac{\|O\|}{\varepsilon}\right)^{\frac{1}{2}+\frac{3}{4k}} \left(\Gamma\beta_{\max}^2\right)^{\frac{1}{2}+\frac{1}{4k}} + m\nu\Big). \tag{5.2.24}$$

However, as mentioned previously, only low-order Trotter methods are preferred for near-term implementation. In particular, for the second-order Trotter method ($k=1$), this becomes

$$\tau_d = \mathcal{O}\Big(L(mt)^{9/4} \left(\frac{\|O\|}{\varepsilon}\right)^{5/4} \Gamma\beta_{\max}^{3/2}\Big). \tag{5.2.25}$$

Compared to the circuit depth obtained by Hamiltonian simulation by SA-LCU, the second-order Trotter method has a better dependence on m, t and $\beta_{\max}$, and a worse dependence on $\|O\|$ and $1/\varepsilon$, in addition to scaling with L. Thus, the circuit depth in Eq. (5.2.13) is shorter in settings where $L \ll \beta_{\max}$, and a high precision of the desired expectation value is demanded.

In summary, SA-LCU achieves significantly shorter circuit depths than second-order Trotterization for high-precision simulations over short time scales. More precisely, the ratio between the circuit depths per coherent run of second-order Trotter and SA-LCU scales as $O(\varepsilon^{1/4}/t^{3/4})$ (ignoring the dependence on all other parameters). Therefore, for simulating Lindblad dynamics over very long time durations where $\varepsilon^{1/4}/t^{3/4} \ll 1$, second-order Trotterization can offer shorter circuit depths.

For higher orders of this method ($k > 2$), the additive term $m\nu$ starts to dominate, and in such cases, the asymptotic circuit depth is

$$\tau_d = \widetilde{\mathcal{O}}\left(\frac{Lm^2t^2\|O\|}{\varepsilon}\Gamma\beta_{\max}\right).$$

Thus, even at very high Trotter orders, the dependence on $1/\varepsilon$, $\|O\|$, and Γ can be no better than Eq. (5.2.13). However, the dependence on m, t and $\beta_{\max}$ is quadratically better. So, the circuit depth in Eq. (5.2.13) is shorter when $L \gg m\beta_{\max}t$. This happens when we wish to simulate Lindblad dynamics for short t, and moreover, the maximum number of terms in the underlying Hamiltonians $\overline{H}_j$ (L) is substantially large [47]. As mentioned before, we have listed the worst-case complexity for Trotterization. It is possible that for specific Hamiltonians, the scaling of the prefactor is better than the worst-case [48].

Finally, qubitization requires $\mathcal{O}(\log L)$ ancilla qubits, coherent access to a block encoding of the underlying Hamiltonians $\bar{H}_j$, and sophisticated controlled operations. The circuit depth is given by

$$\tau_d = \mathcal{O}\left(L\beta mt + m\nu \log\left(m\nu\|O\|/\varepsilon\right)\right) \tag{5.2.26}$$

$$= \widetilde{\mathcal{O}}\left(\frac{Lm^2t^2\|O\|}{\varepsilon}\Gamma\beta_{\max}\right). \tag{5.2.27}$$

Thus, scaling of the circuit depth is similar to a very high order Trotter (up to logarithmic factors).

From the above discussion, it is clear that any procedure would at least require a circuit depth of $m\nu = \mathcal{O}(mt^2\|O\|\Gamma/\varepsilon)$, simply because the sub-environments are prepared a total of $m\nu$ times. This can be seen as a lower bound for the circuit depth of estimating $\mathrm{Tr}[O\, e^{\mathcal{L}t}[\rho_S]]$ using incoherent measurements of O, and matches with the lower bound of Ref. [13]. In the next section, we apply these methods to a concrete problem.

Overall, our methods provide qubit-efficient, end-to-end quantum algorithms for simulating Lindbladian dynamics via the quantum collision model. It is, however, important to distinguish them from direct approaches such as Refs. [13, 101, 102, 103]. These methods assume access to specialized oracles such as block encodings [99, 44], i.e. unitaries that embed the system Hamiltonian (say U_{H_S}) and each of the Lindblad jump operators (say U_{A_j}), in their top-left block. The complexity is expressed in terms of the number of queries made to the oracles U_H and U_{A_j}, with the query complexity scaling as $O(t \cdot \mathrm{polylog}(t/\varepsilon))$ (ignoring dependence on other parameters). The actual circuit depth and gate counts depend on the detailed structure of H_S and A_j, making a direct comparison with our end-to-end methods infeasible. Moreover, constructing such block-encodings often requires substantial overhead (in terms of ancilla qubits, multi-qubit controlled operations), rendering these methods impractical for near-term quantum devices.

5.2.4 Numerical Benchmark: The Transverse-Field Ising Model

We numerically benchmark the performance of the Markovian quantum collision model for simulating Lindblad dynamics by applying it to a concrete problem. We consider the one-dimensional transverse-field Ising model with nearest-neighbor interactions (also known as the Heisenberg XXX model), a widely used testbed for benchmarking Hamiltonian simulation techniques due to its physical significance in condensed matter physics [104]. We look at the dynamics of this system when the environment is a discrete sum of sub-environments, each corresponding to a single-qubit amplitude-damping channel. The j-th collision corresponds to the j-th sub-environment qubit interacting non-trivially with site j of the system Hamiltonian. Thus, the total number of sub-environments is the same as the number of sites in the Ising chain. Consequently, let us define the system Hamiltonian as follows:

$$H_S = -J \sum_{j=1}^{m-1} \sigma_i^z \sigma_{i+1}^z - h \sum_{j=1}^{m} \sigma_i^x, \tag{5.2.28}$$

where J denotes the coupling strength between nearest neighbors, h is the transverse magnetic field strength, and σ_j^z and σ_j^x are Pauli operators acting on site j. The total number of sites m, is the same as the number of sub-environments. Finally, we accommodate for the fact that the system interacts with each sub-environment one at a time by considering the rescaled system Hamiltonian H_S/m, with $\beta_S = (J+h)$. The environment is a discrete sum of m single-qubit number operators, with each sub-environment being in the state $|0\rangle$, corresponding to a thermal state at zero temperature ($\omega \to \infty$), i.e., $\rho_{E_j} = |0\rangle\langle 0|$, for all $j \in [1, m]$.

We define the interaction Hamiltonian corresponding to the j-th collision as

$$\begin{aligned} H_{I_j} = \sqrt{\gamma}(&\mathbb{I}^{j-1} \otimes \hat{\sigma}_j^+ \otimes \mathbb{I}^{m-j-1} \otimes \hat{\sigma}_a^- \\ &+ \mathbb{I}^{j-1} \otimes \hat{\sigma}_j^- \otimes \mathbb{I}^{m-j-1} \otimes \hat{\sigma}_a^+), \end{aligned} \tag{5.2.29}$$

where a denotes the environment qubit, γ denotes the damping strength and σ_- denotes the lowering operator. We estimate the average transverse-field magnetization,

$$M_z = \frac{1}{m} \sum_{j=1}^{m} \sigma_j^z, \tag{5.2.30}$$

with respect to the reduced state $e^{\mathcal{L}t}[\rho_S]$, i.e., we obtain μ such that

$$\left| \mu - \mathrm{Tr}\left[M_z \; e^{\mathcal{L}t}[\rho_S] \right] \right| \leq \varepsilon.$$

Note that the Lindblad master equation dynamics, which we numerically simulate, is given as:

$$\mathcal{L}[\rho_S] \equiv \frac{d\rho_S}{dt} = -i[H_S, \rho] + \sum_j \left(A_j \rho A_j^\dagger - \frac{1}{2}\{A_j^\dagger A_j, \rho\} \right), \tag{5.2.31}$$

where the jump operator for the amplitude damping on jth qubit,

$$A_j = \sqrt{\gamma}\, \mathbb{I}^{j-1} \otimes \sigma_- \otimes \mathbb{I}^{n-j}. \tag{5.2.32}$$

We perform numerical benchmarking for estimating the desired expectation value on a 10-qubit ($m = 10$) transverse field Ising model under amplitude damping via the quantum collision model using different Hamiltonian simulation procedures (first and second order Trotter methods, SA-LCU, and qDRIFT). In Fig. 5.3a, we compare the CNOT gate counts per coherent run to fix the Lindblad evolution time to $t = 1$ and estimate μ for different values of ε. In Fig. 5.3b, we fix $\varepsilon = 0.01$ and vary t instead. To obtain these plots, we set the coupling strength $J = 1$ and the transverse magnetic field strength $h = 0.1$ in the system Hamiltonian H_S. The strength of the amplitude damping channel for each interaction Hamiltonian is fixed to $\gamma = 1$. The time of each collision, Δt, depends on the precision ε and t. Hence, Δt differs for different values of ε and t in these figures.

To obtain the CNOT gate count for each Hamiltonian simulation procedure, we construct the entire circuit on Qiskit using a fully-connected circuit architecture. This corresponds to a composition of Hamiltonian simulations. For this purpose, we use the Solovay-Kitaev theorem (available in Qiskit) to decompose the circuit into a basis comprising single-qubit rotations and CNOT gates. The usual gate optimizations available on Qiskit are applied to all the circuits to obtain a non-trivial total CNOT count. Finally, for a fair comparison, we choose the number of Trotter steps based on the tighter commutator bounds [48] for the first and second-order Trotter methods.

Fig. 5.3a shows that, for a fixed t, the Hamiltonian simulation by the SA-LCU method performs better than the first and second-order Trotter methods and qDRIFT. This is because it has a better dependence on the precision. On the other hand, in Fig. 5.3b, the second-order Trotter method outperforms the rest when ε is fixed and t is increased.

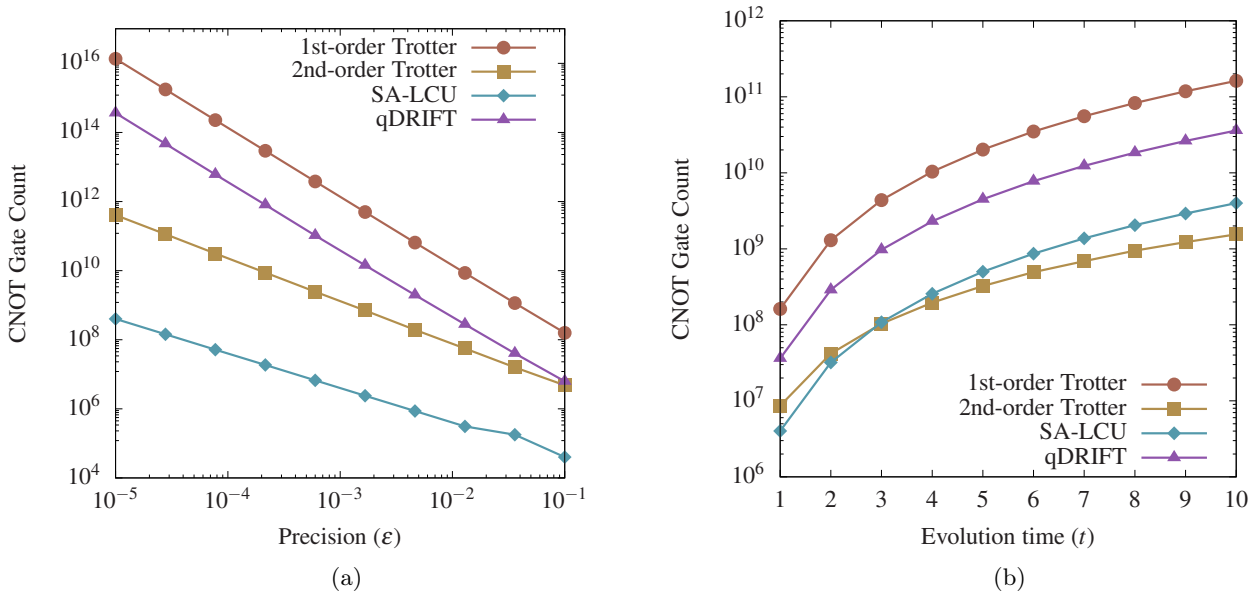


Figure 5.3: We consider the problem of estimating the average transverse-field magnetization of a 10-qubit Heisenberg XXX model under amplitude damping. The corresponding Lindbladian dynamics can be approximated by quantum collision models. The randomized quantum algorithms we develop for simulating quantum collision models can be used to estimate the desired expectation value. In these plots, we show the CNOT gate count per coherent run of our algorithm for different near-term Hamiltonian simulation procedures: the First-order Trotter method (brown circles), the Second-order Trotter method (olive squares), Hamiltonian simulation by Single-Ancilla LCU (SA-LCU, blue diamonds), and qDRIFT (purple triangles). In (a), we vary the precision (ε) for a fixed evolution time ($t = 1$) of the underlying Lindbladian. The Hamiltonian simulation by SA-LCU outperforms the first- and second-order Trotter methods and qDRIFT. In (b), we vary the evolution time (t) for a fixed precision, $\varepsilon = 0.01$, where second-order Trotter outperforms the other methods.

Chapter 6

Simulating Non-Markovian Quantum Collision Models

Chapter Summary

This chapter extends the collision-model framework for open quantum dynamics from the Markovian regime of Sec. 5.1 to the non-Markovian setting, where information can flow back from the environment to the system. Our aim is to develop a practical and scalable simulation methodology that remains compatible with early fault-tolerant devices, while faithfully capturing memory effects that arise from structured system–environment interactions.

We focus on a widely studied mechanism for inducing non-Markovianity in collision models: introducing interactions among environmental subsystems between consecutive system–environment collisions. Concretely, after each collision the participating sub-environment interacts with its neighbour via a CPTP channel, enabling correlations to propagate through the environment and return to the system at later times. This simple modification preserves complete positivity at each step while providing a controllable “memory knob" through the strength of the intra-environment interaction.

On the algorithmic front, we adapt the single-ancilla Linear Combination of Unitaries (SA-LCU) paradigm to implement the non-Markovian K-collision dynamics efficiently. As in the Markovian case, the key subroutine is Hamiltonian simulation for each system–environment interaction; the additional intra-environment coupling is incorporated at the channel level without altering correctness. We formalize the non-Markovian K-collision map, present a randomized single-ancilla algorithm to estimate expectation values after K collisions, and derive accuracy requirements on the underlying Hamiltonian simulation primitives to guarantee an overall additive error ε.

We also compare this SA-LCU-based approach with near-term Hamiltonian simulation baselines (e.g., Trotter formulas and qDRIFT), highlighting trade-offs in coherent circuit depth and repetitions. Finally, we demonstrate the practicality of our method via a numerical benchmark on a ten-qubit transverse-field Ising chain with memory induced by partial-swap interactions between sub-environments, focusing on gate-count scaling with the precision ε.

6.1 Non-Markovian collision models

In the previous chapter (Sec. 5.1), we developed a framework for Markovian collision models in which, at each step, the system interacts with a fresh and uncorrelated sub-environment for a duration Δt and the sub-environment is then traced out. This repeated interaction with independent ancillas ensures CP-divisible, memoryless dynamics: information that flows from the system into a sub-environment does not return, as the sub-environment is discarded and never interacts again.

Non-Markovianity can be introduced into collision models in several ways, including (i) recycling sub-environments so that the same ancilla collides multiple times with the system, (ii) introducing interactions among different sub-environments (intra-environment coupling), (iii) preparing initially correlated environment states (or system–environment correlations) see, e.g., [17]. In this chapter we focus on a particularly simple mechanism: nearest-neighbour interactions between consecutive sub-environments, implemented using successive system–environment collisions via a CPTP channel [18]. This mechanism preserves complete positivity and composability at every step while allowing information to propagate through the environment and back to the system, thereby inducing memory effects.

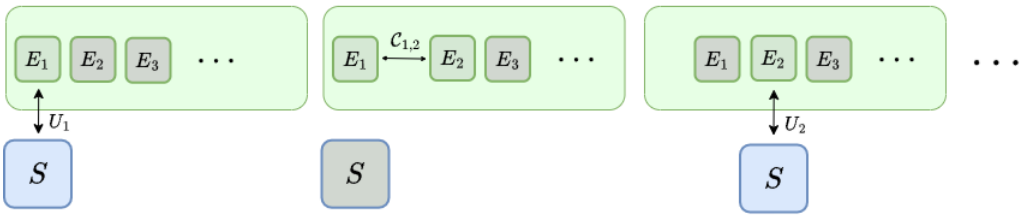


Figure 6.1: Non-Markovian evolution via the collision model illustrating the interleaved dynamics between system-environment and the additional intra-environment interactions. The system S (blue) interacts with the environmental subsystems E_i (green) through unitary operations U_i, while only the adjacent environmental subsystems interact via channel $C_{i,i+1}$. This sequential structure may create a propagating chain of correlations, where information flows not only between the system and environment but also via nearest-neighbor interactions. The three panels represent consecutive time steps of the evolution, demonstrating how correlations may build up and propagate through the environmental subsystems, capturing the memory effects and non-Markovian behavior of the quantum dynamics.

6.1.1 The Non-Markovian K-Collision Map

We consider an n-qubit system with Hamiltonian H_S prepared in state ρ_S on Hilbert space $\mathcal{H}_S$. The environment comprises m sub-environments with Hamiltonians H_{E_j}, prepared in states ρ_{E_j}, for $j \in [1, m]$, with $\mathcal{H}_E = \bigotimes_{j=1}^{m} \mathcal{H}_{E_j}$. For simplicity, we assume all $\mathcal{H}_{E_j}$ have the same dimension. The interaction Hamiltonian for the j-th system–sub-environment collision is H_{I_j}, so the total Hamiltonian governing that collision is

$$H_j = H_S + H_{I_j} + H_{E_j}.$$

After each system–environment collision, we insert an intra-environment interaction between consecutive sub-environments via a CPTP channel $\mathcal{C}_{j,j+1}$. Concretely, the system in state ρ_S first collides with E_j for time Δt, and then E_j interacts with E_{j+1} via $\mathcal{C}_{j,j+1}$; only then is E_j traced out. Hence, a single iteration comprises two steps: a system–sub-environment collision followed by an interaction between neighbouring sub-environments.

This sequence of interactions repeats for K iterations (Fig. 6.1), giving rise to a non-Markovian K-collision map. A convenient choice for the intra-environment channel is the partial swap between consecutive sub-environments [18], which yields a non-Markovian master equation in the continuous-time limit. More precisely, for $p \in [0, 1]$ and two states ρ_j and σ_{j+1} of the same dimension,

$$\begin{aligned}\mathcal{C}_{j,j+1}[\rho_j \otimes \sigma_{j+1}] = \\ (1-p)(\rho_j \otimes \sigma_{j+1}) + S_{j,j+1}(\rho_j \otimes \sigma_{j+1})\mathrm{S}_{j,j+1}^{\dagger}),\end{aligned} \tag{6.1.1}$$

where $S_{j,j+1}$ swaps the two states ρ_j and σ_{j+1}. The parameter p controls the memory strength, ranging from no information transfer (effectively memoryless) at $p = 0$ to perfect swapping between consecutive sub-environments at $p = 1$.

We now formalize a single iteration of this dynamics. Because there are two sub-environment registers in play, we use the parity of j to indicate which register stores $\rho_{E_{j+1}}$ and which collides with the system and is traced out. For $j \in [1, m]$, define

$$U_{S_j} = \bar{U}_j \otimes I_E, \tag{6.1.2}$$

where $\bar{U}_j = e^{-i\Delta t \beta_j \overline{H}_j}$ denotes the collision between the system S and the j-th sub-environment. Here $\overline{H}_j$ is the normalized hamiltonian to make it's decomposition a convex combination of pauli operators and β_j is the normalizing factor. Furthermore, we define the CPTP map,

$$\mathcal{V}_{j,j+1} = I_S \otimes C_{j,j+1}, \tag{6.1.3}$$

where $C_{j,j+1}$ is the partial swap operation defined in Eq. (6.1.1). The collision map is applied to a composite state of the system and the two sub-environments E_j and E_{j+1}, with the state of the latter prepared in $\rho_{E_{j+1}}$. As explained above, the sub-environment register storing $\rho_{E_{j+1}}$ changes depending on whether j is even or odd. Thus, we can now formally define the j-th non-Markovian collision map $\Phi_j^{\mathcal{N}}$ as:

$$\Phi_j^{\mathcal{N}}[\cdot] \equiv \begin{cases} \mathrm{Tr}_{E_j}\left[\mathcal{V}_{j,j+1}\left[U_{S_j}\left(\cdot \otimes \cdot \otimes \rho_{E_{j+1}}\right) U_{S_j}^{\dagger}\right]\right], & j \text{ is odd}, \\ \mathrm{Tr}_{E_j}\left[\mathcal{V}_{j,j+1}\left[U_{S_j}\left(\cdot \otimes \rho_{E_{j+1}} \otimes \cdot\right) U_{S_j}^{\dagger}\right]\right], & j \text{ is even}. \end{cases} \tag{6.1.4}$$

Algorithm 4 Estimator for $\mathrm{Tr}[O\mathcal{N}_K[\rho_S]]$ via the non-Markovian K-collision map

Input: System state ρ_S; sub-environments $\{\rho_{E_j}\}_{j=1}^K$; observable O; unitaries $\{\widetilde{U}_j\}_{j=1}^K$ with LCU decompositions $\widetilde{U}_j = \sum_k \alpha_{jk} W_{jk}$ satisfying $\|\widetilde{U}_j - e^{-i\Delta t H_j}\| \leq \varepsilon'$; precision ε'.
Output: Estimate μ such that $|\mu - \mathrm{Tr}[O\mathcal{N}_K[\rho_S]]| \leq \varepsilon$ with high probability.

Prepare ancilla in $|+\rangle$, system in ρ_S, and first environment register in ρ_{E_1}.
for $j = 1$ to $K-1$ **do**
  Sample $X_j, Y_j \sim \mathcal{D}_j = \{(W_{j\ell},\, \alpha_{j\ell}/\alpha^{(j)})\}$ where $\alpha^{(j)} = \sum_\ell |\alpha_{j\ell}|$.
  Apply $X_j^{(c)}$ and $Y_j^{(a)}$ (controlled by ancilla) on system and the environment register chosen by the parity of j.
  Initialise the other environment register in $\rho_{E_{j+1}}$.
  With probability p apply swap $S_{j,j+1}$ between the two environment registers.
  Trace out the environment register that interacted at step j.
end for
Sample $X_K, Y_K \sim \mathcal{D}_K$ and apply $X_K^{(c)}$ and $Y_K^{(a)}$.
Trace out the final environment register.
Measure $\sigma^x \otimes O$; record outcome μ_i.
Repeat the above procedure T times to obtain $\{\mu_i\}_{i=1}^T$.
Compute $\mu \leftarrow \dfrac{\zeta^2}{T}\displaystyle\sum_{i=1}^{T} \mu_i$ where $\zeta = \prod_{j=1}^K \alpha^{(j)}$.
return μ.

Note that irrespective of the parity of j, the sub-environment E_j is always traced out. The overall non-Markovian evolution over K iterations is obtained by composing these maps for the first $K-1$ steps and then performing a final system–E_K collision and trace-out. We encapsulate this as a definition:

Definition 6.1.1 (Non-Markovian K-collision map)**.** *Let $\Phi_1^{\mathcal{N}}, \ldots, \Phi_{K-1}^{\mathcal{N}}$ be the maps defined in Eq. (6.1.4). The non-Markovian K-collision map $\mathcal{N}_K$ is*

$$\mathcal{N}_K[\cdot] \equiv \mathrm{Tr}_{E_K}\Big[U_{S_K}\big(\bigcirc_{j=1}^{K-1} \Phi_j^{\mathcal{N}}[\cdot]\big)U_{S_K}^\dagger\Big]. \tag{6.1.5}$$

To develop quantum algorithms for implementing the non-Markovian K-collision map $\mathcal{N}_K[.]$, there are a few things to consider. First, it is possible to implement the partial swap $C_{j,j+1}$ efficiently in a probabilistic manner: we apply $S_{j,j+1}$ with probability p and with probability $1-p$, we do not perform any operation. Second, as in the Markovian case, the system environment collisions U_{S_j} can be implemented with any Hamiltonian simulation procedure. We assume that the swap operation can be implemented perfectly so that the only source of error is the underlying Hamiltonian simulation procedure. That is, for each iteration, we implement an approximate map

$$\widetilde{\Phi}_j^{\mathcal{N}}[\cdot] \equiv \begin{cases} \mathrm{Tr}_{E_j}\big[\mathcal{V}_{j,j+1}\big[\widetilde{U}_{S_j}\,(\cdot\otimes\cdot\otimes\rho_{E_{j+1}})\,\widetilde{U}_{S_j}^\dagger\big]\big], & j \text{ is odd,} \\ \mathrm{Tr}_{E_j}\big[\mathcal{V}_{j,j+1}\big[\widetilde{U}_{S_j}\,(\cdot\otimes\rho_{E_{j+1}}\otimes\cdot)\,\widetilde{U}_{S_j}^\dagger\big]\big], & j \text{ is even.} \end{cases} \tag{6.1.6}$$

Analogously, for the overall map, we define

$$\widetilde{\mathcal{N}}_K[.] \equiv \mathrm{Tr}_{E_K}\left[\widetilde{U}_{S_K}\left(\bigcirc_{j=1}^{K-1}\widetilde{\Phi}_j^{\mathcal{N}}[.]\right)\widetilde{U}_{S_K}^\dagger\right]. \tag{6.1.7}$$

Next, we show that the required precision for a Hamiltonian simulation procedure to estimate $\mathrm{Tr}[O\,\mathcal{N}_K[\rho_S]]$ with ε-additive accuracy remains the same as in the Markovian case. For this, we prove the following Lemma:

Lemma 6.1.2 (Bounds on the non-Markovian approximate collision Map)**.** *Let O be an observable and $\widetilde{\mathcal{N}}_K$ be the approximate non-Markovian K-collision map in Eq. (6.1.7), where*

$$\max_{1\leq j\leq K}\left\|U_j - \widetilde{U}_j\right\| \leq \frac{\varepsilon}{3K\|O\|}. \tag{6.1.8}$$

Then, the expectation value of O with respect to the state transformed under the approximate map is ε-close to the expectation value under the exact map. That is, for any ρ,

$$\left|\mathrm{Tr}\left[O\;\mathcal{N}_K[\rho]\right] - \mathrm{Tr}\left[O\;\widetilde{\mathcal{N}}_K[\rho]\right]\right| \leq \varepsilon. \tag{6.1.9}$$

Proof. We consider the error between the operations performed on the state ρ under U_{S_j} and $\widetilde{U}_{S_j}$. More precisely, let $\left\|U_j - \widetilde{U}_j\right\| \leq \xi_j$. Then, immediately, we have $\|U_{S_j} - \widetilde{U}_{S_j}\| \leq \xi_j$. As before, we denote the maximum error in any of the Hamiltonian simulation procedures in the definition of the approximate K-collision map by $\xi_{\max}$, i.e., $\xi_{\max} = \max_{1\leq j\leq K} \xi_j$. Then, using Theorem A.1.2 for any quantum state ρ, we have

$$\left\|U_{S_j}\rho U_{S_j}^\dagger - \widetilde{U}_{S_j}\rho\widetilde{U}_{S_j}^\dagger\right\|_1 \leq 3\xi_j \leq 3\xi_{\max}. \tag{6.1.10}$$

Now, since partial trace and $\mathcal{V}_{j,j+1}$ are CPTP maps, we can make use of the distance between the composition of CPTP maps (Theorem A.1.3) to get

$$\left\|\bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}}[\rho] - \bigcirc_{j=1}^{K-1}\widetilde{\Phi}_j^{\mathcal{N}}[\rho]\right\|_1 \leq 3(K-1)\xi_{\max}. \tag{6.1.11}$$

For the final step, we need to bound

$$\begin{aligned}\left\|[\mathcal{N}_K[\rho]] - \widetilde{\mathcal{N}}_K[\rho]\right\|_1 &= \left\|\mathrm{Tr}_{E_K}\left[U_{S_K}\left(\bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}}[\rho]\right)U_{S_K}^\dagger\right] - \mathrm{Tr}_{E_K}\left[\widetilde{U}_{S_K}\left(\bigcirc_{j=1}^{K-1}\widetilde{\Phi}_j^{\mathcal{N}}[\rho]\right)\widetilde{U}_{S_K}^\dagger\right]\right\|_1 \\ &\leq \left\|U_{S_K}\left(\bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}}[\rho]\right)U_{S_K}^\dagger - \widetilde{U}_{S_K}\left(\bigcirc_{j=1}^{K-1}\widetilde{\Phi}_j^{\mathcal{N}}[\rho]\right)\widetilde{U}_{S_K}^\dagger\right\|_1 \\ &\leq \left\|\bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}}[\rho] - \bigcirc_{j=1}^{K-1}\widetilde{\Phi}_j^{\mathcal{N}}[\rho]\right\|_1 + 2\left\|U_{S_K} - \widetilde{U}_{S_K}\right\| \leq 3(K-1)\xi_{\max} + 2\xi_{\max} < 3K\xi_{\max},\end{aligned} \tag{6.1.12}$$

where we have used Theorem A.1.4 to arrive at the third line from the second. Finally, as in the proof of Lemma 5.1.3, we can use Theorem A.1.2 and set $\xi_{\max} = \varepsilon/(3K\|O\|)$ and get:

$$\left|\mathrm{Tr}\left[O\,\mathcal{N}_K[\rho]\right] - \mathrm{Tr}\left[O\,\widetilde{\mathcal{N}}_K[\rho]\right]\right| \leq \varepsilon. \tag{6.1.13}$$

□

This implies that for estimating the desired expectation value, we need to implement a Hamiltonian simulation procedure with precision $\varepsilon' = \mathcal{O}(\varepsilon/K\|O\|)$, which is the same as in the Markovian case. Thus, we can develop a randomized quantum algorithm that can incorporate any near-term Hamiltonian simulation procedure. In Algorithm 4, we use Hamiltonian simulation by the SA-LCU method to estimate the desired expectation value. We show the circuit corresponding to each run of the algorithm in Fig. 6.2. We now need two sub-environment registers, along with the system register and a single qubit ancilla register. For the first $K-1$ iterations, a composition of the map $\widetilde{\Phi}^{\mathcal{N}}[.]$ is implemented. Note that at odd (even) iterations, the second (first) register stores the state of the subsequent sub-environment. Following the interaction between the two sub-environment registers, the first (second) environment register is traced out.

Following an approach similar to Theorem 5.1.4, we prove the correctness of Algorithm 4.

Theorem 6.1.3. *Let $\varepsilon, \delta \in (0,1)$. Then, for $\varepsilon' = \varepsilon/(6K\|O\|)$, with probability at least $1-\delta$, Algorithm 4 outputs μ, such that*

$$|\mu - \mathrm{Tr}[O\,\mathcal{N}_K[\rho_S]]| \leq \varepsilon,$$

using T repetitions of the circuit shown in Figure 6.2, where

$$T = \mathcal{O}\left(\frac{\|O\|^2\ln(2/\delta)}{\varepsilon^2}\right). \tag{6.1.14}$$

Each such coherent run has a circuit depth of

$$\tau_d = \mathcal{O}\left(\beta^2K^2\Delta t^2\frac{\log(\beta K\|O\|\Delta t/\varepsilon)}{\log\log(\beta K\|O\|\Delta t/\varepsilon)} + K\tau_{\rho_E}\right) \tag{6.1.15}$$

where, $\beta = \max_j \beta_j$, and $\tau_{\rho_E} = \max_j \tau_{\rho_{E_j}}$, where $\tau_{\rho_{E_j}}$ is the circuit depth of the unitary preparing the sub-environment in the state ρ_{E_j}.

Proof. The proof is similar to Theorem 5.1.4. We first initialize the system and ancilla registers. Subsequently, we prepare the first environment register and apply the operations $X_1^{(c)}$ and $Y_1^{(a)}$ obtained by sampling X_1 and Y_1 from $\mathcal{D}_1$. Additionally, we also initialize another environment register that interacts with the first environment register via the partial swap operation represented by the map $\mathcal{V}_{1,2}$. After this interaction, the first environment

register is traced out, leaving the combined system, ancilla and the second environment registers in a state ready for subsequent interactions.

We define the map $\Phi_j^{\mathcal{N}\ (PQ)}$ as:

$$\widetilde{\Phi}_j^{\mathcal{N}^{(PQ)}}[.] \equiv \begin{cases} \mathrm{Tr}_{E_j}\left[\mathcal{V}_{j,j+1}\left[P(.\otimes .\otimes \rho_{E_{j+1}})Q^\dagger\right]\right], \text{ j is odd} \\ \\ \mathrm{Tr}_{E_j}\left[\mathcal{V}_{j,j+1}\left[P(.\otimes \rho_{E_{j+1}}\otimes .)Q^\dagger\right]\right], \text{ j is even} \end{cases} \tag{6.1.16}$$

Here, the map $\widetilde{\Phi}_j^{\mathcal{N}^{(PQ)}}[.]$ represents applying operator P from the left and $Q^\dagger$ from the right, then applying the interaction between environment register followed by tracing out of the first (second) environment register E_j if j is odd (even). Thus, the state of the combined system-ancilla register after the initial collision can succinctly be expressed as:

$$\rho_1 = \frac{1}{2}\Big[|0\rangle\langle 0|\otimes \Phi_1^{\mathcal{N}^{(Y_1Y_1)}}[\rho_S] + |0\rangle\langle 1|\otimes \Phi_1^{\mathcal{N}^{(Y_1X_1)}}[\rho_S] + |1\rangle\langle 0|\otimes \Phi_1^{\mathcal{N}^{(X_1Y_1)}}[\rho_S] + |1\rangle\langle 1|\otimes \Phi_1^{\mathcal{N}^{(X_1X_1)}}[\rho_S]\Big].$$

We use the definition of the controlled and anti-controlled operator to simplify the combined state of the system and ancilla.

To continue the process, we perform the next collision step analogously. We apply the next set of unitaries (obtained by sampling from $\mathcal{D}_2$), then we simulate the intra-environmental interaction by first initializing the next environment register in the state ρ_{E_3} and interacting it with the previous environment and then tracing out the previous environment. At this stage, the cross terms involving different operators, such as $\Phi_2^{\mathcal{N}^{(Y_2Y_2)}}\Phi_1^{\mathcal{N}^{(X_1X_1)}}$ vanish. Thus, after tracing out the appropriate environment register, the state simplifies neatly to:

$$\begin{aligned}\rho_2 = \frac{1}{2}\Big[&|0\rangle\langle 0|\otimes \Phi_2^{\mathcal{N}^{(Y_2Y_2)}}\Phi_1^{\mathcal{N}^{(Y_1Y_1)}}[\rho_S] + |0\rangle\langle 1|\otimes \Phi_2^{\mathcal{N}^{(Y_2X_2)}}\Phi_1^{\mathcal{N}^{(Y_1X_1)}}[\rho_S] \\ &+ |1\rangle\langle 0|\otimes \Phi_2^{\mathcal{N}^{(X_2Y_2)}}\Phi_1^{\mathcal{N}^{(X_1Y_1)}}[\rho_S] + |1\rangle\langle 1|\otimes \Phi_2^{\mathcal{N}^{(X_2X_2)}}\Phi_1^{\mathcal{N}^{(X_1X_1)}}[\rho_S]\Big]\end{aligned} \tag{6.1.17}$$

We continue this $K-1$ times, where in each step the mismatched terms will cancel out and we will be left with $K-1$ composition of the map as follows:

$$\begin{aligned}\rho_{K-1} = \frac{1}{2}\Big[&|0\rangle\langle 0|\otimes \bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}^{(Y_jY_j)}}[\rho_S] + |0\rangle\langle 1|\otimes \bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}^{(Y_jX_j)}}[\rho_S] \\ &+ |1\rangle\langle 0|\otimes \bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}^{(X_jY_j)}}[\rho_S] + |1\rangle\langle 1|\otimes \bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}^{(X_jX_j)}}[\rho_S]\Big]\end{aligned} \tag{6.1.18}$$

For the last collision we apply the control and anti-control operators $X_K^{(c)}$ and $Y_K^{(a)}$ and trace out the first (second) environment register if K is odd (even). Resulting in the final state to be

$$\begin{aligned}\rho_K = \frac{1}{2}\Big[&|0\rangle\langle 0|\otimes \mathrm{Tr}_{E_K}\Big[Y_K\big(\bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}^{(Y_jY_j)}}[\rho_S]\big)Y_K^\dagger\Big] + |0\rangle\langle 1|\otimes \mathrm{Tr}_{E_K}\Big[Y_K\big(\bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}^{(Y_jX_j)}}[\rho_S]\big)X_K^\dagger\Big] \\ &+ |1\rangle\langle 0|\otimes \mathrm{Tr}_{E_K}\Big[X_K\big(\bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}^{(X_jY_j)}}[\rho_S]\big)Y_K^\dagger\Big] + |1\rangle\langle 1|\otimes \mathrm{Tr}_{E_K}\Big[X_K\big(\bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}^{(X_jX_j)}}[\rho_S]\big)X_K^\dagger\Big]\Big]\end{aligned} \tag{6.1.19}$$

Finally, we measure the ancilla and system register with the observable $\sigma^x\otimes O$. This constitutes one run of Algorithm 4. Now measuring the ancilla on σ^x, the first and last terms of Eq. (6.1.19) disappear, and so, the output of the k-th run,

$$\mu_k = \frac{1}{2}\mathrm{Tr}\Big[O\Big[\mathrm{Tr}_{E_K}\Big[Y_K\big(\bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}^{(Y_jX_j)}}[\rho_S]\big)X_K^\dagger\Big] + \mathrm{Tr}_{E_K}\Big[Y_K\big(\bigcirc_{j=1}^{K-1}\Phi_j^{\mathcal{N}^{(Y_jX_j)}}[\rho_S]\big)X_K^\dagger\Big]\Big]\Big] \tag{6.1.20}$$

Then, by the linearity of expectation, we have

$$\mathbb{E}[\mu_k] = \frac{1}{\zeta^2}\Big[\mathrm{Tr}_{E_K}\Big[\widetilde{U}_{S_K}\big(\bigcirc_{j=1}^{K-1}\widetilde{\Phi}_j^{\mathcal{N}}[\rho_S]\big)\widetilde{U}_{S_K}^\dagger\Big]\Big] = \frac{1}{\zeta^2}\mathrm{Tr}\Big[O\widetilde{\mathcal{N}}_K[\rho_S]\Big], \tag{6.1.21}$$

where ζ is as defined in Algorithm 4. Thus, the outcome of each run is a random variable that in expectation value estimates the desired quantity (upto a multiplicative factor of $1/\zeta^2$).

Since the observable O has eigenvalues bounded within $[-\|O\|, \|O\|]$, each individual outcome μ_k satisfies:

$$-\|O\|\zeta^2 \leq \zeta^2 \mu_k \leq \|O\|\zeta^2. \tag{6.1.22}$$

After performing the experiment for T independent runs, we have a collection of random variables $\{\mu_k\}_{k=1}^T$. Then, from Hoeffding's inequality,

$$\mu = \frac{\zeta^2}{T} \sum_{k=1}^{T} \mu_k,$$

satisfies

$$\Pr\left[\left|\mu - \operatorname{Tr}\left[O\, \widetilde{\mathcal{N}}_K[\rho_S]\right]\right| \geq \varepsilon/2\right] \leq 2\exp\left[-\frac{T\varepsilon^2}{8\zeta^4\|O\|^2}\right].$$

Thus, with probability at least $1-\delta$,

$$\left|\mu - \operatorname{Tr}\left[O\widetilde{\mathcal{N}}_K[\rho_S]\right]\right| \leq \varepsilon/2, \tag{6.1.23}$$

for

$$T \geq \frac{8\|O\|^2 \ln(2/\delta)\zeta^4}{\varepsilon^2}.$$

Now from the statement of the Lemma, for any $j \in [1, K]$.

$$\left\|U_j - \widetilde{U}_j\right\| \leq \varepsilon' = \frac{\varepsilon}{6K\|O\|}.$$

Then using Lemma 6.1.2 and the triangle inequality, we obtain

$$\left|\mu - \operatorname{Tr}\left[O\, \mathcal{N}_K[\rho_S]\right]\right| \leq \left|\mu - \operatorname{Tr}\left[O\widetilde{\mathcal{N}}_K[\rho_S]\right]\right| + \left|\operatorname{Tr}\left[O\widetilde{\mathcal{N}}_K[\rho_S]\right] - \operatorname{Tr}\left[O\mathcal{N}_K[\rho_S]\right]\right| \tag{6.1.24}$$

$$\leq \varepsilon/2 + \varepsilon/2 = \varepsilon. \tag{6.1.25}$$

In order to estimate the circuit depth of Algorithm 4 and the number of classical repetitions T, we need to find ζ. Analogous to the Theorem 5.1.4, we use the Lemma 4.6.3 for each collision unitary and choose the maximum repetitions to be $r = \mathcal{O}(K\beta^2\Delta t^2)$, which ensure $\zeta = \mathcal{O}(1)$ Consequently, the number of classical repetitions needed is

$$T = \mathcal{O}\left(\frac{\|O\|^2 \log(1/\delta)}{\varepsilon^2}\right). \tag{6.1.26}$$

Similarly, analogous to Theorem 5.1.4, for the appropriate choices of r and q, we have the overall circuit depth per coherent run as

$$\tau_d = \mathcal{O}\left(\beta^2 K^2 \Delta t^2 \frac{\log(\beta K\|O\|\Delta t/\varepsilon)}{\log\log(\beta K\|O\|\Delta t/\varepsilon)} + K\tau_{\rho_E}\right), \tag{6.1.27}$$

where the additive term $K\tau_{\rho_E}$ again appears as in each run of the circuit, on account of preparing the individual sub-environments, a total of K times. This completes the proof. □

If we assume that the cost of implementing a swap gate between two consecutive sub-environments is constant, the circuit depth of the procedure using the different simulation techniques is the same as those listed in Table 5.1. Thus, this is a unified framework to simulate non-Markovian collisions using near-term Hamiltonian simulation procedures. Again, we do not require many ancilla qubits or need access to block encodings.

It would be interesting to explore whether continuous-time non-Markovian master equations can be approximated by this collision model for finite K. This would require obtaining the number of collisions K for which it is ε-close (in, say, induced 1-norm) to the generator of the underlying non-Markovian master equation. While this problem has been investigated in the Markovian setting (closeness of the K-collision map and exponential of the Lindbladian [19, 49]), very little is known in the non-Markovian case. Ciccarello et al. [18], showed that the collision model we consider gives rise to a non-Markovian master equation for $K \to \infty$ and $p = e^{-\lambda t}$, where λ is a continuous parameter, determining the *memory rate*. However, we leave the question of the precise scaling of the error in this approximation open with (finite) K.

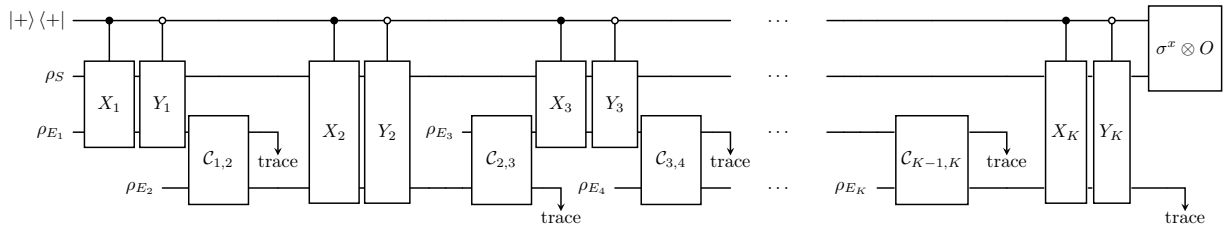


Figure 6.2: Quantum circuit corresponding to each run of Algorithm 4, simulating a non-Markovian K-collision map, using Hamiltonian simulation by SA-LCU. The algorithm applies controlled and anti-controlled sampled unitaries (X_j, Y_j) for the interaction between the system and each sub-environment, followed by an interaction between consecutive sub-environments using the channel $(\mathcal{C}_{j,j+1})$. This sequence is repeated for K collisions. At the end of the process, the ancilla qubit and the system are measured.

6.1.2 Numerical Experiments

We benchmark non-Markovian simulation on a ten-qubit transverse-field Ising chain with nearest-neighbour interactions. Memory effects are induced via sub-environment partial-swap interactions within the collision model. The system–environment interaction is simulated using different Hamiltonian-simulation techniques. To ensure a fair comparison, the gate cost of the environment–environment partial swaps is excluded from the totals, since it is identical across all methods.

Figure 6.3 reports the CNOT gate count per coherent run as a function of the number of collisions K for four methods: single-ancilla LCU, first- and second-order Trotterization, and qDRIFT. Across the tested range of K, the single-ancilla LCU method achieves the lowest CNOT count despite the additional controlled operations on the ancilla. The Trotter baselines are evaluated using tight commutator-based bounds, and qDRIFT lies between first- and second-order Trotter in this regime. The asymptotic precision advantage of single-ancilla LCU persists in the non-Markovian setting, mirroring the Lindbladian study.

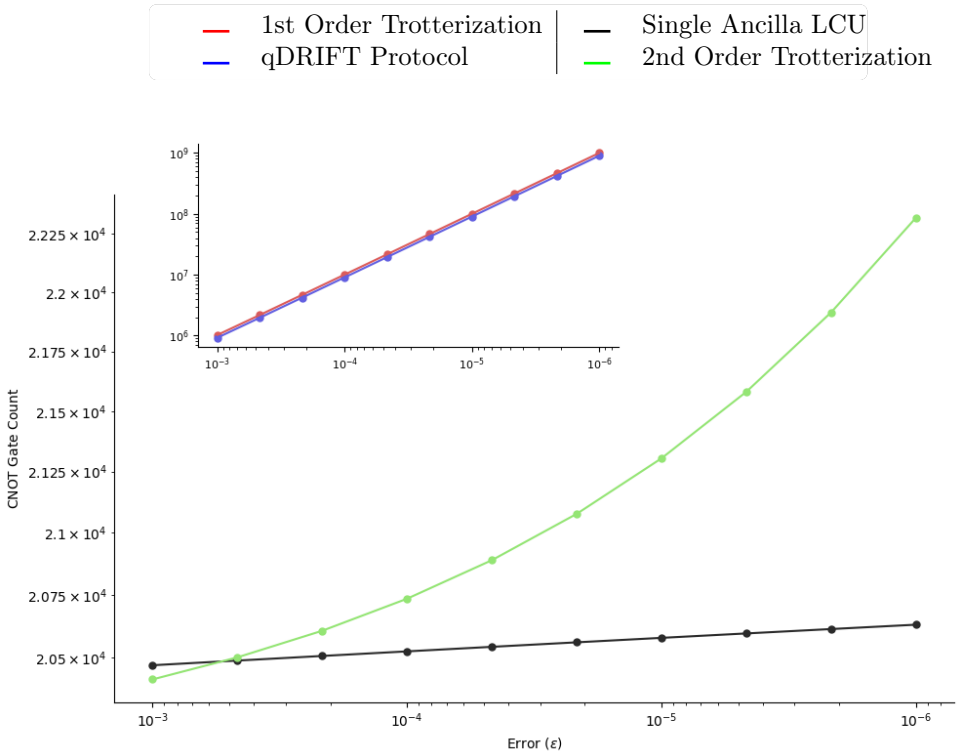


Figure 6.3: Scaling of CNOT gate count per coherent run versus target error ε for a ten-qubit non-Markovian simulation. The main panel uses a linear y-axis (CNOTs) with a logarithmic x-axis (ε); the inset shows the data on a log–log scale. Across the tested regime, the single-ancilla LCU (black) attains the lowest gate counts and the most favorable scaling in ε. First-order Trotterization (red), and qDRIFT (blue) are comparable in this range. The second-order Trotterization (green) is comparable to the single-ancilla LCU in the low precision regime, but quickly becomes less competitive as ε decreases. Gate costs for environment–environment partial swaps are excluded and identical across methods.

Chapter 7

Discussion and Outlook

7.1 Conclusion

This thesis presents a comprehensive framework for simulating open quantum systems on early fault-tolerant quantum computers using collision models. The central challenge we address is the simulation of dissipative quantum dynamics under the stringent resource constraints of near-term fault-tolerant devices.

Our approach leverages collision models, which decompose complex system-environment interactions into a sequence of discrete, manageable collisions between the system and individual environmental subsystems. This framework naturally accommodates both Markovian dynamics, where the environment has no memory, and non-Markovian dynamics, where memory effects arise through correlations between environmental subsystems. The key algorithmic innovation is the development of randomized quantum algorithms that can estimate expectation values of observables after an arbitrary number of such collisions.

The operational advantages of collision models are substantial. Unlike most direct approaches to Lindbladian simulation, which typically require complex block-encodings and deep quantum circuits, collision models offer a modular structure that translates directly into shallow, resource-efficient quantum circuits. Each collision involves only the system and a single environmental subsystem, avoiding the need for large ancilla registers and complex control operations that plague alternative methods.

We conducted a comprehensive analysis of different Hamiltonian simulation techniques within the collision model framework. By comparing Trotterization, qDRIFT, and the single-ancilla Linear Combination of Unitaries (SA-LCU) method, we identified the optimal approach for different parameter regimes. Our analysis reveals that SA-LCU provides superior scaling for precision-critical applications, outperforming low-order Trotter methods and qDRIFT in terms of overall complexity. However, higher-order Trotter formulas achieve the best asymptotic circuit depth scaling of $\widetilde{O}(t^2/\epsilon)$, which matches fundamental lower bounds for collision-model-based simulation of Lindbladian dynamics.

The framework extends naturally to non-Markovian scenarios through a simple but powerful mechanism: allowing interactions between consecutive environmental subsystems before they are discarded. This creates a controllable pathway for information to flow back from the environment to the system, generating memory effects while preserving the modular structure that makes collision models computationally tractable. Importantly, our non-Markovian algorithms maintain similar resource requirements to their Markovian counterparts, contrasting sharply with direct non-Markovian simulation methods that remain prohibitively expensive for early fault-tolerant hardware.

These contributions establish collision models as a practical and versatile platform for open-system simulation. The framework bridges the gap between the heuristic approaches of the NISQ era and the resource-intensive algorithms designed for fully fault-tolerant quantum computers, providing a rigorous yet implementable solution for the crucial intermediate regime of early fault-tolerant quantum computing.

7.2 Future Directions and Open Problems

The results presented here open up several promising avenues for future research, both in improving existing methods and in extending their scope.

Improving simulation efficiency

While our results match known lower bounds for collision-model-based simulation, there is scope to improve the performance based on the recently developed extrapolation techniques have been shown to reduce the depth of Trotterization [105, 106], qDRIFT [107], and more general randomized algorithms [108]. An important open question is whether such extrapolation can be combined with collision models to achieve near-optimal complexity even with low-order methods. In this way, one might approach the optimal t^2/ϵ scaling without resorting to high-order product formulas. Recently, the work by [109] showed this to be indeed possible. But a more detailed analysis with end-to-end method is still needed.

Direct Lindbladian simulation

A new approaches to Lindbladian simulation has been recently proposed Ref. [110]. This work demonstrates that if the Lindblad jump operators can be expressed as linear combinations of Pauli strings, then $e^{\mathcal{L}t}$ can be written as a linear combination of unitaries, similar to SA-LCU for Hamiltonians [50, 51, 52]. Their algorithm estimates expectation values with ϵ-additive accuracy in $O(\|O\|^2/\epsilon^2)$ repetitions and achieves circuit depth $\mathcal{O}(t^2 \log(1/\epsilon))$ using only $n + 4$ qubits. This represents an exponentially better depth compared to procedures based on collision models. Understanding the trade-offs between such direct Lindbladian approaches and the modularity of collision models is a compelling direction for further study.

Time-dependent Lindbladian dynamics

Another natural extension is the simulation of Lindblad equations with time-dependent coefficients. Establishing convergence between such dynamics and time-dependent collision models would allow one to leverage algorithms for simulating time-dependent Hamiltonians [111, 112, 113, 114]. This would broaden the applicability of our framework to scenarios involving driven open systems or systems coupled to environments with explicitly time-varying properties.

Thermalization and state preparation

Beyond dynamics, collision models provide a natural setting for studying thermalization and preparing Gibbs states. Since repeated collisions can drive a system toward equilibrium with an effective bath, one could design early fault-tolerant algorithms for preparing thermal states using only shallow circuits and few ancillas. Thermal state preparation is a crucial primitive for quantum chemistry, condensed matter, and statistical mechanics, yet existing methods such as quantum Metropolis sampling or LCU-based Gibbs sampling demand significant resources. Developing collision-model-based protocols for Gibbs state preparation could provide the first practical algorithms for thermalization in the early fault-tolerant era. Recent results [115] have shown more work in this direction.

Other open problems

Several broader questions remain. Are there efficient ways to incorporate error mitigation directly into the collision framework? Can collision models achieve near optimal scaling for open-system problems? What are the more general classes of open-system problems that collision models can simulate? What classes of open-system problems provide separations between collision-model algorithms and classical methods, pointing toward the possibility of demonstrating genuine quantum advantage in this domain? Can such problems also find relevant physical applications in the near-term?

In summary, this thesis has shown that collision models can serve as a foundation for early fault-tolerant algorithms for open-system simulation. They strike a balance between physical relevance and algorithmic feasibility, and they point toward a future in which rigorous simulation of dissipative quantum dynamics is achievable on the first generations of error-corrected quantum computers. Exploring the open problems outlined above will help determine how far this framework can be pushed, and how it might connect to broader goals of the field.

Bibliography

[1] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936. doi:10.1112/plms/s2-42.2.230.

[2] Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, 1936. doi:10.2307/2371045.

[3] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985. doi:10.1098/rspa.1985.0070.

[4] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992. doi:10.1098/rspa.1992.0167.

[5] Daniel R Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. doi:10.1137/S0097539796298637.

[6] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip*= re. *Communications of the ACM*, 64(11):131–138, 2021. doi:10.1145/3485628.

[7] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 212–219, 1996. doi:10.1145/237814.237866. URL https://dl.acm.org/doi/10.1145/237814.237866.

[8] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *SIAM Journal on Computing*, volume 26, pages 1484–1509, 1997. doi:10.1137/S0097539795293172. URL https://doi.org/10.1137/S0097539795293172.

[9] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103(15):150502, 2009. URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.103.150502.

[10] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6–7):467–488, June 1982. ISSN 1572-9575. doi:10.1007/bf02650179. URL http://dx.doi.org/10.1007/BF02650179.

[11] Daniel Jaschke, Simone Montangero, and Lincoln D Carr. One-dimensional many-body entangled open quantum systems with tensor network methods. *Quantum Science and Technology*, 4(1):013001, November 2018. ISSN 2058-9565. doi:10.1088/2058-9565/aae724. URL http://dx.doi.org/10.1088/2058-9565/aae724.

[12] Dawid A. Hryniuk and Marzena H. Szymańska. Tensor-network-based variational monte carlo approach to the non-equilibrium steady state of open quantum systems. *Quantum*, 8:1475, September 2024. ISSN 2521-327X. doi:10.22331/q-2024-09-17-1475. URL http://dx.doi.org/10.22331/q-2024-09-17-1475.

[13] Richard Cleve and Chunhao Wang. Efficient quantum algorithms for simulating Lindblad evolution. In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80, pages 17:1–17:14. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.17.

[14] Haoya Li, Hongkang Ni, and Lexing Ying. On adaptive low-depth quantum algorithms for robust multiple-phase estimation, 2023. URL https://arxiv.org/abs/2303.08099.

[15] Zhiyan Ding, Xiantao Li, and Lin Lin. Simulating open quantum systems using hamiltonian simulations. *PRX Quantum*, 5(2), May 2024. ISSN 2691-3399. doi:10.1103/prxquantum.5.020332. URL http://dx.doi.org/10.1103/PRXQuantum.5.020332.

[16] Laurent Bruneau, Alain Joye, and Marco Merkli. Repeated interactions in open quantum systems. *Journal of Mathematical Physics*, 55:075204, 2014. doi:10.1063/1.4879240.

[17] Francesco Ciccarello, Salvatore Lorenzo, Vittorio Giovannetti, and G. Massimo Palma. Quantum collision models: Open system dynamics from repeated interactions. *Physics Reports*, 954:1–70, 2022. ISSN 0370-1573. doi:10.1016/j.physrep.2022.01.001.

[18] F. Ciccarello, G. M. Palma, and V. Giovannetti. Collision-model-based approach to non-Markovian quantum dynamics. *Phys. Rev. A*, 87:040103, Apr 2013. doi:10.1103/PhysRevA.87.040103.

[19] Marco Cattaneo, Gabriele De Chiara, Sabrina Maniscalco, Roberta Zambrini, and Gian Luca Giorgi. Collision models can efficiently simulate any multipartite Markovian quantum dynamics. *Phys. Rev. Lett.*, 126:130403, Apr 2021. doi:10.1103/PhysRevLett.126.130403.

[20] Philipp Strasberg, Gernot Schaller, Tobias Brandes, and Massimiliano Esposito. Quantum and information thermodynamics: A unifying framework based on repeated interactions. *Physical Review X*, 7(2):021003, 2017. doi:10.1103/PhysRevX.7.021003.

[21] Felipe Barra. Dissipative charging of a quantum battery. *Phys. Rev. Lett.*, 122:210601, May 2019. doi:10.1103/PhysRevLett.122.210601. URL https://link.aps.org/doi/10.1103/PhysRevLett.122.210601.

[22] Stella Seah, Martí Perarnau-Llobet, Géraldine Haack, Nicolas Brunner, and Stefan Nimmrichter. Quantum speed-up in collisional battery charging. *Phys. Rev. Lett.*, 127:100601, Aug 2021. doi:10.1103/PhysRevLett.127.100601. URL https://link.aps.org/doi/10.1103/PhysRevLett.127.100601.

[23] S. Lorenzo, R. McCloskey, F. Ciccarello, M. Paternostro, and G. M. Palma. Landauer's principle in multipartite open quantum system dynamics. *Phys. Rev. Lett.*, 115:120403, Sep 2015. doi:10.1103/PhysRevLett.115.120403. URL https://link.aps.org/doi/10.1103/PhysRevLett.115.120403.

[24] Valerio Scarani, Mário Ziman, Peter Štelmachovič, Nicolas Gisin, and Vladimír Bužek. Thermalizing quantum machines: Dissipation and entanglement. *Phys. Rev. Lett.*, 88:097905, Feb 2002. doi:10.1103/PhysRevLett.88.097905.

[25] Angsar Manatuly, Wolfgang Niedenzu, Ricardo Román-Ancheyta, Barış Çakmak, Özgür E. Müstecaplıoğlu, and Gershon Kurizki. Collectively enhanced thermalization via multiqubit collisions. *Phys. Rev. E*, 99:042145, Apr 2019. doi:10.1103/PhysRevE.99.042145.

[26] Philipp Strasberg, Gernot Schaller, Tobias Brandes, and Massimiliano Esposito. Quantum and information thermodynamics: A unifying framework based on repeated interactions. *Phys. Rev. X*, 7:021003, Apr 2017. doi:10.1103/PhysRevX.7.021003.

[27] Franklin L. S. Rodrigues, Gabriele De Chiara, Mauro Paternostro, and Gabriel T. Landi. Thermodynamics of weakly coherent collisional models. *Phys. Rev. Lett.*, 123:140601, Oct 2019. doi:10.1103/PhysRevLett.123.140601.

[28] Heather Leitch, Nicolò Piccione, Bruno Bellomo, and Gabriele De Chiara. Driven quantum harmonic oscillators: A working medium for thermal machines. *AVS Quantum Science*, 4:012001, Feb 2022. ISSN 2639-0213. doi:10.1116/5.0072067.

[29] Daniel Grimmer, Eric Brown, Achim Kempf, Robert B. Mann, and Eduardo Martín-Martínez. Gaussian ancillary bombardment. *Phys. Rev. A*, 97:052120, May 2018. doi:10.1103/PhysRevA.97.052120.

[30] Kenza Hammam, Heather Leitch, Yassine Hassouni, and Gabriele De Chiara. Exploiting coherence for quantum thermodynamic advantage. *New Journal of Physics*, 24:113053, November 2022. ISSN 1367-2630. doi:10.1088/1367-2630/aca49b.

[31] Francesco Ciccarello. Collision models in quantum optics. *Quantum Measurements and Quantum Metrology*, 4(1):53–63, 2017. doi:doi:10.1515/qmetro-2017-0007. URL https://doi.org/10.1515/qmetro-2017-0007.

[32] Arne L. Grimsmo. Time-delayed quantum feedback control. *Phys. Rev. Lett.*, 115:060402, Aug 2015. doi:10.1103/PhysRevLett.115.060402.

[33] S J Whalen, A L Grimsmo, and H J Carmichael. Open quantum systems with delayed coherent feedback. *Quantum Science and Technology*, 2(4):044008, August 2017. ISSN 2058-9565. doi:10.1088/2058-9565/aa8331.

[34] Hannes Pichler and Peter Zoller. Photonic circuits with time delays and quantum feedback. *Phys. Rev. Lett.*, 116:093601, Mar 2016. doi:10.1103/PhysRevLett.116.093601.

[35] Kevin A. Fischer, Rahul Trivedi, Vinay Ramasesh, Irfan Siddiqi, and Jelena Vučković. Scattering into one-dimensional waveguides from a coherently-driven quantum-optical system. *Quantum*, 2:69, May 2018. ISSN 2521-327X. doi:10.22331/q-2018-05-28-69.

[36] Kevin Fischer. Derivation of the quantum-optical master equation based on coarse-graining of time. *Journal of Physics Communications*, 2(9):091001, sep 2018. doi:10.1088/2399-6528/aadaf8.

[37] Dario Cilluffo, Angelo Carollo, Salvatore Lorenzo, Jonathan A. Gross, G. Massimo Palma, and Francesco Ciccarello. Collisional picture of quantum optics with giant emitters. *Phys. Rev. Res.*, 2:043070, Oct 2020. doi:10.1103/PhysRevResearch.2.043070. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.043070.

[38] Jonathan A Gross, Carlton M Caves, Gerard J Milburn, and Joshua Combes. Qubit models of weak continuous measurements: Markovian conditional and open-system dynamics. *Quantum Science and Technology*, 3(2):024005, February 2018. ISSN 2058-9565. doi:10.1088/2058-9565/aaa39f.

[39] Stella Seah, Stefan Nimmrichter, Daniel Grimmer, Jader P. Santos, Valerio Scarani, and Gabriel T. Landi. Collisional quantum thermometry. *Phys. Rev. Lett.*, 123:180602, Oct 2019. doi:10.1103/PhysRevLett.123.180602. URL https://link.aps.org/doi/10.1103/PhysRevLett.123.180602.

[40] Angeline Shu, Stella Seah, and Valerio Scarani. Surpassing the thermal cramér-rao bound with collisional thermometry. *Phys. Rev. A*, 102:042417, Oct 2020. doi:10.1103/PhysRevA.102.042417. URL https://link.aps.org/doi/10.1103/PhysRevA.102.042417.

[41] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 792–809, 2015. doi:10.1109/FOCS.2015.54.

[42] Dominic W Berry, Andrew M Childs, Richard Cleve, Robin Kothari, and Rolando D Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Phys. Rev. Lett.*, 114(9):090502, 2015. doi:10.1103/PhysRevLett.114.090502.

[43] Guang Hao Low and Isaac L Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.*, 118(1):010501, 2017. doi:10.1103/PhysRevLett.118.010501.

[44] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019. doi:10.22331/q-2019-07-12-163.

[45] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996. doi:10.1126/science.273.5278.1073. URL https://doi.org/10.1126/science.273.5278.1073.

[46] Andrew M Childs, Dmitri Maslov, Yunseong Nam, Neil J Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018. doi:10.1073/pnas.1801723115.

[47] Earl Campbell. Random compiler for fast hamiltonian simulation. *Phys. Rev. Lett.*, 123(7):070503, 2019. doi:10.1103/PhysRevLett.123.070503.

[48] 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):011020, 2021. doi:10.1103/PhysRevX.11.011020.

[49] Matthew Pocrnic, Dvira Segal, and Nathan Wiebe. Quantum simulation of Lindbladian dynamics via repeated interactions. *arXiv:2312.05371*, 2023. doi:10.48550/arXiv.2312.05371.

[50] Kianna Wan, Mario Berta, and Earl T. Campbell. Randomized quantum algorithm for statistical phase estimation. *Phys. Rev. Lett.*, 129:030503, Jul 2022. doi:10.1103/PhysRevLett.129.030503.

[51] Samson Wang, Sam McArdle, and Mario Berta. Qubit-efficient randomized quantum algorithms for linear algebra. *PRX Quantum*, 5:020324, Apr 2024. doi:10.1103/PRXQuantum.5.020324.

[52] Shantanav Chakraborty. Implementing any linear combination of unitaries on intermediate-term quantum computers. *Quantum*, 8:1496, 2024. doi:10.22331/q-2024-10-10-1496.

[53] Daniel Gottesman. The heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006*, 1998. URL https://arxiv.org/abs/quant-ph/9807006.

[54] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70 (5):052328, 2004. doi:10.1103/PhysRevA.70.052328.

[55] Alexei Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, 1997. doi:10.1070/RM1997v052n06ABEH002155.

[56] Daniel A. Lidar. Lecture notes on the theory of open quantum systems, 2020. URL https://arxiv.org/abs/1902.00967.

[57] Francesco Ciccarello, Salvatore Lorenzo, Vittorio Giovannetti, and G. Massimo Palma. Quantum collision models: Open system dynamics from repeated interactions. *Physics Reports*, 954:1–70, April 2022. ISSN 0370-1573. doi:10.1016/j.physrep.2022.01.001. URL http://dx.doi.org/10.1016/j.physrep.2022.01.001.

[58] Matthew Pocrnic, Dvira Segal, and Nathan Wiebe. Quantum simulation of lindbladian dynamics via repeated interactions, 2025. URL https://arxiv.org/abs/2312.05371.

[59] Vittorio Gorini, Andrzej Kossakowski, and E C G Sudarshan. Completely positive dynamical semigroups of n-level systems. *Journal of Mathematical Physics*, 17(5):821–825, 1976. doi:10.1063/1.522979.

[60] Göran Lindblad. On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics*, 48(2):119–130, 1976. doi:10.1007/BF01608499.

[61] Heinz-Peter Breuer, Elsi-Mari Laine, and Jyrki Piilo. Measure for the degree of non-markovian behavior of quantum processes in open systems. *Physical Review Letters*, 103(21), November 2009. ISSN 1079-7114. doi:10.1103/physrevlett.103.210401. URL http://dx.doi.org/10.1103/PhysRevLett.103.210401.

[62] Elsi-Mari Laine, Jyrki Piilo, and Heinz-Peter Breuer. Measure for the non-markovianity of quantum processes. *Physical Review A*, 81(6), June 2010. ISSN 1094-1622. doi:10.1103/physreva.81.062115. URL http://dx.doi.org/10.1103/PhysRevA.81.062115.

[63] Ángel Rivas, Susana F. Huelga, and Martin B. Plenio. Entanglement and non-markovianity of quantum evolutions. *Physical Review Letters*, 105(5), July 2010. ISSN 1079-7114. doi:10.1103/physrevlett.105.050403. URL http://dx.doi.org/10.1103/PhysRevLett.105.050403.

[64] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985. doi:10.1098/rspa.1985.0070. URL https://royalsocietypublishing.org/doi/10.1098/rspa.1985.0070.

[65] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992. URL https://royalsocietypublishing.org/doi/10.1098/rspa.1992.0167.

[66] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5): 1474–1483, 1997. doi:10.1137/S0097539796298637. URL https://epubs.siam.org/doi/10.1137/S0097539796298637.

[67] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 124–134, 1994. doi:10.1109/SFCS.1994.365700. URL https://doi.org/10.1109/SFCS.1994.365700.

[68] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing (STOC '96)*, pages 212–219, 1996. doi:10.1145/237814.237866. URL https://doi.org/10.1145/237814.237866.

[69] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations, 2012. URL https://arxiv.org/abs/1202.5822.

[70] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 193–204, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367059. doi:10.1145/3313276.3316366. URL https://doi.org/10.1145/3313276.3316366.

[71] Li Wan, Xin Zhang, Xin Wang, and Guilu Long. Block-encoding-based quantum algorithm for linear systems. *Physical Review A*, 104(6):062414, 2021. doi:10.1103/PhysRevA.104.062414. URL https://journals.aps.org/pra/abstract/10.1103/PhysRevA.104.062414.

[72] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002. doi:10.1090/conm/305/05215. URL https://doi.org/10.1090/conm/305/05215.

[73] Ashley Montanaro. Quantum speedup of monte carlo methods. *Proceedings of the Royal Society A*, 471 (2181):20150301, 2015. doi:10.1098/rspa.2015.0301. URL https://royalsocietypublishing.org/doi/10.1098/rspa.2015.0301.

[74] Guang Hao Low and Isaac L. Chuang. Optimal hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118(1):010501, 2017. doi:10.1103/PhysRevLett.118.010501. URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.118.010501.

[75] Andrew M Childs, Robin Kothari, and Rolando D Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6): 1920–1950, 2017. doi:10.1137/16m1087072. URL http://dx.doi.org/10.1137/16M1087072.

[76] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13):130503, 2014. doi:10.1103/PhysRevLett.113.130503. URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.113.130503.

[77] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995. URL https://arxiv.org/abs/quant-ph/9511026.

[78] Robert B. Griffiths and Chi-Sheng Niu. Semiclassical fourier transform for quantum computation. *Physical Review Letters*, 76(17):3228–3231, 1996. doi:10.1103/PhysRevLett.76.3228. URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.76.3228.

[79] Daniel Gottesman. Stabilizer codes and quantum error correction. *arXiv preprint quant-ph/9705052*, 1997. URL https://arxiv.org/abs/quant-ph/9705052.

[80] John Preskill. Reliable quantum computers. *Proceedings of the Royal Society A*, 454(1969):385–410, 1998. doi:10.1098/rspa.1998.0167. URL https://royalsocietypublishing.org/doi/10.1098/rspa.1998.0167.

[81] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and John Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, 2012. doi:10.1103/PhysRevA.86.032324. URL https://journals.aps.org/pra/abstract/10.1103/PhysRevA.86.032324.

[82] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Physical Review A*, 71(2):022316, 2005. doi:10.1103/PhysRevA.71.022316. URL https://journals.aps.org/pra/abstract/10.1103/PhysRevA.71.022316.

[83] Craig Gidney and Austin G. Fowler. How to factor 2048-bit rsa integers in 8 hours using 20 million noisy qubits. *Quantum*, 3:163, 2019. doi:10.22331/q-2019-04-08-135. URL https://quantum-journal.org/papers/q-2019-04-08-135/.

[84] Daniel Litinski. A game of surface codes: Large-scale quantum computing with lattice surgery. *Quantum*, 3:128, 2019. doi:10.22331/q-2019-03-05-128. URL https://quantum-journal.org/papers/q-2019-03-05-128/.

[85] Guang Hao Low and Isaac L. Chuang. Methodology of resonant equiangular composite quantum gates and qubitization. *arXiv preprint arXiv:1610.06546*, 2016. URL https://arxiv.org/abs/1610.06546.

[86] Masuo Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations. *Physics Letters A*, 146(6):319–323, 1990. doi:10.1016/0375-9601(90)90962-N. URL https://doi.org/10.1016/0375-9601(90)90962-N.

[87] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 116(17):201907306, 2019. doi:10.1073/pnas.1907306116. URL https://doi.org/10.1073/pnas.1907306116.

[88] Andrew M. Childs and Yuan Su. Nearly optimal lattice simulation by product formulas. *Physical Review Letters*, 123(5):050503, 2019. doi:10.1103/PhysRevLett.123.050503.

[89] 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):011020, 2021. doi:10.1103/PhysRevX.11.011020.

[90] Qiyao Liang, Yiqing Zhou, Archismita Dalal, and Peter D. Johnson. Modeling the performance of early fault-tolerant quantum algorithms. *Physical Review Research*, 6(2):023118, 2024. doi:10.1103/PhysRevResearch.6.023118. Case study on Randomized Fourier Estimation (RFE) vs. traditional QPE.

[91] John S. Nelson et al. Assessment of quantum phase estimation protocols for early fault-tolerant quantum computers. *Physical Review A*, 110(4):042420, 2024. doi:10.1103/PhysRevA.110.042420.

[92] A. Katabarwa et al. Early fault-tolerant quantum computing. *PRX Quantum*, 5:020101, 2024. doi:10.1103/PRXQuantum.5.020101.

[93] Shantanav Chakraborty, Soumyabrata Hazra, Tongyang Li, Changpeng Shao, Xinzhao Wang, and Yuxin Zhang. Quantum singular value transformation without block encodings: Near-optimal complexity with minimal ancilla. *arXiv:2504.02385*, 2025. URL https://arxiv.org/abs/2504.02385.

[94] Yutaro Akahoshi, Riki Toshio, Jun Fujisaki, Hirotaka Oshima, Shintaro Sato, and Keisuke Fujii. Compilation of trotter-based time evolution for partially fault-tolerant quantum computing architecture, 2024. URL https://arxiv.org/abs/2408.14929.

[95] Raul Campos, Benjamin J. Brown, and et al. Early fault-tolerant quantum error correction: bridging the gap between nisq and ftqc. *arXiv preprint arXiv:2408.14929*, 2024. URL https://arxiv.org/abs/2408.14929.

[96] Giacomo Cimini et al. Early fault-tolerant quantum algorithms in practice: Application to ground-state energy estimation. *arXiv:2405.03754*, 2025. URL https://arxiv.org/abs/2405.03754.

[97] Scott Aaronson and Patrick Rall. Quantum approximate counting, simplified. In *Symposium on simplicity in algorithms*, pages 24–32. SIAM, 2020. doi:10.1137/1.9781611976014.5.

[98] Dmitry Grinko, Julien Gacon, Christa Zoufal, and Stefan Woerner. Iterative quantum amplitude estimation. *npj Quantum Information*, 7(1):52, 2021. doi:10.1038/s41534-021-00379-1.

[99] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. The Power of Block-Encoded Matrix Powers: Improved Regression Techniques via Faster Hamiltonian Simulation. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2019.33.

[100] Andrew M Childs and Tongyang Li. Efficient simulation of sparse Markovian quantum dynamics. *Quantum Information & Computation*, 17(11-12):901–947, 2017. doi:10.26421/QIC17.11-12-1.

[101] Xiantao Li and Chunhao Wang. Simulating Markovian open quantum systems using higher-order series expansion. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261, pages 87:1–87:20. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2023. doi:10.4230/LIPIcs.ICALP.2023.87.

[102] Zhiyan Ding, Xiantao Li, and Lin Lin. Simulating open quantum systems using Hamiltonian simulations. *PRX Quantum*, 5(2):020332, 2024. doi:10.1103/PRXQuantum.5.020332.

[103] Evan Borras and Milad Marvian. Quantum algorithm to simulate lindblad master equations. *Phys. Rev. Res.*, 7:023076, Apr 2025. doi:10.1103/PhysRevResearch.7.023076. URL https://link.aps.org/doi/10.1103/PhysRevResearch.7.023076.

[104] Subir Sachdev. *Quantum Phase Transitions*. Cambridge University Press, 2nd edition, 2011. doi:10.1017/CBO9780511973765.

[105] James D. Watson and Jacob Watkins. Exponentially reduced circuit depths using Trotter error mitigation. *arXiv:2408.14385*, 2024. doi:10.48550/arXiv.2408.14385.

[106] Gumaro Rendon, Jacob Watkins, and Nathan Wiebe. Improved accuracy for trotter simulations using chebyshev interpolation. *Quantum*, 8:1266, February 2024. ISSN 2521-327X. doi:10.22331/q-2024-02-26-1266. URL http://dx.doi.org/10.22331/q-2024-02-26-1266.

[107] James D Watson. Randomly compiled quantum simulation with exponentially reduced circuit depths. *arXiv preprint arXiv:2411.04240*, 2024. doi:10.48550/arXiv.2411.04240.

[108] Shantanav Chakraborty, Soumyabrata Hazra, Tongyang Li, Changpeng Shao, Xinzhao Wang, and Yuxin Zhang. Quantum singular value transformation without block encodings: Near-optimal complexity with minimal ancilla. *arXiv preprint arXiv:2504.02385*, 2025. doi:10.48550/arXiv.2504.02385.

[109] Pegah Mohammadipour and Xiantao Li. Reducing circuit depth in lindblad simulation via step-size extrapolation, 2025.

[110] Jumpei Kato, Kaito Wada, Kosuke Ito, and Naoki Yamamoto. Exponentially accurate open quantum simulation via randomized dissipation with minimal ancilla. *arXiv preprint arXiv:2412.19453*, 2024. doi:10.48550/arXiv.2412.19453.

[111] Dominic W. Berry, Andrew M. Childs, Yuan Su, Xin Wang, and Nathan Wiebe. Time-dependent Hamiltonian simulation with L^1-norm scaling. *Quantum*, 4:254, April 2020. ISSN 2521-327X. doi:10.22331/q-2020-04-20-254. URL https://doi.org/10.22331/q-2020-04-20-254.

[112] Yi-Hsiang Chen, Amir Kalev, and Itay Hen. Quantum algorithm for time-dependent hamiltonian simulation by permutation expansion. *PRX Quantum*, 2:030342, Sep 2021. doi:10.1103/PRXQuantum.2.030342. URL https://link.aps.org/doi/10.1103/PRXQuantum.2.030342.

[113] Jacob Watkins, Nathan Wiebe, Alessandro Roggero, and Dean Lee. Time-dependent hamiltonian simulation using discrete-clock constructions. *PRX Quantum*, 5:040316, Oct 2024. doi:10.1103/PRXQuantum.5.040316. URL https://link.aps.org/doi/10.1103/PRXQuantum.5.040316.

[114] Di Fang, Diyi Liu, and Rahul Sarkar. Time-dependent hamiltonian simulation via magnus expansion: Algorithm and superconvergence. *Communications in Mathematical Physics*, 406(6):1–36, 2025. doi:10.1007/s00220-025-05314-5. URL https://doi.org/10.1007/s00220-025-05314-5.

[115] Zhiyan Ding, Yongtao Zhan, John Preskill, and Lin Lin. End-to-end efficient quantum thermal and ground state preparation made simple, 2025. URL https://arxiv.org/abs/2508.05703.

[116] Mary Beth Ruskai. Inequalities for traces on von Neumann algebras. *Communications in Mathematical Physics*, 26:280–289, 1972. doi:10.1007/BF01645523.

Appendix

A.1 Distances between quantum states

In this section, we prove some results concerning the distance between operators/ CPTP maps applied to quantum states. First, consider that there exist two operators P and Q such that $\|P-Q\| \leq \gamma$. We demonstrate that the expectation value of O with respect to $P\rho P^\dagger$ is not far off from the expectation value of O with respect to $Q\rho Q^\dagger$, for any density matrix ρ. More precisely, we prove that

$$\left|\mathrm{Tr}[O\ P\rho P^\dagger] - \mathrm{Tr}[O\ Q\rho Q^\dagger]\right| \leq 3\|P\|\|O\|\gamma.$$

The result was proven in Refs. [52, 108], and we state this here for completeness. Let us recall the tracial version of Hölder's inequality, which is stated below for completeness:

Lemma A.1.1 (Tracial version of Hölder's inequality [116])**.** *Define two operators A and B and parameters $p, q \in [1,\infty]$ such that $1/p + 1/q = 1$. Then the following holds:*

$$\mathrm{Tr}[A^\dagger B] \leq \|A\|_p \|B\|_q.$$

Here $\|X\|_p$ corresponds to the Schatten p-norm of the operator X. For the special case of $p=\infty$ and $q=1$, the statement of Lemma A.1.1 can be rewritten as

$$\mathrm{Tr}[A^\dagger B] = \|A^\dagger B\|_1 \leq \|A\|_\infty \|B\|_1 = \|A\|\|B\|_1. \tag{A.1.1}$$

Now we are in a position to formally state the main result.

Theorem A.1.2. *Suppose P and Q are operators such that $\|P-Q\| \leq \gamma$ for some $\gamma \in [0,1]$. Furthermore, let ρ be any density matrix and O be some Hermitian operator with spectral norm $\|O\|$. Then, if $\|P\| \geq 1$, the following holds:*

$$\left|\mathrm{Tr}[O\ P\rho P^\dagger] - \mathrm{Tr}[O\ Q\rho Q^\dagger]\right| \leq 3\|O\|\|P\|\gamma.$$

Proof. Using Lemma A.1.1 with $p=\infty$ and $q=1$, we obtain

$$|\mathrm{Tr}[O\ P\rho P^\dagger] - \mathrm{Tr}[O\ Q\rho Q^\dagger]| \leq \|O\| \cdot \|P\rho P^\dagger - Q\rho Q^\dagger\|_1 \tag{A.1.2}$$

For the second term in the RHS of the above equation, we can successively apply the tracial version of Hölder's inequality (Lemma A.1.1 with $p=\infty$ and $q=1$) the triangle inequality to obtain:

$$\begin{aligned}
\left\|P\rho P^\dagger - Q\rho Q^\dagger\right\|_1 &= \left\|P\rho P^\dagger - P\rho Q^\dagger + P\rho Q^\dagger - Q\rho Q^\dagger\right\|_1 && \text{(A.1.3)}\\
&\leq \|P\rho\|_1\|P-Q\| + \|P-Q\|\|\rho Q\|_1 && \text{(A.1.4)}\\
&\leq \|P\|\|P-Q\| + \|Q\|\|P-Q\| \qquad [\text{ As } \|\rho\|_1 = 1] && \text{(A.1.5)}\\
&\leq (\|P\| + \|Q\|)\cdot\|P-Q\| && \text{(A.1.6)}\\
&\leq (\|P\| + \|Q-P+P\|)\cdot\|P-Q\| && \text{(A.1.7)}\\
&\leq (\|P\| + \|P-Q\| + \|P\|)\cdot\|P-Q\| && \text{(A.1.8)}\\
&\leq 2\|P\|\|P-Q\| + \|P-Q\|^2. && \text{(A.1.9)}
\end{aligned}$$

Now, substituting this upper bound back in the RHS of Eq. (A.1.2), we obtain

$$\begin{aligned} \left|\mathrm{Tr}[O\ P\rho P^\dagger] - \mathrm{Tr}[O\ Q\rho Q^\dagger]\right| &\leq \|O\|\|P-Q\|^2 + 2\|O\|\|P\|\|P-Q\| && \text{(A.1.10)}\\ &\leq \|O\|\|P-Q\|^2 + 2\|O\|\|P\|\|P-Q\| \\ &\leq \gamma^2\|O\| + 2\|O\|\|P\|\gamma \\ &\leq 3\gamma\|O\|\|P\| && \text{(A.1.11)} \end{aligned}$$

□

Next, we bound the distance between two quantum states that have been transformed by a composition of two Completely Positive Trace Preserving (CPTP) maps. We have the following Lemma:

Lemma A.1.3 (Distance between quantum states obtained by applying a composition of CPTP maps)**.** *Let $\{\mathcal{A}_i\}_{i=1}^K$ and $\{\mathcal{B}_i\}_{i=1}^K$ be two sets of maps acting on any density operator ρ, such that each $\mathcal{A}_i$ and $\mathcal{B}_i$ are CPTP maps. Assume that for all $i \in [1, K]$, the following bound holds:*

$$\|\mathcal{A}_i[\rho] - \mathcal{B}_i[\rho]\|_1 \leq \varepsilon. \tag{A.1.12}$$

Then, the compositions of these maps satisfy:

$$\left\|\bigcirc_{i=1}^K \mathcal{A}_i[\rho] - \bigcirc_{i=1}^K \mathcal{B}_i[\rho]\right\|_1 \leq K\varepsilon. \tag{A.1.13}$$

Proof. Expanding the composition, we write:

$$\left\|\bigcirc_{i=1}^K \mathcal{A}_i[\rho] - \bigcirc_{i=1}^K \mathcal{B}_i[\rho]\right\|_1 = \|\mathcal{A}_K\mathcal{A}_{K-1}\ldots\mathcal{A}_1[\rho] - \mathcal{B}_K\mathcal{B}_{K-1}\ldots\mathcal{B}_1[\rho]\|_1. \tag{A.1.14}$$

Adding and subtracting intermediate terms iteratively, and using the triangle inequality, we obtain:

$$\begin{aligned} \|\mathcal{A}_K\mathcal{A}_{K-1}\ldots\mathcal{A}_1[\rho] - \mathcal{B}_K\mathcal{B}_{K-1}\ldots\mathcal{B}_1[\rho]\|_1 \leq\ & \|\mathcal{A}_K\mathcal{A}_{K-1}\ldots\mathcal{A}_1[\rho] - \mathcal{B}_K\mathcal{A}_{K-1}\ldots\mathcal{A}_1[\rho]\|_1 + \\ & \|\mathcal{B}_K\mathcal{A}_{K-1}\ldots\mathcal{A}_1[\rho] - \mathcal{B}_K\mathcal{B}_{K-1}\ldots\mathcal{A}_1[\rho]\|_1 + \cdots + \\ & \|\mathcal{B}_K\ldots\mathcal{B}_2\mathcal{A}_1[\rho] - \mathcal{B}_K\ldots\mathcal{B}_2\mathcal{B}_1[\rho]\|_1. \end{aligned} \tag{A.1.15}$$

Now, let $\rho_k^{\mathcal{A}} = \bigcirc_{i=1}^k \mathcal{A}_i[\rho]$ and $\rho_k^{\mathcal{B}} = \bigcirc_{i=1}^k \mathcal{B}_i[\rho]$, representing the intermediate states obtained after the application of k maps. Then due to the initial assumption Eq. (A.1.12)

$$\left\|\mathcal{A}_j[\rho_{j-1}^{\mathcal{A}}] - \mathcal{B}_j[\rho_{j-1}^{\mathcal{B}}]\right\|_1 \leq \varepsilon \tag{A.1.16}$$

Using the contractivity of completely positive maps,

$$\|\mathcal{B}_{j+1}\mathcal{A}_j[\rho_{j-1}^{\mathcal{A}}] - \mathcal{B}_{j+1}\mathcal{B}_j[\rho_{j-1}^{\mathcal{B}}]\|_1 \leq \varepsilon, \tag{A.1.17}$$

Using this repeatedly, we bound each term on the right side of the Eq. (A.1.15) by ε. Thus, by summing the contributions across all maps, we obtain:

$$\left\|\bigcirc_{i=1}^K \mathcal{A}_i[\rho] - \bigcirc_{i=1}^K \mathcal{B}_i[\rho]\right\|_1 \leq K\varepsilon. \tag{A.1.18}$$

□

Now, consider two copies of the quantum state ρ, such that the CPTP map $\mathcal{A}$ has been applied to one copy, while another CPTP map $\mathcal{B}$ has been applied to the second copy to obtain $\mathcal{A}[\rho]$ and $\mathcal{B}[\rho]$, respectively. Now consider unitaries U and $\widetilde{U}$ such that they are close (in spectral norm). Then, we find the distance (1-norm) between the quantum states obtained by applying U to $\mathcal{A}[\rho]$, and $\widetilde{U}$ to $\mathcal{A}[\rho]$, via the following theorem:

Theorem A.1.4 (Distance between quantum states)**.** *Suppose U and $\widetilde{U}$ are unitary while $\mathcal{A}$, $\mathcal{B}$ are CPTP maps. Then for any density operator ρ,*

$$\left\|U\ \mathcal{A}[\rho]U^\dagger - \widetilde{U}\ \mathcal{B}[\rho]\widetilde{U}^\dagger\right\|_1 \leq 2\left\|U - \widetilde{U}\right\| + \|\mathcal{A}[\rho] - \mathcal{B}[\rho]\|_1.$$

Proof. Using triangle inequality, and Tracial version of Hölder's inequality (Lemma A.1.1), we obtain

$$\left\|U\mathcal{A}[\rho]U^\dagger - \widetilde{U}\mathcal{B}[\rho]\widetilde{U}^\dagger\right\|_1 \leq \left\|U\mathcal{A}[\rho]U^\dagger - U\mathcal{A}[\rho]\widetilde{U}^\dagger\right\|_1 + \left\|U\mathcal{A}[\rho]\widetilde{U}^\dagger - \widetilde{U}\mathcal{B}[\rho]\widetilde{U}^\dagger\right\|_1 \tag{A.1.19}$$

$$\leq \left\|U\mathcal{A}[\rho]U^\dagger - U\mathcal{A}[\rho]\widetilde{U}^\dagger\right\|_1 + \left\|U\mathcal{A}[\rho] - \widetilde{U}\mathcal{B}[\rho]\right\|_1 \cdot \left\|\widetilde{U}\right\| \tag{A.1.20}$$

$$= \left\|U\mathcal{A}[\rho]U^\dagger - U\mathcal{A}[\rho]\widetilde{U}^\dagger\right\|_1 + \left\|U\mathcal{A}[\rho] - \widetilde{U}\mathcal{B}[\rho]\right\|_1 \tag{A.1.21}$$

For the first term in the RHS we can use Lemma A.1.1 to obtain

$$\left\|U\mathcal{A}[\rho]U^\dagger - U\mathcal{A}[\rho]\widetilde{U}^\dagger\right\|_1 \leq \|U\mathcal{A}[\rho]\|_1 \cdot \left\|U - \widetilde{U}\right\| \tag{A.1.22}$$

$$\leq \|U\| \cdot \|\mathcal{A}[\rho]\|_1 \cdot \left\|U - \widetilde{U}\right\| \tag{A.1.23}$$

$$\leq \left\|U - \widetilde{U}\right\|. \tag{A.1.24}$$

On the other hand, for the second term,

$$\left\|U\mathcal{A}[\rho] - \widetilde{U}\mathcal{B}[\rho]\right\|_1 = \left\|U\mathcal{A}[\rho] - U\mathcal{B}[\rho] + U\mathcal{B}[\rho] - \widetilde{U}\mathcal{B}[\rho]\right\|_1. \tag{A.1.25}$$

By using triangle inequality once again to the RHS, followed by Lemma A.1.1, we obtain

$$\left\|U\mathcal{A}[\rho] - \widetilde{U}\mathcal{B}[\rho]\right\|_1 \leq \|U\| \cdot \|\mathcal{A}[\rho] - \mathcal{B}[\rho]\|_1 + \left\|U - \widetilde{U}\right\| \cdot \|\mathcal{B}[\rho]\|_1 \tag{A.1.26}$$

$$\leq \|\mathcal{A}[\rho] - \mathcal{B}[\rho]\|_1 + \left\|U - \widetilde{U}\right\|. \tag{A.1.27}$$

So, overall, we have

$$\left\|U\mathcal{A}[\rho]U^\dagger - \widetilde{U}\mathcal{B}[\rho]\widetilde{U}^\dagger\right\|_1 \leq 2\left\|U - \widetilde{U}\right\| + \|\mathcal{A}[\rho] - \mathcal{B}[\rho]\|_1.$$

This completes the proof. □