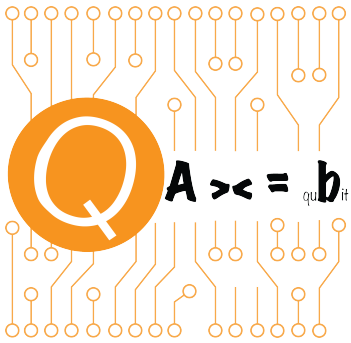
Quantum Algorithms for Scientific Computation

Lin Lin and Nathan Wiebe

March 15, 2026



PRELIMINARY NOTES BEING CONTINUOUSLY UPDATED.

Contents

Part I

Background

Part I of this book sets the stage for our exploration of quantum algorithms for scientific computation by asking two questions: why should we expect quantum computers to offer a computational advantage, and what are the basic mathematical and physical principles that govern them?

Chapter 1 tackles the first question. We begin by tracing the conceptual origins of quantum computing and formalizes the notion of quantum speedup. We then introduce a quantum advantage hierarchy, which classifies applications based on the strength of evidence for quantum speedup.

Chapter 2 addresses the second question by providing a concise overview of elements of quantum computation. We introduce the postulates of quantum mechanics, the circuit model, and the density operator formalism. We also cover concepts such as the no-cloning theorem and the principles of deferred and implicit measurement. The chapter concludes by introducing the operator formalisms for spin, fermionic, and bosonic systems, which are essential for describing the physical problems encountered in scientific applications, and presents several example Hamiltonians that will serve as recurring illustrations throughout the book.

CHAPTER 1

Quantum advantage in scientific computation

In this chapter, we trace the conceptual origins of quantum computing and explain how the physical nature of information suggests that quantum mechanics may offer computational power beyond classical Turing machines. We then formalize the notion of quantum speedup. Any claim of quantum advantage requires accounting for all relevant computational costs, including data input and output. To structure this assessment, we introduce a quantum advantage hierarchy that categorizes problems based on the existing evidence for significant speedups. The chapter concludes with a brief discussion of quantum error correction, and why exponentially large state spaces do not force exponential error accumulation: in fault-tolerant computation, it suffices to implement each gate to an accuracy that scales inversely with the gate count.

1.1. Origin and justification for quantum computing

Our aim in this textbook is to provide a concrete understanding of not only how quantum algorithms work, but more importantly *why* they work and what impact scalable quantum computers are expected to yield in both the scientific and industrial worlds. Underlying this inquiry, however, is a deeper philosophical question about what it means to compute and why probing this question inevitably led to the idea of quantum computing.

Modern computer science traces its roots back to the early 20th century, with luminaries such as Alan Turing, John von Neumann, and Claude Shannon struggling to mathematically describe how information is stored and processed. Turing's great realization was that all such computers could be mathematically modeled by an abstract device called a "Turing Machine". The Turing machine was inspired strongly by the human "computers" (clerks) of the day: it possesses a tape for storing information and a read head that moves along the tape, updating the data on the tape in accordance with a stored program [Tur36].

John von Neumann is often credited with providing the first modern computer architecture that resembles modern computers, featuring dedicated memory, arithmetic and logic units, and input/output capabilities [VN93]. This architecture provided a far more realistic model of the postwar computers that were emerging, but conceptually these devices were no more powerful than the original Turing machine. Specifically, a machine is said to be "Turing Complete" if any function that a Turing machine can compute can be computed on the device. The von Neumann machine (given sufficient memory) can be shown to be Turing Complete, and in fact, a Turing machine can also simulate a machine implementing the von Neumann architecture. In this sense, the device is more than just Turing Complete: it is actually Turing Equivalent. Indeed, all known classical computational systems are Turing equivalent in this sense. This observation means that, effectively, every computational system in the universe could be understood as a Turing machine.

The formal study of algorithms revealed that not all tasks are fundamentally as easy for a Turing Machine. Some tasks, such as deciding whether a program halts, are strictly uncomputable [Tur36].

On the other hand, problems such as multiplying two n-bit numbers can be performed using a number of steps that scales polynomially in n. Still other problems, such as factoring an n-bit integer into a product of primes, can have their solution *verified* in polynomial time, but to date, no efficient algorithm has been found on a Turing machine that can *find* these factors in time that is polynomial in n (despite centuries of study). This suggested that a more fine-grained notion of computability needed to be considered than simply "computable" or "uncomputable". Instead, it was seen to be useful to categorize computational tasks that can be computed on a Turing Machine using a polynomial number of operations as "efficiently computable" and all others as inefficient.

This categorization led to a bold hypothesis, which we will later criticize, known as the **Extended Church-Turing Thesis**. This statement says that any reasonable model of computing can be simulated using a polynomial number of computational steps by a probabilistic Turing machine. The example of von Neumann's model of computing being simulatable in polynomial time by a Turing machine has indeed been reinforced by other models of computing based on physical phenomena, including billiard balls and the Game of Life. However, a challenge would emerge from an unlikely source: fundamental physics.

At the same time as computer science was being developed, a revolution was happening in physics. It had long been observed by physicists such as Planck and Einstein that classical physics could not be used to explain why heated objects (blackbodies) glowed red or how solar panels worked. Indeed, realistic models of these effects based on Newtonian principles failed to predict experimental observations. In the case of the stove elements, this failure was so radical that it predicted that infinite energy would be emitted by a stove burner (the "ultraviolet catastrophe"). A new type of model, formalized by von Neumann and others, was proposed to describe these systems that we now know as quantum mechanics (so named for its prediction that light should be emitted or absorbed in discrete quanta of energy). This language ultimately became the foundation of all fundamental physical law (gravitation being a notable exception).

Subsequent questions from Einstein, Podolsky, Rosen, and developments by Bell showed that quantum mechanics could not reasonably be described by classical local realism. Specifically, a phenomenon known as **entanglement**, which describes the correlations between measurement outcomes of coupled quantum systems, could not be described by classical mechanics without incorporating a non-local mechanism for updating measurement results. This work began to seriously question whether quantum systems could be plausibly described as mechanical systems. This, in turn, would much later be seen to question the Extended Church-Turing Thesis, as a Turing Machine is at its core a classical mechanical object that relies on local interactions.

A surprising feature of quantum mechanics is that its connection to computing seems to have taken several decades to be appreciated, despite us owing John von Neumann a great debt for formalizing both theories. With the benefit of hindsight, it is clear that with the appreciation of the fact that information is physical, quantum computing could have been developed as early as the 1940s.

The physical nature of information was elucidated most clearly by Shannon and Landauer. Shannon showed that the information content of a signal takes the same form as entropy, or disorder, in thermodynamics. Inspired by this connection, Shannon proposed that the two concepts were the same, establishing a link between his mathematical theory of information and thermal physics. Indeed, according to a widely circulated anecdote attributed to Shannon in an article by Tribus, von Neumann may have been agonizingly close to realizing the connection between physics and information processing [TM71]:

> *"What's in a name? In the case of Shannon's measure the naming was not accidental. In 1961 one of us (Tribus) asked Shannon what he had thought about when he had finally confirmed his famous measure. Shannon replied: 'My greatest concern was what to call it. I thought of calling it 'information', but the word was overly used, so I decided to call it 'uncertainty'. When I discussed it with John von Neumann, he had a better idea. Von Neumann told me, 'You should call it entropy, for two reasons. In the first place your uncertainty function has been used in statistical mechanics under that name. In the second place, and more importantly, no one knows what entropy really is, so in a debate you will always have the advantage."*

Indeed, Shannon's work provided strong evidence that the two concepts are in fact the same and that thermodynamics had been telling us a secret lesson about information all along.

Landauer took this insight one step further by showing that thermodynamics places limitations on computers. Specifically, he showed that any computer that performs a calculation at finite temperature must pay an energy price for every bit of information erased to avoid violating the laws of thermodynamics [Lan61]. Similar work studying Maxwell's Demon, a hypothetical agent that can raise and lower a gate that allows fast gas molecules through while blocking slow molecules, revealed that if the thermodynamic cost of measuring and computing were ignored, the laws of thermodynamics could be violated by such an agent [Ben87]. These works showed a strong link between information and physics and laid the foundation for the link to quantum computing that would soon follow.

It took the insight that information is physical to begin to motivate incorporating the formalism of quantum mechanics into the language of computer science. Quantum computing was born of this synthesis and was articulated independently by Manin [Man80] and Feynman [Fey82]. The justification that they had was the fact that the description of the state space of even small quantum systems scales exponentially with the number of quantum bits. This means that a naïve simulation of the laws of quantum mechanics would require exponentially more time on a classical computer than the physical system itself requires to evolve. This work opened the possibility that a computer that exploited the full capabilities of quantum mechanics may be, for certain problems, exponentially more powerful than the Turing machine. This in turn caused the scientific community to begin to doubt that the Extended Church-Turing Thesis holds, and now the belief that any realistic model of computing is polynomially equivalent to a quantum computer has become widespread after the discovery of the fast factoring algorithm of Shor [Sho99], the quantum simulation algorithms of Lloyd and others [Llo96], as well as the quantum advantage proposals of Aaronson and Arkhipov [AA11].

At a high level though, quantum computing suggests something potentially even stronger. If the Extended Church-Turing Thesis is replaced by a quantum version, then all of nature could be described or simulated in polynomial time by a massive quantum computer. In this sense, the strong link between information and physics reaches a crescendo with quantum computing, which suggests that all of physical law could be thought of as an algorithm that is run on a quantum computer, and the set of tasks that a quantum computer cannot perform efficiently are precisely those that nature also cannot solve at scale. For this reason, the search for exponential algorithmic advantage plays a central role in quantum computing, not only because it provides us with new opportunities for our computers, but also because it reveals the limitations that physical systems impose on information processing, and in turn, the limitations that information processing places on physical systems. Indeed, the main purpose of this text is to shed light on the origin and utility of quantum speedups for scientific applications.

1.2. Quantum speedup

The primary aim of exploring quantum computation is to attain a **quantum speedup** or **quantum advantage**, thereby enhancing problem-solving capabilities in scientific computation. At first glance, it seems that n qubits can be used to represent a superposition over 2^n classical basis states, and significant quantum speedups should be expected everywhere. However, the situation is much more ambiguous: does the quantum algorithm require an exponential amount of classical information to be passed into the quantum computer? Does the quantum algorithm generate an exponential amount of information that needs to be extracted out of the quantum computer? If the size of the classical state space is 2^n, is it mandatory for the classical algorithm to go through all states in order to find an approximate solution to a desired precision? If the size of the classical state space is only n but the computational cost of an existing algorithm is 2^n, is it possible for a future classical algorithm to reduce this cost to poly(n)? Readers may be curious about how to evaluate and answer these questions before dedicating substantial time to learning quantum computation. Indeed, these discussions can occur at a relatively broad level, largely circumventing the need for intricate quantum jargon.

One way to formulate the quantum speedup (as a function of the system size n) is

$$\text{Quantum speedup} = \frac{\log(\min \text{Cost(classical)})}{\log \text{Cost(quantum)}}. \tag{1.1}$$

The presence of the logarithm can be intuitively understood as follows. For a task with a "system size" n, assume that the classical and quantum costs are (asymptotically) proportional to n^{α_c} and n^{α_q}, respectively. Then as $n \to \infty$, the quantum speedup defined according to Eq. (1.1) is α_c/α_q. For instance, a *quadratic* quantum speedup means $\alpha_c/\alpha_q = 2$, a *cubic* quantum speedup means $\alpha_c/\alpha_q = 3$, and so on. If $\alpha_c \to \infty$ as $n \to \infty$ but α_q remains bounded, the quantum speedup is *superpolynomial*. There is also a concept called "exponential quantum advantage" (EQA), which suggests that the classical cost increases at least exponentially in n but the quantum cost increases only polynomially.

Rigorous proof of EQA can be extraordinarily difficult for practical problems. For example, given two prime numbers p, q, the product $m = p \cdot q$ can be easily carried out on a classical computer. However, if we are only given the integer m, finding the prime factors p, q can be very challenging. This is called the prime factorization problem and has wide applications in cryptography. The difficulty of the prime factorization problem can be measured in terms of the number of bits in m. An integer m can always be expressed in binary format. For instance, $12 = 2^3 + 2^2$ can be represented as 1100 in binary format, where the number of bits n is 4. The most efficient classical algorithm, judged by asymptotic scaling in n, is the General Number Field Sieve method [Bri98]. The computational scaling is proportional to $\exp[cn^{\frac{1}{3}}(\log n)^{\frac{2}{3}}]$, which increases superpolynomially with n. Shor's celebrated algorithm [Sho94, Sho99] addresses the same problem on a quantum computer, with its cost being proportional to $n^2 \log n \log\log n$, i.e., only polynomial in n. On one hand, this provides a very clean (and so far the cleanest) quantum solution with a significant quantum speedup that is superpolynomial in n. On the other hand, even for this problem, the speedup is not yet exponential in the strict sense above. For practical purposes, we will be (more than) content with a superpolynomial quantum speedup.

In principle, the classical cost should be minimized with respect to *all* classical algorithms, including algorithms that exist today, and those that will ever be developed in the future. A useful lower bound of the cost of classical algorithms may be obtained for some simple problems. However, this undertaking is exceedingly challenging for the majority of scientific computing problems. For

instance, we do not know whether the problem of prime factorization can or cannot be performed in polynomial time. Therefore, for practical purposes, we will further be satisfied with an estimate of $\min \text{Cost}(\text{classical})$ by weighing both theoretical and empirical evidence, based on *existing* classical algorithms.

Although quantum mechanics is frequently described as a probabilistic theory, a key component is actually the quantum wavefunction (or quantum amplitude). This can be roughly equated to the square root of a probability density, along with phase information. This difference between probability density and quantum amplitude often forms the basis of the quadratic speedup, i.e., $\alpha_c/\alpha_q = 2$. The most prominent example of this is Grover's algorithm for unstructured search (see Chapter 11). Although a quadratic speedup is valuable, it is unlikely that this speedup alone will be the most groundbreaking application of early fault-tolerant quantum computers. Hence, we use the loose term *significant* quantum speedup to refer to speedups greater than quadratic (such as cubic or quartic), or better, to superpolynomial speedups.

The quantum cost can be roughly calculated as the total gate complexity multiplied by the number of repetitions due to the measurement process. It is also conceptually useful to divide it into the following three components:

(1) Input cost, or the cost for preparing the input quantum state. Without loss of generality, the quantum algorithm starts from a clean quantum state such as $|0^n\rangle$, and the input state to the quantum algorithm, denoted by $|\psi_I\rangle$, can be prepared using a unitary matrix U_I as $|\psi_I\rangle = U_I|0^n\rangle$. Then the input cost is the gate complexity for implementing U_I. Sometimes a quantum algorithm requires multiple accesses to the input oracle U_I in a coherent fashion. In this case, the input cost is given by the gate complexity for implementing U_I multiplied by the number of coherent initial state preparations.
(2) Output cost, or the cost of quantum measurement. Without loss of generality, after an appropriate basis change the measurement can be taken to be performed on one or multiple qubits in the computational basis at the end of an algorithm. Then the output cost is the number of repetitions M needed to run the quantum algorithm.
(3) Running cost, or the cost of coherently running the quantum algorithm once. This is given by the gate complexity for implementing the algorithm (excluding the cost for implementing U_I).

One reason for separating the total gate complexity into the input cost and the running cost is that it allows us to distinguish the case when the overall cost is dominated by preparing the input, rather than by coherently executing the rest of the algorithm. In many settings, the input information is classical, and the nature of its complexity can be very different from that in the quantum algorithm. There is also an important scenario in which the input state $|\psi_I\rangle$ is not generated by a known circuit U_I, but is produced by a quantum experiment. In this case, the relevant input cost is often the number of times the experiment must be repeated to prepare $|\psi_I\rangle$ (a sample complexity), rather than the gate complexity of a circuit. For instance, **quantum learning theory** studies how efficiently one can infer properties of an unknown quantum state from state preparations and measurements. Throughout this book, we focus on computational tasks in which quantum and classical algorithms have access to the same amount of classical input information and are required to output classical information, and we will not discuss quantum learning theory in detail (except basic concepts such as parameter estimation in Chapter 8).

Ultimately, all quantum algorithms must output information that can be processed through classical means via quantum measurements. If the quantum state itself is the end product, the procedure to recover the quantum state on a classical computer is called quantum state tomography.

The cost of the state tomography procedure usually grows exponentially relative to the size of the quantum system. Therefore, it is unlikely that significant quantum speedup can be achieved for problems involving a tomography procedure on a large number of qubits. Instead, we should focus on problems whose end result can be obtained by measuring a small number of observables related to the quantum state to a desired accuracy, for which the measurement overhead can sometimes be reduced substantially.

In summary, a quantum computer should not be viewed as an all-purpose computational device destined to replace classical computers. Rather, it should be seen as an accelerator, capable of providing significant speedups for specific computational tasks. As emphasized in [Aar14], one must "read the fine print" when evaluating claims of quantum advantage. Several criteria must be met: the problem under consideration should be computationally intensive on classical hardware; the task must be solvable efficiently on a quantum device; and the overhead associated with data input and output (i.e., loading and extracting data) should not dominate the overall cost. Furthermore, several proposed quantum speedups for linear algebra and machine learning on classical data rely on strong data-access assumptions, and in some cases comparable scaling can be achieved by quantum-inspired classical algorithms under similar assumptions. Meeting all of these conditions is far from trivial. It represents a significant theoretical, experimental, and algorithmic challenge for the entire scientific community.

1.3. Quantum advantage hierarchy

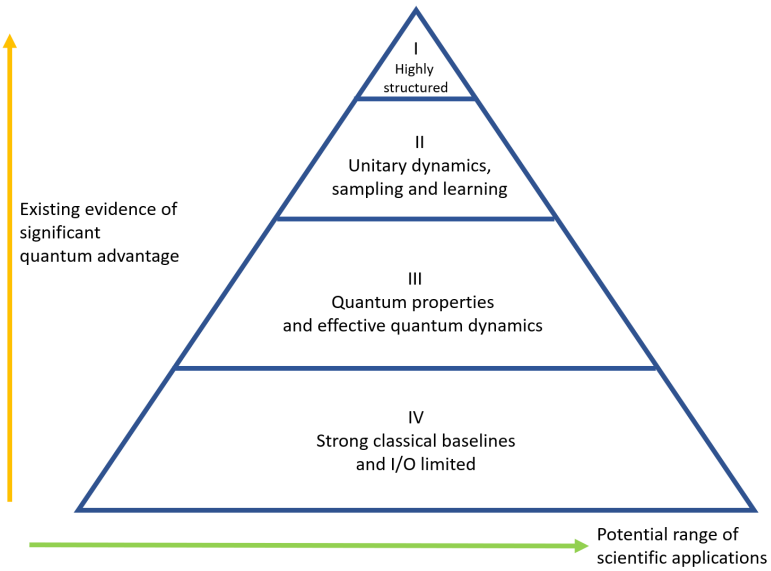


FIGURE 1.1. Quantum advantage hierarchy. The vertical level is determined by the most compelling application in each category, as demonstrated by the available evidence of quantum advantage.

Based on the aforementioned definition of quantum speedups, Fig. 1.1 organizes various quantum applications related to this book using a pyramid structure of 4 levels: Level I (Highly structured problems), Level II (Unitary dynamics, sampling, and learning problems), Level III (Quantum properties and effective dynamics problems), and Level IV (Strong classical baselines and I/O limited problems). Significant quantum speedups may exist across all levels. The vertical axis represents the *existing* amount of evidence supporting significant quantum speedups, while the horizontal axis represents the *potential* range of scientific applications. Besides gate complexity, for learning and sensing tasks the dominant cost can be the sample complexity (the number of experimental repetitions), which we treat as part of the running and output costs in this hierarchy. Now we give some examples at each level of the hierarchy, and the results are summarized in Table 1.1.

Level	Input Cost	Output Cost	Running Cost	Classical Cost	Examples
I Highly structured	✓	✓	✓	Provably expensive	Shor's algorithm for prime factorization and discrete logarithm, decoded quantum interferometry for structured optimization
II Unitary dynamics sampling and learning	✓	✓	✓	Empirically expensive	Hamiltonian simulation, random circuit sampling, learning with quantum memory
III Quantum properties and effective dynamics	?	?	✓	Empirically expensive	Ground state energy estimation, thermal state preparation, Green's function, open quantum system dynamics
IV Strong classical baselines and I/O limited	?	?	?	Efficient (except very large systems)	Classical partial differential equations, stochastic differential equations, unstructured optimization, classical machine learning

TABLE 1.1. Examples of problems in the quantum advantage hierarchy and existing amount of evidence justifying significant quantum speedups.

While prime factorization (and cryptography problems in general) are not typically classified as scientific computing problems, they occupy a unique position (Level I) at the peak of this hierarchy, and serves as a reference point for the ideal demonstration of quantum advantage in highly structured settings. These problems possess specific mathematical structures that allow quantum algorithms to bypass the exhaustive search often required by classical approaches. By describing the classical cost as "provably expensive," we mean that the problem is hard under reasonable complexity-theoretic conjectures or relative to the best-known classical algorithms. For example, Shor's algorithm exploits the periodicity of the modular exponentiation function, a property related

to the hidden subgroup problem. Another reason for placing Shor's algorithm at the top of the hierarchy is that these problems are "verifiable," meaning a candidate solution can be efficiently checked by classical means (e.g., by multiplying the returned factors). The recently developed decoded quantum interferometry (DQI) algorithm [JSW$^+$25] also solves highly structured optimization problems with superpolynomial speedups and has the potential to be classified in Level I. Unfortunately, such structures are rare in general scientific computing settings, as most problems in physics and engineering lack these clean, exploitable properties. To date, only a small number of applications have presented a comparable level of evidence supporting significant quantum speedups, and the list of credible candidates continues to evolve.

The most prominent example in Level II is the time evolution of a quantum state under a Hamiltonian, known as the Hamiltonian simulation problem. Many tasks in quantum physics and chemistry can be cast in this form. This category also includes sampling from unitary evolutions not explicitly defined by a Hamiltonian, such as random circuit sampling used in quantum supremacy experiments. For a physical Hamiltonian acting on n qubits, the description size is typically polynomial in n. We assume simple initial states, such as product states, which can be prepared with polynomial cost. The cost to simulate the dynamics for time t with precision ϵ then scales as $\mathrm{poly}(n, t, 1/\epsilon)$. Under these assumptions, no known classical algorithm is expected to reliably simulate generic many-body dynamics for long times. However, compared to Level I, the theoretical justification for speedup is often less rigorous, and verifying quantum advantage can be more difficult. For instance, verifying the output distribution of random circuit sampling typically demands exponential classical resources. Consequently, evidence for advantage relies heavily on the empirical hardness of classical simulation. Certain quantum learning tasks can demonstrate exponential advantage in sample complexity [HBC$^+$22]. These advantages primarily stem from the quantum nature of the input data and the availability of quantum memory [CCHL22]. This differs from the computational tasks addressed in this book, which focus on problems with classical inputs and outputs. Furthermore, the exponential advantage assumes that we have zero knowledge about the quantum system being learned, which is often not the case in physical applications.

Level III of the hierarchy includes a large class of problems in quantum physics, quantum chemistry, and materials science. By "quantum properties," we refer to static characteristics of the system, such as ground state energy, excited state energy, stationary states, and spectral properties. By "effective dynamics," we refer to processes that are not natively unitary, such as open system dynamics involving dissipation or imaginary time evolution used for cooling. Compared to Level II problems, the mapping from these non-unitary objects to the unitary logic of quantum hardware is indirect. This mapping often introduces overheads, such as the need for linear combinations of unitaries, post-selection, or many ancillary qubits. The amount of information that needs to be extracted from the quantum computer can be comparable to that in the quantum dynamics simulation and is at most polynomial in n. In the case of the ground-state energy estimation, the situation is even clearer since we only need to estimate a single number as the output. Compared to unitary dynamics, there exist a much larger number of powerful classical algorithms for these tasks. These are approximate methods and often cannot be used to converge to the true solution to arbitrary precision. However, for many practical problems, they have been shown to be sufficiently accurate. Input cost is also a major factor placing these problems at Level III. For example, ground state estimation often requires a good initial guess (an ansatz) to succeed; generating this ansatz can be computationally expensive or physically difficult, sometimes leading to QMA-hard bottlenecks. Finally, we may design quantum algorithms to solve problems that are entirely classical. For instance, we can consider quantum solvers for classical partial differential equations

(PDEs), stochastic differential equations, unstructured optimization problems, and sampling tasks. These cover a large variety of problems in scientific computing. However, many such classical problems fall into Level IV because they have strong classical baselines and/or are I/O limited. For example, many PDEs on a grid of size N can be solved classically in time polynomial in N (often approximately linear in N using fast algorithms). Even if a quantum algorithm offers a speedup in the processing stage, it faces the "I/O limit": merely loading an arbitrary input vector of size N into the quantum state takes time linear in N, which negates any potential exponential speedup. One exception arises when the classical data possesses significant structure that allows for efficient loading; a potential advantage in this regime was recently demonstrated for a quantum solver of a large number of classical oscillators [BBK+23]. Regarding unstructured search problems, while Grover's algorithm provides a quadratic speedup, this is often insufficient to overcome the significant constant-factor overheads of fault-tolerant quantum error correction compared to highly optimized classical heuristics. Thus, while the range of applications is vast, securing an end-to-end advantage is difficult. That being said, many cryptography problems can be formulated as classical optimization problems, and the next breakthrough in quantum algorithms *may* emerge from classical problems again.

The ongoing evaluation and pursuit of quantum advantage is a rapidly developing field. When discussing applications, we will only scratch the surface of the potential indications of quantum advantage by examining aspects such as quantum input cost, output cost, running cost, and the cost of classical algorithms, wherever possible. This approach is intended to encourage readers to seek out these elements in their own research. However, it is important to understand that the findings presented, while based on existing literature, are far from exhaustive or conclusive. The rapid pace of advancements means that future developments could significantly alter the current understanding and conclusions.

1.4. Quantum error correction and fault tolerant computation

All previous discussions assume that quantum operations can be perfectly performed. To this end, quantum error correction is necessary. The threshold theorem [ABO97] is a central result in the field of quantum error correction. The theorem essentially states that if the error rate of quantum operations (including gates and measurements) is below a certain threshold value (around 0.001, though the precise value depends on the detailed assumptions), then it is possible to perform quantum computation for an arbitrary length of time with arbitrarily high accuracy (see [NC00, Section 10.6]).

THEOREM 1.1 (Threshold theorem). *There exists an error threshold $p_t > 0$. If the physical error rate p per gate operation satisfies $p < p_t$, there exists a quantum error correction scheme such that the logical error rate q can be made as small as desired. In other words, $q = \mathcal{O}((p/p_t)^\ell)$ for any positive integer ℓ.*

We will not study the details of quantum error correction in this book. In classical computing, modern algorithm design generally does not take error correction into account. Similarly, in the long term, quantum error correction is expected to be largely a separate issue from the design of quantum algorithms. We always assume quantum error correction protocols have been implemented, physical noise has been eliminated, and the resulting quantum computer is **fault-tolerant**. For the purpose of this book, all errors come from either *approximation errors* at the mathematical level, or *Monte Carlo errors* in the readout process due to the probabilistic nature of the measurement process.

Quantum error correction is a dynamic and rapidly progressing field, and will significantly impact the development and potential of quantum algorithms, and the landscape of quantum computing. On a very coarse scale, we can categorize quantum algorithms based on the type of quantum computer architecture they are designed for.

(1) Noisy intermediate-scale quantum (NISQ) computers: These devices represent the current state of quantum computing technology. Characterized by a relatively small number (tens to a few hundreds) of physical qubits, these systems are prone to errors and lack full error correction capabilities. Quantum algorithms designed for NISQ devices, such as the Variational Quantum Eigensolver (VQE), need to be error resilient and must be capable of delivering meaningful results despite the presence of noise. Most of this book will not discuss NISQ algorithms.
(2) Fully fault-tolerant quantum computers: These are the ideal, long-term goal of quantum computing research. In these systems, quantum error correction protocols are fully implemented, allowing quantum algorithms to run for long durations without being overwhelmed by errors. This architecture will enable the execution of complex algorithms that require a large number of qubits and gate operations. Many of the algorithms discussed in this book are designed for this type of architecture. At the current stage, the goal of many fully fault-tolerant quantum algorithms is to minimize the total cost (in an *asymptotic* sense with respect to certain parameters, such as precision, system size etc.) for solving a given task.
(3) Early fault-tolerant quantum computers: This category represents a transitional phase between NISQ devices and fully fault-tolerant quantum computers. These systems would implement some form of quantum error correction, but they may have constraints such as a very limited number of logical ancilla qubits. This means that they can only run quantum algorithms within a certain complexity limit. Despite these constraints, early fault-tolerant quantum computers provide an opportunity to test and refine fault-tolerant designs and protocols, and to run quantum algorithms that are beyond the reach of NISQ devices but do not require the full capabilities of fault-tolerant quantum computers. Some of the algorithms in this book take such constraints into account and can be suitable on early fault-tolerant quantum computers.

1.5. Error accumulation mechanisms in classical and quantum computation

Quantum computation aims at processing objects whose natural dimension is exponential, such as vectors in $\mathbb{C}^{2^n}$ and matrices of size $2^n \times 2^n$. No computation can be carried out exactly, so will the error also accumulate exponentially with the system size? If that were the case, then quantum algorithms would become useless precisely in the regime where they are designed to operate. In this section we give a bird's-eye view of the relevant error accumulation mechanisms.

At first glance, deterministic numerical computation can look discouraging in this respect. Even a basic task such as forming an inner product involves many elementary operations, and Example 1.2 shows a worst-case bound for the accumulated rounding effects that is proportional to $N = 2^n$.

However, scientific computation has long dealt with exponentially large state spaces without requiring errors to grow linearly in the dimension. Randomized algorithms on n bits evolve a probability distribution on a space of size $N = 2^n$, yet the accuracy of the computation is governed by how many transition steps are composed, not by N itself: if each step is implemented to accuracy ϵ, then the overall error is at most $K\epsilon$, where K is the number of steps (see Proposition 3.30).

Quantum computation behaves in the same way at the level of circuit synthesis: a quantum algorithm is a product of elementary unitaries, and the accumulated implementation error is controlled by the number of gates. In particular, if the gate count is K and each gate can be implemented to precision ϵ/K, then the final error is $\mathcal{O}(\epsilon)$. In the fault-tolerant setting assumed above, achieving such per-gate accuracy is a realistic requirement, and the overhead of approximating elementary unitaries to a desired precision is discussed later (see Chapter 4). The distance notions used to make these comparisons precise are developed in Chapter 3, and the Monte Carlo errors arising at readout are discussed further in Chapter 8.

1.5.1. Deterministic classical computation. Modern scientific computation on classical computers is based on floating point arithmetic operations, which express a number in scientific notation. For instance, the number $-.271828 \times 10^5$ involves a sign $(-)$, fraction (271828), base (10), and exponent (5). In binary floating point, one stores a sign bit together with a fixed-length exponent and fraction. For instance, the IEEE single precision uses 1 bit for the sign, 8 bits for the exponent, and 23 bits for the fraction (32 bits long). The IEEE double precision uses 1 bit for the sign, 11 bits for the exponent, and 52 bits for the fraction (64 bits long). For instance, a double precision ranges from 2^{-1022} to 2^{1023}, or about 10^{-308} to 10^{308}. Numbers outside this range yield underflow or overflow error and need to be handled separately. This is much more efficient than the fixed point number representation (see Section 5.3), which would require more than 2046 bits (i.e., more than 2046 logical qubits for a single number) to cover the same range of numbers.

The basic assumption is that any real number a should be represented by $\mathrm{fl}(a)$ using a given number of bits. Similarly, any binary operation $a \odot b$ should be represented by $\mathrm{fl}(a \odot b)$, where $\odot$ is one of the four elementary binary operations $+, -, *, /$. The difference $a \odot b - \mathrm{fl}(a \odot b)$ is called the roundoff error. When the number is rounded correctly, i.e., $\mathrm{fl}(a \odot b)$ is a nearest floating point number to $a \odot b$, we have

$$\mathrm{fl}(a \odot b) = (a \odot b)(1 + \delta), \tag{1.2}$$

where $|\delta|$ is upper bounded by ϵ_{mach} (called the machine precision).

Example 1.2. Given $u, v \in \mathbb{R}^N$, consider the error accumulation of computing an inner product $\sum_{i=1}^N u_i v_i$. The error from each operation in the floating-point arithmetic needs to be counted separately. The floating-point representation of a product $u_i v_i$ is given by $u_i v_i(1 + \epsilon_i)$, where $|\epsilon_i| \leq \epsilon_{\mathrm{mach}}$, and ϵ_{mach} is the machine epsilon.

However, when summing these products, there is an additional error introduced at each addition step. Let us denote by δ'_j the relative rounding error incurred when adding the j-th term (so $|\delta'_j| \leq \epsilon_{\mathrm{mach}}$). Then the partial sums satisfy

$$\mathrm{fl}\big(s_{j-1} + u_j v_j(1 + \epsilon_j)\big) = \big(s_{j-1} + u_j v_j(1 + \epsilon_j)\big)(1 + \delta'_j), \tag{1.3}$$

where s_{j-1} denotes the computed partial sum from the previous step. After summing over all N terms, we may write

$$1 + \delta_i := (1 + \epsilon_i) \prod_{j=i+1}^{N} (1 + \delta'_j). \tag{1.4}$$

Therefore if overflow or underflow does not occur, then

$$\mathrm{fl}\left(\sum_{i=1}^N u_i v_i\right) = \sum_{i=1}^N u_i v_i(1 + \delta_i), \quad |\delta_i| \leq (1 + \epsilon_{\mathrm{mach}})^N - 1 \leq e^{N\epsilon_{\mathrm{mach}}} - 1. \tag{1.5}$$

◇

When $N\epsilon_{\text{mach}} < 1$, we have $|\delta_i| \leq 2N\epsilon_{\text{mach}}$. So the error grows linearly in N. This is due to the step of adding N numbers following a linear order. For computing the inner product, the error accumulation in the summation step can be significantly reduced using a technique called the pair summation (or cascade summation) to $\mathcal{O}((\log N)\epsilon_{\text{mach}})$. However, such a more accurate summation method is more difficult to implement in broader scenarios such as matrix-matrix multiplication. For most of the tasks, the poly(N) factor in the error accumulation is unavoidable. For instance, for solving a triangular linear system, the error accumulation is $\mathcal{O}(N\epsilon_{\text{mach}})$ [GVL13, Chapter 3.1]. For Gaussian elimination (or LU factorization), standard backward-error bounds involve the growth factor ρ and scale polynomially in N, typically of order $\mathcal{O}(N\rho\epsilon_{\text{mach}})$ [GVL13, Chapter 3.4].

That being said, not all deterministic computations involving vectors in $\mathbb{C}^N$ necessarily exhibit a poly(N) accumulation of numerical error. Error accumulation is governed not by the ambient dimension N itself, but by the number of elementary operations performed. For instance, tensor network methods provide settings in which certain computations on structured vectors in $\mathbb{C}^N$ can be carried out using only poly(n) operations, where $N = 2^n$. We will not discuss tensor network methods in this book, and classical probabilistic computation provides a more direct analogy to quantum computation for tackling high dimensional problems, as discussed next.

1.5.2. Probabilistic classical computation. A probabilistic computation on n bits evolves a probability distribution on a space of size $N = 2^n$, and hence it can be described by a vector in $\mathbb{R}^N$ acted on by stochastic matrices. The ambient dimension is exponential in n, but the computation is specified by a sequence of local update rules. As a result, neither the cost nor the accumulated implementation error needs to scale exponentially in N. This viewpoint also extends to the comparison between quantum and classical algorithms: a probability distribution can be viewed as a special quantum state, and a transition matrix can be associated with a special quantum channel (see Section 3.2).

If we can implement each transition matrix to precision ϵ, the global error of the overall transition matrix grows at most linearly with respect to the number of transition matrices and is at most $1, K\epsilon$ (see Proposition 3.30). Equivalently, if the gate complexity is K and we can implement each transition matrix to precision ϵ/K, then the final error is upper bounded by ϵ, independent of N. Compared to deterministic classical algorithms, randomized algorithms introduce another error mechanism: even when the transition rule is specified, one often estimates quantities of interest by sampling, and the output is therefore subject to Monte Carlo fluctuations. For example, estimating an expectation value by N_s independent samples typically incurs an error of order $\mathcal{O}(N_s^{-1/2})$, independent of the size of the underlying sample space. The statistical side of this issue is discussed further in Chapter 8.

1.5.3. Quantum computation. Quantum algorithms are designed to handle objects of size $N = 2^n$ without explicitly storing N numbers. As in probabilistic computation, error accumulation depends on how many steps are composed and on the metric used to compare channels (see Chapter 3), and they do not introduce an explicit dependence on N.

Every quantum circuit can be represented by a unitary U, decomposed into a series of simpler unitaries as $U = U_K \cdots U_1$. Each U_i can only be implemented approximately by some $\widetilde{U}_i$ to precision ϵ. The implementation cost of each simple unitary is independent of the Hilbert space dimension N (see Chapter 4). This implies that for any vector $|\psi\rangle$ of size N, the error between $U_i|\psi\rangle$ and $\widetilde{U}_i|\psi\rangle$ is less than ϵ with no explicit dependence on N.

If we can implement each local unitary to precision ϵ, the global error grows at most *linearly* with respect to the number of gates and is at most $K\epsilon$ (see Proposition 3.21). In other words, if the gate complexity is K and we can implement each gate to precision ϵ/K, then the final error is upper bounded by ϵ and is independent of N. The same statement holds for quantum channels (see Section 3.6).

CHAPTER 2

Elements of quantum computation

This chapter lays the groundwork for our journey into quantum algorithms for scientific computation. We will review the mathematical and physical principles that underpin quantum computing. While we assume a basic familiarity with quantum mechanics, our focus will be on establishing the specific concepts and notational conventions used throughout this book. This chapter is not intended as a comprehensive introduction to quantum computing, but rather as a targeted primer on the tools we will need to build and analyze sophisticated quantum algorithms. For a more comprehensive introduction to quantum computation, we refer the reader to standard textbooks such as [NC00, Wat18].

We start with the postulates of quantum mechanics, introducing the Dirac notation and the core principles governing quantum states and their evolution. We then move to the language of quantum circuits, which greatly simplifies the tensor manipulations inherent in multi-qubit systems. To handle scenarios involving noise and subsystems, we introduce the density operator formalism. We will also discuss the no-cloning theorem, which forbids the copying of arbitrary quantum states, and the principles of deferred and implicit measurement, which offer flexibility in circuit design. The latter part of the chapter introduces the representation of structured matrices, including sparse matrices and operators from fermionic and bosonic systems. We conclude with a selected list of Hamiltonians from physics, chemistry, and optimization that will serve as motivating examples in our exploration of quantum simulation and other applications.

2.1. Basic notation

The sets of real and complex numbers are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively. For a complex number $c \in \mathbb{C}$, the notation $\overline{c}$ or c^* denotes its complex conjugate.

A complex vector v of size N is an N-tuple of complex numbers, written as $v \in \mathbb{C}^N$, with its j-th component denoted by v_j. By default, we use 0-based indexing, that is, $j \in [N] := \{0, \ldots, N-1\}$. When 1-based indexing is used, we will explicitly write $j = 1, \ldots, N$.

The **vector 2-norm** of v is denoted by $\|v\| = \sqrt{\sum_{i\in[N]} |v_i|^2}$. Unless otherwise specified, a vector $v \in \mathbb{C}^N$ is considered unnormalized. A nonzero, normalized vector (viewed as a pure quantum state) is written as $|v\rangle = v/\|v\|$. To emphasize that a vector is unnormalized, we sometimes use the notation $|v\rangle$.

A matrix A of size $M \times N$ is denoted by $A \in \mathbb{C}^{M\times N}$, and its (i,j)-th entry is A_{ij} or a_{ij}. For $A \in \mathbb{C}^{M\times N}$, the complex conjugate of A, denoted by $\overline{A}$ or A^*, is obtained by replacing each entry of A with its complex conjugate. The inverse of A (if A is invertible) is denoted by A^{-1}. The transpose of A is denoted by $A^\top$. The Hermitian conjugate (or adjoint) of A, denoted by $A^\dagger$, is the complex conjugate of the transpose of A, which can be expressed as $A^\dagger = (A^\top)^*$. A matrix A is **Hermitian** if it is equal to its Hermitian conjugate, i.e., $A = A^\dagger$. A matrix A is **normal** if it commutes with its Hermitian conjugate, i.e., $AA^\dagger = A^\dagger A$. A matrix U is unitary if its Hermitian

conjugate is its inverse, i.e., $U^\dagger = U^{-1}$. The set of all $N \times N$ unitary matrices forms the unitary group, denoted by $\mathrm{U}(N)$. The set of all $N \times N$ unitary matrices with determinant 1 forms the special unitary group, denoted by $\mathrm{SU}(N)$.

If all eigenvalues of a Hermitian matrix $A \in \mathbb{C}^{N\times N}$ are nonnegative, A is called a **positive semidefinite** matrix, or **positive operator**, denoted by $A \succeq 0$. The notation $A \succeq B$ means $A - B \succeq 0$, and $A \preceq B$ means $B \succeq A$. Similarly, if all eigenvalues of A are positive, then A is called a **positive definite** matrix, denoted by $A \succ 0$. The notation $A \succ B$ means $A - B \succ 0$.

The **operator norm** (also called **induced vector 2-norm**)[1] of a matrix A is

$$\|A\| := \sup_{\|v\|=1} \|Av\| . \tag{2.1}$$

In quantum information theory, it is useful to consider the **Schatten p-norm** of A:

$$\|A\|_p := \left(\mathrm{Tr}(A^\dagger A)^{\frac{p}{2}}\right)^{\frac{1}{p}}, \quad p \geq 1. \tag{2.2}$$

The particularly useful one is the **Schatten 1-norm** (also called the **trace norm**)

$$\|A\|_1 := \mathrm{Tr}\sqrt{A^\dagger A}. \tag{2.3}$$

For instance, any quantum state (density operator) ρ is normalized with respect to the trace norm, i.e., $\|\rho\|_1 = 1$. Furthermore, the Schatten ∞-norm $\|A\|_\infty$ can be shown to coincide with the operator norm $\|A\|$. Many readers may not be familiar with the Schatten norms. We will discuss these norms in detail in Chapter 3.

We adopt the following **asymptotic notations**: Let $\mathbb{R}_+$ be the set of positive real numbers. Consider two functions $f : \mathbb{R} \to \mathbb{C}$ and $g : \mathbb{R} \to \mathbb{R}_+$. For any $a \in \mathbb{R} \cup \{\pm\infty\}$, if $\limsup_{x\to a} \frac{|f(x)|}{g(x)} < \infty$, then we write $f(x) = \mathcal{O}(g(x))$ as $x \to a$, or simply $f = \mathcal{O}(g)$[2] when $x \to a$ is clear from the context. We write $f = \Omega(g)$ if $g = \mathcal{O}(f)$; $f = \Theta(g)$ if $f = \mathcal{O}(g)$ and $g = \mathcal{O}(f)$. Note that $\mathcal{O}(g)$ can also be interpreted as a set, so it is also valid to write $f \in \mathcal{O}(g)$. Similarly we may write $f \in \Omega(g), f \in \Theta(g)$ etc.

The notation $\widetilde{\mathcal{O}}, \widetilde{\Omega}, \widetilde{\Theta}$ are used to suppress subdominant polylogarithmic factors. Specifically, $f = \widetilde{\mathcal{O}}(g)$ if $f = \mathcal{O}(g \,\mathrm{polylog}(g))$; $f = \widetilde{\Omega}(g)$ if $f = \Omega(g \,\mathrm{polylog}(g))$; $f = \widetilde{\Theta}(g)$ if $f = \Theta(g \,\mathrm{polylog}(g))$. Note that these tilde notations usually do not suppress dominant polylogarithmic factors. For instance, if $f = \mathcal{O}(\log g \log\log g)$, then we write $f = \widetilde{\mathcal{O}}(\log g)$ instead of $f = \widetilde{\mathcal{O}}(1)$. However, for simplicity of presentation, we may sometimes use the notation $\widetilde{\mathcal{O}}$ more casually to suppress dominant polylogarithmic factors. When we do so, we will make an explicit mention of this usage.

Throughout the book, the natural logarithm is denoted by ln, and is sometimes written as log without an explicit base when the context is clear. The logarithm to base 2 is denoted by $\log_2$. When N denotes the dimension of $\mathbb{C}^N$, and the notations N and n appear together, it is usually assumed that $N = 2^n$ for some positive integer n, referred to as the number of quantum bits (or **qubits**). Additional notations will be introduced in the book as needed.

---

[1] In matrix analysis, the operator norm is sometimes denoted by $\|A\|_2$ to indicate that this is the induced vector 2-norm. More generally, the induced vector p-norm is $\|A\|_p = \sup_{\|x\|_p=1} \|Ax\|_p$ where $\|x\|_p = \left(\sum_i |x_i|^p\right)^{1/p}$. For example, the induced vector 1-norm is $\|A\|_1 = \sup_{\|x\|_1=1} \|Ax\|_1 = \max_j \sum_i |a_{ij}|$. This book **does not** adopt such a notation.

[2] Sometimes $\mathcal{O}(g)$ is treated as a set of functions, and by this interpretation we can equivalently write $f \in \mathcal{O}(g)$.

2.2. Postulates of quantum mechanics

This section encapsulates some of the most important postulates of quantum mechanics. All postulates concern finite dimensional, closed quantum systems (i.e., systems isolated from environments). For more details, we refer readers to [NC00, Section 2.2].

2.2.1. State space postulate.

Definition 2.1 (Hilbert space)**.** *A (complex) Hilbert space denoted by $\mathcal{H}$ is a complex vector space equipped with an inner product $\langle\cdot|\cdot\rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{C}$ that satisfies the following properties for all $x, y, z \in \mathcal{H}$ and all $\alpha, \beta \in \mathbb{C}$:*

1. *(Conjugate Symmetry) $\langle x|y\rangle = \overline{\langle y|x\rangle}$.*
2. *(Linearity in the second argument) $\langle z|\alpha x + \beta y\rangle = \alpha\langle z|x\rangle + \beta\langle z|y\rangle$.*
3. *(Positive-definiteness) $\langle x|x\rangle \geq 0$ with equality if and only if $x = 0$.*

Furthermore, $\mathcal{H}$ is complete with respect to the norm induced by the inner product, where the norm of a vector $x \in \mathcal{H}$ is given by $\|x\| = \sqrt{\langle x|x\rangle}$.

The state space postulate assumes that the set of all quantum states of a quantum system, called the state space, is a Hilbert space. If the state space $\mathcal{H}$ is finite dimensional, it is isomorphic (i.e., there is a one-to-one mapping) to some $\mathbb{C}^N$, written as $\mathcal{H} \cong \mathbb{C}^N$. Throughout the book, unless otherwise specified, we only consider finite dimensional Hilbert spaces. A **state vector** (also called ket vector, wavefunction, or pure quantum state) $|\psi\rangle \in \mathcal{H}$ can be identified with a column vector in $\mathbb{C}^N$

$$\psi = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{N-1} \end{pmatrix}. \tag{2.4}$$

Let $\{e_i\}$ be the standard basis of $\mathbb{C}^N$. The i-th entry of ψ can be written as an inner product $\psi_i = \langle e_i|\psi\rangle$. We also use the Dirac notation, which uses $|\psi\rangle$ to denote a quantum state. We further postulate that two state vectors $|\psi\rangle$ and $c\,|\psi\rangle$ for some $0 \neq c \in \mathbb{C}$ always refer to the same physical state. Hence without loss of generality we always assume $|\psi\rangle$ is normalized to be a unit vector, i.e., $\langle\psi|\psi\rangle = 1$. Restricting to normalized state vectors, the complex number $c = e^{i\theta}$ for some $\theta \in [0, 2\pi)$ is called the global phase factor.

Throughout the book, unless otherwise specified, an unnormalized state vector is often denoted by ψ without the ket notation $|\cdot\rangle$, and $|\psi\rangle := \psi/\|\psi\|$ denotes the normalized counterpart.

The bra vector $\langle\psi|$ can be interpreted as a linear functional on $\mathcal{H}$, which maps any $|\varphi\rangle \in \mathcal{H}$ to a complex number $\langle\psi|\varphi\rangle$. When $\mathcal{H} = \mathbb{C}^N$, we have $\langle\psi|\varphi\rangle = \sum_{i\in[N]} \overline{\psi_i}\varphi_i$. It can be identified with a row vector, which is the Hermitian conjugate of the column vector ψ:

$$\psi^\dagger = \begin{pmatrix} \overline{\psi_0} & \overline{\psi_1} & \cdots & \overline{\psi_{N-1}} \end{pmatrix}. \tag{2.5}$$

The set of all bra vectors, or linear functionals on $\mathcal{H}$, is denoted by $\mathcal{H}^\star$[3].

Given a state space $\mathcal{H}$, let $L(\mathcal{H})$ denote the set of all linear operators on $\mathcal{H}$. When $\mathcal{H} = \mathbb{C}^N$, $L(\mathbb{C}^N)$ can be identified with the set of $N \times N$ matrices, denoted by $\mathbb{C}^{N\times N}$. The ketbra notation $|\psi\rangle\langle\varphi|$ is an element in $L(\mathcal{H})$, which maps any vector $|\xi\rangle \in \mathcal{H}$ to another state vector in $\mathcal{H}$ as

[3]The star $\star$ acting on a vector space does not mean the complex conjugation of $\mathcal{H}$. This notation is only used occasionally in the book. A Hilbert space satisfies $\mathcal{H} \cong \mathcal{H}^\star$ by the Riesz representation theorem.

$|\psi\rangle \langle\varphi|\xi\rangle$. The matrix representation of $|\psi\rangle\langle\varphi|$ is the product of the column vector ψ and the row vector $\varphi^\dagger$, i.e., $\psi\varphi^\dagger \in \mathbb{C}^{N\times N}$.

Example 2.2 (Single qubit system and Bloch sphere)**.** A (single) qubit corresponds to a state space $\mathbb{C}^2$. We also define

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}. \tag{2.6}$$

Since the state space of the spin-$\frac{1}{2}$ system is also isomorphic to $\mathbb{C}^2$, this is also called the single spin system, where $|0\rangle, |1\rangle$ are referred to as the spin-up and spin-down state, respectively. A general state vector in $\mathbb{C}^2$ takes the form

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle = \begin{pmatrix}a\\b\end{pmatrix}, \quad a, b \in \mathbb{C}, \tag{2.7}$$

and the normalization condition implies $|a|^2 + |b|^2 = 1$. So we may rewrite $|\psi\rangle$ as

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}\,|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}\,|1\rangle\right), \quad \theta, \varphi, \gamma \in \mathbb{R}. \tag{2.8}$$

If we ignore the irrelevant global phase $e^{i\gamma}$ (which also absorbs a minus sign in the coefficient of $|0\rangle$), then it holds

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}\,|1\rangle, \quad 0 \le \theta \le \pi, 0 \le \varphi < 2\pi. \tag{2.9}$$

So we may identify each single qubit quantum state with a unique point on the unit three-dimensional sphere (called the **Bloch sphere**) as

$$\mathbf{a} = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)^\top. \tag{2.10}$$

◇

2.2.2. Quantum operator postulate. The quantum operator postulate states that the evolution of a quantum state from $|\psi\rangle \to |\psi'\rangle \in \mathcal{H}$ is always achieved via a unitary operator U, i.e.,

$$|\psi'\rangle = U\,|\psi\rangle, \quad U^\dagger U = I. \tag{2.11}$$

Here $U^\dagger$ is the Hermitian conjugate of U, and I is the identity map that can be identified with a N-dimensional identity matrix. The set of all $N \times N$ unitary matrices is the unitary group, denoted by $\mathrm{U}(N)$. The set of all $N \times N$ unitary matrices with determinant 1 forms the special unitary group, denoted by $\mathrm{SU}(N)$.

This unitary evolution is derived from the system's **Hamiltonian** $H \in L(\mathcal{H})$, which is a Hermitian matrix that encapsulates the total energy of the system and thus governs its dynamics. For a time-independent Hamiltonian H, the state $|\psi(t)\rangle$ satisfies the Schrödinger equation

$$i\partial_t\,|\psi(t)\rangle = H\,|\psi(t)\rangle. \tag{2.12}$$

The corresponding time evolution operator is

$$U(t_2, t_1) = e^{-iH(t_2 - t_1)}, \quad \forall t_2 \ge t_1. \tag{2.13}$$

In particular, $U(t_2, t_1) = U(t_2 - t_1, 0)$.

More generally, starting from an initial quantum state $|\psi(0)\rangle$, the quantum state can evolve in time, which gives a single parameter family of quantum states denoted by $\{|\psi(t)\rangle\}$. These quantum states are related to each other via a quantum evolution operator U:

$$|\psi(t_2)\rangle = U(t_2, t_1)\,|\psi(t_1)\rangle\,, \tag{2.14}$$

where $U(t_2, t_1)$ is unitary for any given t_1, t_2. Here $t_2 > t_1$ refers to quantum evolution forward in time, $t_2 < t_1$ refers to quantum evolution backward in time, and $U(t_1, t_1) = I$ for any t_1.

In quantum computation, a unitary matrix is often referred to as a **quantum gate**.

Example 2.3. For a single qubit, the **Pauli matrices** are

$$\sigma_x = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.15}$$

Together with the two-dimensional identity matrix, they form a basis of all linear operators on $\mathbb{C}^2$. ◇

Some other commonly used single qubit operators include, to name a few:

- **Hadamard gate**

$$\mathrm{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.16}$$

- **Phase gate**

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{2.17}$$

- T **gate**:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = \sqrt{S}. \tag{2.18}$$

When there are notational conflicts, we will use the roman font such as H, X for these single-qubit gates (for example, to distinguish the Hadamard gate H from a Hamiltonian H). An operator acting on an n-qubit quantum state space is called an n-qubit operator.

Example 2.4. For $P \in \{X, Y, Z\}$, the unitary evolution generated by the Hamiltonian $H = P$ is a rotation about the corresponding Bloch-sphere axis. Concretely,

$$\begin{aligned} R_x(2t) :=& e^{-itX} = \begin{pmatrix} \cos(t) & -i\sin(t) \\ -i\sin(t) & \cos(t) \end{pmatrix}, \\ R_y(2t) :=& e^{-itY} = \begin{pmatrix} \cos(t) & -\sin(t) \\ \sin(t) & \cos(t) \end{pmatrix}, \\ R_z(2t) :=& e^{-itZ} = \begin{pmatrix} e^{-it} & 0 \\ 0 & e^{it} \end{pmatrix}. \end{aligned} \tag{2.19}$$

For instance, starting from an initial state $|\psi(0)\rangle = |0\rangle$, under $R_x(2t)$ at time $t = \pi/2$ the state evolves into $|\psi(\pi/2)\rangle = -i\,|1\rangle$, i.e., the $|1\rangle$ state up to a global phase. ◇

THEOREM 2.5 (Spectral theorem of normal matrices). *Given a matrix $A \in \mathbb{C}^{N\times N}$, the matrix A is normal (i.e., $A^\dagger A = AA^\dagger$), if and only if*

$$A = VDV^\dagger. \tag{2.20}$$

Here, $D \in \mathbb{C}^{N\times N}$ is a diagonal matrix containing the eigenvalues of A, and $V \in \mathrm{U}(N)$ is a unitary matrix whose columns are the eigenvectors of A.

A more general decomposition, which plays a key role throughout the book, is the **singular value decomposition** (SVD).

THEOREM 2.6 (Singular value decomposition). *Given any matrix $A \in \mathbb{C}^{M\times N}$, there exist unitary matrices $U \in \mathrm{U}(M)$ and $V \in \mathrm{U}(N)$, and a diagonal matrix $\Sigma \in \mathbb{C}^{M\times N}$ with non-negative real numbers on the diagonal, such that*

$$A = U\Sigma V^\dagger. \tag{2.21}$$

The diagonal entries of Σ are called the singular values of A, the columns of U are called the left singular vectors of A, and the columns of V are called the right singular vectors of A.

Operator exponentials, also called **matrix exponentials**, gives us a way to express gates as operator exponentials and because the algebra of exponentials makes this representation far easier to work with than explicitly writing the unitary in a matrix representation.

Definition 2.7 (Matrix function)**.** *For $A \in \mathbb{C}^{N\times N}$, and a complex valued function $f : \mathbb{C} \mapsto \mathbb{C}$, the matrix function $f(A)$ is defined as follows:*

1. *If f is an analytic function such that $f(x) = \sum_{j=0}^{\infty} a_j x^j$ then $f(A) := \sum_{j=0}^{\infty} a_j A^j$.*
2. *If f is a complex valued function and A is a normal matrix such that $A = VDV^\dagger$ where V is unitary and $D := \mathrm{diag}(\lambda_0, \ldots, \lambda_{N-1})$ where $f(\lambda_j) \in \mathbb{C}$. Then $f(A) := Vf(D)V^\dagger$ where $f(D) = \mathrm{diag}(f(\lambda_0), \ldots, f(\lambda_{N-1}))$.*

The definition of a matrix exponential can be seen as a direct consequence of either of the above definitions, and both definitions find extensive use in quantum computing. Specifically, using the former definition we have that for any matrix A

$$e^A := \sum_{j=0}^{\infty} \frac{A^j}{j!}. \tag{2.22}$$

Matrix function can also be defined for non-normal matrices using contour integrals (see [Hig08, Chapter 1]).

Lemma 2.8. *Let $A \in \mathbb{C}^{N\times N}$ and let $U \in \mathrm{U}(N)$ be a unitary matrix, then $Ue^AU^\dagger = e^{UAU^\dagger}$.*

The following result can be viewed as the simplest realization of the **Baker–Campbell–Hausdorff formula** (BCH).

Lemma 2.9. *For any $A, B \in \mathbb{C}^{N\times N}$, we have*

1. *if $[A, B] = 0$, then $e^A e^B = e^{A+B}$.*
2. *if $[A, [A, B]] = [B, [A, B]] = 0$, then $e^A e^B = e^{A+B+\frac{1}{2}[A,B]}$.*
3. *if $[A, B] \neq 0$, then $e^A e^B = e^{A+B+\frac{1}{2}[A,B]} + \mathcal{O}(\max(\|A\|, \|B\|)^3)$.*

In general, we can express any unitary operator as an exponential of a Hermitian operator. This result is a direct consequence of the definition of the operator exponential.

Lemma 2.10. *For any unitary matrix $U \in \mathrm{U}(N)$, there exists a Hermitian matrix $H \in \mathbb{C}^{N\times N}$ such that $U = e^{-iH}$.*

PROOF. A unitary matrix U is a normal matrix. According to Theorem 2.5, the unitary matrix U can be diagonalized as

$$U = VDV^\dagger, \tag{2.23}$$

where $V \in \mathbb{C}^{N\times N}$ is a unitary matrix and D is a diagonal matrix. The diagonal entries satisfy $|D_{ii}| = 1$. Without loss of generality we can write $D_{ii} = e^{-i\theta_i}$ where $\theta_i \in [0, 2\pi)$. Then define a diagonal matrix $\Theta_{ii} = \theta_i$, and $H = V\Theta V^\dagger$, we obtain $U = e^{-iH}$. Note that the matrix H is not unique since each θ_i can be chosen modulo 2π. □

In many scenarios such as the analysis of quantum simulation using Trotter-Suzuki formulas, we need to find Taylor series expansions of conjugated operators.

Lemma 2.11. *Let A, B be normal matrices in $\mathbb{C}^{N\times N}$ and let $t \in \mathbb{R}$. We then have that*

$$e^{At}Be^{-At} = B + \frac{[A,B]t}{1!} + \frac{[A,[A,B]]t^2}{2!} + \frac{[A,[A,[A,B]]]t^3}{3!} + \cdots \tag{2.24}$$

PROOF. We note that the above result is a power series in t, which must coincide with the Taylor series expansion of the function $f(t) = e^{At}Be^{-At}$ because the function is analytic. Thus the expression is true if the k-th derivative of $f(t)$ at $t = 0$ is given by the k-fold commutator. We prove by induction that

$$\partial_t^k(e^{At}Be^{-At}) = e^{At}[A,[A,[\cdots,[A,B]\cdots]]]e^{-At}, \tag{2.25}$$

where the commutator is applied k times. The base case $k = 0$ holds trivially. Assume the hypothesis holds for some $k \geq 0$. Then

$$\begin{aligned}\partial_t^{k+1}(e^{At}Be^{-At}) &= \partial_t\left(e^{At}[A,[A,[\cdots,[A,B]\cdots]]]e^{-At}\right)\\ &= e^{At}\left(A[A,[\cdots,[A,B]\cdots]] - [A,[\cdots,[A,B]\cdots]]A\right)e^{-At}\\ &= e^{At}[A,[A,[\cdots,[A,B]\cdots]]]e^{-At},\end{aligned} \tag{2.26}$$

where the final expression contains $k+1$ commutators. This confirms the inductive step. Evaluating at $t = 0$ yields the coefficients of the Taylor series, completing the proof. □

2.2.3. Quantum measurement postulate. In quantum mechanics, a **quantum observable** is always represented by a Hermitian matrix acting on the state space. The reason for using Hermitian matrices is that they have real eigenvalues, which correspond to the outcome of **quantum measurements**.

A quantum observable $O \in L(\mathcal{H})$ has the spectral decomposition

$$O = \sum_m \lambda_m P_m. \tag{2.27}$$

Here $\lambda_m \in \mathbb{R}$ are the eigenvalues of O, and $P_m \in L(\mathcal{H})$ is the projection operator onto the eigenspace associated with λ_m. The quantum measurement postulate states that when conducting a measurement on a quantum state $|\psi\rangle$ with respect to a quantum observable O, the eigenvalues λ_m represent all the possible results of the measurement. Furthermore, the probability of obtaining a particular outcome λ_m is

$$p_m = \langle\psi|P_m|\psi\rangle\,. \tag{2.28}$$

Following the measurement, the quantum state collapses to the corresponding eigenspace

$$|\psi\rangle \to \frac{P_m\,|\psi\rangle}{\sqrt{p_m}}. \tag{2.29}$$

The set of projection operators satisfies the resolution of identity:

$$\sum_m P_m = I. \tag{2.30}$$

This implies the normalization condition

$$\sum_m p_m = \sum_m \langle\psi|P_m|\psi\rangle = \langle\psi|\psi\rangle = 1, \quad \forall\, |\psi\rangle \in \mathcal{H}. \tag{2.31}$$

Together with $p_m \geq 0$, we find that $\{p_m\}$ is indeed a probability distribution.

The expectation value of the measurement outcome can be expressed as

$$\mathbb{E}_\psi(O) = \sum_m \lambda_m p_m = \sum_m \lambda_m \langle\psi|P_m|\psi\rangle = \left\langle \psi \middle| \left(\sum_m \lambda_m P_m\right) \middle| \psi \right\rangle = \langle\psi|O|\psi\rangle\,. \tag{2.32}$$

Example 2.12. Let $O = X$ be the Pauli X operator. From the spectral decomposition of X:

$$X\,|\pm\rangle = \lambda_\pm\,|\pm\rangle\,, \tag{2.33}$$

where $|\pm\rangle := \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$, $\quad \lambda_\pm = \pm 1$, we obtain the eigendecomposition

$$O = X = |+\rangle\langle+| - |-\rangle\langle-|\,. \tag{2.34}$$

Consider a quantum state $|\psi\rangle = |0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)$, then

$$\langle\psi|P_+|\psi\rangle = \langle\psi|P_-|\psi\rangle = \frac{1}{2}. \tag{2.35}$$

Therefore the expectation value of the measurement is $\langle\psi|X|\psi\rangle = 0$. ◇

Exercise 2.1. Prove Eq. (2.34).

2.2.4. Tensor product postulate.

Definition 2.13 (Tensor product)**.** *The tensor product of two finite dimensional Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ is a complex vector space, denoted by $\mathcal{H}_1 \otimes \mathcal{H}_2$, spanned by vectors of the form $v \otimes w$ with $v \in \mathcal{H}_1$ and $w \in \mathcal{H}_2$. The bilinear map $\otimes : \mathcal{H}_1 \times \mathcal{H}_2 \to \mathcal{H}_1 \otimes \mathcal{H}_2$ satisfies for all $v, v' \in \mathcal{H}_1$, $w, w' \in \mathcal{H}_2$, and scalars $\alpha, \beta \in \mathbb{C}$:*

1. *$(\alpha v + \beta v') \otimes w = \alpha(v \otimes w) + \beta(v' \otimes w)$ and $v \otimes (\alpha w + \beta w') = \alpha(v \otimes w) + \beta(v \otimes w')$.*
2. *$(\alpha v) \otimes w = \alpha(v \otimes w)$ and $v \otimes (\beta w) = \beta(v \otimes w)$.*

The tensor product is associative in the sense that the two vector spaces $(\mathcal{H}_1 \otimes \mathcal{H}_2) \otimes \mathcal{H}_3$ and $\mathcal{H}_1 \otimes (\mathcal{H}_2 \otimes \mathcal{H}_3)$ are isomorphic. Let $\mathcal{H}_1, \ldots, \mathcal{H}_k$ be finite-dimensional Hilbert spaces with inner products $\langle\cdot|\cdot\rangle_i$ for $i = 1, 2, \ldots, k$. The tensor product of these k spaces can be recursively defined as $\mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_k := \mathcal{H}_1 \otimes (\mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_k)$, which is spanned by all elements of the form $v_1 \otimes v_2 \otimes \cdots \otimes v_k$ called **product states**, where $v_i \in \mathcal{H}_i$ for $i = 1, 2, \ldots, k$. The inner product of two vectors $v = v_1 \otimes v_2 \otimes \cdots \otimes v_k$ and $w = w_1 \otimes w_2 \otimes \cdots \otimes w_k$ in the tensor product space $\mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_k$ is defined as

$$\langle v|w\rangle = \langle v_1|w_1\rangle_1 \cdot \langle v_2|w_2\rangle_2 \cdots \langle v_k|w_k\rangle_k.$$

This inner product is extended linearly to the entire tensor product space as

$$\left\langle \sum_i a_i v_i \middle| \sum_j b_j w_j \right\rangle = \sum_{i,j} \overline{a_i} b_j \langle v_i|w_j\rangle, \quad a_i, b_j \in \mathbb{C}.$$

The tensor product postulate states that the state space with k components $\mathcal{H}_1 \cong \mathbb{C}^{N_1}, \ldots, \mathcal{H}_k \cong \mathbb{C}^{N_k}$ is the tensor product of these spaces $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_k$. Let $\{|j_i\rangle\}_{j_i \in [N_i]}$ be the basis of $\mathbb{C}^{N_i}$, then a general state vector in $\mathcal{H}$ takes the form

$$|\psi\rangle = \sum_{j_1 \in [N_1], \ldots, j_k \in [N_k]} \psi_{j_1 \cdots j_k} |j_1\rangle \otimes \cdots \otimes |j_k\rangle . \tag{2.36}$$

Here $\psi_{j_1 \cdots j_k} \in \mathbb{C}$ is an entry of a k-way tensor. Given another state vector

$$|\varphi\rangle = \sum_{j_1 \in [N_1], \ldots, j_k \in [N_k]} \varphi_{j_1 \cdots j_k} |j_1\rangle \otimes \cdots \otimes |j_k\rangle , \tag{2.37}$$

the inner product takes the form

$$\langle \psi | \varphi \rangle = \sum_{j_1 \in [N_1], \ldots, j_k \in [N_k]} \overline{\psi_{j_1 \cdots j_k}} \varphi_{j_1 \cdots j_k} . \tag{2.38}$$

The state space of n-qubits is $\mathcal{H} = (\mathbb{C}^2)^{\otimes n} \cong \mathbb{C}^{2^n}$. We also use a shorthand notation: the tensor product $\otimes$ may be omitted when the context is clear.

$$|01\rangle \equiv |0, 1\rangle \equiv |0\rangle |1\rangle \equiv |0\rangle \otimes |1\rangle , \quad |0^{\otimes n}\rangle \equiv |0^n\rangle \equiv |0\rangle^{\otimes n} . \tag{2.39}$$

The tensor product operation provides us with a powerful way to describe two independent copies of different vector spaces as a single larger vector space. Further, the tensor product when viewed through this lens does not care about the nature of the form of the Hilbert spaces that are being combined. In fact a particularly important case that we need to consider is the tensor product between two operators.

Definition 2.14 (Tensor products of linear operators)**.** *Given two finite dimensional Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, the tensor product of $L(\mathcal{H}_1)$ and $L(\mathcal{H}_2)$, denoted by $L(\mathcal{H}_1) \otimes L(\mathcal{H}_2)$, is a complex vector space spanned by linear operators of the form $A \otimes B$ with $A \in L(\mathcal{H}_1)$ and $B \in L(\mathcal{H}_2)$. The bilinear map $\otimes : L(\mathcal{H}_1) \times L(\mathcal{H}_2) \to L(\mathcal{H}_1) \otimes L(\mathcal{H}_2)$ satisfies for all $A, B \in L(\mathcal{H}_1)$ and $C, D \in L(\mathcal{H}_2)$, $v \in \mathcal{H}_1$, $w \in \mathcal{H}_2$ and scalars $\alpha, \beta \in \mathbb{C}$:*

1. $(\alpha A + \beta B) \otimes C = \alpha A \otimes C + \beta B \otimes C$ *and* $A \otimes (\alpha C + \beta D) = \alpha A \otimes C + \beta A \otimes D$.
2. $(\alpha A) \otimes B = \alpha A \otimes B = A \otimes (\alpha B)$.

The space $L(\mathcal{H}_1) \otimes L(\mathcal{H}_2)$ is isomorphic to $L(\mathcal{H}_1 \otimes \mathcal{H}_2)$. The tensor product is also associative in the sense that $L(\mathcal{H}_1) \otimes (L(\mathcal{H}_2) \otimes L(\mathcal{H}_3))$ is isomorphic to $(L(\mathcal{H}_1) \otimes L(\mathcal{H}_2)) \otimes L(\mathcal{H}_3)$. A consequence of this definition is further that the application of multiple tensor products of linear operators on matching tensor products of vectors distributes across the tensor product via

$$(A_1 \otimes A_2 \otimes \cdots \otimes A_k)(v_1 \otimes v_2 \otimes \cdots \otimes v_k) = (A_1 v_1) \otimes (A_2 v_2) \otimes \cdots \otimes (A_k v_k). \tag{2.40}$$

Example 2.15 (Two qubit system)**.** The state space is $\mathcal{H} = (\mathbb{C}^2)^{\otimes 2} \cong \mathbb{C}^4$. The standard basis is (row-major order, i.e., last index is the fastest changing one)

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{2.41}$$

There are many important quantum operators on the two-qubit quantum system. One of them is the **CNOT gate**, with matrix representation

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.42}$$

In other words, when acting on the standard basis, we have

$$\text{CNOT} \begin{cases} |00\rangle & = |00\rangle \\ |01\rangle & = |01\rangle \\ |10\rangle & = |11\rangle \\ |11\rangle & = |10\rangle \end{cases}. \tag{2.43}$$

This can be compactly written as

$$\text{CNOT} \, |a\rangle \, |b\rangle = |a\rangle \, |a \oplus b\rangle \, . \tag{2.44}$$

Here $a \oplus b = (a + b) \mod 2$ is the "exclusive or" (XOR) operation. ◇

Definition 2.16 (Controlled unitaries)**.** *A controlled unitary operation is a quantum gate that applies a specified unitary operation U to a set of target qubits only when the control qubits are in a particular state, typically the $|1\rangle$ state for each control qubit. The single qubit controlled unitary operation can be represented as:*

$$CU = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U.$$

An n-qubit controlled unitary can be written as:

$$C^n U = (I - |1^n\rangle\langle 1^n|) \otimes I + |1^n\rangle\langle 1^n| \otimes U.$$

The CNOT gate is the same as CX. Controlled unitaries are ubiquitous in quantum algorithms. In particular, it enables conditional logic within quantum circuits.

Example 2.17 (Multi-qubit Pauli operators)**.** For a n-qubit quantum system, the Pauli operator acting on the i-th qubit is denoted by P_i ($P = X, Y, Z$), i.e.,

$$\begin{aligned} X_i &:= I^{\otimes(i-1)} \otimes X \otimes I^{\otimes(n-i)}, \\ Y_i &:= I^{\otimes(i-1)} \otimes Y \otimes I^{\otimes(n-i)}, \\ Z_i &:= I^{\otimes(i-1)} \otimes Z \otimes I^{\otimes(n-i)}. \end{aligned} \tag{2.45}$$

For example, in a 2-qubit system, following the row-major convention, the matrix representation of X_1, X_2 are

$$X_1 = X \otimes I = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad X_2 = I \otimes X = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.46}$$

◇

Definition 2.18 (Pauli group)**.** *The n-qubit Pauli group, denoted as $\mathcal{P}_n$, is a group that consists of all possible tensor products of n-qubit Pauli matrices along with multiplicative factors of ± 1 and $\pm i$. Each element of the n-qubit Pauli group can be represented as*

$$i^k(P_1 \otimes P_2 \otimes \cdots \otimes P_n),$$

where $k \in \{0, 1, 2, 3\}$, each P_i is one of the Pauli matrices X, Y, Z, or the identity matrix I, and $\otimes$ denotes the tensor product.

The n-qubit Pauli group contains 4^{n+1} elements due to the 4^n possible tensor products of Pauli matrices and identity matrices, each multiplied by one of the four possible phase factors $\pm 1, \pm i$. It plays a key role in quantum simulation and quantum error correction. Note that the product of any two elements is another element of the group (up to a phase factor), and every element is its own inverse (up to a phase factor).

Definition 2.19 (Clifford group)**.** *The n-qubit Clifford group, denoted as $\mathcal{C}_n$, is a group of unitary operators that normalizes the n-qubit Pauli group $\mathcal{P}_n$. This means that for every Clifford operator $C \in \mathcal{C}_n$ and every Pauli operator $P \in \mathcal{P}_n$, there exists a Pauli operator $P' \in \mathcal{P}_n$ such that*

$$CPC^\dagger = P'.$$

The Clifford group includes all elements of the Pauli group, the Hadamard gate H, the phase gate S, and the CNOT gate. It can be generated by $\{\mathrm{H}, S, \mathrm{CNOT}\}$.

Example 2.20. The single-qubit Pauli group $\mathcal{P}_1$ is defined as the group generated by the Pauli matrices X, Y, Z together with the phase factor i:

$$\mathcal{P}_1 = \{i^k P \mid k \in \{0, 1, 2, 3\}, P \in \{I, X, Y, Z\}\}.$$

We show that $\mathcal{P}_1$ can be generated by the set $\{\mathrm{H}, S\}$. First, we obtain the Pauli Z operator by squaring the phase gate:

$$S^2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}^2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z.$$

Next, we utilize the property that the Hadamard gate transforms Z into X under conjugation. Since H is Hermitian and unitary, we have:

$$X = \mathrm{H}\, Z\, \mathrm{H} = \mathrm{H}\, S^2\, \mathrm{H}\,.$$

The Pauli Y operator can be generated by conjugating X by S. We compute the conjugate transpose $S^\dagger = \mathrm{diag}(1, -i)$ and verify the relation:

$$\begin{aligned} SXS^\dagger &= \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \\ i & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = Y. \end{aligned}$$

Since S is unitary and $S^4 = I$, we have $S^\dagger = S^{-1} = S^3$. Thus, $Y = SXS^3$.

Finally, since $XYZ = iI$, we conclude that $\{\mathrm{H}, S\}$ generates the entire Pauli group $\mathcal{P}_1$.

Since

$$T^2 = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}^2 = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix} = S,$$

it immediately follows that $\{\mathrm{H}, T\}$ also generates $\mathcal{P}_1$. ◇

The Clifford group plays an important role in many areas. In quantum error correction, Clifford operations can transform certain errors into forms that are more easily correctable. This makes it desirable to choose Clifford gates to be part of a universal gate set (the most common one is Clifford + T). Additionally, the Gottesman–Knill theorem states that any quantum circuit using only Clifford gates on computational basis states and measurements in the computational basis can be efficiently simulated classically.

Example 2.21. We can concisely describe block matrices stored within a larger matrix. The matrix representation of $T \in L(\mathbb{C}^{NM} \otimes \mathbb{C}^{NM})$, when writing in the block form,

$$T = \begin{pmatrix} T_{0,0} & \cdots & T_{0,N-1} \\ \vdots & \ddots & \vdots \\ T_{N-1,0} & \cdots & T_{N-1,N-1} \end{pmatrix}, \quad T_{ij} \in \mathbb{C}^{M \times M}. \tag{2.47}$$

can be rewritten as

$$T = \sum_{i,j \in [N]} |e_i\rangle\langle e_j| \otimes T_{ij}. \tag{2.48}$$

◇

The notation for partial inner products and partial applications of operators is used throughout this book, particularly in the context of block-encoding.

Definition 2.22 (Partial inner product)**.** *Consider two finite dimensional Hilbert spaces $\mathcal{H}_A \cong \mathbb{C}^N$ with an orthonormal basis $\{|e_i\rangle\}_{i\in[N]}$, and $\mathcal{H}_B \cong \mathbb{C}^M$ with an orthonormal basis $\{|f_i\rangle\}_{i\in[M]}$. The partial inner product $\langle\cdot|\cdot\rangle$ is a map $\mathcal{H}_A \times (\mathcal{H}_A \otimes \mathcal{H}_B) \to \mathcal{H}_B$ defined as follows. For any $v \in \mathcal{H}_A$, $w \in \mathcal{H}_A \otimes \mathcal{H}_B$*

$$\langle v|w\rangle = \sum_{ij} \left(\langle v|e_i\rangle \langle e_i, f_j|w\rangle\right) |f_j\rangle \in \mathcal{H}_B. \tag{2.49}$$

With some abuse of notation, the partial inner product $\langle\cdot|\cdot\rangle$ also denotes a map: $(\mathcal{H}_A \otimes \mathcal{H}_B) \times \mathcal{H}_A \to \mathcal{H}_B^\star$ according to

$$\langle w|v\rangle = \sum_{ij} \left(\langle e_i|v\rangle \langle w|e_i, f_j\rangle\right) \langle f_j| \in \mathcal{H}_B^\star. \tag{2.50}$$

This definition of a partial inner product has been used in the literature in several works such as [LC17b]. A problem with the notation though is that it requires that the reader pay close attention to the dimensions of the objects in question in order to infer the dimension of the output with a partial inner product. This runs counter to the advantages of Dirac notation which can be confusing when used in the context of conventional Dirac notation where the inner product is always a scalar. While its brevity is an advantage, great care must be taken when using the above notation to avoid making mistakes about the shape of the output.

Example 2.23. Let $|v\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ be a one-qubit state, $|w\rangle = |0\rangle \otimes (|00\rangle + |11\rangle) + |1\rangle \otimes (|01\rangle + |10\rangle)$ be a three-qubit state, then the partial inner product

$$\langle v|w\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) + \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \tag{2.51}$$

is a two-qubit state. ◇

Example 2.24. Let $w = \sum_{i\in[N]} |e_i\rangle \otimes |w_i\rangle$ be reshaped into a matrix

$$W = \begin{pmatrix} w_0 & \cdots & w_{N-1} \end{pmatrix} \in \mathbb{C}^{N\times M}. \tag{2.52}$$

Then the partial inner product $\langle e_i|w\rangle$ for $i \in [N]$ picks out the i-th column w_i. Similarly, the partial inner product $\langle w|e_i\rangle$ picks out the ith row of $W^\dagger$, which is $w_i^\dagger$. ⋄

The partial inner product between pure states provides a natural way to focus our attention on one of the subspaces involved. Sometimes however, we will wish to apply a transformation on the system in question. This generalizes the concept of the partial inner product, and will be vital in our later discussion on block-encoding in Chapter 9.

Definition 2.25 (Partial application of operators)**.** *Consider two finite dimensional Hilbert spaces $\mathcal{H}_A \cong \mathbb{C}^N$ with an orthonormal basis $\{|e_i\rangle\}_{i\in[N]}$, and $\mathcal{H}_B \cong \mathbb{C}^M$ with an orthonormal basis $\{|f_i\rangle\}_{i\in[M]}$. A partial application is a map $(\mathcal{H}_A^\star \otimes L(\mathcal{H}_B)) \times (\mathcal{H}_A \otimes \mathcal{H}_B) \to \mathcal{H}_B$ so that for $|v\rangle \in \mathcal{H}_A$, $C \in L(\mathcal{H}_B)$, $|u\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$,*

$$(\langle v| \otimes C)\,|u\rangle = \sum_{jk} \left(\langle v|e_j\rangle \,\langle e_j, f_k|u\rangle\right)(C\,|f_k\rangle) \in \mathcal{H}_B. \tag{2.53}$$

Similarly we define

$$\langle u|\,(|v\rangle \otimes C) = \sum_{jk} \left(\langle e_j|v\rangle \,\langle u|e_j, f_k\rangle\right)(\langle f_k|\,C) \in \mathcal{H}_B^\star. \tag{2.54}$$

Example 2.26. The partial inner product can also be viewed as a partial application of the identity gate, i.e., for $|v\rangle \in \mathcal{H}_A$, $I \in L(\mathcal{H}_B)$, $|u\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$

$$\langle v|u\rangle = (\langle v| \otimes I)\,|u\rangle\,, \quad \langle u|v\rangle = \langle u|\,(|v\rangle \otimes I). \tag{2.55}$$

For $T = \sum_{jk} |e_j\rangle\langle e_k| \otimes T_{jk}$, the quantity $(\langle e_i| \otimes I)T$ can be represented as a rectangular matrix that consists of the i-th block row of T:

$$\begin{aligned}(\langle e_i| \otimes I)T &= (\langle e_i| \otimes I) \sum_{jk} |e_j\rangle\langle e_k| \otimes T_{jk} \\ &= \sum_k \langle e_k| \otimes T_{ik} \equiv \begin{pmatrix} T_{i,0} & \cdots & T_{i,N-1} \end{pmatrix},\end{aligned} \tag{2.56}$$

Similarly, $T(|e_j\rangle \otimes I)$ picks out the j-th block column of the matrix T.

$$\begin{aligned}T(|e_j\rangle \otimes I) &= \sum_{ik} \delta_{jk}\,|e_i\rangle \otimes T_{ik} \\ &= \sum_i |e_i\rangle \otimes T_{ij} \equiv \begin{pmatrix} T_{0,j} \\ \vdots \\ T_{N-1,j} \end{pmatrix},\end{aligned} \tag{2.57}$$

and $(\langle e_i| \otimes I)T(|e_j\rangle \otimes I)$ returns the (i,j)-th block T_{ij} as can be seen via

$$(\langle e_i| \otimes I)T(|e_j\rangle \otimes I) = (\langle e_i| \otimes I) \sum_k |e_k\rangle \otimes T_{kj} = T_{ij}. \tag{2.58}$$

With some abuse of notation, we may omit the $\otimes I$ notation, so $(\langle e_i| \otimes I)T(|e_j\rangle \otimes I)$ may be written simply as $\langle e_i|\,T\,|e_j\rangle$. ⋄

2.3. Density operator

So far all quantum states encountered have been described by a single state vector $|\psi\rangle$. How to describe a classical mixture of state vectors, such as the state after a measurement process? How can the state of a subsystem within a larger quantum system be defined? The answer to these questions requires the formulation of the density operator (also called density matrix).

Definition 2.27 (Density operator)**.** *A linear operator $\rho \in L(\mathbb{C}^N)$ is called a density operator, if $\rho \succeq 0$, and $\operatorname{Tr}\rho = 1$. The set of all density operators is denoted by $\mathcal{D}(\mathbb{C}^N)$.*

The density operator corresponding to a state vector $|\psi\rangle$ is a rank-1 matrix

$$\rho = |\psi\rangle\langle\psi|. \tag{2.59}$$

Recall that quantum mechanics postulates that $|\psi\rangle$ and $|\psi'\rangle = e^{i\theta}|\psi\rangle$ represent the same physical state. This statement is more natural from the perspective of the density operator, since

$$\rho' = |\psi'\rangle\langle\psi'| = e^{i\theta}|\psi\rangle\, e^{-i\theta}\langle\psi| = \rho. \tag{2.60}$$

In physics, such an irrelevant phase factor is referred to as a gauge degree of freedom. The density operator ρ encapsulates the same physical information as is present in $|\psi\rangle$, but with the added benefit of being invariant to the gauge choice.

With some abuse of terminology, throughout this book, both the density operator ρ and the state vector $|\psi\rangle$ are called quantum states. A rank-1 density operator is called a **pure state**.

Exercise 2.2. Prove that all eigenvalues of a density operator ρ belong to $[0, 1]$. Furthermore, $\rho^2 \preceq \rho$, and the equality holds if and only if ρ is a pure state.

If ρ is not a pure state, then it is called a **mixed state**. We can diagonalize the density matrix as

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| =: \sum_i p_i \rho_i, \tag{2.61}$$

where all state vectors $|\psi_i\rangle$ are orthogonal to each other, and each ρ_i is a pure state. On the other hand, if we have the ability to prepare each pure state ρ_i, then to create the mixed state ρ, all we need to do is prepare a state ρ_i randomly, with the probability of preparing each state given by p_i. In essence, a mixed state can be seen as a **classical ensemble** of pure quantum states. In particular, an n-qubit state $\rho = \frac{I}{2^n}$ is called the **maximally mixed state**.

Let $\{\rho_j\}$ be a set of density operators. With any discrete probability distribution $\{p_j\}$, define $\rho' = \sum_j p_j \rho_j$. Then $\rho' \succeq 0$ and $\operatorname{Tr}[\rho'] = \sum_j p_j \operatorname{Tr}[\rho_j] = \sum_j p_j = 1$. Therefore ρ' is a density operator. In other words, a classical ensemble of (pure or mixed) density operators is also a density operator.

Example 2.28 (Expectation value of a quantum observable)**.** Let us consider the expectation value of an observable O with respect to a mixed state ρ. Since the expectation value with respect to a pure state is

$$\langle O\rangle_{\rho_i} = \langle\psi_i|O|\psi_i\rangle = \operatorname{Tr}[O\rho_i], \tag{2.62}$$

if we obtain the expectation value for a mixed state that obtains a pure state ρ_i with probability p_i, the expectation value is concisely written as

$$\langle O\rangle_{\rho} = \sum_i p_i \operatorname{Tr}[O\rho_i] = \operatorname{Tr}[O\rho]. \tag{2.63}$$

◇

The measurement process can be described without referring to a quantum observable. A quantum measurement can be described by a set of measurement operators $\{M_m\}$, where m labels the different possible outcomes of the measurement. The operators M_m act on the state space $\mathcal{H}$ of the system and satisfy the completeness relation: $\sum_m M_m^\dagger M_m = I$. After a measurement described by M_m is made on a quantum system in a state ρ, the probability of getting result m is given by

$$p_m = \mathrm{Tr}[M_m \rho M_m^\dagger]. \tag{2.64}$$

If outcome m occurs, then the state of the quantum system collapses to a new state

$$\rho'_m = \frac{M_m \rho M_m^\dagger}{\mathrm{Tr}[M_m \rho M_m^\dagger]}. \tag{2.65}$$

The density operator of the resulting ensemble is

$$\rho' = \sum_m p_m \rho'_m = \sum_m M_m \rho M_m^\dagger. \tag{2.66}$$

If each M_m is a projection operator denoted by P_m, then $\{P_m\}$ is called a **projective measurement**. When a quantum observable is measured, the action that is performed on the quantum system is a projective measurement. That is, the state of the system is projected onto an eigenstate of the observable, corresponding to the obtained result of the measurement.

Example 2.29 (Projective measurement)**.** Let the initial state $\rho = |\psi\rangle\langle\psi|$ be a pure state subject to a projective measurement $\{P_m\}_m$. After measurement, the system collapses into a state $|\psi_m\rangle = P_m|\psi\rangle / \sqrt{p_m}$ with probability $p_m = \langle\psi|P_m|\psi\rangle$. If we attempt to represent it by a pure state, one natural choice seems to be $|\psi'\rangle = \sum_m \sqrt{p_m}|\psi_m\rangle$. However, using the normalization condition of the projective measurement in Eq. (2.30)

$$\sum_m \sqrt{p_m}|\psi_m\rangle = \sum_m \sqrt{p_m} P_m|\psi\rangle / \sqrt{p_m} = \sum_m P_m|\psi\rangle = |\psi\rangle. \tag{2.67}$$

In other words, state before and after the measurement is exactly the same! This clearly does not make sense.

Instead, the resulting state should be be represented by a mixed state

$$\rho' = \sum_m p_m |\psi_m\rangle\langle\psi_m| = \sum_m P_m|\psi\rangle\langle\psi| P_m = \sum_m P_m \rho P_m. \tag{2.68}$$

◇

The partial trace is an operation on a joint quantum state (often representing a composite system), which effectively "traces out" one or more subsystems to leave a reduced density operator for the remaining subsystem(s). The operation is widely used in quantum mechanics, especially in the study of open quantum systems, quantum information, and quantum computation.

Definition 2.30 (Partial trace)**.** *Consider two finite dimensional Hilbert spaces $\mathcal{H}_A \cong \mathbb{C}^N$ with an orthonormal basis $\{|e_i\rangle\}_{i\in[N]}$, and $\mathcal{H}_B \cong \mathbb{C}^M$, and $T \in L(\mathcal{H}_A \otimes \mathcal{H}_B)$. The partial trace over $\mathcal{H}_A$, denoted by $\mathrm{Tr}_A(T)$ is an element in $L(\mathcal{H}_B)$ defined as:*

$$\mathrm{Tr}_A(T) = \sum_{i\in[N]} (\langle e_i| \otimes I) T (|e_i\rangle \otimes I). \tag{2.69}$$

The partial trace $\mathrm{Tr}_B(T)$ is defined similarly.

Example 2.31. The matrix representation of $T \in L(\mathbb{C}^N \otimes \mathbb{C}^M)$ takes the form of a block matrix

$$T = \begin{pmatrix} T_{0,0} & \cdots & T_{0,N-1} \\ \vdots & \ddots & \vdots \\ T_{N-1,0} & \cdots & T_{N-1,N-1} \end{pmatrix}, \quad T_{ij} \in \mathbb{C}^{M\times M}. \tag{2.70}$$

Then

$$\mathrm{Tr}_A(T) = \sum_{i\in[N]} T_{ii} \tag{2.71}$$

is the sum of all diagonal blocks. ◇

Given a density operator $\rho \in \mathcal{D}(\mathcal{H}_A \otimes \mathcal{H}_B)$, the partial trace

$$\rho_A = \mathrm{Tr}_B[\rho] \in \mathcal{D}(\mathcal{H}_A), \quad \rho_B = \mathrm{Tr}_A[\rho] \in \mathcal{D}(\mathcal{H}_B) \tag{2.72}$$

are called **reduced density operators**. In particular, if $\rho = \rho_1 \otimes \rho_2$, then

$$\mathrm{Tr}_B[\rho] = \rho_1, \quad \mathrm{Tr}_A[\rho] = \rho_2. \tag{2.73}$$

Note that even if ρ is a pure state, in general, the reduced density operators ρ_A, ρ_B are mixed states.

If a quantum observable is defined only on the subsystem A, i.e., $O = O_A \otimes I_B$ and $O_A = \sum_m \lambda_m P_m$, then when measuring a quantum state ρ with respect to O, the probability of obtaining λ_m, and the expectation value only depend on the reduced density matrix ρ_A:

$$p_m = \mathrm{Tr}[(P_m \otimes I)\rho] = \mathrm{Tr}[P_m \,\mathrm{Tr}_B[\rho]] = \mathrm{Tr}[P_m \rho_A], \quad \mathbb{E}_\rho[O] = \mathrm{Tr}[(O_A \otimes I)\rho] = \mathrm{Tr}[O_A \rho_A]. \tag{2.74}$$

Exercise 2.3. The **Bell state** (also called the EPR pair) is defined to be

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{2.75}$$

Use the partial trace over the second qubit to prove that the Bell state cannot be written as any product state $|a\rangle \otimes |b\rangle$.

Example 2.32 (Purification of mixed state)**.** Any mixed state can always be dilated to a pure state using ancilla qubits. In particular, any n-qubit mixed state ρ can be expressed as $\sum_j p_j |\lambda_j\rangle\langle\lambda_j|$ where $|\lambda_j\rangle$ are the eigenvectors of ρ, and p_j is the corresponding eigenvalue. Given this we can construct a $2n$-qubit pure state

$$|\rho\rangle := \sum_j \sqrt{p_j}\, |\lambda_j\rangle_A \, |\lambda_j\rangle_B \,. \tag{2.76}$$

Then $\mathrm{Tr}_B(|\rho\rangle\langle\rho|) = \rho$. ◇

A more general concept than projective measurement is called **generalized measurement**, also called positive operator-valued measure (POVM).

Definition 2.33. *A **positive operator-valued measure** (POVM) is a set of positive semidefinite operators $\{E_m\}$ that sum to the identity:*

$$\sum_m E_m = I, \quad E_m \succeq 0. \tag{2.77}$$

If a quantum system is in state ρ, the probability of obtaining outcome m is given by

$$p_m = \mathrm{Tr}[E_m \rho]. \tag{2.78}$$

Unlike projective measurements, the elements E_m of a POVM are not necessarily orthogonal, nor are they required to be projection operators (i.e., E_m^2 need not equal E_m). However, POVMs provide the most general description of quantum measurements. On the other hand, the Naimark's dilation theorem (see e.g. [Wat18, Chapter 2.3]) tells us that any generalized measurement can be implemented by coupling the system of interest to an ancilla system and performing a standard projective measurement on the composite system.

THEOREM 2.34 (Naimark's dilation theorem). *Every POVM can be realized as a projective measurement on a larger Hilbert space. Specifically, given a POVM $\{E_m\}$ on $\mathcal{H}_A$, there exists an auxiliary Hilbert space $\mathcal{H}_B$, a pure state $|0\rangle_B \in \mathcal{H}_B$, and a projective measurement $\{P_m\}$ on $\mathcal{H}_A \otimes \mathcal{H}_B$ such that for any state ρ on $\mathcal{H}_A$:*

$$\mathrm{Tr}[E_m \rho] = \mathrm{Tr}[P_m(\rho \otimes |0\rangle\langle 0|_B)]. \tag{2.79}$$

PROOF. Since each E_m is positive semidefinite, we can define $M_m = \sqrt{E_m}$ such that $M_m^\dagger M_m = E_m$. Let $\mathcal{H}_B$ be a Hilbert space with an orthonormal basis $\{|m\rangle\}$ corresponding to the indices of the POVM elements. We define a linear operator $V : \mathcal{H}_A \to \mathcal{H}_A \otimes \mathcal{H}_B$ by its action on an arbitrary state $|\psi\rangle \in \mathcal{H}_A$:

$$V|\psi\rangle = \sum_m M_m |\psi\rangle \otimes |m\rangle_B . \tag{2.80}$$

This operator is an isometry because

$$\langle V\psi|V\psi\rangle = \sum_{m,n} \langle\psi| M_m^\dagger M_n |\psi\rangle \langle m|n\rangle = \sum_m \langle\psi| M_m^\dagger M_m |\psi\rangle = \langle\psi| \left(\sum_m E_m\right) |\psi\rangle = \langle\psi|\psi\rangle . \tag{2.81}$$

We can extend this isometry to a unitary operator U acting on $\mathcal{H}_A \otimes \mathcal{H}_B$ such that $U(|\psi\rangle \otimes |0\rangle_B) = V|\psi\rangle$. Now, define the projective measurement on the composite system by the projectors $\Pi_m = I_A \otimes |m\rangle\langle m|_B$. Let $P_m = U^\dagger \Pi_m U$. Since U is unitary and $\{\Pi_m\}$ are orthogonal projectors summing to identity, $\{P_m\}$ is a valid projective measurement. Finally, we verify the probability condition:

$$\begin{aligned} \mathrm{Tr}[P_m(\rho \otimes |0\rangle\langle 0|_B)] &= \mathrm{Tr}[U^\dagger \Pi_m U(\rho \otimes |0\rangle\langle 0|_B)] \\ &= \mathrm{Tr}[\Pi_m U(\rho \otimes |0\rangle\langle 0|_B)U^\dagger] \\ &= \mathrm{Tr}[(I_A \otimes |m\rangle\langle m|_B) V \rho V^\dagger]. \end{aligned} \tag{2.82}$$

Using the definition of V, we have $V\rho V^\dagger = \sum_{k,l} M_k \rho M_l^\dagger \otimes |k\rangle\langle l|_B$. Substituting this back,

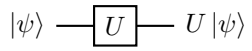
$$\begin{aligned} \mathrm{Tr}[P_m(\rho \otimes |0\rangle\langle 0|_B)] &= \mathrm{Tr}\left[(I_A \otimes |m\rangle\langle m|_B) \sum_{k,l} M_k \rho M_l^\dagger \otimes |k\rangle\langle l|_B\right] \\ &= \mathrm{Tr}[M_m \rho M_m^\dagger] = \mathrm{Tr}[M_m^\dagger M_m \rho] = \mathrm{Tr}[E_m \rho]. \end{aligned} \tag{2.83}$$

□

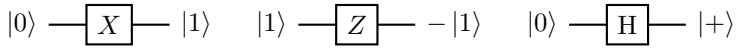
2.4. Quantum circuit

Nearly all quantum algorithms operate on multi-qubit quantum systems. When quantum operators operate on two or more qubits, writing down quantum states in terms of its components as in Eq. (2.36) quickly becomes cumbersome. The language of **quantum circuit** offers a graphical and compact manner for writing down the procedure of applying a sequence of quantum operators to a quantum state. For more details see [NC00, Section 4.2, 4.3].

In the quantum circuit language, time flows from the left to right, i.e., the input quantum state appears on the left, and the quantum operator appears on the right, and each "wire" represents a qubit i.e.,



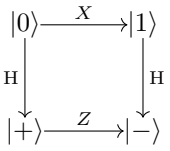
Here are a few examples:



which is a graphical way of writing

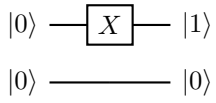
$$X|0\rangle = |1\rangle\,, \quad Z|1\rangle = -|1\rangle\,, \quad \mathrm{H}|0\rangle = |+\rangle\,. \tag{2.84}$$

The relation between these states can be expressed in terms of the following diagram



(2.85)

Also verify that

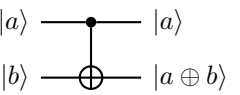


which is a graphical way of writing

$$(X \otimes I)|00\rangle = |10\rangle\,. \tag{2.86}$$

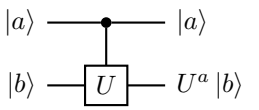
Note that the input state can be general, and in particular does not need to be a product state. For example, if the input is a Bell state (2.75), we just apply the quantum operator to $|00\rangle$ and $|11\rangle$, respectively and multiply the results by $1/\sqrt{2}$ and add together. To distinguish with other symbols, these single qubit gates may be either written as X, Y, Z, H or (using the roman font) $\mathrm{X}, \mathrm{Y}, \mathrm{Z}, \mathrm{H}$.

The quantum circuit for the CNOT gate is

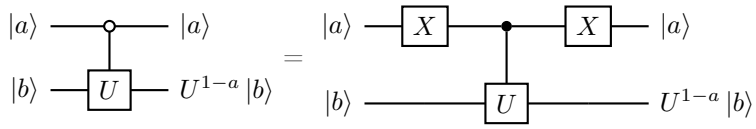


Here the "dot" means that the quantum gate connected to the dot only becomes active if the state of the qubit 0 (called the control qubit) is $a = 1$. This justifies the name of the CNOT gate (controlled NOT).

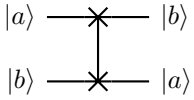
Similarly,



is the controlled U gate for some unitary U. Here $U^a = I$ if $a = 0$. The CNOT gate can be obtained by setting $U = X$. Sometimes we want to control a unitary only if the control qubit is zero rather than 1. In this case, we represent the control using a hollow circle as shown below.

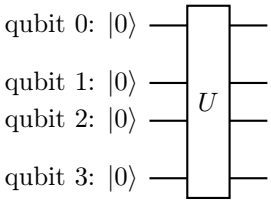


Another commonly used two-qubit gate is the **SWAP gate**, which swaps the state in the 0-th and the 1-st qubits.



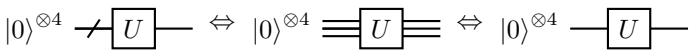
Exercise 2.4. Write down the matrix representation of the SWAP gate.

Quantum operators applied to multiple qubits can be written in a similar manner:

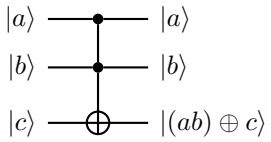


For a multi-qubit quantum circuit, unless stated otherwise, the first qubit will be referred to as the qubit 0, and the second qubit as the qubit 1, etc.

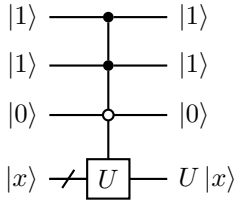
When the context is clear, we may also use a more compact notation for the multi-qubit quantum operators:



One useful multiple qubit gate is the **Toffoli gate** (or controlled-controlled-NOT, CCNOT gate).



We may also want to apply a n-qubit unitary U only when certain conditions are met

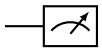


where the empty circle means that the gate being controlled only becomes active when the value of the control qubit is 0. This can be used to write down the quantum "if" statements, i.e., when the qubits $0, 1$ are at the $|1\rangle$ state and the qubit 2 is at the $|0\rangle$ state, then apply U to $|x\rangle$.

A set of qubits is often called a **quantum register** (or register for short). For example, in the picture above, the main quantum state of interest (an n qubit quantum state $|x\rangle$) is called the system register. The first 3 qubits can be called the control register. When multiple registers are present, we can distinguish them by writing $|x\rangle_A |y\rangle_B$, so that we can refer to the quantum state associated with the qubits in registers A and B, respectively.

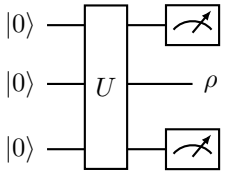
In quantum computation, a classical bit-string is denoted as $x \in \{0,1\}^n$, and the corresponding $|x\rangle$ is called a **classical state**. The set of all classical states form the **computational basis** of an n-qubit system. It is worth noting that $\{|x\rangle \langle x| \, | x \in \{0,1\}^n\}$ forms a set of projective measurement operators, which can be identified with the simultaneous measurement with respect to Pauli-Z operators $Z_1, \ldots, Z_n$. Consequently, when a measurement is performed with respect to the Pauli-Z operator, it is called a measurement in the computational basis.

The circuit symbol for the quantum measurement with respect to a single Pauli-Z is



Example 2.35 (Measure Pauli-Z operators)**.** For a quantum state $|\psi\rangle$, the measurement of a multi-qubit Pauli-Z operator of the form $(Z_1)^{a_1} \cdots (Z_n)^{a_n}$, where $a_1, \ldots, a_n \in \{0,1\}$ can be directly implemented at the circuit level. For example, for a 3-qubit system, the following circuit

(2.87)



measures the outcome of Z_1 and Z_3, yielding 4 possible outcomes $\{00, 01, 10, 11\}$ with respective probabilities $\{p(00), p(01), p(10), p(11)\}$. Now consider an observable $O = Z_1 Z_3$ whose eigenvalues are 1 and -1. The probability of obtaining each eigenvalue is

$$p(O = 1) = p(00) + p(11), \quad p(O = -1) = p(01) + p(10). \tag{2.88}$$

◇

Example 2.36 (**Hadamard test circuit**)**.** The Hadamard test is a useful tool for computing the expectation value of an unitary operator with respect to a state, i.e., $\langle\psi|U|\psi\rangle$. It can be used to solve the phase estimation problem. The Hadamard test uses two circuits to estimate the real and imaginary part of the expectation value separately.

The (real) Hadamard test is the quantum circuit in Fig. 2.1 for estimating $\mathrm{Re}\,\langle\psi|U|\psi\rangle$.

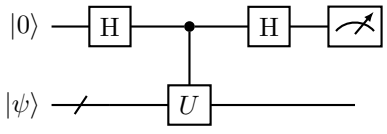


FIGURE 2.1. Hadamard test for $\mathrm{Re}\,\langle\psi|U|\psi\rangle$.

To verify this, we find that the circuit transforms $|0\rangle|\psi\rangle$ as

$$\begin{aligned}|0\rangle|\psi\rangle &\xrightarrow{\mathrm{H}\otimes I} \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|\psi\rangle \\ &\xrightarrow{c\text{-}U} \frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle+|1\rangle U|\psi\rangle) \\ &\xrightarrow{\mathrm{H}\otimes I} \frac{1}{2}|0\rangle(|\psi\rangle+U|\psi\rangle)+\frac{1}{2}|1\rangle(|\psi\rangle-U|\psi\rangle).\end{aligned}$$

The probability of measuring the qubit 0 to be in state $|0\rangle$ is

$$p(0)=\frac{1}{2}(1+\operatorname{Re}\langle\psi|U|\psi\rangle). \tag{2.89}$$

This is well defined since $-1\le \operatorname{Re}\langle\psi|U|\psi\rangle\le 1$.

To obtain the imaginary part, we can use the circuit in Fig. 2.2 called the (imaginary) Hadamard test, where

$$S=\begin{pmatrix}1&0\\0&i\end{pmatrix} \tag{2.90}$$

is called the phase gate.

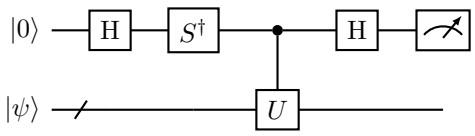


FIGURE 2.2. Hadamard test for $\operatorname{Im}\langle\psi|U|\psi\rangle$.

Similar calculation shows the circuit transforms $|0\rangle|\psi\rangle$ to the state

$$\frac{1}{2}|0\rangle(|\psi\rangle-iU|\psi\rangle)+\frac{1}{2}|1\rangle(|\psi\rangle+iU|\psi\rangle). \tag{2.91}$$

Therefore the probability of measuring the qubit 0 to be in state $|0\rangle$ is

$$p(0)=\frac{1}{2}(1+\operatorname{Im}\langle\psi|U|\psi\rangle). \tag{2.92}$$

Combining the results from the two circuits, we obtain the estimate to $\langle\psi|U|\psi\rangle$.

◇

Example 2.37 (Overlap estimate using the SWAP test)**.** A special case of the Hadamard test is called the **SWAP test**, which can be used to estimate the overlap of two quantum states $|\langle\varphi|\psi\rangle|$. The quantum circuit for the swap test is

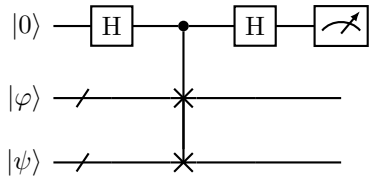


FIGURE 2.3. Circuit for the SWAP test.

Note that this is exactly the Hadamard test with U being the swap gate. Direct calculation shows that the probability of measuring the first qubit and obtaining outcome 0 is

$$p(0) = \frac{1}{2}(1 + \mathrm{Re}\,\langle\varphi, \psi|\psi, \varphi\rangle) = \frac{1}{2}(1 + |\langle\varphi|\psi\rangle|^2). \tag{2.93}$$

◇

2.5. Copy operation and no-cloning theorem

Most computer programs on classical computers have an assignment of the form $y = x$, or $y = \mathrm{copy}(x)$, which stores the value in the variable x in a new location in memory as a variable y. In scientific computation, this is the foundation of iterative methods, which solve a problem by making progress gradually. For example, classical iterative algorithms for solving linear systems require storing intermediate variables. It is therefore striking that such a basic step is explicitly ruled out by quantum mechanics.

The **no-cloning theorem** is an early result in quantum computation: it forbids a universal quantum copy operation (see also [NC00, Section 12.1]).

THEOREM 2.38 (No cloning). *Given a fixed state $|s\rangle$ (e.g. $|s\rangle = |0^n\rangle$), there is no unitary operator U that acts as a copy operation, in the sense that for every state $|x\rangle$,*

$$U\,|x\rangle \otimes |s\rangle = |x\rangle \otimes |x\rangle\,. \tag{2.94}$$

PROOF. Assume such a U exists. Take two states $|x_1\rangle, |x_2\rangle$ such that $0 < |\langle x_1|x_2\rangle| < 1$. Then

$$U\big(|x_1\rangle \otimes |s\rangle\big) = |x_1\rangle \otimes |x_1\rangle\,, \quad U\big(|x_2\rangle \otimes |s\rangle\big) = |x_2\rangle \otimes |x_2\rangle\,. \tag{2.95}$$

Taking the inner product of the two equations and using unitarity,

$$\langle x_1|x_2\rangle = \langle x_1, s|x_2, s\rangle = \langle x_1, s|U^\dagger U|x_2, s\rangle = \langle x_1, x_1|x_2, x_2\rangle = \langle x_1|x_2\rangle^2\,. \tag{2.96}$$

Hence $\langle x_1|x_2\rangle \in \{0, 1\}$, contradicting $0 < |\langle x_1|x_2\rangle| < 1$. □

There are two important special cases in which copying is possible without contradicting Theorem 2.38. The first is that $|x\rangle$ is not arbitrary: it is a specific state for which we know a preparation procedure, i.e., $|x\rangle = U_x\,|s\rangle$ for a known unitary U_x and some fixed state $|s\rangle$. Then we can prepare a second copy of $|x\rangle$ via

$$(I \otimes U_x)\,|x\rangle \otimes |s\rangle = |x\rangle \otimes |x\rangle\,. \tag{2.97}$$

The second is copying classical information in the computational basis, using the CNOT gate. i.e.,

$$\mathrm{CNOT}\,|x, 0\rangle = |x, x\rangle\,, \quad x \in \{0, 1\}. \tag{2.98}$$

The same principle applies to copying classical information from multiple qubits. Fig. 2.4 gives an example of copying the classical information stored in 3 bits.

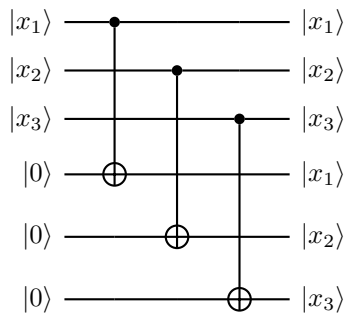


FIGURE 2.4. Copying classical information using multi-qubit CNOT gates.

In general, multi-qubit CNOT operations can be used to perform classical copying in the computational basis. Note that in the circuit model, this can be implemented with a depth-1 circuit, since these CNOT gates act on disjoint sets of qubits.

The copying of classical information is compatible with Theorem 2.38 in the following sense. The proof of Theorem 2.38 uses two non-orthogonal states $|x_1\rangle, |x_2\rangle$ to obtain a contradiction. However, all states in the computational basis are orthogonal to each other. Therefore, there exist unitaries that copy a specified orthonormal set of states, but a universal quantum copy operation is impossible.

Example 2.39. Let us verify that the CNOT gate does not violate the no-cloning theorem, i.e., it cannot be used to copy a general superposition $|x\rangle = a|0\rangle + b|1\rangle$. Direct calculation shows

$$\text{CNOT}\,|x\rangle \otimes |0\rangle = a|00\rangle + b|11\rangle \neq |x\rangle \otimes |x\rangle \tag{2.99}$$

unless $ab = 0$. In particular, if $|x\rangle = |+\rangle$, then CNOT creates a Bell state. ◇

Similar to the quantum no-cloning theorem, there does not exist a unitary U that performs a "deleting" operation which resets an unknown state $|x\rangle$ to $|0^n\rangle$:

$$U|0^n\rangle \otimes |x\rangle = |0^n\rangle \otimes |0^n\rangle \tag{2.100}$$

for all $|x\rangle$. Indeed, if $|x_1\rangle, |x_2\rangle$ are orthogonal, then unitarity implies

$$0 = \langle 0^n, x_1|0^n, x_2\rangle = \langle 0^n, x_1|U^\dagger U|0^n, x_2\rangle = \langle 0^n, 0^n|0^n, 0^n\rangle = 1, \tag{2.101}$$

a contradiction.

A more general version of the no-deleting theorem is as follows: given two copies of an arbitrary quantum state, it is impossible to delete one of the copies. Specifically, there is no unitary U performing the following operation using fixed known states $|s\rangle, |s'\rangle$,

$$U|x\rangle|x\rangle|s\rangle = |x\rangle|0^n\rangle|s'\rangle \tag{2.102}$$

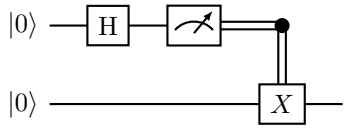
for an arbitrary unknown state $|x\rangle$.

Exercise 2.5. Prove the version of the no-deleting theorem in Eq. (2.102).

2.6. Deferred and implicit measurements

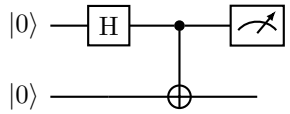
There are two important principles related to quantum measurements: the principle of deferred measurement, and the principle of implicit measurement. At first glance, both principles may seem counterintuitive.

Example 2.40 (Deferring quantum measurements)**.** Consider the circuit



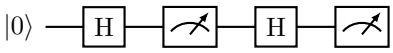
Here the double line denotes a classical control operation. The outcome is that qubit 0 has probability 1/2 of outputting 0, and qubit 1 is in the state $|0\rangle$. Qubit 0 also has probability 1/2 of outputting 1, and qubit 1 is in the state $|1\rangle$.

However, we may replace the classical control operation after the measurement by a quantum controlled X (i.e. CNOT), and measure qubit 0 afterwards:

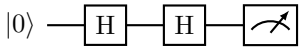


It can be verified that the result is the same. In this sense, CNOT copies the measurement outcome of qubit 0 to qubit 1 in the computational basis. ◇

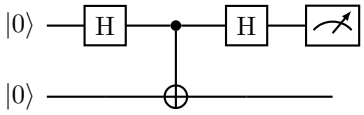
Example 2.41 (Deferring measurement requires extra qubits)**.** The procedure of deferring quantum measurements using CNOTs is general, and important. Consider the following circuit:



The probability of obtaining 0 or 1 is 1/2. However, if we simply "defer" the measurement to the end by removing the intermediate measurement, we obtain

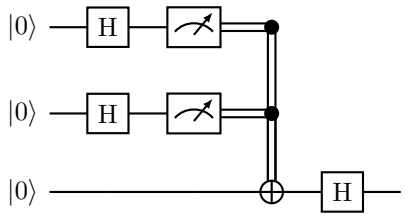


The result of the measurement is deterministically 0! The correct way of deferring the intermediate quantum measurement is to introduce another qubit



Measuring the qubit 0, we obtain 0 or 1 w.p. 1/2, respectively. Hence when deferring quantum measurements, it is necessary to store the intermediate information in extra (ancilla) qubits, even if such information is not used afterwards. ◇

Exercise 2.6. Consider a quantum circuit with three qubits, initially all in state $|0\rangle$. The circuit is as follows:

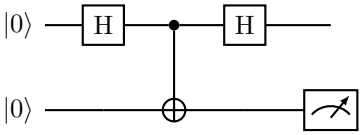


Design a quantum circuit that defers the measurements of the first two qubits to the end, using two additional ancilla qubits to store the intermediate measurement information. After the deferred measurements, describe the final states of all qubits. Ensure the overall effect on the ancilla qubits is the same as if the measurements were performed immediately.

The **principle of deferred measurement** states that in a quantum circuit, measurement operations can be postponed from an intermediate stage to the end of the circuit. This remains true even when a measurement at an intermediate step determines the conditional control of subsequent gates: such classical controls can be replaced by quantum controls. One use of this principle is to simplify quantum circuits and their analysis, by expressing the computation as a unitary circuit (possibly using ancilla qubits) followed by measurements at the end.

The **principle of implicit measurements** states that, for predicting the statistics of the qubits that are measured at the end of a circuit, it is irrelevant whether other qubits are explicitly measured at the end or simply left unmeasured.

Example 2.42. Consider the circuit:



Before the measurement, the final state is $\frac{1}{2}(|00\rangle + |01\rangle) + \frac{1}{2}(|10\rangle - |11\rangle)$. So measuring qubit 1 yields 0 and 1 with equal probability.

If we measure qubit 0 first, verify that qubit 1 will be in the mixed state

$$\rho = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|, \tag{2.103}$$

so if we measure qubit 1 afterwards, we again obtain 0 and 1 with equal probability. ◇

Why does the principle of implicit measurement hold? Assume the quantum system consists of two subsystems A and B. Recall from Eq. (2.74) that a measurement on subsystem A only depends on the reduced density matrix ρ_A. Thus it suffices to show that ρ_A does not depend on whether B is measured. Let $\{P_i\}$ be the projectors onto the computational basis of B, and let the joint state be ρ. If we measure subsystem B and discard the outcome, the joint state becomes

$$\rho' = \sum_i (I \otimes P_i)\rho(I \otimes P_i). \tag{2.104}$$

Then

$$\rho'_A = \mathrm{Tr}_B[\rho'] = \sum_i \mathrm{Tr}_B[(I \otimes P_i)\rho(I \otimes P_i)] = \sum_i \mathrm{Tr}_B[\rho(I \otimes P_i)] = \mathrm{Tr}_B\left[\rho\big(I \otimes \sum_i P_i\big)\right] = \mathrm{Tr}_B[\rho] = \rho_A. \tag{2.105}$$

Therefore, if the qubits in A are to be measured at the end of the circuit, the measurement statistics do not depend on whether the qubits in B are measured or not.

2.7. Sparse matrix, Majorana, fermionic, and bosonic operators

Sparse matrices are among the most important examples of very large matrices that can be efficiently encoded on quantum computers. They are also closely related to many physical Hamiltonians in practical applications.

Definition 2.43 (s-sparse matrix)**.** *A matrix $A \in \mathbb{C}^{M\times N}$ is called s-sparse if each row and column of the matrix contains at most s non-zero entries.*

Example 2.44. A diagonal matrix is 1-sparse. Any diagonal matrix $A \in \mathbb{C}^{2^n\times 2^n}$ can be written as a linear combination of Pauli Z-operators

$$A = \sum_{i_1,\ldots,i_n\in\{0,1\}} J_{i_1,\ldots,i_n}\sigma_{i_1,1}\cdots\sigma_{i_n,n}, \tag{2.106}$$

where $\sigma_{s,k}$ is equal to Z_k if $s = 1$ and I if $s = 0$. Any permutation matrix Π is 1-sparse. A row and column permutation of a 1-sparse matrix is 1-sparse. Any 1-sparse matrix A can be written as ΠD or $D\Pi'$, where D is a diagonal matrix and Π, Π' are permutation matrices. A tridiagonal matrix is 3-sparse. The following matrix

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix} = \left(\sum_{i=1}^{N} e_i\right) e_1^\top \in \mathbb{R}^{N\times N} \tag{2.107}$$

has only one nonzero entry per row, but it is **not** 1-sparse since the first column has N nonzero entries. ◇

Definition 2.45. *The maximal absolute value of the entries of $A \in \mathbb{C}^{M\times N}$, also called the **max norm**, is defined as:*

$$\|A\|_{\max} := \max_{i,j} |A_{ij}| . \tag{2.108}$$

Lemma 2.46. *Let $A \in \mathbb{C}^{N\times N}$ be s-sparse. Then*

$$\|A\| \le s \|A\|_{\max} . \tag{2.109}$$

Proof. For any row i of A, the set of nonzero column indices is denoted by $\mathcal{C}_i$. By Cauchy-Schwarz,

$$|(Ax)_i|^2 = \left|\sum_{j\in\mathcal{C}_i} A_{ij}x_j\right|^2 \le \sum_{j\in\mathcal{C}_i} |A_{ij}|^2 \sum_{j\in\mathcal{C}_i} |x_j|^2 \le s \|A\|_{\max}^2 \sum_{j\in\mathcal{C}_i} |x_j|^2 . \tag{2.110}$$

Then

$$\|Ax\|^2 \le s \|A\|_{\max}^2 \sum_i \sum_{j\in\mathcal{C}_i} |x_j|^2 . \tag{2.111}$$

The condition that A is s-sparse implies that for each j, there are at most s indices i such that $A_{ij} \neq 0$, i.e., j belongs to at most s sets among $\{\mathcal{C}_i\}_i$. Therefore each j can appear at most s times in the double sum. This means

$$\|Ax\|^2 \leq s^2 \|A\|_{\max}^2 \sum_j |x_j|^2 = s^2 \|A\|_{\max}^2 \|x\|^2 . \tag{2.112}$$

Taking the supremum over $x \neq 0$ yields $\|A\| \leq s \|A\|_{\max}$. □

The equality in Lemma 2.46 can be reached by considering a matrix B whose upper left $s \times s$ block is $\|A\|_{\max} ee^\top$, where e is an all 1 vector of length s. Direct computation shows that $\|B\| = s \|A\|_{\max}$.

A useful lemma is that the product of any 1-sparse matrices is 1-sparse.

Lemma 2.47. *Let A and B be $N \times N$ 1-sparse matrices. Then $C = AB$ is also 1-sparse.*

Proof. Since A, B are 1-sparse, there exists permutation matrices Π, Π' and diagonal matrices D, D' so that $A = \Pi D, B = D'\Pi'$. Therefore

$$C = \Pi(DD')\Pi' \tag{2.113}$$

is a permutation of a diagonal matrix, and is therefore 1-sparse. □

Example 2.48. All Pauli gates in $\mathcal{P}_n$ are 1-sparse. This can be proved by induction. First, all Pauli matrices I, X, Y, Z are 1-sparse matrices. Assume all Pauli gates in $\mathcal{P}_{n-1}$ are 1-sparse, then an element in $\mathcal{P}_n$ can always be constructed (up to a reordering of qubits) as

$$P \otimes P_1, \quad P \in \mathcal{P}_{n-1}, P_1 \in \mathcal{P}_1. \tag{2.114}$$

This replaces a nonzero entry in P by a 2×2 matrix that is 1-sparse, so the overall matrix is still 1-sparse. ◇

Example 2.49 (Majorana operator)**.** For a fermionic system defined on n modes, the state space $\mathcal{F} = \otimes_{i=1}^n \mathbb{C}^2 \cong \mathbb{C}^{2^n}$ is called the Fock space. The Majorana fermion operators (or Majorana operators for short) denoted by $\{\gamma_i\}_{i=1}^{2n}$, are Hermitian operators in $L(\mathbb{C}^{2^n})$ satisfying the anticommutation relations:

$$\{\gamma_i, \gamma_j\} := \gamma_i\gamma_j + \gamma_j\gamma_i = 2\delta_{ij}, \quad i, j = 1, \ldots, 2n. \tag{2.115}$$

The canonical realization of Majorana operators is through Pauli operators. When $n = 1$, we simply have

$$\gamma_1 = X, \quad \gamma_2 = Y. \tag{2.116}$$

For the n mode system, the Majorana operators can be defined using the **Jordan–Wigner transformation**,

$$\gamma_{2j-1} = \left(\prod_{k=1}^{j-1} Z_k\right) X_j, \quad \gamma_{2j} = \left(\prod_{k=1}^{j-1} Z_k\right) Y_j, \quad j = 1, \ldots, n. \tag{2.117}$$

So Majorana operators are also 1-sparse matrices. Furthermore, any product of Majorana operators $\gamma_{i_1} \cdots \gamma_{i_k}$, $i_1, \cdots, i_k \in \{1, \ldots, 2n\}$ is 1-sparse. ◇

Example 2.50 (Fermionic operator)**.** For a fermionic system defined on n modes with the Fock space $\mathcal{F} = \otimes_{i=1}^{n} \mathbb{C}^2 \cong \mathbb{C}^{2^n}$, the fermionic creation and annihilation operators, denoted by $a_i^\dagger$ and a_i respectively, are operators in $L(\mathbb{C}^{2^n})$ that satisfy the **canonical anticommutation relations** (CAR):

$$\{a_i, a_j^\dagger\} := a_i a_j^\dagger + a_j^\dagger a_i = \delta_{ij},\ \{a_i, a_j\} = \{a_i^\dagger, a_j^\dagger\} = 0, \quad i, j = 1, \ldots, n. \tag{2.118}$$

The creation operator $a_i^\dagger$ adds a fermion to the mode i, while the annihilation operator a_i removes a fermion from the mode i.

For a single mode system,

$$a = X^+ = \frac{1}{2}(X + iY) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad a^\dagger = X^- = \frac{1}{2}(X - iY) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}. \tag{2.119}$$

In this convention $a^\dagger |0\rangle = |1\rangle, a^\dagger |1\rangle = 0$, $a |1\rangle = |0\rangle, a |0\rangle = 0$. Here $|s\rangle$ denotes the state with s fermions ($s = 0, 1$). The number operator $\hat{n} = a^\dagger a = \frac{1}{2}(1 - Z)$ satisfies $\hat{n} |s\rangle = s |s\rangle$.

For an n-mode system, the fermionic operators are related to the Majorana operators according to the relation:

$$a_i = \frac{1}{2}(\gamma_{2i-1} + i\gamma_{2i}), \quad a_i^\dagger = \frac{1}{2}(\gamma_{2i-1} - i\gamma_{2i}), \quad i = 1, \ldots, n, \tag{2.120}$$

where γ_{2i-1} and γ_{2i} are the Majorana operators associated with the i-th fermionic mode. Therefore any operator defined using a linear combination of fermionic creation and annihilation operators can be expressed as a linear combination of Majorana operators, and vice versa.

From the Jordan–Wigner transformation,

$$a_j = \left(\prod_{k=1}^{j-1} Z_k\right) X_j^+, \quad a_j^\dagger = \left(\prod_{k=1}^{j-1} Z_k\right) X_j^-, \tag{2.121}$$

with

$$X_j^+ = \frac{1}{2}(X_j + iY_j), \quad X_j^- = \frac{1}{2}(X_j - iY_j). \tag{2.122}$$

Since $X^\pm$ are 1-sparse matrices, $a_j^\dagger, a_j, a_j^\dagger a_j, a_j a_j^\dagger$ are also 1-sparse. Furthermore, any product of fermionic operators $a_{i_1}^\dagger \cdots a_{i_k}^\dagger a_{j_1} \cdots a_{j_l}$ is 1-sparse. ◇

Example 2.51 (Bosonic operator)**.** For an n-mode bosonic systems, the bosonic creation and annihilation operators, denoted by $b_i^\dagger$ and b_i respectively, are operators that satisfy the **canonical commutation relations** (CCR):

$$[b_i, b_j^\dagger] := b_i b_j^\dagger - b_j^\dagger b_i = \delta_{ij}, \quad [b_i, b_j] = [b_i^\dagger, b_j^\dagger] = 0, \quad i, j = 1, \ldots, n. \tag{2.123}$$

The creation operator $b_i^\dagger$ adds a boson to the mode i, while the annihilation operator b_i removes a boson from the mode i.

When $n = 1$, these operators satisfy

$$b |0\rangle = 0, \quad b |s\rangle = \sqrt{s} |s - 1\rangle, s = 1, 2, \ldots, \tag{2.124}$$

and

$$b^\dagger |s\rangle = \sqrt{s + 1} |s + 1\rangle, s = 0, 1, 2, \ldots. \tag{2.125}$$

Here $|s\rangle$ denotes a state with s bosons. We also have

$$b^\dagger b |s\rangle = s |s\rangle. \tag{2.126}$$

In the matrix form, we can write

$$b = \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & \cdots \\ 0 & 0 & \sqrt{2} & 0 & \cdots \\ 0 & 0 & 0 & \sqrt{3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \quad b^\dagger = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots \\ \sqrt{1} & 0 & 0 & 0 & \cdots \\ 0 & \sqrt{2} & 0 & 0 & \cdots \\ 0 & 0 & \sqrt{3} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \tag{2.127}$$

These operators are infinite dimensional operators, i.e., operators defined in an infinite dimensional space. They are also 1-sparse. Furthermore, $\|b\|, \left\|b^\dagger\right\| = \infty$, so unlike any finite dimensional matrices, these operators are unbounded. The physical reason is that a single bosonic mode can accommodate an infinite number of bosons, and the energy of a system with an infinite number of bosons in a single mode is infinity.

Due to the commutation relation, multi-mode bosonic operators can be defined using tensor products:

$$b_i = I^{\otimes(i-1)} \otimes b \otimes I^{\otimes(n-i)}, \quad b_i^\dagger = I^{\otimes(i-1)} \otimes b^\dagger \otimes I^{\otimes(n-i)}, \quad i = 1, \ldots, n, \tag{2.128}$$

where the identity operator $I |s\rangle = |s\rangle$ also acts on an infinite dimensional space.

The precise characterization of the Hilbert space for unbounded operators is beyond the scope of this book. However, if we truncate the state space of each bosonic mode to a finite dimensional space with d levels, i.e., $\mathbb{C}^d$, the state space of a bosonic system defined on n modes with d levels per mode is $\mathcal{F} = \otimes_{i=1}^n \mathbb{C}^d \cong \mathbb{C}^{d^n}$ and is finite dimensional.

In a single-mode truncated bosonic system, $b, b^\dagger$ are finite dimensional matrices:

$$b = \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \sqrt{2} & 0 & \cdots & 0 \\ 0 & 0 & 0 & \sqrt{3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \sqrt{d-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \quad b^\dagger = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ \sqrt{1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sqrt{2} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sqrt{3} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sqrt{d-1} & 0 \end{pmatrix}. \tag{2.129}$$

These are 1-sparse matrices of size $d \times d$. Then the multi-mode operators defined in Eq. (2.128) are 1-sparse matrices. Using Lemma 2.47, the product of any multi-mode bosonic operators $b_{i_1}^\dagger \cdots b_{i_k}^\dagger b_{j_1} \cdots b_{j_l}$, where $b, b^\dagger$ are truncated bosonic creation and annihilation operators defined in Eq. (2.129) are 1-sparse matrices. ◇

Exercise 2.7. Prove that the truncated bosonic creation and annihilation operators defined in Eq. (2.129) satisfy the modified commutation relation

$$[b, b^\dagger] = 1 - \frac{d}{(d-1)!} \left(b^\dagger\right)^{d-1} (b)^{d-1}. \tag{2.130}$$

2.8. Selected examples of Hamiltonians in physics, chemistry, and optimization

With the introduction of spin, Majorana, fermionic, and bosonic operators, we can provide several examples of Hamiltonians encountered in applications. Although we will not use all of these examples to illustrate the performance of quantum algorithms, the algorithms in this book can be applied to any of them.

2.8.1. Condensed matter physics.

Example 2.52 (Transverse field Ising model)**.** The Hamiltonian for the one dimensional transverse field Ising model (TFIM) with nearest neighbor interaction of length n is

$$H = -\sum_{i=1}^{n-1} Z_i Z_{i+1} - g\sum_{i=1}^{n} X_i, \tag{2.131}$$

where g is the coupling constant. ◇

Example 2.53 (1D Heisenberg model)**.** The Hamiltonian for the 1D Heisenberg model with nearest neighbor interaction is given by

$$H = -J\sum_{i=1}^{n-1} \mathbf{S}_i \cdot \mathbf{S}_{i+1} \tag{2.132}$$

where J is the interaction strength and $\mathbf{S}_i$ represents the spin operator at site i, defined as

$$\mathbf{S}_i = \frac{1}{2}\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}. \tag{2.133}$$

We can decompose this Hamiltonian into three terms, each associated with the x, y, and z components of the spins:

$$H_x = -\frac{J}{4}\sum_{i=1}^{n-1} X_i X_{i+1}, \quad H_y = -\frac{J}{4}\sum_{i=1}^{n-1} Y_i Y_{i+1}, \quad H_z = -\frac{J}{4}\sum_{i=1}^{n-1} Z_i Z_{i+1}. \tag{2.134}$$

When $J > 0$ the problem is called ferromagnetic, and when $J < 0$ it is called anti-ferromagnetic. ◇

Example 2.54 (2D Heisenberg model)**.** The Hamiltonian for the 2D Heisenberg model on a square lattice is given by:

$$H = -J\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}(\mathbf{S}_{i,j}\cdot\mathbf{S}_{i+1,j} + \mathbf{S}_{i,j}\cdot\mathbf{S}_{i,j+1}) \tag{2.135}$$

We decompose this Hamiltonian into three terms associated with the x, y, and z components of the spins:

$$\begin{aligned} H_x &= -\frac{J}{4}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}(X_{i,j}X_{i+1,j} + X_{i,j}X_{i,j+1}), \\ H_y &= -\frac{J}{4}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}(Y_{i,j}Y_{i+1,j} + Y_{i,j}Y_{i,j+1}), \\ H_z &= -\frac{J}{4}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}(Z_{i,j}Z_{i+1,j} + Z_{i,j}Z_{i,j+1}). \end{aligned} \tag{2.136}$$

◇

Example 2.55 (k-Local Hamiltonian)**.** A k-local Hamiltonian is a quantum Hamiltonian where each term acts nontrivially on at most k qubits. One convenient way to write such a Hamiltonian on n qubits is as a linear combination of Pauli strings of weight at most k. For example, one may write

$$H = \sum_{S\subseteq[n],\,|S|\leq k} \sum_{\alpha\in\{0,1,2,3\}^S} J_{S,\alpha} \prod_{i\in S} \sigma_{\alpha(i),i}, \tag{2.137}$$

where $\sigma_{0,i} = I$, $\sigma_{1,i} = X_i$, $\sigma_{2,i} = Y_i$, and $\sigma_{3,i} = Z_i$.

For example, consider a 2-local Hamiltonian for an n-qubit system:

$$H = \sum_{i<j} J_{ij}\sigma_i\sigma_j, \tag{2.138}$$

where σ_i, σ_j are Pauli operators acting on qubits i and j, respectively. Transverse Ising models and Heisenberg models are 2-local Hamiltonians. ◇

Example 2.56 (Quadratic fermionic Hamiltonians)**.** Consider the following n-mode fermionic Hamiltonian

$$H = \sum_{k=1}^{n} \lambda_k c_k^\dagger c_k = \sum_{k=1}^{n} \frac{\lambda_k}{2}(1 - Z_k), \tag{2.139}$$

where $c_k^\dagger$ and c_k are new fermionic creation and annihilation operators, and λ_k are real eigenvalues representing the energy levels of the system. The Hamiltonian H is a linear combination of Pauli Z operators and is thus a diagonal matrix.

Now, consider a general quadratic fermionic Hamiltonian of the form:

$$H = \sum_{i,j=1}^{n} A_{ij} a_i^\dagger a_j, \tag{2.140}$$

where A is a Hermitian matrix. Since A is Hermitian, we can diagonalize it using a unitary transformation U such that:

$$U^\dagger A U = \Lambda, \tag{2.141}$$

where Λ is a diagonal matrix containing the eigenvalues λ_k. Then define

$$c_k = \sum_{i=1}^{n} (U^\dagger)_{ki} a_i, \quad c_k^\dagger = \sum_{i=1}^{n} a_i^\dagger U_{ik}, \quad k = 1, \ldots, n. \tag{2.142}$$

Direct calculation shows that the new set of creation and annihilation operators $\{c_k^\dagger, c_k\}$ satisfy the canonical anticommutation relation. Substituting these transformations into the Hamiltonian,

$$H = \sum_{i,j,k} U_{ik}\lambda_k (U^\dagger)_{kj} a_i^\dagger a_j = \sum_{k=1}^{n} \lambda_k c_k^\dagger c_k, \tag{2.143}$$

we have transformed H into a diagonal Hamiltonian. ◇

Example 2.57 (1D spinless Hubbard model)**.** The Hamiltonian for the 1D spinless Hubbard model with nearest-neighbor interaction is given by:

$$H = -t \sum_{i=1}^{n-1} (a_i^\dagger a_{i+1} + a_{i+1}^\dagger a_i) + U \sum_{i=1}^{n-1} n_i n_{i+1}, \tag{2.144}$$

where t is the hopping parameter, representing the kinetic energy term, and U is the nearest-neighbor interaction strength. The operators $a_i^\dagger$ and a_i are the fermionic creation and annihilation operators at site i, respectively, and $n_i = a_i^\dagger a_i$ is the number operator at site i. When $U = 0$, the Hamiltonian is a quadratic in the fermionic operators and can be turned into a diagonalized form. When $U \neq 0$, the Hamiltonian is no longer quadratic and cannot be turned into a diagonalized Hamiltonian using the same strategy. ◇

Example 2.58 (Uniform electron gas in a plane wave basis)**.** In a plane wave basis, the Hamiltonian for a box of uniform electron gas can be expressed in second quantization as follows:

$$H = \sum_{\mathbf{k}} \epsilon_{\mathbf{k}} c_{\mathbf{k}}^\dagger c_{\mathbf{k}} + \frac{1}{2} \sum_{\mathbf{k}_1, \mathbf{k}_2, \mathbf{q}} V(\mathbf{q}) c_{\mathbf{k}_1+\mathbf{q}}^\dagger c_{\mathbf{k}_2-\mathbf{q}}^\dagger c_{\mathbf{k}_2} c_{\mathbf{k}_1}, \tag{2.145}$$

where $c_{\mathbf{k}}^\dagger$ and $c_{\mathbf{k}}$ are fermionic creation and annihilation operators for an electron with wave vector $\mathbf{k} \in \mathbb{R}^3$, $\epsilon_{\mathbf{k}} = |\mathbf{k}|^2/2$ is the kinetic energy. The interaction potential $V(\mathbf{q}) = 4\pi/\mathbf{q}^2$ in a plane wave basis is the Fourier transform of the Coulomb potential. ◇

Example 2.59 (Harmonic oscillator)**.** The Hamiltonian for a quantum harmonic oscillator in the first quantization (i.e., real space representation) is given by

$$H = \frac{p^2 + x^2}{2}, \tag{2.146}$$

where $p = -i\partial_x$ is the momentum operator and x is the position operator. Define

$$b = \frac{1}{\sqrt{2}}(x + ip), \quad b^\dagger = \frac{1}{\sqrt{2}}(x - ip), \tag{2.147}$$

then $b, b^\dagger$ satisfy the canonical commutation relation $[b, b^\dagger] = 1$. Furthermore, the Hamiltonian takes the form

$$H = b^\dagger b + \frac{1}{2}. \tag{2.148}$$

If we truncate the bosonic mode to include d levels, the state space is $\mathcal{F} = \mathbb{C}^d$, and H is a diagonal matrix of size $d \times d$. ◇

2.8.2. Quantum chemistry.

Example 2.60 (Quantum chemistry in first quantization)**.** In first quantization, the Hamiltonian for a many-electron system is given in terms of the coordinates and momenta of the electrons. The non-relativistic electronic Hamiltonian for a molecule in atomic units can be expressed as:

$$H = -\sum_{i=1}^{N} \frac{\nabla_i^2}{2} - \sum_{i=1}^{N} \sum_{A=1}^{M} \frac{Z_A}{|\mathbf{r}_i - \mathbf{R}_A|} + \sum_{i<j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{A<B} \frac{Z_A Z_B}{|\mathbf{R}_A - \mathbf{R}_B|}, \tag{2.149}$$

where N is the number of electrons, M is the number of nuclei, $\mathbf{r}_i$ and $\mathbf{R}_A$ are the positions of the i-th electron and the A-th nucleus, respectively, Z_A is the atomic number of the A-th nucleus. This is an unbounded operator. ◇

Example 2.61 (Quantum chemistry in second quantization)**.** In quantum chemistry, the electronic structure of molecules can be described using the formalism of second quantization with n molecular orbitals. The state space $\mathcal{F} = \otimes_{i=1}^n \mathbb{C}^2$ is finite dimensional. The use of second quantization allows

for a compact and efficient representation of the Hamiltonian and facilitates the expression of the Hamiltonian on quantum computers via the Jordan–Wigner transformation. The Hamiltonian of a many-electron system in second quantization is given by

$$H = \sum_{p,q=1}^{n} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{p,q,r,s=1}^{n} V_{pqrs} a_p^\dagger a_q^\dagger a_r a_s, \tag{2.150}$$

where $a_p^\dagger$ and a_q are fermionic creation and annihilation operators, respectively. The creation operator $a_p^\dagger$ adds an electron to the molecular orbital p, and the annihilation operator a_q removes an electron from the molecular orbital q. For simplicity we only consider the spatial part of the orbital and omit the spin part. The indices p, q, r, and s label the molecular orbitals, h_{pq} are the one-electron integrals, and V_{pqrs} are the two-electron integrals.

The one-electron integrals h_{pq} are given by

$$h_{pq} = \int \psi_p^*(\mathbf{r}) \left(-\frac{\nabla^2}{2} + V_{\text{ext}}(\mathbf{r}) \right) \psi_q(\mathbf{r})\, d\mathbf{r}, \tag{2.151}$$

where $\psi_p(\mathbf{r})$ is the spatial part of the molecular orbital and $V_{\text{ext}}(\mathbf{r}) = -\sum_{A=1}^{M} \frac{Z_A}{|\mathbf{r}-\mathbf{R}_A|}$ is the external potential due to the nuclei. The two-electron integrals V_{pqrs} are given by

$$V_{pqrs} = \int \int \psi_p^*(\mathbf{r}_1) \psi_q^*(\mathbf{r}_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \psi_r(\mathbf{r}_2) \psi_s(\mathbf{r}_1)\, d\mathbf{r}_1 d\mathbf{r}_2. \tag{2.152}$$

The nuclei-nuclei interaction is a constant and is dropped for simplicity. ◇

Example 2.62 (PPP Model)**.** The Pariser-Parr-Pople (PPP) model is used in quantum chemistry to describe the π-electron systems in conjugated organic molecules. The Hamiltonian for the PPP model can be written as

$$H = \sum_{p,q=1}^{n} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{p,q=1}^{n} V_{pq} n_p n_q, \tag{2.153}$$

where h_{pq} are hopping integral elements, V_{pq} are Coulomb interaction elements, $a_p^\dagger$ and a_p are the fermionic creation and annihilation operators at site p, and $n_p = a_p^\dagger a_p$ is the number operator. The Hubbard model is a special case of the PPP model with short ranged hopping and Coulomb interaction elements. Compared to the full chemistry Hamiltonian in second quantization, the two-body interaction coefficients V_{pq} have only $\mathcal{O}(n^2)$ entries but can still represent long range interactions. ◇

2.8.3. Quantum field theory.

Example 2.63 (Schwinger Model in 1D)**.** The Schwinger model describes quantum electrodynamics in $1+1$ dimensions. The state space for the Schwinger model is the tensor product of two spaces: a tensor product of $n+1$ fermionic spaces and a product of n gauge field spaces. The total Fock space is given by

$$\mathcal{F} = \left(\bigotimes_{i=1}^{n+1} \mathbb{C}^2 \right) \otimes \left(\bigotimes_{j=1}^{n} \mathbb{C}^d \right), \tag{2.154}$$

where $d = 2L+1$ is the number of levels the gauge field can take. There are two operators that we need to define that act on the gauge field space. The first is E_j^2, which is a diagonal operator that

counts the energy stored in the gauge field with index $j \in \{1, \ldots, n\}$. The second is U_j, which adds one to the value stored in the gauge field register and is analogous to a bosonic creation operator. The action of these operators is given formally below:

$$E_j^2 = \sum_{\varepsilon=-L}^{L} \varepsilon^2 \left|\varepsilon\right\rangle_j \left\langle\varepsilon\right|_j, \quad U_j = \sum_{\varepsilon=-L}^{L} \left|\varepsilon+1\right\rangle_j \left\langle\varepsilon\right|_j, \quad U_j^\dagger = \sum_{\varepsilon=-L}^{L} \left|\varepsilon-1\right\rangle_j \left\langle\varepsilon\right|_j. \tag{2.155}$$

Here we assume for U_j and its adjoint that the gauge field satisfies periodic boundary conditions at the cutoff located at $\varepsilon = \pm L$.

The Hamiltonian for the Schwinger model is given by:

$$H = \sum_{j=1}^{n} E_j^2 \otimes I_2^{\otimes(n+1)} + \nu \sum_{j=1}^{n} \left[U_j \otimes a_j^\dagger a_{j+1} - U_j^\dagger \otimes a_j a_{j+1}^\dagger\right] + \mu \sum_{j=1}^{n} (-1)^j I_d^{\otimes n} \otimes a_j^\dagger a_j, \tag{2.156}$$

where a_i and $a_i^\dagger$ are the fermionic annihilation and creation operators at site i, and I_m denotes the identity operator of dimension m. The parameters μ, ν are related to parameters such as the lattice spacing. ◇

Example 2.64 (Quadratic Majorana operators)**.** From the Jordan–Wigner transformation in Eq. (2.117), and use the fact that $XY = iZ$, we find that

$$H = -i \sum_{k=1}^{n} \lambda_k \gamma_{2k-1} \gamma_{2k} = \sum_{k=1}^{n} \lambda_k Z_k, \quad \lambda_k \in \mathbb{R} \tag{2.157}$$

is a diagonal Hamiltonian.

Consider a quadratic Hamiltonian of the form:

$$H = -i \sum_{1 \le p < q \le 2n} A_{pq} \zeta_p \zeta_q = -\frac{i}{2} \sum_{p,q=1}^{2n} A_{pq} \zeta_p \zeta_q, \tag{2.158}$$

where A is a real antisymmetric matrix, and $\{\zeta_p\}_{p=1}^{2n}$ is a set of Majorana operators. There exists an orthogonal matrix O such that:

$$O^\top A O = \bigoplus_{k=1}^{n} \begin{pmatrix} 0 & \lambda_k \\ -\lambda_k & 0 \end{pmatrix} =: \Lambda \tag{2.159}$$

where λ_k are the singular values of A. Now define a set of transformed Majorana operators

$$\gamma_j = \sum_p \zeta_p O_{pj} = \sum_p (O^\top)_{jp} \zeta_p, \quad j = 1, \ldots, 2n, \tag{2.160}$$

then we still have

$$\{\gamma_j, \gamma_{j'}\} = 2\delta_{j,j'}. \tag{2.161}$$

The transformed Hamiltonian takes a diagonal form

$$H = -\frac{i}{2} \sum_{1 \le j, j' \le 2n} \gamma_j (\Lambda)_{jj'} \gamma_{j'} = -i \sum_{k=1}^{n} \lambda_k \gamma_{2k-1} \gamma_{2k}. \tag{2.162}$$

The quadratic fermionic Hamiltonian in Example 2.56 is a special case of this example. ◇

Example 2.65 (SYK Model)**.** The Sachdev-Ye-Kitaev (SYK) model is a quantum mechanical model of n Majorana fermions with random all-to-all interactions. The Hamiltonian for the SYK model is given by

$$H = \sum_{1 \le i < j < k < l \le 2n} J_{ijkl} \gamma_i \gamma_j \gamma_k \gamma_l, \tag{2.163}$$

where γ_i are the Majorana fermion operators, and J_{ijkl} are random coupling constants, typically drawn from a Gaussian distribution. The SYK model is of particular interest due to its connections to quantum chaos, holography, and black hole physics. ◇

2.8.4. Optimization.

Example 2.66 (k-SAT problem)**.** Classical optimization problems, such as the k-SAT problem, can be represented using a Hamiltonian. The k-SAT problem is a type of Boolean satisfiability problem where each clause contains exactly k literals. The goal is to find an assignment to the Boolean variables that satisfies all the clauses. The most famous examples are 2-SAT (classically easy), and 3-SAT (NP-complete).

Consider a k-SAT problem with n Boolean variables $x_1, x_2, \ldots, x_n$ and m clauses $C_1, C_2, \ldots, C_m$. Each clause C_i is a disjunction of exactly k literals.

We can construct a Hamiltonian H such that its ground state corresponds to the solution of the k-SAT problem. The Hamiltonian for the k-SAT problem can be written as:

$$H = \sum_{i=1}^{m} H_{C_i}, \tag{2.164}$$

where H_{C_i} is the Hamiltonian for the i-th clause. Each clause Hamiltonian H_{C_i} is designed to be zero if the clause is satisfied and positive otherwise. For clauses involving single literals, such as $C_k = (x_p)$ or $C_l = (\bar{x}_q)$, the Hamiltonians H_{C_k} and H_{C_l} are:

$$H_{C_k} = \frac{1}{2}\left(1 + Z_p\right), \quad H_{C_l} = \frac{1}{2}\left(1 - Z_q\right). \tag{2.165}$$

For a clause $C_i = (x_p \vee \bar{x}_q)$, the corresponding Hamiltonian H_{C_i} can be written using the product

$$H_{C_i} = \frac{1}{4}\left(1 + Z_p\right)\left(1 - Z_q\right). \tag{2.166}$$

For a general clause $C_i = (l_1 \vee l_2 \vee \cdots \vee l_k)$, where l_j represents either x_{p_j} or $\bar{x}_{p_j}$, the corresponding Hamiltonian H_{C_i} can be written using the Pauli-Z operator Z:

$$H_{C_i} = \prod_{j=1}^{k} \frac{1 + z_j Z_{p_j}}{2}, \tag{2.167}$$

where $z_j = +1$ if $l_j = x_{p_j}$ and $z_j = -1$ if $l_j = \bar{x}_{p_j}$. The Hamiltonian H is diagonal and positive semidefinite. If the smallest eigenvalue (called the ground state energy) of H is 0, then the associated eigenvector (called the ground state, which may not be unique) corresponds to the Boolean variable assignment that satisfies all the clauses of the k-SAT problem. ◇

Example 2.67 (MAX-CUT problem)**.** The MAX-CUT problem is a well-known combinatorial optimization problem. Given a graph $G = (V, E)$ with a set of vertices V and a set of edges E, the

goal is to partition the vertices into two subsets such that the number of edges between the subsets is maximized. Assume the graph has n vertices, and the Hamiltonian for the MAX-CUT problem can be written as:

$$H = -\sum_{(i,j)\in E} \frac{1}{2}\left(1 - Z_i Z_j\right). \tag{2.168}$$

Each term $-\frac{1}{2}(1 - Z_i Z_j)$ equals -1 if vertices i and j are in different subsets and 0 if they are in the same subset. Therefore, minimizing H is equivalent to maximizing the number of edges that are cut by the partition. ◇

Part II

Foundation

CHAPTER 3

Probability, quantum channel, and distances

We begin by reviewing basic concepts in classical probability theory, which provides intuition for how errors propagate in randomized processes. We then introduce quantum channels as the general framework for quantum dynamics. Unlike ideal quantum circuits which are unitary, real-world quantum processes often involve noise and decoherence. Quantum channels allow us to model these effects, as well as measurements and interactions with the environment. We explain the requirements (specifically, the concept of complete positivity) for a map to be a valid quantum channel and describe standard representations such as the Kraus and Stinespring forms.

With this framework in place, we introduce distance measures for quantum states. For pure states, we use norms that account for the global phase. For mixed states, we introduce the trace distance and fidelity. These two measures are complementary: trace distance relates to the distinguishability of states via measurement, while fidelity captures their overlap and behaves well under quantum operations.

Finally, we discuss how to compare quantum channels. This requires norms that are stable even when the channels act on part of an entangled system. This leads us to the diamond norm, which is the standard metric for quantifying the error of quantum operations.

3.1. Basic notions in probability theory

Probability theory is a subject that carries nearly as many profound surprises as quantum theory itself. In this section, we introduce some basic concepts in probability theory, focusing on finite-dimensional spaces. In quantum computing, the probability distributions associated with an n-qubit system reside in 2^n-dimensional spaces.

Definition 3.1. *Let Σ be a finite set called a state space, or sample space, where each element of Σ is called an event. A **probability distribution** is a function $\mathbb{P} : \Sigma \to [0, 1]$, which can be represented as a vector in a Euclidean space, and satisfies $\sum_{s\in\Sigma} \mathbb{P}(s) = 1$.*

Let Σ_A and Σ_B be sample spaces and let $\mathbb{P}_A$ and $\mathbb{P}_B$ be probability distributions on the two sample spaces. These distributions are said to be independent if the joint distribution, $\mathbb{P}_{AB}$ on the set $\Sigma_A \times \Sigma_B$ obeys $\mathbb{P}_{AB} = \mathbb{P}_A \otimes \mathbb{P}_B$. The expectation value (or average value) of a function mapping $f : \Sigma \mapsto \mathbb{C}$ is defined to be $\mathbb{E}(f) := \sum_{s\in\Sigma} f(s)\mathbb{P}(s) = \langle f, \mathbb{P}\rangle$.

Example 3.2. As an example, let us consider rolling a four-sided die. Here the random variable is the outcome of the experiment; the sample space is $\{1, 2, 3, 4\}$ and the probability distribution (for a fair die) is $1/4$ for each of these outcomes. The random variable, x, in this case corresponds to the result of the die.

In the event that we wanted to find the probability that the sample is a prime number, we could redefine the sample space and the underlying distribution but it is easier to use the indicator-function property of the distribution to see that

$$\mathbb{P}(x \in \{2,3\}) = \mathbb{E}(\mathbf{1}_{\{2,3\}}) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}. \tag{3.1}$$

In general, this approach is often the easiest way to compute a probability because it constructs an indicator function which projects onto the fraction of the sample space that we want to measure. Note this is also true in quantum theory wherein the probability of measuring a mixed state, ρ, to be a pure state $|\psi\rangle$ is

$$\mathbb{P}(|\psi\rangle) = \mathrm{Tr}(|\psi\rangle\langle\psi|\rho) = \langle\psi|\,\rho\,|\psi\rangle\,. \tag{3.2}$$

Here the projector $|\psi\rangle\langle\psi|$ plays the same role as the indicator function used above, and further illustrates the close ties between probability theory and quantum theory. ⋄

Similar to the amplitude of the wave function in quantum theory, there is not a single unifying interpretation of probability. For this reason we recommend that the reader be well versed in both interpretations as each can convey useful intuitions.

The following bound, known as the union bound, is very useful for estimating probabilities of events. We provide it as well as its proof as an elementary example of probability theory.

THEOREM 3.3 (Union Bound). *Let Σ be a sample space and let $A, B \subseteq \Sigma$ and let $\mathbb{P}$ be a probability distribution on Σ. We then have*

$$\mathbb{P}(A \cup B) = \mathbb{E}(\mathbf{1}_A + \mathbf{1}_B - \mathbf{1}_A\mathbf{1}_B) \leq \mathbb{P}(A) + \mathbb{P}(B). \tag{3.3}$$

PROOF. Intuitively, by looking at a Venn diagram for events A and B it is clear that the region $A \cup B$ contains region A and region B but also may include region $A \cap B$. Thus the upper bound given above overcounts the probability in the intersection and therefore it is an upper bound. Formally, we use linearity of expectation:

$$\mathbb{E}(\mathbf{1}_A + \mathbf{1}_B - \mathbf{1}_A\mathbf{1}_B) = \mathbb{E}(\mathbf{1}_A) + \mathbb{E}(\mathbf{1}_B) - \mathbb{E}(\mathbf{1}_A\mathbf{1}_B). \tag{3.4}$$

Next, $\mathbb{E}(\mathbf{1}_A\mathbf{1}_B) = \sum_{s\in\Sigma} \mathbb{P}(s)(\mathbf{1}_A(s)\mathbf{1}_B(s)) \geq 0$, and $\mathbb{E}(\mathbf{1}_A) = \mathbb{P}(A)$, $\mathbb{E}(\mathbf{1}_B) = \mathbb{P}(B)$. Combining these gives the claim. □

Example 3.4 (Failure Propagation Bound)**.** Consider the following problem: you have a quantum algorithm that succeeds with probability $1 - \delta$ and fails with probability δ. Suppose we run the algorithm independently N times; determine a value of δ that guarantees the probability of at least one failure is at most $1/3$. This problem appears ubiquitously in quantum computing in problems such as phase estimation or quantum error correction where the probability of failure needs to be considered and extra computational resources are needed to suppress them.

The N events each have a probability of δ assigned to them and so we expect that the total probability of at least one error happening will be from the union bound $N\delta$. We can validate this inductively. For the base case we see trivially that the claim holds for $N = 1$. For the induction step, let us assume that the probability of at least one error occuring in the first $N-1$ steps is at most $(N-1)\delta$. From the union bound the probability of failing in the next sample is δ and thus the total failure probability is at most $(N-1)\delta + \delta = N\delta$. Thus if we want to see a failure probability of $1/3$ it suffices to take

$$\delta \leq \frac{1}{3N}. \tag{3.5}$$

This example shows that worst case scenario that the failure probability for our algorithm grows linearly. This actually might seem strange to the reader since the error probability compounds exponentially in practice; however, linear growth of error is actually in this context worse than exponential because for large enough N the union bound will be greater than 1 whereas the exponential upper bound is always less than 1. In this context, surprisingly, linear growth is worse than exponential but nonetheless the simplicity and generality of union bounds often provide good enough bounds that are easy to manipulate. ◇

The natural operations on probability distributions are stochastic transformations, which can be represented as transition matrices. We define these transformations below.

Definition 3.5. *Let Σ be a sample space of size N and let $p \in \mathbb{R}^N$ be the column vector representation of a probability distribution. A valid transformation on the state space of the register X to itself has a matrix representation $P : \mathbb{R}^N \to \mathbb{R}^N$, which maps p to Pp. The matrix P is called a* ***transition matrix*** *and satisfies*

(1) $P_{ij} \geq 0, \quad \forall i, j \in [N]$,
(2) $\sum_{i \in [N]} P_{ij} = 1, \quad \forall j \in [N]$.

Remark 3.6. In classical probability theory, the probability distribution is often written as a row vector. Then the transition matrix is applied from the right as pP, and the transition matrix needs to be **right stochastic** or **row stochastic**, i.e., $\sum_{j \in [N]} P_{ij} = 1$ for all $i \in [N]$. Given a probability distribution $p \in \mathbb{R}^N$, a natural quantum state encoding the distribution p (also called a coherent version of p) is

$$|\sqrt{p}\rangle = \sum_i \sqrt{p_i}\, |i\rangle \,. \tag{3.6}$$

This is a normalized state. It is thus more natural to view p as a column vector so that the usual rule of applying an operator to a state vector applies. A matrix satisfying the properties in Definition 3.5 is also called **left stochastic** or **column stochastic**. Any j-th column of P, denoted by $P_{:,j}$, is a probability distribution. If P is both left and right stochastic, then it is called a **doubly stochastic** matrix. ◇

Example 3.7. Let us consider how we would represent an AND gate in this language. The AND gate has the property that for any $x, y \in \{0, 1\}$, $\mathrm{AND}(x, y) = xy$. This operation is an example of an irreversible operation, meaning that it cannot be inverted from the outputs to find the inputs. In this case the natural vector space for probability distributions for two bits can be represented as a probability vector in $\mathbb{R}^2 \otimes \mathbb{R}^2$. As we are using square matrices to represent these transformations we will take $\mathrm{AND}(e_x \otimes e_y) = e_0 \otimes e_{xy}$ for computational basis vectors e_0, e_1 and $x, y \in \{0, 1\}$. Specifically then we have that the gate can be represented as a stochastic matrix P_{AND}

$$P_{\mathrm{AND}} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.7}$$

We see that the matrix representation is stochastic, but not doubly stochastic.

If we consider taking two distributions for our bits $p_x = [a, 1-a]^\top$ and $p_y = [b, 1-b]^\top$ for $a, b \in [0, 1]$ then we can see that the distribution that we get from applying the AND operation to

the distribution on the bits is

$$P_{\mathrm{AND}}(p_x \otimes p_y) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} ab \\ a(1-b) \\ b(1-a) \\ (1-a)(1-b) \end{pmatrix} = \begin{bmatrix} (a+b)-ab \\ 1-(a+b)+ab \\ 0 \\ 0 \end{bmatrix}. \tag{3.8}$$

This output distribution makes intuitive sense. The AND output is 1 only if both inputs are 1, which occurs with probability $(1-a)(1-b)$, corresponding to the second entry above. Equivalently, the probability that the AND output is 0 is the probability that at least one input is 0, namely $a+b-ab$, corresponding to the first entry. ◇

3.2. Quantum channels

The concept of a quantum channel generalizes both the unitary evolution of isolated quantum systems, as governed by the Schrödinger equation, and the stochastic evolution of classical probability distributions. It provides a unified framework for describing the most general physically permissible evolution of quantum states, encompassing coherent dynamics (e.g., unitary transformations) and incoherent processes such as measurement, decoherence, and interactions with an environment.

We begin by defining the mathematical objects under consideration. A **superoperator** is a linear map $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$. We denote the action of $\mathcal{Q}$ on an operator $A \in L(\mathbb{C}^N)$ by $\mathcal{Q}[A]$ or $\mathcal{Q}(A)$.

Given two superoperators $\mathcal{Q}_1 : L(\mathbb{C}^{N_1}) \to L(\mathbb{C}^{M_1})$ and $\mathcal{Q}_2 : L(\mathbb{C}^{N_2}) \to L(\mathbb{C}^{M_2})$, their **tensor product** $\mathcal{Q}_1 \otimes \mathcal{Q}_2$ is the unique linear map $L(\mathbb{C}^{N_1} \otimes \mathbb{C}^{N_2}) \to L(\mathbb{C}^{M_1} \otimes \mathbb{C}^{M_2})$ satisfying

$$(\mathcal{Q}_1 \otimes \mathcal{Q}_2)[A_1 \otimes A_2] = \mathcal{Q}_1[A_1] \otimes \mathcal{Q}_2[A_2] \tag{3.9}$$

for all $A_1 \in L(\mathbb{C}^{N_1})$ and $A_2 \in L(\mathbb{C}^{N_2})$. This definition extends to all operators by linearity.

Just as a unitary transformation maps a state vector to another state vector while preserving its norm, a **quantum channel** is a superoperator intended to map a density operator to another density operator. A fundamental example is the **identity channel** $\mathcal{I} : L(\mathbb{C}^N) \to L(\mathbb{C}^N)$, defined by $\mathcal{I}[A] = A$ for any $A \in L(\mathbb{C}^N)$.

Example 3.8. The action of the tensor product of superoperators is particularly important when analyzing local operations on composite systems. Let $\mathcal{I}_K : L(\mathbb{C}^K) \to L(\mathbb{C}^K)$ be the identity map and $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ be a linear map. Consider an operator $A \in L(\mathbb{C}^K \otimes \mathbb{C}^N)$. We can represent A in block form with respect to an orthonormal basis $\{|i\rangle\}$ of $\mathbb{C}^K$:

$$A = \sum_{i,j \in [K]} |i\rangle\langle j| \otimes A_{ij}, \quad A_{ij} \in L(\mathbb{C}^N). \tag{3.10}$$

The action of $\mathcal{I}_K \otimes \mathcal{Q}$ is given by applying $\mathcal{Q}$ to each block:

$$(\mathcal{I}_K \otimes \mathcal{Q})[A] = \sum_{i,j \in [K]} |i\rangle\langle j| \otimes \mathcal{Q}[A_{ij}]. \tag{3.11}$$

For instance, if $K = 2$, the matrix representation is

$$(\mathcal{I}_2 \otimes \mathcal{Q}) \left[\begin{pmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{pmatrix} \right] = \begin{pmatrix} \mathcal{Q}[A_{00}] & \mathcal{Q}[A_{01}] \\ \mathcal{Q}[A_{10}] & \mathcal{Q}[A_{11}] \end{pmatrix} \in L(\mathbb{C}^2 \otimes \mathbb{C}^M). \tag{3.12}$$

◇

To ensure that a superoperator maps density operators (which are positive semidefinite and have unit trace) to density operators, it must satisfy certain constraints.

Definition 3.9. *A linear map $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ is called:* ***positive*** *if $\mathcal{Q}[A]$ is positive semidefinite for every positive semidefinite $A \in L(\mathbb{C}^N)$. $\mathcal{Q}$ is called* ***trace preserving*** *(TP) if $\mathrm{Tr}(\mathcal{Q}[A]) = \mathrm{Tr}(A)$ for every $A \in L(\mathbb{C}^N)$.*

While it might seem sufficient to define a quantum channel simply as a positive, trace-preserving map, the structure of quantum mechanics demands a stronger condition. Quantum systems often exist as subsystems of larger, composite systems. If $\mathcal{Q}$ describes the evolution of a system S, and S is potentially entangled with an ancillary system A, the evolution of the joint system is described by $\mathcal{I}_A \otimes \mathcal{Q}$. For this joint evolution to be physically valid, $\mathcal{I}_A \otimes \mathcal{Q}$ must also map density operators to density operators, meaning it must be a positive map, regardless of the dimension of the ancilla A. This requirement leads to the concept of complete positivity.

Definition 3.10. *A linear map $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ is* ***completely positive*** *(CP) if for all integers $K \geq 1$, the map $\mathcal{I}_K \otimes \mathcal{Q} : L(\mathbb{C}^K \otimes \mathbb{C}^N) \to L(\mathbb{C}^K \otimes \mathbb{C}^M)$ is positive.*

It is worth noting that the ordering of the tensor product in the definition is immaterial. One could equivalently require that $\mathcal{Q} \otimes \mathcal{I}_K$ be positive for all K. Physically, this reflects the fact that the labeling of the ancillary system is arbitrary. Mathematically, the maps $\mathcal{I} \otimes \mathcal{Q}$ and $\mathcal{Q} \otimes \mathcal{I}$ are related via the SWAP operator (the isomorphism that exchanges the tensor factors). Specifically, they are unitarily equivalent:

$$\mathcal{Q} \otimes \mathcal{I} = \mathcal{U}_{\mathrm{SWAP}} \circ (\mathcal{I} \otimes \mathcal{Q}) \circ \mathcal{U}_{\mathrm{SWAP}}^{-1}, \tag{3.13}$$

where the superoperator $\mathcal{U}_{\mathrm{SWAP}}$ acts as $\mathcal{U}_{\mathrm{SWAP}}[X] = \mathrm{SWAP} \cdot X \cdot \mathrm{SWAP}^\dagger$. Since $X \succeq 0$ if and only if $UXU^\dagger \succeq 0$ for any unitary U, it follows that $\mathcal{I} \otimes \mathcal{Q}$ is positive if and only if $\mathcal{Q} \otimes \mathcal{I}$ is positive.

While positivity ensures that the channel acts correctly on the system itself, complete positivity is strictly stronger, ensuring correct action even when the system is entangled with an ancilla.

Example 3.11 (Positive map that is not completely positive)**.** Consider the transpose map $\mathcal{T} : L(\mathbb{C}^2) \to L(\mathbb{C}^2)$, defined by $\mathcal{T}[A] = A^\top$ with respect to the computational basis. If A is positive, its eigenvalues are non-negative. Since A and $A^\top$ share the same spectrum, $A^\top$ is also positive. Thus, $\mathcal{T}$ is a positive map.

However, $\mathcal{T}$ is not completely positive. To illustrate this, consider a two-qubit system in the maximally entangled Bell state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. The corresponding density operator is:

$$\rho = |\psi\rangle\langle\psi| = \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|). \tag{3.14}$$

We apply the map $\mathcal{I} \otimes \mathcal{T}$ (the partial transpose with respect to the second subsystem) to this state:

$$(\mathcal{I} \otimes \mathcal{T})[\rho] = \frac{1}{2}(|00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 11|). \tag{3.15}$$

In the standard basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, the matrix representation is

$$\frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.16}$$

This matrix has eigenvalues $\{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, -\frac{1}{2}\}$. Since one eigenvalue is negative, the resulting operator is not positive. Thus, $\mathcal{T}$ is not completely positive. ◇

We now arrive at the formal definition of a quantum channel.

Definition 3.12 (Quantum channel, or CPTP map)**.** *A **quantum channel** $\mathcal{Q}$ is a linear map $L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ that is completely positive (CP) and trace preserving (TP).*

If $\mathcal{Q}$ is a quantum channel, it maps any density operator $\rho \in \mathcal{D}(\mathbb{C}^N)$ to a density operator $\mathcal{Q}[\rho] \in \mathcal{D}(\mathbb{C}^M)$. The complete positivity condition ensures that if $\mathcal{Q}$ acts locally on a subsystem of a larger entangled state $\widetilde{\rho} \in \mathcal{D}(\mathbb{C}^K \otimes \mathbb{C}^N)$, the resulting state $(\mathcal{I}_K \otimes \mathcal{Q})[\widetilde{\rho}]$ remains a valid density operator in $\mathcal{D}(\mathbb{C}^K \otimes \mathbb{C}^M)$. This property is fundamental to the consistency of quantum mechanics.

Example 3.13. A fundamental class of quantum channels is the **unitary channel**. This requires the input and output dimensions to be equal, $N = M$. Given a unitary matrix $U \in \mathrm{U}(N)$, the corresponding channel $\mathcal{U} : L(\mathbb{C}^N) \to L(\mathbb{C}^N)$ acts by conjugation:

$$\mathcal{U}[\rho] = U\rho U^\dagger. \tag{3.17}$$

This map is trace-preserving, as $\mathrm{Tr}[U\rho U^\dagger] = \mathrm{Tr}[\rho U^\dagger U] = \mathrm{Tr}[\rho]$. It is also completely positive, as we will see shortly. The identity channel $\mathcal{I}$ is a unitary channel with $U = I$. ◇

A powerful way to characterize and construct quantum channels is through the Kraus representation.

Proposition 3.14. *Let $\{K_j\}_{j\in[R]}$ be a set of matrices in $\mathbb{C}^{M\times N}$ satisfying the completeness relation*

$$\sum_{j\in[R]} K_j^\dagger K_j = I_N. \tag{3.18}$$

Then the linear map $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ defined by

$$\mathcal{Q}[\rho] = \sum_{j\in[R]} K_j \rho K_j^\dagger \tag{3.19}$$

is a quantum channel (CPTP).

Proof. We first verify complete positivity. Let L be an arbitrary integer and consider any positive operator $X \in L(\mathbb{C}^L \otimes \mathbb{C}^N)$. The action of the extended map is

$$(\mathcal{I}_L \otimes \mathcal{Q})[X] = \sum_{j\in[R]} (I_L \otimes K_j) X (I_L \otimes K_j)^\dagger. \tag{3.20}$$

For any operator A, if X is positive, then $AXA^\dagger$ is also positive. Thus, each term in the summation is a positive operator. Since the sum of positive operators is positive, $\mathcal{I}_L \otimes \mathcal{Q}$ is a positive map for all L. Thus, $\mathcal{Q}$ is completely positive.

Next, we verify the trace-preserving property. For any $\rho \in L(\mathbb{C}^N)$, using the linearity and the cyclic property of the trace, we have

$$\mathrm{Tr}[\mathcal{Q}[\rho]] = \sum_{j\in[R]} \mathrm{Tr}[K_j \rho K_j^\dagger] = \mathrm{Tr}\left[\rho \left(\sum_{j\in[R]} K_j^\dagger K_j\right)\right]. \tag{3.21}$$

Substituting the completeness relation $\sum_{j\in[R]} K_j^\dagger K_j = I_N$, we obtain $\mathrm{Tr}[\rho I_N] = \mathrm{Tr}[\rho]$. Therefore, $\mathcal{Q}$ is trace-preserving. □

The representation in Eq. (3.19) is called the **Kraus form** or the **operator sum representation** of the channel. The operators $\{K_j\}$ are known as Kraus operators. For example, the unitary channel in Example 3.13 is in Kraus form with a single Kraus operator $K_0 = U$.

We can now explore the connection between classical stochastic evolution and quantum channels. This correspondence highlights that quantum mechanics is a generalization of classical probability theory.

For any probability distribution $p \in \mathbb{R}^N$, we can embed it into a quantum state

$$\rho = \sum_{i\in[N]} p_i |i\rangle\langle i|. \tag{3.22}$$

This diagonal density matrix is called a **classical state** or **probabilistic state**.

Given a (column) stochastic matrix $P \in \mathbb{R}^{M\times N}$ (i.e., $P_{ij} \geq 0$ and $\sum_{i\in[M]} P_{ij} = 1$ for all $j \in [N]$), which defines a classical Markov process mapping distribution p to $p' = Pp$, we can construct a corresponding **classical channel** $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ defined by

$$\mathcal{Q}[\rho] = \sum_{i\in[M],j\in[N]} P_{ij} |i\rangle\langle j|\rho|j\rangle\langle i|. \tag{3.23}$$

If ρ is a classical state, $\mathcal{Q}[\rho]$ is also a classical state corresponding to the evolved probability distribution p'.

Exercise 3.1. Prove that the classical channel $\mathcal{Q}$ defined in Eq. (3.23) is indeed a quantum channel (CPTP).

The fact that classical channels are a subset of quantum channels suggests that any advantage offered by quantum computation must stem from the utilization of the off-diagonal entries of the density matrix (coherence) and the structure of non-classical channels.

We now present several examples of important quantum channels, typically modeling different types of noise processes in qubits ($N = M = 2$).

Example 3.15 (Bit flip and phase flip channels)**.** The bit flip channel $\mathcal{Q}_{\text{bf}}$ describes a process where the qubit state is flipped (i.e., X gate applied) with probability $1-p$, and remains unchanged with probability p:

$$\mathcal{Q}_{\text{bf}}[\rho] = p\rho + (1-p)X\rho X, \quad 0 \leq p \leq 1. \tag{3.24}$$

This is in Kraus form with $K_0 = \sqrt{p}I$ and $K_1 = \sqrt{1-p}X$.

Similarly, the phase flip channel $\mathcal{Q}_{\text{pf}}$ flips the relative phase (i.e., Z gate applied) with probability $1-p$:

$$\mathcal{Q}_{\text{pf}}[\rho] = p\rho + (1-p)Z\rho Z, \quad 0 \leq p \leq 1. \tag{3.25}$$

This channel is also known as the **dephasing channel**, as it suppresses coherences while leaving populations unchanged. ◇

Example 3.16 (Depolarizing channel)**.** The depolarizing channel $\mathcal{Q}_{\text{dp}} : L(\mathbb{C}^N) \to L(\mathbb{C}^N)$ models a process where the state remains intact with probability p, and is replaced by the maximally mixed state I/N with probability $1-p$:

$$\mathcal{Q}_{\text{dp}}[\rho] = p\rho + \frac{1-p}{N} I, \quad 0 \leq p \leq 1. \tag{3.26}$$

◇

Example 3.17 (Amplitude damping channel)**.** The amplitude damping channel $\mathcal{Q}_{\text{ad}} : L(\mathbb{C}^2) \to L(\mathbb{C}^2)$ models energy dissipation, such as spontaneous emission, where an excited state $|1\rangle$ decays to the ground state $|0\rangle$ with probability γ. It is described by the Kraus operators

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad K_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}, \quad 0 \leq \gamma \leq 1. \tag{3.27}$$

◇

Perhaps surprisingly, the converse of Proposition 3.14 is also true: every quantum channel can be written in the Kraus form. This fundamental result demonstrates that the abstract definition of a CPTP map is equivalent to the constructive definition provided by the operator sum representation.

THEOREM 3.18 (Choi–Kraus Representation). *A linear map $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ is a quantum channel if and only if there exists a set of matrices $\{K_j\}_{j\in[R]}$ in $\mathbb{C}^{M\times N}$, with $R \leq NM$, satisfying the completeness relation $\sum_{j\in[R]} K_j^\dagger K_j = I_N$, such that $\mathcal{Q}$ takes the form*

$$\mathcal{Q}(\rho) = \sum_{j\in[R]} K_j \rho K_j^\dagger. \tag{3.28}$$

PROOF. The "if" part is established by Proposition 3.14. We now prove the "only if" part using a technique known as the **Choi-Jamiołkowski isomorphism**.

Let $\mathcal{Q}$ be a quantum channel. Define an unnormalized maximally entangled state on $\mathbb{C}^N \otimes \mathbb{C}^N$:

$$|\gamma\rangle = \sum_{i\in[N]} |i\rangle \otimes |i\rangle . \tag{3.29}$$

Let $\mathcal{I}_N$ denote the identity map on the first N-dimensional register (the ancilla). By the complete positivity of $\mathcal{Q}$, the map $\mathcal{I}_N \otimes \mathcal{Q}$ is positive. Therefore, the **Choi matrix** defined as

$$\sigma = (\mathcal{I}_N \otimes \mathcal{Q})[|\gamma\rangle\langle\gamma|] \in L(\mathbb{C}^N \otimes \mathbb{C}^M) \tag{3.30}$$

is a positive operator.

The Choi matrix completely characterizes the channel $\mathcal{Q}$. To see this, we use a key property of the maximally entangled state. For any vector $|\psi\rangle = \sum_i \psi_i |i\rangle \in \mathbb{C}^N$, let $|\widetilde{\psi}\rangle = \sum_i \overline{\psi_i} |i\rangle$ be its element-wise conjugate in the computational basis. We can verify the identity:

$$(\langle\widetilde{\psi}| \otimes I_N) |\gamma\rangle = \sum_{i,j} \psi_j (\langle j|i\rangle \otimes |i\rangle) = \sum_i \psi_i |i\rangle = |\psi\rangle . \tag{3.31}$$

We can recover the action of the channel on $|\psi\rangle\langle\psi|$ by taking the partial inner product of σ with $|\widetilde{\psi}\rangle$ on the first register. By the definition of the tensor product map and the identity above, we have:

$$\begin{aligned} (\langle\widetilde{\psi}| \otimes I_M)\sigma(|\widetilde{\psi}\rangle \otimes I_M) &= (\langle\widetilde{\psi}| \otimes I_M)(\mathcal{I}_N \otimes \mathcal{Q})[|\gamma\rangle\langle\gamma|](|\widetilde{\psi}\rangle \otimes I_M) \\ &= \mathcal{Q}\left[(\langle\widetilde{\psi}| \otimes I_N)|\gamma\rangle\langle\gamma|(|\widetilde{\psi}\rangle \otimes I_N)\right] \\ &= \mathcal{Q}(|\psi\rangle\langle\psi|). \end{aligned} \tag{3.32}$$

Since σ is positive, we can perform its eigendecomposition. Let $R = \text{rank}(\sigma) \leq NM$. We write

$$\sigma = \sum_{j\in[R]} |s_j\rangle\langle s_j|, \tag{3.33}$$

where $|s_j\rangle \in \mathbb{C}^N \otimes \mathbb{C}^M$ are (potentially unnormalized) eigenvectors scaled by the square root of the eigenvalues.

For each $j \in [R]$, we define a linear operator $K_j : \mathbb{C}^N \to \mathbb{C}^M$ via the relation (sometimes called vectorization or flattening):

$$K_j |\psi\rangle := (\langle\widetilde{\psi}| \otimes I_M) |s_j\rangle . \tag{3.34}$$

Substituting the decomposition of σ back into the recovery formula:

$$\begin{aligned} \mathcal{Q}(|\psi\rangle\langle\psi|) &= (\langle\widetilde{\psi}| \otimes I_M) \left(\sum_{j\in[R]} |s_j\rangle\langle s_j| \right) (|\widetilde{\psi}\rangle \otimes I_M) \\ &= \sum_{j\in[R]} \left[(\langle\widetilde{\psi}| \otimes I_M) |s_j\rangle \right] \left[\langle s_j| (|\widetilde{\psi}\rangle \otimes I_M) \right] \\ &= \sum_{j\in[R]} (K_j |\psi\rangle)(K_j |\psi\rangle)^\dagger = \sum_{j\in[R]} K_j |\psi\rangle\langle\psi| K_j^\dagger. \end{aligned} \tag{3.35}$$

Since this holds for arbitrary $|\psi\rangle$, by linearity it holds for all operators $\rho \in L(\mathbb{C}^N)$.

Finally, we must verify the completeness relation. The trace-preserving property $\mathrm{Tr}[\mathcal{Q}(\rho)] = \mathrm{Tr}[\rho]$ implies

$$\mathrm{Tr}\left[\sum_{j\in[R]} K_j \rho K_j^\dagger \right] = \mathrm{Tr}\left[\left(\sum_{j\in[R]} K_j^\dagger K_j \right) \rho \right] = \mathrm{Tr}[I_N \rho]. \tag{3.36}$$

Since this equality holds for all ρ, we must have $\sum_{j\in[R]} K_j^\dagger K_j = I_N$. □

The definition of complete positivity in Definition 3.10 requires verifying positivity for all dimensions K, which is operationally cumbersome. However, the proof of the Choi–Kraus theorem reveals that a much simpler criterion suffices. Let $\mathcal{I}_N$ denote the identity channel on $L(\mathbb{C}^N)$. If we assume only that the map $\mathcal{I}_N \otimes \mathcal{Q}$ is positive, then the Choi matrix σ (defined in the proof of Theorem 3.18) must be positive, as it is the image of the positive operator $|\gamma\rangle\langle\gamma|$ under this map. As shown in the proof, the positivity of σ guarantees the existence of a Kraus representation for $\mathcal{Q}$. Finally, by Proposition 3.14, any map with a Kraus representation is completely positive (i.e., $\mathcal{I}_K \otimes \mathcal{Q}$ is positive for all K). This establishes the equivalence between the original definition and a condition involving only an ancilla of the input dimension:

Proposition 3.19 (Choi's Theorem)**.** *A linear map $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ is completely positive if and only if its Choi matrix σ is positive semidefinite. Equivalently, $\mathcal{Q}$ is CP if and only if the map $\mathcal{I}_N \otimes \mathcal{Q}$ is positive.*

The Kraus representation provides deep insight into the structure of quantum channels. Another fundamental structural result is the Stinespring dilation theorem, which connects general quantum channels (which may involve decoherence or dissipation) to coherent evolution on a larger Hilbert space.

THEOREM 3.20 (Stinespring dilation). *Given any quantum channel $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$, there exists an ancilla system A of dimension $R \leq NM$, and an isometry $V : \mathbb{C}^N \to \mathbb{C}^M \otimes \mathbb{C}^R$ (i.e., $V^\dagger V = I_N$) such that*

$$\mathcal{Q}(\rho) = \mathrm{Tr}_A \left[V \rho V^\dagger \right] . \tag{3.37}$$

Furthermore, this isometry can always be realized by a unitary evolution U on a sufficiently large joint system initialized with the ancilla in a fixed state $|0\rangle$:

$$\mathcal{Q}(\rho) = \mathrm{Tr}_A \left[U(\rho \otimes |0\rangle\langle 0|)U^\dagger \right]. \tag{3.38}$$

Proof. By the Choi–Kraus theorem (Theorem 3.18), $\mathcal{Q}$ has a Kraus representation $\mathcal{Q}(\rho) = \sum_{j\in[R]} K_j \rho K_j^\dagger$, where $R \le NM$.

We construct the isometry $V : \mathbb{C}^N \to \mathbb{C}^M \otimes \mathbb{C}^R$. Let $\{|j\rangle\}$ be an orthonormal basis for the ancilla space $\mathbb{C}^R$. Define V by

$$V|\psi\rangle = \sum_{j\in[R]} (K_j|\psi\rangle) \otimes |j\rangle. \tag{3.39}$$

We verify that V is an isometry. For any $|\psi\rangle \in \mathbb{C}^N$:

$$\begin{aligned} \langle\psi|V^\dagger V|\psi\rangle &= \|V|\psi\rangle\|^2 = \sum_{j\in[R]} \|K_j|\psi\rangle\|^2 \\ &= \sum_{j\in[R]} \langle\psi|K_j^\dagger K_j|\psi\rangle = \langle\psi| \left(\sum_j K_j^\dagger K_j \right) |\psi\rangle. \end{aligned} \tag{3.40}$$

By the completeness relation, this equals $\langle\psi|\psi\rangle$. Thus $V^\dagger V = I_N$.

Now we verify the representation in Eq. (3.37). We compute $V\rho V^\dagger$. It is helpful to view V formally as $V = \sum_j K_j \otimes |j\rangle$. Then

$$V\rho V^\dagger = \left(\sum_i K_i \otimes |i\rangle \right) \rho \left(\sum_j K_j^\dagger \otimes \langle j| \right) = \sum_{i,j} (K_i \rho K_j^\dagger) \otimes |i\rangle\langle j|. \tag{3.41}$$

Tracing over the ancilla (the second register) yields

$$\mathrm{Tr}_A[V\rho V^\dagger] = \sum_{i,j} (K_i \rho K_j^\dagger) \,\mathrm{Tr}[|i\rangle\langle j|] = \sum_j K_j \rho K_j^\dagger = \mathcal{Q}(\rho). \tag{3.42}$$

To realize this via a unitary evolution, we define U such that its action on the subspace corresponding to the initial state $\rho \otimes |0\rangle\langle 0|$ matches the isometry V. Let the joint space be large enough (e.g., dimension $D = \max(N, M)R$). We define U such that

$$U(|\psi\rangle \otimes |0\rangle) = V|\psi\rangle, \quad \forall |\psi\rangle \in \mathbb{C}^N. \tag{3.43}$$

(We might need to embed $\mathbb{C}^N$ and $\mathbb{C}^M \otimes \mathbb{C}^R$ into the larger space $\mathbb{C}^D$). Since V is an isometry, this definition is norm-preserving. We can always extend this definition to a full unitary U on the joint space.

Finally, we verify the representation in Eq. (3.38). Let $\rho = \sum_k p_k |\psi_k\rangle\langle\psi_k|$ be the spectral decomposition.

$$\begin{aligned} U(\rho \otimes |0\rangle\langle 0|)U^\dagger &= \sum_k p_k U(|\psi_k\rangle \otimes |0\rangle)(\langle\psi_k| \otimes \langle 0|)U^\dagger \\ &= \sum_k p_k (V|\psi_k\rangle)(V|\psi_k\rangle)^\dagger = V\rho V^\dagger. \end{aligned} \tag{3.44}$$

Therefore, $\mathcal{Q}(\rho) = \mathrm{Tr}_A[V\rho V^\dagger] = \mathrm{Tr}_A \left[U(\rho \otimes |0\rangle\langle 0|)U^\dagger \right]$. □

Theorem 3.20 states that any quantum channel, no matter how noisy or irreversible it appears, can always be modeled as a unitary interaction between the system and an environment (ancilla), followed by discarding the environment. This provides a powerful conceptual tool, showing that all quantum evolution is fundamentally unitary if we consider a large enough closed system.

3.3. Distance between state vectors and unitaries

A **distance** (also called a **metric**) on a set X is a function $d : X \times X \to \mathbb{R}$ that assigns a real number $d(x, y)$ to each pair of points $x, y \in X$. This function satisfies the following properties for all $x, y, z \in X$:

(1) (Non-negativity) $d(x, y) \geq 0$.
(2) (Identity of indiscernibles) $d(x, y) = 0$ if and only if $x = y$.
(3) (Symmetry) $d(x, y) = d(y, x)$.
(4) (Triangle inequality) $d(x, y) \leq d(x, z) + d(z, y)$.

For example, the vector 2-norm defines a metric on $\mathbb{C}^N : (x, y) \to \|x - y\|$, and the operator norm defines a metric on $\mathrm{U}(N)$: $(U, V) \to \|U - V\|$.

The difference for the product of K unitaries can be bounded using a simple technique sometimes referred to as a "hybrid argument". This technique is used to bound the distance between two states by considering a sequence of "hybrid" unitaries, each of which differs from the next in the sequence by a small amount.

Proposition 3.21 (Linear error growth for products of unitaries)**.** *Given unitaries* $U_1, \widetilde{U}_1, \ldots, U_K, \widetilde{U}_K \in \mathrm{U}(N)$ *satisfying*

$$\left\|U_i - \widetilde{U}_i\right\| \leq \epsilon, \quad \forall i = 1, \ldots, K, \tag{3.45}$$

we have

$$\left\|U_K \cdots U_1 - \widetilde{U}_K \cdots \widetilde{U}_1\right\| \leq K\epsilon. \tag{3.46}$$

Proof. Use a telescoping series

$$\begin{aligned} & U_K \cdots U_1 - \widetilde{U}_K \cdots \widetilde{U}_1 \\ =& (U_K \cdots U_2 U_1 - U_K \cdots U_2 \widetilde{U}_1) + (U_K \cdots U_3 U_2 \widetilde{U}_1 - U_K \cdots U_3 \widetilde{U}_2 \widetilde{U}_1) + \cdots \\ & + (U_K U_{K-1} \cdots \widetilde{U}_1 - \widetilde{U}_K \widetilde{U}_{K-1} \cdots \widetilde{U}_1) \\ =& U_K \cdots U_2 (U_1 - \widetilde{U}_1) + U_K \cdots U_3 (U_2 - \widetilde{U}_2) \widetilde{U}_1 + \cdots + (U_K - \widetilde{U}_K) \widetilde{U}_{K-1} \cdots \widetilde{U}_1. \end{aligned} \tag{3.47}$$

Since all $U_i, \widetilde{U}_i$ are unitary matrices, we readily have

$$\left\|U_K \cdots U_1 - \widetilde{U}_K \cdots \widetilde{U}_1\right\| \leq \sum_{i=1}^{K} \left\|U_i - \widetilde{U}_i\right\| \leq K\epsilon. \tag{3.48}$$

□

It is worth mentioning that the **Duhamel principle** (also called the variation of constants) can be viewed as a continuous-time analogue of the linear error growth property in discrete-time settings.

Proposition 3.22 (Duhamel's principle for Hamiltonian simulation)**.** *Let $H \in \mathbb{C}^{N\times N}$ be a Hermitian matrix and let $B(t) \in \mathbb{C}^{N\times N}$ be an arbitrary time-dependent matrix. Suppose $U(t), \widetilde{U}(t) \in \mathbb{C}^{N\times N}$ form the solutions to the following initial value problems:*

$$i\partial_t U(t) = HU(t), \quad i\partial_t \widetilde{U}(t) = H\widetilde{U}(t) + B(t), \quad \text{with } U(0) = \widetilde{U}(0) = I. \tag{3.49}$$

Then, the solution $\widetilde{U}(t)$ satisfies the integral equation

$$\widetilde{U}(t) = U(t) - i\int_0^t U(t-s)B(s)\,\mathrm{d}s. \tag{3.50}$$

Furthermore, the difference between the propagators is bounded by

$$\left\|\widetilde{U}(t) - U(t)\right\| \le \int_0^t \|B(s)\|\,\mathrm{d}s. \tag{3.51}$$

Proof. We verify Eq. (3.50) by direct differentiation. Let $V(t)$ denote the right-hand side of Eq. (3.50). At $t = 0$, the integral vanishes, so $V(0) = U(0) = I$, which satisfies the initial condition.

Next, we differentiate $V(t)$ with respect to time. Using $\partial_t U(t) = -iHU(t)$ and Leibniz's integral rule, we obtain:

$$\begin{aligned}
\partial_t V(t) &= \partial_t U(t) - i\left(U(0)B(t) + \int_0^t \left[\partial_t U(t-s)\right] B(s)\,\mathrm{d}s\right) \\
&= -iHU(t) - iB(t) - i\int_0^t \left[-iHU(t-s)\right] B(s)\,\mathrm{d}s \\
&= -iH\left(U(t) - i\int_0^t U(t-s)B(s)\,\mathrm{d}s\right) - iB(t) \\
&= -iHV(t) - iB(t).
\end{aligned} \tag{3.52}$$

Multiplying by i, we obtain $i\partial_t V(t) = HV(t) + B(t)$. Thus, $V(t)$ satisfies the defining differential equation for $\widetilde{U}(t)$.

Taking the norm of the integral term yields

$$\left\|\widetilde{U}(t) - U(t)\right\| = \left\|-i\int_0^t U(t-s)B(s)\,\mathrm{d}s\right\| \le \int_0^t \|U(t-s)\|\,\|B(s)\|\,\mathrm{d}s = \int_0^t \|B(s)\|\,\mathrm{d}s, \tag{3.53}$$

which completes the proof. □

An important application of Proposition 3.22 arises when the inhomogeneity takes the form $B(t) = E(t)\widetilde{U}(t)$, where $E(t)$ represents an error term or perturbation. In this case, Eq. (3.50) becomes

$$\widetilde{U}(t) = U(t) - i\int_0^t U(t-s)E(s)\widetilde{U}(s)\,\mathrm{d}s. \tag{3.54}$$

If the perturbed dynamics remains unitary (for example, if $\widetilde{U}$ solves $i\partial_t\widetilde{U}(t) = (H + E(t))\widetilde{U}(t)$ with $E(t)$ Hermitian), then $\left\|\widetilde{U}(s)\right\| = 1$ and the error bound simplifies to

$$\left\|\widetilde{U}(t) - U(t)\right\| \le \int_0^t \left\|E(s)\widetilde{U}(s)\right\|\,\mathrm{d}s \le \int_0^t \|E(s)\|\,\mathrm{d}s. \tag{3.55}$$

This is a continuous-time analogue of Proposition 3.21.

For most of this book, the vector 2-norm and the operator norm distances are both convenient and sufficient. However, they are only applicable to pure states. For measuring the distance between mixed states, new tools will be needed. Even for pure states, unitaries may differ by a phase which should be inconsequential for measuring physical observables. These require the introduction of new metrics.

Two state vectors $|\psi\rangle, |\varphi\rangle \in \mathbb{C}^N$ are physically indistinguishable if they only differ by a global phase. Similarly, two unitary matrices $U, V \in \mathrm{U}(N)$ induce the same evolution on density operators if they only differ by a global phase. Consider the matrices

$$I_+ := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad I_- := \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{3.56}$$

We have in this case that $\|I_+ - I_-\| = 2$. However, for an arbitrary density matrix ρ, the induced evolution of the density operator under these two operators is

$$\|I_+ \rho I_+ - I_- \rho I_-\| = \left\|\rho - (-1)^2 \rho\right\| = 0. \tag{3.57}$$

This motivates the definition of the global phase invariant distance for vectors and unitary matrices. The subscript p in D_p stands for phase.

Definition 3.23. *Let $|\psi\rangle, |\varphi\rangle \in \mathbb{C}^N$ be two state vectors, their **global phase invariant distance** is*

$$D_p(|\psi\rangle, |\varphi\rangle) := \min_{\phi \in \mathbb{R}} \left\| |\psi\rangle - e^{i\phi} |\varphi\rangle \right\|. \tag{3.58}$$

Definition 3.24. *For two unitaries $U, V \in \mathrm{U}(N)$, their global phase invariant distance is*

$$D_p(U, V) = \min_{\phi \in \mathbb{R}} \left\| U - e^{i\phi} V \right\|. \tag{3.59}$$

An **equivalence relation** on a set X is a binary relation $\sim$ that satisfies the following three properties for all $a, b, c \in X$:

(1) (Reflexivity) $a \sim a$.
(2) (Symmetry) If $a \sim b$, then $b \sim a$.
(3) (Transitivity) If $a \sim b$ and $b \sim c$, then $a \sim c$.

A relation that satisfies these properties is called an equivalence relation, and it partitions the set X into disjoint equivalence classes.

Definition 3.25. *Let X be a set and $\sim$ be an equivalence relation on X. The quotient space (or quotient set) $X/\sim$ is defined as the set of equivalence classes of X under the relation $\sim$. An equivalence class $[x]$ of an element $x \in X$ is the set of all elements in X that are equivalent to x, i.e.,*

$$[x] = \{y \in X \mid y \sim x\}. \tag{3.60}$$

The quotient space $X/\sim$ is the set of all such equivalence classes:

$$X/\sim = \{[x] \mid x \in X\}. \tag{3.61}$$

Example 3.26. Define an equivalence relation on $\mathbb{C}^N$:

$$x \sim y \iff x = \lambda y \text{ for some } \lambda \in \mathbb{C} \setminus \{0\}, \quad x, y \in \mathbb{C}^N \setminus \{0\}. \tag{3.62}$$

Then $\mathrm{P}\mathbb{C}^N := \mathbb{C}^N \setminus \{0\}/\sim$ is called the **complex projective space**, which is isomorphic to the set of all nonzero physical states. The **real dimension** of a manifold M is the number of real coordinates needed to locally describe the manifold. For example, the real dimension of $\mathbb{C}^N$ is $2N$, and the real dimension of $\mathrm{P}\mathbb{C}^N$ is $2N-2$.

We may identify each single qubit quantum state with a unique point on the Bloch sphere as

$$\mathbf{a} = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)^\top, \quad \theta, \varphi \in \mathbb{R}. \tag{3.63}$$

This agrees with the previous statement that the real dimension of $\mathrm{P}\mathbb{C}^2$ is 2. ◇

Exercise 3.2. Prove that the global phase invariant distance is a distance on the complex projective space $\mathrm{P}\mathbb{C}^N$.

Example 3.27. Define an equivalence relation on $\mathrm{U}(N)$:

$$U \sim V \iff U = e^{i\theta}V \text{ for some } \theta \in \mathbb{R}, \quad U, V \in \mathrm{U}(N). \tag{3.64}$$

Then $\mathrm{PU}(N) := \mathrm{U}(N)/\sim$ is called the **projective unitary group**. The real dimension of $\mathrm{U}(N)$ is N^2, and the real dimension of $\mathrm{PU}(N)$ is N^2-1.

Recall that the special unitary group $\mathrm{SU}(N)$ consists of all unitary matrices with determinant 1. So the real dimension of $\mathrm{SU}(N)$ is N^2-1. However, the equivalence relation on $\mathrm{SU}(N)$ is

$$U \sim V \iff U = e^{i2\pi k/N}V \text{ for some } k \in [N], \quad U, V \in \mathrm{SU}(N). \tag{3.65}$$

So each equivalence class only consists of N discrete elements and does not reduce the dimension. Therefore the real dimension of the projective special unitary group denoted by $\mathrm{PSU}(N)$ is still N^2-1. ◇

Exercise 3.3. Prove that the global phase invariant distance is a distance on the projective unitary group $\mathrm{PU}(N)$.

Exercise 3.4. Given unitaries $U_1, \widetilde{U}_1, \ldots, U_K, \widetilde{U}_K \in \mathrm{U}(N)$ satisfying

$$D_p(U_i, \widetilde{U}_i) \le \epsilon, \quad \forall i = 1, \ldots, K, \tag{3.66}$$

prove that

$$D_p(U_K \cdots U_1, \widetilde{U}_K \cdots \widetilde{U}_1) \le K\epsilon. \tag{3.67}$$

Let $|\varphi\rangle = e^{i\alpha}\cos\theta\,|\psi\rangle + \sin\theta\,|\perp\rangle$, where $\langle\psi|\perp\rangle = 0$ and $0 \le \theta \le \pi/2$. Then $\cos\theta = |\langle\varphi|\psi\rangle|$ is the overlap between the two vectors. We can perform a unitary operation that rotates $e^{i\alpha}|\psi\rangle$ to $|0\rangle$ and $|\perp\rangle$ to $|1\rangle$. Direct calculation shows

$$D_p(|\psi\rangle, |\varphi\rangle) = \min_{\phi\in\mathbb{R}} \big\| |0\rangle - e^{i\phi}(e^{i\alpha}\cos\theta\,|0\rangle + \sin\theta\,|1\rangle)\big\| = \sqrt{2(1-\cos\theta)} = \sqrt{2(1-|\langle\varphi|\psi\rangle|)}. \tag{3.68}$$

Therefore the global phase invariant distance between two vectors can be directly computed from the overlap.

Exercise 3.5. For $U, V \in \mathrm{U}(N)$, prove that

$$D_p(U, V) = 2\min_{\phi}\max_{j}\left|\sin\frac{\lambda_j - \phi}{2}\right|, \tag{3.69}$$

where $\{e^{i\lambda_j}\}$ are eigenvalues of $V^\dagger U$.

Exercise 3.6. For $U, V \in \mathrm{U}(N)$, another distance that is invariant to the global phase is

$$D_{p,F}(U,V) = \frac{1}{\sqrt{2N}} \min_{\phi} \left\| U - e^{i\phi} V \right\|_F . \tag{3.70}$$

Prove that

$$D_{p,F}(U,V) = \sqrt{1 - \frac{|\mathrm{Tr}[U^\dagger V|}{N}}. \tag{3.71}$$

3.4. Distance between classical states and classical channels

In this section, we provide a connection between concepts in classical probabilistic computation and density operators and quantum channels in quantum computation. For two probability distributions $p, q \in \mathbb{R}^N$, the **total variation distance** is

$$D(p,q) := \frac{1}{2} \sum_{i \in [N]} |p_i - q_i| . \tag{3.72}$$

The name total variation distance comes from that it measures the largest difference between p and q for some subset (also called event) S. The total variation distance is the default metric we will use between probability distributions and will be denoted by D without subscripts.

Proposition 3.28. *For any two classical probability distributions $p, q \in \mathbb{R}^N$,*

$$D(p,q) = \max_S (p(S) - q(S)) := \max_S \left(\sum_{i \in S} p_i - \sum_{i \in S} q_i \right), \tag{3.73}$$

where the maximization is over all subsets S.

Proof. For any subset S, let $\overline{S}$ be its complement. Then

$$0 = \sum_i p_i - \sum_i q_i = \sum_{i \in S} (p_i - q_i) + \sum_{i \in \overline{S}} (p_i - q_i). \tag{3.74}$$

Hence

$$\sum_{i \in S} p_i - \sum_{i \in S} q_i = \frac{1}{2} \left(\sum_{i \in S} (p_i - q_i) - \sum_{i \in \overline{S}} (p_i - q_i) \right) \leq D(p,q). \tag{3.75}$$

Now let $S = \{i | p_i \geq q_i\}$. Then

$$\frac{1}{2} \left(\sum_{i \in S} (p_i - q_i) - \sum_{i \in \overline{S}} (p_i - q_i) \right) = \frac{1}{2} \sum_i |p_i - q_i| = D(p,q), \tag{3.76}$$

and the equality is achieved. □

We now prove that the application of a transition matrix does not increase the total variation distance.

Proposition 3.29. *Given a transition matrix $P \in \mathbb{R}^{N \times N}$, and any two classical probability distributions $p, q \in \mathbb{R}^N$,*

$$D(Pp, Pq) \leq D(p,q). \tag{3.77}$$

If the equality holds for any $p, q \in \mathbb{R}^N$, then P is a permutation matrix.

PROOF. Use the left stochasticity of the transition matrix, we have

$$D(Pp, Pq) = \frac{1}{2}\sum_i \left|\sum_j P_{ij}(p_j - q_j)\right| \leq \frac{1}{2}\sum_i \sum_j P_{ij}\,|p_j - q_j| = \frac{1}{2}\sum_j |p_j - q_j| = D(p, q). \tag{3.78}$$

If the equality holds for any $p, q \in \mathbb{R}^N$, we prove that each row of P has only one nonzero entry. If this is not the case, assume that there exists a row index i and two distinct column indices $j_1 \neq j_2$ such that $P_{ij_1} > 0$ and $P_{ij_2} > 0$. Choose $p = e_{j_1}$ and $q = e_{j_2}$. Then for this row i,

$$\left|\sum_j P_{ij}(p_j - q_j)\right| = |P_{i,j_1} - P_{i,j_2}| < P_{i,j_1} + P_{i,j_2} = \sum_j P_{ij}\,|p_j - q_j|\,, \tag{3.79}$$

which contradicts equality in the triangle inequality step above. Hence each row has exactly one nonzero entry. By left stochasticity, each column must also have exactly one nonzero entry, which must equal 1. This proves that P is a permutation matrix. □

The **induced total variation distance** between two transition matrices $P, Q \in \mathbb{R}^{N\times N}$ is defined as

$$D(P, Q) = \max_{j\in[N]} D(P_{:,j}, Q_{:,j}). \tag{3.80}$$

Exercise 3.7. Prove that $D(\cdot,\cdot)$ is a distance on the set of $N \times N$ transition matrices.

Finally, we prove that the difference for the composition of K classical channels grows linearly.

Proposition 3.30 (Linear error growth for product of transition matrices)**.** *Given the transition matrices $P_1, \widetilde{P}_1, \ldots, P_K, \widetilde{P}_K \in \mathbb{R}^{N\times N}$, the induced total variation distance satisfies*

$$D(P_K \cdots P_1, \widetilde{P}_K \cdots \widetilde{P}_1) \leq \sum_{i=1}^{K} D(P_i, \widetilde{P}_i). \tag{3.81}$$

PROOF. Using the telescope series Proposition 3.21, it is sufficient to consider the case for $K = 2$. Then

$$\begin{aligned}
D(P_2P_1, \widetilde{P}_2\widetilde{P}_1) \leq& D(P_2P_1, P_2\widetilde{P}_1) + D(P_2\widetilde{P}_1, \widetilde{P}_2\widetilde{P}_1)\\
=& \max_{j\in[N]} D((P_2P_1)_{:,j}, (P_2\widetilde{P}_1)_{:,j}) + \max_{j\in[N]} D((P_2\widetilde{P}_1)_{:,j}, (\widetilde{P}_2\widetilde{P}_1)_{:,j})\\
\leq& \max_{j\in[N]} D((P_1)_{:,j}, (\widetilde{P}_1)_{:,j}) + \max_{j\in[N]} \left(\max_{l\in[N]} D((P_2)_{:,l}, (\widetilde{P}_2)_{:,l})\right) \sum_k (\widetilde{P}_1)_{kj}\\
\leq& \max_{j\in[N]} D((P_1)_{:,j}, (\widetilde{P}_1)_{:,j}) + \max_{l\in[N]} D((P_2)_{:,l}, (\widetilde{P}_2)_{:,l})\\
=& D(P_1, \widetilde{P}_1) + D(P_2, \widetilde{P}_2).
\end{aligned} \tag{3.82}$$

Here we have used Proposition 3.29 and the left stochasticity of $\widetilde{P}_1$. □

3.5. Distance between quantum states

Quantifying the similarity or difference between quantum states is fundamental to quantum information theory. It allows us to analyze the performance of quantum algorithms, assess the errors in quantum communication protocols, and understand the distinguishability of quantum states through measurements. In this section, we introduce the two most widely used measures: the trace distance and the fidelity. These generalize the corresponding concepts for classical probability distributions, such as the total variation distance discussed in Section 3.4. For a comprehensive treatment, we refer readers to [NC00, Chapter 9] and [Wat18, Chapter 3].

3.5.1. Schatten norms and the trace norm. To define distances between density operators, which are matrices, we first need appropriate matrix norms. The Schatten norms provide a family of norms generalizing the ℓ^p norms for vectors to the space of operators.

Let $A \in \mathbb{C}^{M\times N}$. The singular values of A, denoted $\sigma_i(A)$, are the square roots of the non-negative eigenvalues of $A^\dagger A$. The matrix norm!Schatten p-norm of A for $p \geq 1$ is defined as the ℓ^p norm of its singular values:

$$\|A\|_p := \left(\sum_i \sigma_i(A)^p\right)^{\frac{1}{p}}. \tag{3.83}$$

This can also be expressed using the trace function. Let $|A| := \sqrt{A^\dagger A}$ denote the positive semidefinite square root of $A^\dagger A$. Then

$$\|A\|_p = (\mathrm{Tr}[|A|^p])^{\frac{1}{p}}. \tag{3.84}$$

The following choices of p are particularly important:

- The Schatten 1-norm, also known as the trace norm, is the sum of the singular values:

$$\|A\|_1 = \mathrm{Tr}[|A|] = \sum_i \sigma_i(A). \tag{3.85}$$

  If A is positive semidefinite, $|A| = A$, so $\|A\|_1 = \mathrm{Tr}[A]$.
- The Schatten 2-norm (also called the Hilbert-Schmidt norm or Frobenius norm) is the Euclidean norm of the singular values:

$$\|A\|_2 = \sqrt{\mathrm{Tr}[A^\dagger A]} = \left(\sum_i \sigma_i(A)^2\right)^{\frac{1}{2}}. \tag{3.86}$$

- The Schatten ∞-norm is the maximum singular value:

$$\|A\|_\infty = \lim_{p\to\infty} \|A\|_p = \max_i \sigma_i(A). \tag{3.87}$$

  This is identical to the standard operator norm (the induced $\ell^2 \to \ell^2$ norm), often denoted $\|A\|$ (equivalently $\|A\|_\infty$).

A basic but useful property relates the trace of a matrix to its trace norm.

Proposition 3.31. *For any square matrix $A \in L(\mathbb{C}^N)$,*

$$|\mathrm{Tr}[A]| \leq \|A\|_1. \tag{3.88}$$

PROOF. Consider the singular value decomposition $A = U\Sigma V^\dagger$, where U, V are unitary and $\Sigma = \mathrm{diag}(\sigma_i)$ contains the singular values. Using the cyclic property of the trace:

$$\mathrm{Tr}[A] = \mathrm{Tr}[U\Sigma V^\dagger] = \mathrm{Tr}[\Sigma V^\dagger U]. \tag{3.89}$$

Let $W = V^\dagger U$. Since W is unitary, its entries satisfy $|W_{ii}| \leq 1$ for all i. Therefore, by the triangle inequality,

$$|\mathrm{Tr}[A]| = \left|\sum_i \sigma_i W_{ii}\right| \leq \sum_i \sigma_i |W_{ii}| \leq \sum_i \sigma_i = \|A\|_1 . \tag{3.90}$$

□

The Schatten norms share many properties with the ℓ^p norms for vectors, including the triangle inequality and Hölder's inequality. We state these fundamental results without proof, referring the reader to texts on matrix analysis such as [Bha97].

Proposition 3.32 (Properties of Schatten p-norms)**.** *Let A, B be operators.*

1. *(Triangle inequality) For $1 \leq p \leq \infty$, $\|A + B\|_p \leq \|A\|_p + \|B\|_p$.*
2. *(Hölder's inequality, [Bha97, Corollary IV.2.6]) For $1 \leq p, q \leq \infty$ satisfying $\frac{1}{p} + \frac{1}{q} = 1$, if the product AB is defined, then $\|AB\|_1 \leq \|A\|_p \|B\|_q$.*

We are primarily interested in the trace norm ($p = 1$) and the operator norm ($p = \infty$). An important specialization of Hölder's inequality is the case $p = \infty, q = 1$:

$$\|AB\|_1 \leq \|A\|_\infty \|B\|_1 . \tag{3.91}$$

This inequality is frequently used to bound the trace norm of a product. Another useful variation involves the trace of a product, which can be viewed as a generalization of the Cauchy-Schwarz inequality. We provide a self-contained proof of this specific case.

Lemma 3.33 (Hölder's inequality for trace)**.** *For any operators $A, B \in L(\mathbb{C}^N)$, the following inequality holds:*

$$\left|\mathrm{Tr}(A^\dagger B)\right| \leq \|A\|_\infty \|B\|_1 . \tag{3.92}$$

PROOF. Let $B = U\Sigma V^\dagger$ be the SVD of B, with singular values s_i. By definition, $\|B\|_1 = \sum_i s_i$. Using the cyclic property of the trace:

$$\mathrm{Tr}(A^\dagger B) = \mathrm{Tr}(A^\dagger U\Sigma V^\dagger) = \mathrm{Tr}(V^\dagger A^\dagger U\Sigma). \tag{3.93}$$

Let $W = V^\dagger A^\dagger U$. Since U and V are unitary, the operator norm is invariant under unitary multiplication: $\|W\|_\infty = \left\|A^\dagger\right\|_\infty$. Furthermore, $\left\|A^\dagger\right\|_\infty = \|A\|_\infty$ as they share the same singular values. The trace is the sum of the diagonal elements of W weighted by the singular values:

$$\mathrm{Tr}(W\Sigma) = \sum_i W_{ii} s_i. \tag{3.94}$$

We can now bound the magnitude of the trace using the triangle inequality:

$$\begin{aligned} \left|\mathrm{Tr}(A^\dagger B)\right| &= \left|\sum_i W_{ii} s_i\right| \leq \sum_i |W_{ii}| s_i \\ &\leq \sum_i \|W\|_\infty s_i = \|A\|_\infty \sum_i s_i = \|A\|_\infty \|B\|_1 . \end{aligned} \tag{3.95}$$

□

We now consider how the trace norm behaves under the partial trace operation, which often arises when dealing with composite systems.

Exercise 3.8. Let $|u\rangle, |v\rangle$ be normalized state vectors in $\mathcal{H}_A \otimes \mathcal{H}_B$. Show that

$$\|\mathrm{Tr}_B\, |u\rangle\langle v|\|_1 \leq 1. \tag{3.96}$$

(Hint: use Hölder's inequality for the Schatten 2-norm.)

More generally, the partial trace is a contraction with respect to the trace norm.

Proposition 3.34 (Partial trace does not increase the trace norm)**.** *For any operator $O \in L(\mathcal{H}_A \otimes \mathcal{H}_B)$,*

$$\|\mathrm{Tr}_B\, O\|_1 \leq \|O\|_1 . \tag{3.97}$$

Proof. Consider the singular value decomposition of the operator O:

$$O = \sum_k \sigma_k |u_k\rangle\langle v_k|, \tag{3.98}$$

where $\sigma_k > 0$ are the singular values, and $\{|u_k\rangle\}, \{|v_k\rangle\}$ are sets of orthonormal vectors in $\mathcal{H}_A \otimes \mathcal{H}_B$. The trace norm is $\|O\|_1 = \sum_k \sigma_k$.

Applying the partial trace and using the triangle inequality (Proposition 3.32):

$$\|\mathrm{Tr}_B\, O\|_1 = \left\| \sum_k \sigma_k \,\mathrm{Tr}_B\, |u_k\rangle\langle v_k| \right\|_1 \leq \sum_k \sigma_k \,\|\mathrm{Tr}_B\, |u_k\rangle\langle v_k|\|_1 . \tag{3.99}$$

By Exercise 3.8, $\|\mathrm{Tr}_B\, |u_k\rangle\langle v_k|\|_1 \leq 1$. Therefore,

$$\|\mathrm{Tr}_B\, O\|_1 \leq \sum_k \sigma_k = \|O\|_1 . \tag{3.100}$$

□

The trace norm and the operator norm are dual to each other with respect to the trace inner product, a property that is frequently exploited in optimization problems and for deriving operational interpretations of these norms.

Lemma 3.35 (Duality of Trace and Operator Norms)**.** *For any operator $Y \in L(\mathbb{C}^N)$, the following identities hold:*

$$\|Y\|_1 = \sup_{\|Z\|_\infty \leq 1} |\,\mathrm{Tr}(Z^\dagger Y)|, \tag{3.101}$$

and

$$\|Y\|_\infty = \sup_{\|X\|_1 \leq 1} |\,\mathrm{Tr}(Y^\dagger X)|. \tag{3.102}$$

Proof. We first prove Eq. (3.101). Let S_1 denote the right-hand side. Applying Hölder's inequality (Lemma 3.33), we have $|\,\mathrm{Tr}(Z^\dagger Y)| \leq \|Z\|_\infty \|Y\|_1$. If we restrict the optimization to $\|Z\|_\infty \leq 1$, then $|\,\mathrm{Tr}(Z^\dagger Y)| \leq \|Y\|_1$. Taking the supremum yields $S_1 \leq \|Y\|_1$.

To show $S_1 \geq \|Y\|_1$, we construct an operator Z that achieves the bound. Let $Y = U\Sigma V^\dagger$ be the SVD of Y. Define $Z = UV^\dagger$. Since Z is unitary, $\|Z\|_\infty = 1$. We compute the trace:

$$\begin{aligned} \mathrm{Tr}(Z^\dagger Y) &= \mathrm{Tr}((UV^\dagger)^\dagger (U\Sigma V^\dagger)) = \mathrm{Tr}(VU^\dagger U\Sigma V^\dagger) \\ &= \mathrm{Tr}(V\Sigma V^\dagger) = \mathrm{Tr}(\Sigma) = \|Y\|_1 . \end{aligned} \tag{3.103}$$

Thus, $S_1 \geq \|Y\|_1$.

Next, we prove Eq. (3.102). Let S_∞ denote the right-hand side. Applying Lemma 3.33, we have $|\operatorname{Tr}(Y^\dagger X)| \leq \|Y\|_\infty \|X\|_1$. Restricting to $\|X\|_1 \leq 1$ and taking the supremum yields $S_\infty \leq \|Y\|_\infty$.

To show $S_\infty \geq \|Y\|_\infty$, we construct an optimal X. Let $Y = \sum_i s_i |u_i\rangle\langle v_i|$ be the SVD of Y, ordered such that $s_1 = \|Y\|_\infty$. Define the rank-1 operator $X = |u_1\rangle\langle v_1|$. Since $|u_1\rangle, |v_1\rangle$ are normalized, $\|X\|_1 = 1$. We compute the trace:

$$\begin{aligned} \operatorname{Tr}(Y^\dagger X) &= \operatorname{Tr}\left(\left(\sum_i s_i |v_i\rangle\langle u_i|\right)|u_1\rangle\langle v_1|\right) \\ &= \operatorname{Tr}\left(\sum_i s_i |v_i\rangle\langle v_1| \langle u_i|u_1\rangle\right). \end{aligned} \tag{3.104}$$

Due to the orthonormality of $\{|u_i\rangle\}$, only the $i = 1$ term survives:

$$\operatorname{Tr}(Y^\dagger X) = \operatorname{Tr}(s_1 |v_1\rangle\langle v_1|) = s_1 = \|Y\|_\infty . \tag{3.105}$$

Thus, $S_\infty \geq \|Y\|_\infty$. □

When the operator Y is Hermitian, the optimization domains in these duality relations can also be restricted to Hermitian operators.

Lemma 3.36 (Duality for Hermitian Operators)**.** *Let $H \in L(\mathbb{C}^N)$ be a Hermitian operator.*

(1) *The trace norm is achieved by maximizing over Hermitian operators in the unit operator-norm ball (i.e., $-I \preceq Z \preceq I$):*

$$\|H\|_1 = \sup\{|\operatorname{Tr}(ZH)| : Z = Z^\dagger, \|Z\|_\infty \leq 1\}. \tag{3.106}$$

(2) *The operator norm is achieved by maximizing over density operators:*

$$\|H\|_\infty = \sup\{|\operatorname{Tr}(H\rho)| : \rho \in \mathcal{D}(\mathbb{C}^N)\}. \tag{3.107}$$

Proof. In both cases, the inequality $\leq$ (for the left-hand side) follows immediately from Lemma 3.35, as the restricted optimization domains are subsets of the original domains. We only need to show that the bounds can be achieved within these restricted domains.

1. Proof of Eq. (3.106). Let $H = \sum_i \lambda_i |\psi_i\rangle\langle\psi_i|$ be the spectral decomposition, where $\lambda_i \in \mathbb{R}$. The trace norm is $\|H\|_1 = \sum_i |\lambda_i|$. Define the sign operator $Z = \sum_i \operatorname{sgn}(\lambda_i)|\psi_i\rangle\langle\psi_i|$. Z is Hermitian, and its eigenvalues are in $\{-1, 0, 1\}$, so $\|Z\|_\infty \leq 1$.

$$\operatorname{Tr}(ZH) = \sum_i \operatorname{sgn}(\lambda_i)\lambda_i = \sum_i |\lambda_i| = \|H\|_1 . \tag{3.108}$$

2. Proof of Eq. (3.107). The operator norm is $\|H\|_\infty = \max_i |\lambda_i|$. Let k be an index achieving the maximum. Define the pure state $\rho = |\psi_k\rangle\langle\psi_k|$, which is a density operator.

$$|\operatorname{Tr}(H\rho)| = |\langle\psi_k|H|\psi_k\rangle| = |\lambda_k| = \|H\|_\infty . \tag{3.109}$$

□

3.5.2. Trace distance. The trace norm provides a natural way to define a distance metric on the space of quantum states, generalizing the classical total variation distance.

Definition 3.37 (Trace distance)**.** *The **trace distance** between two quantum states $\rho, \sigma \in \mathcal{D}(\mathbb{C}^N)$ is defined as*

$$D(\rho, \sigma) := \frac{1}{2} \|\rho - \sigma\|_1 . \tag{3.110}$$

The factor of $1/2$ ensures that the distance lies in the range $[0, 1]$. Since $\|\rho\|_1 = 1$ and $\|\sigma\|_1 = 1$, the triangle inequality (Proposition 3.32) gives $\|\rho - \sigma\|_1 \leq \|\rho\|_1 + \|\sigma\|_1 = 2$.

Example 3.38 (Trace distance for classical states)**.** Consider classical probability distributions $p, s \in \mathbb{R}^N$ embedded as classical states:

$$\rho = \sum_{i \in [N]} p_i |i\rangle\langle i|, \quad \sigma = \sum_{i \in [N]} s_i |i\rangle\langle i|. \tag{3.111}$$

The difference $\rho - \sigma$ is a diagonal matrix with entries $p_i - s_i$. The trace norm is the sum of the absolute values of the eigenvalues:

$$D(\rho, \sigma) = \frac{1}{2} \|\rho - \sigma\|_1 = \frac{1}{2} \sum_i |p_i - s_i| . \tag{3.112}$$

This is exactly the total variation distance $D(p, s)$ between the probability distributions p and s. $\diamond$

The trace distance has an operational interpretation related to the distinguishability of quantum states through measurement. This is the quantum generalization of Proposition 3.28.

Proposition 3.39 (Operational interpretation of trace distance)**.** *For any quantum states $\rho, \sigma \in \mathcal{D}(\mathbb{C}^N)$, the trace distance satisfies*

$$D(\rho, \sigma) = \max_{0 \preceq M \preceq I} \operatorname{Tr}[M(\rho - \sigma)]. \tag{3.113}$$

The maximum is achieved when M is the projector onto the subspace where $\rho - \sigma$ is positive.

Proof. Let $\Delta = \rho - \sigma$. Δ is Hermitian and $\operatorname{Tr}[\Delta] = 0$. We want to maximize $\operatorname{Tr}[M\Delta]$ over $0 \preceq M \preceq I$.

We utilize the duality results established earlier. Consider an operator M such that $0 \preceq M \preceq I$. Define $Z = 2M - I$. Then Z is Hermitian, and $-I \preceq Z \preceq I$, which means $\|Z\|_\infty \leq 1$. We have

$$\operatorname{Tr}[Z\Delta] = \operatorname{Tr}[(2M - I)\Delta] = 2 \operatorname{Tr}[M\Delta]. \tag{3.114}$$

By the Hermitian duality relation (Lemma 3.36, Eq. (3.106)), $\|\Delta\|_1 = \sup\{|\operatorname{Tr}(Z'\Delta)| : Z' = Z'^\dagger, \|Z'\|_\infty \leq 1\}$. Since Z is admissible for this optimization, we have

$$2 \operatorname{Tr}[M\Delta] = \operatorname{Tr}[Z\Delta] \leq \|\Delta\|_1 . \tag{3.115}$$

Thus, $\operatorname{Tr}[M\Delta] \leq \frac{1}{2} \|\Delta\|_1 = D(\rho, \sigma)$.

To show equality, we construct an optimal M. Let $\Delta = \Delta_+ - \Delta_-$, where Δ_+, Δ_- are positive semidefinite operators with orthogonal support. Since $\operatorname{Tr}[\Delta] = 0$, we have $\operatorname{Tr}[\Delta_+] = \operatorname{Tr}[\Delta_-]$. The trace norm is

$$\|\Delta\|_1 = \operatorname{Tr}[\Delta_+] + \operatorname{Tr}[\Delta_-] = 2 \operatorname{Tr}[\Delta_+]. \tag{3.116}$$

So $D(\rho, \sigma) = \operatorname{Tr}[\Delta_+]$.

Let P be the projector onto the support of Δ_+ with $P\Delta_+ = \Delta_+$. We evaluate the trace:

$$\mathrm{Tr}[P\Delta] = \mathrm{Tr}[P(\Delta_+ - \Delta_-)] = \mathrm{Tr}[\Delta_+] = D(\rho, \sigma). \tag{3.117}$$

Therefore, the maximum is achieved. □

Proposition 3.39 implies that $D(\rho, \sigma)$ is the maximum difference in the probability of obtaining a specific measurement outcome when measuring ρ versus σ.

A fundamental property of the trace distance is that it cannot increase under the action of a quantum channel. This reflects the physical intuition that noise or information loss (modeled by the channel) makes states harder to distinguish. This result parallels Proposition 3.29 for classical channels.

THEOREM 3.40 (Quantum channels are contractive). *Let $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ be a quantum channel. For any $\rho, \sigma \in \mathcal{D}(\mathbb{C}^N)$,*

$$D(\mathcal{Q}[\rho], \mathcal{Q}[\sigma]) \leq D(\rho, \sigma). \tag{3.118}$$

PROOF. Let $\rho' = \mathcal{Q}[\rho]$ and $\sigma' = \mathcal{Q}[\sigma]$. By Proposition 3.39, there exists a projector P (specifically, onto the positive subspace of $\rho' - \sigma'$) such that

$$D(\rho', \sigma') = \mathrm{Tr}[P(\rho' - \sigma')] = \mathrm{Tr}[P\mathcal{Q}[\rho - \sigma]]. \tag{3.119}$$

Consider the decomposition $\rho - \sigma = \Delta_+ - \Delta_-$, where $\Delta_+, \Delta_- \succeq 0$ are the positive and negative parts, respectively. As shown in the proof of Proposition 3.39, $D(\rho, \sigma) = \mathrm{Tr}[\Delta_+] = \mathrm{Tr}[\Delta_-]$.

Substituting the decomposition and using linearity:

$$D(\rho', \sigma') = \mathrm{Tr}[P\mathcal{Q}[\Delta_+ - \Delta_-]] = \mathrm{Tr}[P\mathcal{Q}[\Delta_+]] - \mathrm{Tr}[P\mathcal{Q}[\Delta_-]]. \tag{3.120}$$

We analyze the two terms. Since $\mathcal{Q}$ is a positive map, and $\Delta_- \succeq 0$, the output $\mathcal{Q}[\Delta_-]$ is positive semidefinite. Since $P \succeq 0$, the trace of the product of two positive operators is non-negative: $\mathrm{Tr}[P\mathcal{Q}[\Delta_-]] \geq 0$. Therefore,

$$D(\rho', \sigma') \leq \mathrm{Tr}[P\mathcal{Q}[\Delta_+]]. \tag{3.121}$$

Next, since $\mathcal{Q}[\Delta_+] \succeq 0$ and $P \preceq I$, we have $I - P \succeq 0$. Thus $\mathrm{Tr}[(I - P)\mathcal{Q}[\Delta_+]] \geq 0$, which implies $\mathrm{Tr}[P\mathcal{Q}[\Delta_+]] \leq \mathrm{Tr}[\mathcal{Q}[\Delta_+]]$. Therefore,

$$D(\rho', \sigma') \leq \mathrm{Tr}[\mathcal{Q}[\Delta_+]]. \tag{3.122}$$

Finally, since $\mathcal{Q}$ is trace-preserving, $\mathrm{Tr}[\mathcal{Q}[\Delta_+]] = \mathrm{Tr}[\Delta_+]$. Combining the inequalities, we obtain

$$D(\mathcal{Q}[\rho], \mathcal{Q}[\sigma]) \leq \mathrm{Tr}[\Delta_+] = D(\rho, \sigma). \tag{3.123}$$

□

3.5.3. Fidelity. While the trace distance is an operationally useful metric for the distance between quantum states, another widely used measure is the fidelity. Fidelity quantifies the "overlap" between two quantum states, and generalizes the inner product between pure state vectors.

Definition 3.41 (Fidelity)**.** *The **fidelity** between two quantum states $\rho, \sigma \in \mathcal{D}(\mathbb{C}^N)$ is defined as*

$$F(\rho, \sigma) := \mathrm{Tr}\left[\sqrt{\rho^{\frac{1}{2}}\sigma\rho^{\frac{1}{2}}}\right]. \tag{3.124}$$

This definition can be rewritten using the trace norm. A more symmetric expression involves the operator $A = \rho^{1/2}\sigma^{1/2}$. Recall that the trace norm of A is $\|A\|_1 = \mathrm{Tr}[|A|] = \mathrm{Tr}[\sqrt{A^\dagger A}]$. Here $A^\dagger A = \sigma^{1/2}\rho\sigma^{1/2}$. The singular values of A are the square roots of the eigenvalues of $A^\dagger A$ (and also $AA^\dagger = \rho^{1/2}\sigma\rho^{1/2}$). Thus,

$$F(\rho,\sigma) = \left\|\rho^{1/2}\sigma^{1/2}\right\|_1. \tag{3.125}$$

This immediately establishes that fidelity is symmetric: $F(\rho,\sigma) = F(\sigma,\rho)$, since $\|A\|_1 = \left\|A^\dagger\right\|_1$.

Remark 3.42. Nomenclature can be confusing. Sometimes the quantity defined above is called the square root fidelity, and $F(\rho,\sigma)^2$ is called the fidelity. The **infidelity** is then defined as $1-F(\rho,\sigma)^2$. We will adhere to Definition 3.41. ◇

Fidelity satisfies $0 \le F(\rho,\sigma) \le 1$. The upper bound follows from Hölder's inequality (Proposition 3.32, $p = q = 2$):

$$F(\rho,\sigma) = \left\|\rho^{1/2}\sigma^{1/2}\right\|_1 \le \left\|\rho^{1/2}\right\|_2 \left\|\sigma^{1/2}\right\|_2. \tag{3.126}$$

Since $\left\|\rho^{1/2}\right\|_2^2 = \mathrm{Tr}[\rho^{1/2}\rho^{1/2}] = \mathrm{Tr}[\rho] = 1$, we have $F(\rho,\sigma) \le 1$. Furthermore, $F(\rho,\sigma) = 1$ if and only if $\rho = \sigma$.

Fidelity itself is not a distance metric (it does not satisfy the triangle inequality). However, it can be converted into a metric known as the angle or Bures angle.

Definition 3.43 (Angle between quantum states)**.** *The **angle** between two quantum states $\rho,\sigma \in \mathcal{D}(\mathbb{C}^N)$ is*

$$\theta(\rho,\sigma) := \arccos(F(\rho,\sigma)) \in [0,\pi/2]. \tag{3.127}$$

Example 3.44 (Pure states)**.** If $\rho = |\psi\rangle\langle\psi|$ and $\sigma = |\varphi\rangle\langle\varphi|$ are two pure states.

$$\rho^{1/2}\sigma\rho^{1/2} = |\psi\rangle\langle\psi||\varphi\rangle\langle\varphi||\psi\rangle\langle\psi| = |\langle\psi|\varphi\rangle|^2|\psi\rangle\langle\psi|. \tag{3.128}$$

This is a rank-1 operator. Its only non-zero eigenvalue is $|\langle\psi|\varphi\rangle|^2$. The square root of this eigenvalue is $|\langle\psi|\varphi\rangle|$. Thus,

$$F(\rho,\sigma) = |\langle\psi|\varphi\rangle|. \tag{3.129}$$

The fidelity is the absolute value of the overlap between the state vectors.

More generally, if only one state is pure, say $\rho = |\psi\rangle\langle\psi|$, then

$$F(\rho,\sigma) = \sqrt{\langle\psi|\sigma|\psi\rangle}. \tag{3.130}$$

It is the square root of the overlap between the pure state $|\psi\rangle$ and the mixed state σ.

Let us relate the trace distance and fidelity for pure states ρ,σ. Let the angle be $\theta = \theta(\rho,\sigma)$, so $F(\rho,\sigma) = \cos\theta$. We can choose a basis such that $|\psi\rangle = |0\rangle$ and $|\varphi\rangle = \cos\theta\,|0\rangle + \sin\theta\,|1\rangle$ (by adjusting global phase). In this 2D subspace, the difference $\rho - \sigma$ is represented by the matrix:

$$\Delta = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} \cos^2\theta & \cos\theta\sin\theta \\ \cos\theta\sin\theta & \sin^2\theta \end{pmatrix} = \begin{pmatrix} \sin^2\theta & -\cos\theta\sin\theta \\ -\cos\theta\sin\theta & -\sin^2\theta \end{pmatrix}. \tag{3.131}$$

The eigenvalues of Δ are $\pm\sin\theta$. The trace norm is $\|\Delta\|_1 = |\sin\theta| + |-\sin\theta| = 2\sin\theta$ (since $0 \le \theta \le \pi/2$).

$$D(\rho,\sigma) = \frac{1}{2}\|\rho-\sigma\|_1 = \sin\theta. \tag{3.132}$$

We can express this in terms of fidelity $F = \cos\theta$:

$$D(\rho, \sigma) = \sqrt{1 - F(\rho, \sigma)^2}. \tag{3.133}$$

◇

Example 3.45 (Classical states)**.** Let ρ, σ be classical states corresponding to probability distributions p, q (see Example 3.38). Since the operators are diagonal, the definition simplifies:

$$F(\rho, \sigma) = \sum_j \sqrt{p_j q_j}. \tag{3.134}$$

This is the classical Bhattacharyya coefficient. The relationship between trace distance and fidelity for classical states is characterized by the inequality:

$$\begin{aligned} D(\rho, \sigma) &= \frac{1}{2} \sum_j |p_j - q_j| \geq \frac{1}{2} \sum_j (\sqrt{p_j} - \sqrt{q_j})^2 \\ &= \frac{1}{2} \sum_j (p_j + q_j - 2\sqrt{p_j q_j}) = 1 - \sum_j \sqrt{p_j q_j} = 1 - F(\rho, \sigma). \end{aligned} \tag{3.135}$$

The inequality step uses $|a^2 - b^2| \geq (a - b)^2$ for $a, b \geq 0$. ◇

We have seen two extremes: for pure states $D = \sqrt{1 - F^2}$, while for classical states $D \geq 1 - F$. These relationships are generalized by the Fuchs–van de Graaf inequalities (see [NC00, Section 9.2]), which provide tight bounds relating the two measures for arbitrary quantum states.

THEOREM 3.46 (Fuchs–van de Graaf inequalities). *For any $\rho, \sigma \in \mathcal{D}(\mathbb{C}^N)$,*

$$1 - F(\rho, \sigma) \leq D(\rho, \sigma) \leq \sqrt{1 - F(\rho, \sigma)^2}. \tag{3.136}$$

We state a few important properties of fidelity without proof. Their proofs typically rely on a powerful result known as Uhlmann's theorem, which relates the fidelity between two mixed states to the maximum overlap between their purifications (see [NC00, Chapter 9], [Wat18, Chapter 3]).

Proposition 3.47 (Properties of Fidelity and Angle)**.** *Let $\rho, \sigma \in \mathcal{D}(\mathbb{C}^N)$.*

1. *(Metric property) The angle $\theta(\rho, \sigma)$ is a distance metric on $\mathcal{D}(\mathbb{C}^N)$.*
2. *(Contractivity) For any quantum channel $\mathcal{Q}$, the angle is contractive:*

$$\theta(\mathcal{Q}[\rho], \mathcal{Q}[\sigma]) \leq \theta(\rho, \sigma). \tag{3.137}$$

Equivalently, fidelity increases (or stays the same) under quantum channels:

$$F(\mathcal{Q}[\rho], \mathcal{Q}[\sigma]) \geq F(\rho, \sigma). \tag{3.138}$$

The Fuchs–van de Graaf inequalities (Theorem 3.46) can be rewritten in terms of the angle $\theta = \theta(\rho, \sigma)$:

$$2\sin^2\frac{\theta}{2} \leq D(\rho, \sigma) \leq \sin\theta. \tag{3.139}$$

When the states are close ($\theta 1$), we can use the approximations $\sin\theta \approx \theta$ and $2\sin^2(\theta/2) \approx \theta^2/2$. This gives

$$\frac{1}{2}\theta^2 \lesssim D(\rho, \sigma) \lesssim \theta. \tag{3.140}$$

This quadratic difference in scaling suggests that while the different distance metrics are related, they can behave very differently.

Example 3.48. Consider a target state $\rho = |0\rangle\langle 0|$. Let $\theta \in [0, \pi/2]$ and define two pure states:

$$|\theta_+\rangle = \cos\theta\,|0\rangle + \sin\theta\,|1\rangle\,, \quad |\theta_-\rangle = \cos\theta\,|0\rangle - \sin\theta\,|1\rangle\,. \tag{3.141}$$

Let σ_+ and σ_- be the corresponding density operators. We also consider the mixed state $\sigma_M = \frac{1}{2}(\sigma_+ + \sigma_-)$.

$$\sigma_M = \cos^2\theta|0\rangle\langle 0| + \sin^2\theta|1\rangle\langle 1|. \tag{3.142}$$

We compare the fidelities and trace distances to the target state ρ. The fidelities are identical:

$$F(\rho, \sigma_+) = F(\rho, \sigma_-) = F(\rho, \sigma_M) = \cos\theta. \tag{3.143}$$

However, the trace distances differ significantly. For the pure states (using Example 3.44):

$$D(\rho, \sigma_\pm) = \sin\theta. \tag{3.144}$$

For the mixed state σ_M (using Example 3.38):

$$D(\rho, \sigma_M) = \sin^2\theta. \tag{3.145}$$

If θ is small, $D(\rho, \sigma_\pm) \approx \theta$ while $D(\rho, \sigma_M) \approx \theta^2$. The mixed state is quadratically closer to the target state in trace distance than its pure components, even though they all share the same fidelity. The coherent superpositions in σ_+ and σ_- (the off-diagonal terms) cancel out in the incoherent mixture σ_M, leading to a state that is statistically closer to ρ. ◇

Which measure, fidelity or trace distance, is more physically relevant? The answer depends on the context. Fidelity can often be estimated experimentally (e.g., via the SWAP test), while estimating the trace distance generally requires full quantum state tomography.

On the other hand, the trace distance directly bounds the difference in measurement statistics. According to Proposition 3.39, the maximum difference in the probability of any measurement outcome M is bounded by the trace distance:

$$|\mathrm{Tr}[M\rho] - \mathrm{Tr}[M\sigma]| \leq D(\rho, \sigma). \tag{3.146}$$

If the trace distance is small, the states are statistically indistinguishable by any measurement.

3.6. Distance between quantum channels

Quantifying the distance between quantum channels is important for analyzing the precision of quantum gates, the robustness of quantum algorithms, and the distinguishability of physical processes. This section introduces the primary tools used for this purpose: the induced trace norm and the diamond norm.

3.6.1. Induced trace norm. We begin by considering norms induced on the space of linear maps (superoperators) by the Schatten norms on the input and output spaces.

Definition 3.49. *For a linear map $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$, the **induced trace norm** (or the induced $1 \to 1$ norm) is defined as*

$$\|\mathcal{Q}\|_{1\to 1} := \sup_{X \in L(\mathbb{C}^N), \|X\|_1 \leq 1} \|\mathcal{Q}[X]\|_1\,. \tag{3.147}$$

This norm quantifies the maximum amplification of the trace norm under the action of $\mathcal{Q}$.

*Analogously, the **induced operator norm** (or the induced $\infty \to \infty$ norm) is defined using the operator norm $\|\cdot\|_\infty$:*

$$\|\mathcal{Q}\|_{\infty\to\infty} := \sup_{X \in L(\mathbb{C}^N), \|X\|_\infty \leq 1} \|\mathcal{Q}[X]\|_\infty\,. \tag{3.148}$$

Induced norms are inherently submultiplicative, a property useful when analyzing compositions of maps.

Proposition 3.50 (Submultiplicativity)**.** *Let $\mathcal{R} : L(\mathbb{C}^N) \to L(\mathbb{C}^{N'})$ and $\mathcal{Q} : L(\mathbb{C}^{N'}) \to L(\mathbb{C}^M)$ be linear maps. Then*

$$\|\mathcal{Q} \circ \mathcal{R}\|_{1\to 1} \leq \|\mathcal{Q}\|_{1\to 1} \|\mathcal{R}\|_{1\to 1}. \tag{3.149}$$

Proof. For any $X \in L(\mathbb{C}^N)$, by the definition of the induced norm:

$$\|(\mathcal{Q} \circ \mathcal{R})(X)\|_1 = \|\mathcal{Q}[\mathcal{R}[X]]\|_1 \leq \|\mathcal{Q}\|_{1\to 1} \|\mathcal{R}[X]\|_1 \leq \|\mathcal{Q}\|_{1\to 1} \|\mathcal{R}\|_{1\to 1} \|X\|_1. \tag{3.150}$$

Taking the supremum over X with $\|X\|_1 \leq 1$ yields the result. □

To analyze these norms, we introduce the concept of the adjoint map. The space of linear operators $L(\mathbb{C}^N)$ forms a Hilbert space under the Hilbert-Schmidt inner product $\langle A, B\rangle = \mathrm{Tr}(A^\dagger B)$. The **adjoint map** $\mathcal{Q}^\dagger : L(\mathbb{C}^M) \to L(\mathbb{C}^N)$ is uniquely defined by the relation

$$\langle Y, \mathcal{Q}(X)\rangle = \langle \mathcal{Q}^\dagger(Y), X\rangle, \tag{3.151}$$

for all $X \in L(\mathbb{C}^N)$ and $Y \in L(\mathbb{C}^M)$.

The induced trace norm and the induced operator norm exhibit a duality relationship analogous to the duality between the trace norm and operator norm for matrices (Lemma 3.35).

Proposition 3.51 (Duality of Induced Norms)**.** *For any linear map $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$, the following duality relation holds:*

$$\|\mathcal{Q}\|_{1\to 1} = \left\|\mathcal{Q}^\dagger\right\|_{\infty\to\infty}. \tag{3.152}$$

Proof. We begin with the definition of the induced trace norm and apply the variational characterization of the trace norm (Lemma 3.35, Eq. (3.101)):

$$\begin{aligned} \|\mathcal{Q}\|_{1\to 1} &= \sup_{\|X\|_1 \leq 1} \|\mathcal{Q}(X)\|_1 \\ &= \sup_{\|X\|_1 \leq 1} \left(\sup_{\|Y\|_\infty \leq 1} |\,\mathrm{Tr}(Y^\dagger \mathcal{Q}[X])| \right). \end{aligned} \tag{3.153}$$

We exchange the order of the suprema and employ the definition of the adjoint map (Eq. (3.151)):

$$\|\mathcal{Q}\|_{1\to 1} = \sup_{\|Y\|_\infty \leq 1} \left(\sup_{\|X\|_1 \leq 1} |\,\mathrm{Tr}((\mathcal{Q}^\dagger(Y))^\dagger X)| \right). \tag{3.154}$$

The inner supremum is the characterization of the operator norm via duality (Lemma 3.35, Eq. (3.102)), applied to the operator $W = \mathcal{Q}^\dagger(Y)$. That is, $\sup_{\|X\|_1 \leq 1} |\,\mathrm{Tr}(W^\dagger X)| = \|W\|_\infty$.

$$\|\mathcal{Q}\|_{1\to 1} = \sup_{\|Y\|_\infty \leq 1} \left\|\mathcal{Q}^\dagger(Y)\right\|_\infty = \left\|\mathcal{Q}^\dagger\right\|_{\infty\to\infty}. \tag{3.155}$$

□

To compute the induced trace norm, it is helpful to characterize the inputs that achieve the maximum. We first establish that for general linear maps, the maximum is attained on rank-1 operators.

Lemma 3.52. *For any linear map $\mathcal{Q}$, the induced $1 \to 1$ norm is achieved by a rank-1 operator:*

$$\|\mathcal{Q}\|_{1\to 1} = \sup\{\|\mathcal{Q}(|u\rangle\langle v|)\|_1 : \|u\|_2 = 1, \|v\|_2 = 1\}. \tag{3.156}$$

PROOF. Let $C_1 = \{X : \|X\|_1 \leq 1\}$ be the unit ball in the trace norm. The function $f(X) = \|\mathcal{Q}(X)\|_1$ is convex. Since C_1 is a compact, convex set, the maximum of $f(X)$ over C_1 must be achieved at an extreme point of C_1. The extreme points of C_1 are precisely the rank-1 operators of the form $|u\rangle\langle v|$ with normalized vectors $|u\rangle, |v\rangle$.

Explicitly, let X maximize the norm, with $\|X\|_1 = 1$. Its SVD $X = \sum_i s_i |u_i\rangle\langle v_i|$ is a convex combination (since $\sum s_i = 1, s_i > 0$) of the rank-1 operators $X_i = |u_i\rangle\langle v_i|$. By the triangle inequality:

$$\|\mathcal{Q}(X)\|_1 = \left\| \sum_i s_i \mathcal{Q}(X_i) \right\|_1 \leq \sum_i s_i \|\mathcal{Q}(X_i)\|_1 \leq \max_i \|\mathcal{Q}(X_i)\|_1 . \tag{3.157}$$

Thus, the maximum is achieved by one of the rank-1 operators X_i. □

We now investigate how these norms behave for positive maps. We first state the following result for positive maps without proof [Wat18, Eq. (3.329)].

Lemma 3.53. *Let $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ be a positive linear map. Then*

$$\|\mathcal{Q}\|_{1\to 1} = \left\|\mathcal{Q}^\dagger(I_M)\right\|_\infty . \tag{3.158}$$

A celebrated result known as the Russo–Dye theorem [Wat18, Theorem 3.39] simplifies the calculation of the induced norm for such maps.

THEOREM 3.54 (Russo–Dye). *Let $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ be a positive linear map. Then*

$$\|\mathcal{Q}\|_{1\to 1} = \max_{\|u\|_2 = 1} \mathrm{Tr}\left(\mathcal{Q}\left(|u\rangle\langle u|\right)\right). \tag{3.159}$$

PROOF. Since $\mathcal{Q}^\dagger(I_M)$ is Hermitian (in fact positive semidefinite), its operator norm is the largest eigenvalue:

$$\left\|\mathcal{Q}^\dagger(I_M)\right\|_\infty = \sup_{\|u\|_2=1} \langle u| \mathcal{Q}^\dagger(I_M) |u\rangle = \sup_{\|u\|_2=1} \mathrm{Tr}(\mathcal{Q}(|u\rangle\langle u|)). \tag{3.160}$$

The result follows from Lemma 3.53. □

As an immediate consequence, if $\mathcal{Q}$ is a quantum channel, it is positive and trace-preserving. Thus,

$$\|\mathcal{Q}\|_{1\to 1} = \max_u \mathrm{Tr}\left(\mathcal{Q}\left(|u\rangle\langle u|\right)\right) = \max_u \mathrm{Tr}\left(|u\rangle\langle u|\right) = 1. \tag{3.161}$$

The fact that quantum channels have an induced trace norm of 1 leads to an important stability property for compositions of channels.

Proposition 3.55. *Let $\mathcal{Q}_1, \ldots, \mathcal{Q}_K$ and $\widetilde{\mathcal{Q}}_1, \ldots, \widetilde{\mathcal{Q}}_K$ be sequences of quantum channels. Then*

$$\left\| \mathcal{Q}_K \circ \cdots \circ \mathcal{Q}_1 - \widetilde{\mathcal{Q}}_K \circ \cdots \circ \widetilde{\mathcal{Q}}_1 \right\|_{1\to 1} \leq \sum_{i=1}^K \left\| \mathcal{Q}_i - \widetilde{\mathcal{Q}}_i \right\|_{1\to 1} . \tag{3.162}$$

PROOF. We use a telescoping sum argument. For $K = 2$:

$$\mathcal{Q}_2 \circ \mathcal{Q}_1 - \widetilde{\mathcal{Q}}_2 \circ \widetilde{\mathcal{Q}}_1 = (\mathcal{Q}_2 - \widetilde{\mathcal{Q}}_2) \circ \mathcal{Q}_1 + \widetilde{\mathcal{Q}}_2 \circ (\mathcal{Q}_1 - \widetilde{\mathcal{Q}}_1). \tag{3.163}$$

By the triangle inequality and submultiplicativity (Proposition 3.50):

$$\begin{aligned} \left\| \mathcal{Q}_2 \circ \mathcal{Q}_1 - \widetilde{\mathcal{Q}}_2 \circ \widetilde{\mathcal{Q}}_1 \right\|_{1\to 1} \leq & \left\| \mathcal{Q}_2 - \widetilde{\mathcal{Q}}_2 \right\|_{1\to 1} \|\mathcal{Q}_1\|_{1\to 1} \\ & + \left\| \widetilde{\mathcal{Q}}_2 \right\|_{1\to 1} \left\| \mathcal{Q}_1 - \widetilde{\mathcal{Q}}_1 \right\|_{1\to 1} . \end{aligned} \tag{3.164}$$

Since $\mathcal{Q}_1$ and $\widetilde{\mathcal{Q}}_2$ are quantum channels, their induced trace norms are 1.

$$\left\|\mathcal{Q}_2 \circ \mathcal{Q}_1 - \widetilde{\mathcal{Q}}_2 \circ \widetilde{\mathcal{Q}}_1\right\|_{1\to 1} \leq \left\|\mathcal{Q}_2 - \widetilde{\mathcal{Q}}_2\right\|_{1\to 1} + \left\|\mathcal{Q}_1 - \widetilde{\mathcal{Q}}_1\right\|_{1\to 1}. \tag{3.165}$$

The general case follows by induction. □

3.6.2. The diamond norm. The induced trace norm quantifies how much a map $\mathcal{Q}$ acting on a system S changes the state of S. However, this is insufficient in quantum mechanics due to entanglement. If S is entangled with an auxiliary system A, the action of $\mathcal{Q}$ on S (described by $\mathcal{Q} \otimes \mathcal{I}_A$) might alter the joint state of SA significantly more than predicted by $\|\mathcal{Q}\|_{1\to 1}$. To capture the true behavior of the map in the presence of arbitrary entanglement, we must consider its stabilized action. This leads to the **diamond norm**, also known as the **completely bounded trace norm**.

Definition 3.56 (Diamond Norm)**.** *Let $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$ be a linear map. The diamond norm of $\mathcal{Q}$ is defined as*

$$\|\mathcal{Q}\|_\diamond := \sup_{k\geq 1} \|\mathcal{Q} \otimes \mathcal{I}_k\|_{1\to 1} = \sup_{k\geq 1} \|\mathcal{I}_k \otimes \mathcal{Q}\|_{1\to 1}, \tag{3.166}$$

where $\mathcal{I}_k$ denotes the identity map on $L(\mathbb{C}^k)$.

If $\mathcal{Q}$ is a quantum channel, then for every k the map $\mathcal{Q} \otimes \mathcal{I}_k$ is also a quantum channel, and hence has induced trace norm 1. Therefore,

$$\|\mathcal{Q}\|_\diamond = \sup_{k\geq 1} \|\mathcal{Q} \otimes \mathcal{I}_k\|_{1\to 1} = 1. \tag{3.167}$$

While the definition involves a supremum over all dimensions k, a remarkable result shows that the supremum is achieved when the auxiliary dimension matches the input dimension of the map.

Proposition 3.57 (Stabilization of the Diamond Norm)**.** *For any linear map $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^M)$, the supremum in Eq.* (3.166) *is achieved for $k = N$. That is,*

$$\|\mathcal{Q}\|_\diamond = \|\mathcal{Q} \otimes \mathcal{I}_N\|_{1\to 1}. \tag{3.168}$$

Proof. We aim to show that for any $k \geq 1$, $\|\mathcal{Q} \otimes \mathcal{I}_k\|_{1\to 1} \leq \|\mathcal{Q} \otimes \mathcal{I}_N\|_{1\to 1}$.

Let $k \geq 1$. By Lemma 3.52, the induced norm is achieved by a rank-1 input. There exist normalized vectors $|\alpha\rangle, |\beta\rangle \in \mathbb{C}^N \otimes \mathbb{C}^k$ such that

$$\|\mathcal{Q} \otimes \mathcal{I}_k\|_{1\to 1} = \|(\mathcal{Q} \otimes \mathcal{I}_k)(|\alpha\rangle\langle\beta|)\|_1. \tag{3.169}$$

Consider the Schmidt decompositions of $|\alpha\rangle$ and $|\beta\rangle$. The Schmidt ranks r, s are at most N.

$$|\alpha\rangle = \sum_{i=1}^{r} \sqrt{p_i}\,|a_i\rangle \otimes |x_i\rangle, \quad |\beta\rangle = \sum_{j=1}^{s} \sqrt{q_j}\,|b_j\rangle \otimes |y_j\rangle. \tag{3.170}$$

Here, $\{|a_i\rangle\}, \{|b_j\rangle\} \subset \mathbb{C}^N$ and $\{|x_i\rangle\}, \{|y_j\rangle\} \subset \mathbb{C}^k$ are orthonormal sets. Let $Y = (\mathcal{Q} \otimes \mathcal{I}_k)(|\alpha\rangle\langle\beta|)$.

$$Y = \sum_{i,j} \sqrt{p_i q_j}\,\mathcal{Q}(|a_i\rangle\langle b_j|) \otimes |x_i\rangle\langle y_j|. \tag{3.171}$$

We construct corresponding vectors in $\mathbb{C}^N \otimes \mathbb{C}^N$. Let $\{|e_i\rangle\}_{i=1}^N$ be a basis for $\mathbb{C}^N$. Define normalized vectors $|\alpha'\rangle, |\beta'\rangle \in \mathbb{C}^N \otimes \mathbb{C}^N$ by replacing $|x_i\rangle$ with $|e_i\rangle$ and $|y_j\rangle$ with $|e_j\rangle$. Let $Y' = (\mathcal{Q} \otimes \mathcal{I}_N)(|\alpha'\rangle\langle\beta'|)$.

$$Y' = \sum_{i,j} \sqrt{p_i q_j}\,\mathcal{Q}(|a_i\rangle\langle b_j|) \otimes |e_i\rangle\langle e_j|. \tag{3.172}$$

We show that $\|Y\|_1 = \|Y'\|_1$. Define partial isometries $V, W : \mathbb{C}^N \to \mathbb{C}^k$. Let V map $\text{span}\{|e_i\rangle\}_{i=1}^r$ isometrically onto $\text{span}\{|x_i\rangle\}_{i=1}^r$, and similarly for W and $\{|y_j\rangle\}$. We can relate Y and Y':

$$Y = (I_M \otimes V)Y'(I_M \otimes W^\dagger). \tag{3.173}$$

Extend V and W to unitaries $\widetilde{V}, \widetilde{W}$ on $\mathbb{C}^k$ (by choosing orthonormal complements). Since Y' only has support on the subspaces where V and W act isometrically, we have

$$Y = (I_M \otimes \widetilde{V})Y'(I_M \otimes \widetilde{W}^\dagger). \tag{3.174}$$

By unitary invariance of the trace norm, $\|Y\|_1 = \|Y'\|_1$.

We have established $\|\mathcal{Q} \otimes \mathcal{I}_k\|_{1\to 1} = \|Y'\|_1$. Since $\||\alpha'\rangle\langle\beta'|\|_1 = 1$, we have $\|Y'\|_1 \leq \|\mathcal{Q} \otimes \mathcal{I}_N\|_{1\to 1}$. This completes the proof. □

The diamond norm inherits the submultiplicativity property from the induced trace norm.

Proposition 3.58 (Submultiplicativity of the Diamond Norm)**.** *Let $\mathcal{R} : L(\mathbb{C}^N) \to L(\mathbb{C}^{N'})$ and $\mathcal{Q} : L(\mathbb{C}^{N'}) \to L(\mathbb{C}^M)$ be linear maps. Then*

$$\|\mathcal{Q} \circ \mathcal{R}\|_\diamond \leq \|\mathcal{Q}\|_\diamond \|\mathcal{R}\|_\diamond . \tag{3.175}$$

Proof. We use the definition of the diamond norm and the property that $(\mathcal{Q} \circ \mathcal{R}) \otimes \mathcal{I}_k = (\mathcal{Q} \otimes \mathcal{I}_k) \circ (\mathcal{R} \otimes \mathcal{I}_k)$.

$$\|\mathcal{Q} \circ \mathcal{R}\|_\diamond = \sup_k \|(\mathcal{Q} \otimes \mathcal{I}_k) \circ (\mathcal{R} \otimes \mathcal{I}_k)\|_{1\to 1} . \tag{3.176}$$

By the submultiplicativity of the induced trace norm (Proposition 3.50):

$$\begin{aligned} \|\mathcal{Q} \circ \mathcal{R}\|_\diamond &\leq \sup_k \left(\|\mathcal{Q} \otimes \mathcal{I}_k\|_{1\to 1} \|\mathcal{R} \otimes \mathcal{I}_k\|_{1\to 1}\right) \\ &\leq \left(\sup_k \|\mathcal{Q} \otimes \mathcal{I}_k\|_{1\to 1}\right)\left(\sup_k \|\mathcal{R} \otimes \mathcal{I}_k\|_{1\to 1}\right) \\ &= \|\mathcal{Q}\|_\diamond \|\mathcal{R}\|_\diamond . \end{aligned} \tag{3.177}$$

□

We can derive useful bounds on the diamond norm for specific types of maps. We start with maps defined by a single Kraus operator.

Lemma 3.59. *Let $\mathcal{Q}_A(X) = AXA^\dagger$ and $\mathcal{Q}_B(X) = BXB^\dagger$. Then the diamond norm of their difference is bounded by*

$$\|\mathcal{Q}_A - \mathcal{Q}_B\|_\diamond \leq (\|A\|_\infty + \|B\|_\infty) \|A - B\|_\infty . \tag{3.178}$$

Proof. Let $\Phi = \mathcal{Q}_A - \mathcal{Q}_B$. By stabilization (Proposition 3.57), we evaluate $\|\Phi \otimes \mathcal{I}_N\|_{1\to 1}$. Let X_{SR} be an input operator with $\|X_{SR}\|_1 = 1$.

$$(\Phi \otimes \mathcal{I}_N)(X_{SR}) = (A \otimes I)X_{SR}(A^\dagger \otimes I) - (B \otimes I)X_{SR}(B^\dagger \otimes I). \tag{3.179}$$

We use the identity $AA^\dagger - BB^\dagger = A(A^\dagger - B^\dagger) + (A - B)B^\dagger$.

$$\begin{aligned} (\Phi \otimes \mathcal{I}_N)(X_{SR}) = &(A \otimes I)X_{SR}((A^\dagger - B^\dagger) \otimes I) \\ &+ ((A - B) \otimes I)X_{SR}(B^\dagger \otimes I). \end{aligned} \tag{3.180}$$

We bound the trace norm using the triangle inequality and Hölder's inequality ($\|Y_1 X Y_2\|_1 \leq \|Y_1\|_\infty \|X\|_1 \|Y_2\|_\infty$). Since $\|X_{SR}\|_1 = 1$ and $\|Y \otimes I\|_\infty = \|Y\|_\infty$:

$$\|(\Phi \otimes \mathcal{I}_N)(X_{SR})\|_1 \leq \|A\|_\infty \left\|A^\dagger - B^\dagger\right\|_\infty + \|A - B\|_\infty \left\|B^\dagger\right\|_\infty . \tag{3.181}$$

Using $\left\|A^\dagger - B^\dagger\right\|_\infty = \|A - B\|_\infty$ and $\left\|B^\dagger\right\|_\infty = \|B\|_\infty$, we obtain the bound. □

Example 3.60 (Distance between unitary channels)**.** Consider unitary channels $\mathcal{U}(X) = UXU^\dagger$ and $\mathcal{V}(X) = VXV^\dagger$. Since $\|U\|_\infty = \|V\|_\infty = 1$, Lemma 3.59 yields the bound:

$$\|\mathcal{U} - \mathcal{V}\|_\diamond \leq 2 \|U - V\|_\infty . \tag{3.182}$$

◇

While the bound in Eq. (3.182) is widely used, it is not always tight. Furthermore, one might expect that stabilization is necessary for unitary channels. However, the difference between unitary channels exhibits a special structure that renders stabilization unnecessary.

Proposition 3.61. *Let $\mathcal{U}, \mathcal{V}$ be two unitary channels defined by unitaries U and V. Then the diamond norm of their difference is equal to the induced trace norm:*

$$\|\mathcal{U} - \mathcal{V}\|_\diamond = \|\mathcal{U} - \mathcal{V}\|_{1\to 1} . \tag{3.183}$$

This norm can be computed explicitly using the numerical range of $W = U^\dagger V$:

$$\|\mathcal{U} - \mathcal{V}\|_\diamond = 2\sqrt{1 - d_{\min}^2}, \tag{3.184}$$

where $d_{\min} = \inf\{|z| : z \in \mathcal{W}(W)\}$ is the minimum distance from the origin to the numerical range $\mathcal{W}(W) = \{\langle x| W |x\rangle : \|x\|_2 = 1\}$.

Proof. Let $\Phi = \mathcal{U} - \mathcal{V}$. We first establish a lower bound for the induced trace norm $\|\Phi\|_{1\to 1}$. According to Lemma 3.52, the induced trace norm is defined by the supremum over rank-1 inputs. Restricting the optimization to pure states $\rho = |x\rangle\langle x|$ yields a lower bound:

$$\|\Phi\|_{1\to 1} \geq \sup_{|x\rangle} \|\Phi(|x\rangle\langle x|)\|_1 . \tag{3.185}$$

The output is

$$\Phi(|x\rangle\langle x|) = U|x\rangle\langle x|U^\dagger - V|x\rangle\langle x|V^\dagger = |\psi_U\rangle\langle\psi_U| - |\psi_V\rangle\langle\psi_V|, \tag{3.186}$$

where $|\psi_U\rangle = U|x\rangle$ and $|\psi_V\rangle = V|x\rangle$. The trace norm of the difference between two pure states is determined by their overlap (see Example 3.44):

$$\||\psi_U\rangle\langle\psi_U| - |\psi_V\rangle\langle\psi_V|\|_1 = 2\sqrt{1 - |\langle\psi_U|\psi_V\rangle|^2}. \tag{3.187}$$

The overlap is $\langle\psi_U|\psi_V\rangle = \langle x| U^\dagger V |x\rangle = \langle x| W |x\rangle$. To maximize the norm, we must minimize the magnitude of the overlap. The set of values $\{\langle x| W |x\rangle : \|x\|_2 = 1\}$ is the numerical range $\mathcal{W}(W)$. Thus, the supremum over pure states is

$$2\sqrt{1 - \inf_{z\in\mathcal{W}(W)} |z|^2} = 2\sqrt{1 - d_{\min}^2}. \tag{3.188}$$

Next, we consider the diamond norm $\|\Phi\|_\diamond$. By the stabilization property (Proposition 3.57), $\|\Phi\|_\diamond = \|\Phi \otimes \mathcal{I}_N\|_{1\to 1}$. Unlike the induced trace norm, the diamond norm is achieved on pure states (see [Wat18, Theorem 3.51]). Let $|\Psi\rangle \in \mathbb{C}^N \otimes \mathbb{C}^N$ be a normalized pure state. The action of the map

on $\rho = |\Psi\rangle\langle\Psi|$ yields the difference of two pure states $|\Psi_U\rangle = (U \otimes I)|\Psi\rangle$ and $|\Psi_V\rangle = (V \otimes I)|\Psi\rangle$. The norm is again given by $2\sqrt{1 - |\langle\Psi_U|\Psi_V\rangle|^2}$. The overlap is

$$\langle\Psi_U|\Psi_V\rangle = \langle\Psi|(U^\dagger \otimes I)(V \otimes I)|\Psi\rangle = \langle\Psi|(W \otimes I)|\Psi\rangle. \tag{3.189}$$

We express this overlap in terms of the reduced density operator $\rho_A = \mathrm{Tr}_B[|\Psi\rangle\langle\Psi|]$:

$$\langle\Psi|(W \otimes I)|\Psi\rangle = \mathrm{Tr}[(W \otimes I)|\Psi\rangle\langle\Psi|] = \mathrm{Tr}[W\rho_A]. \tag{3.190}$$

As $|\Psi\rangle$ varies over all pure states in the joint space, ρ_A varies over all density operators in $\mathcal{D}(\mathbb{C}^N)$. The set of achievable overlaps is therefore the set of expectation values $\{\mathrm{Tr}[W\rho] : \rho \in \mathcal{D}(\mathbb{C}^N)\}$. This set is the convex hull of the numerical range $\mathcal{W}(W)$. By the Toeplitz–Hausdorff theorem (see [Bha97, Chapter 1]), the numerical range $\mathcal{W}(W)$ is a convex set. Therefore, the convex hull of $\mathcal{W}(W)$ is $\mathcal{W}(W)$ itself. This implies that allowing entanglement does not extend the range of possible overlaps:

$$\inf_{\||\Psi\rangle\|_2=1} |\langle\Psi|(W \otimes I)|\Psi\rangle| = \inf_{z \in \mathcal{W}(W)} |z| = d_{\min}. \tag{3.191}$$

Consequently,

$$\|\Phi\|_\diamond = 2\sqrt{1 - d_{\min}^2}. \tag{3.192}$$

Combining this with Eq. (3.188) and the inequality $\|\Phi\|_{1\to1} \leq \|\Phi\|_\diamond$, we conclude $\|\Phi\|_\diamond = \|\Phi\|_{1\to1}$. □

Example 3.62. Consider the 2×2 unitaries $U = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ and $V = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. We calculate the operator norm of their difference:

$$U - V = \begin{pmatrix} -1 & 1 \\ -1 & -1 \end{pmatrix}. \tag{3.193}$$

The singular values are the square roots of the eigenvalues of $(U-V)^\dagger(U-V) = \mathrm{diag}(2,2)$. Thus, $\|U - V\|_\infty = \sqrt{2}$. The general bound in Eq. (3.182) gives $\|\mathcal{U} - \mathcal{V}\|_\diamond \leq 2\sqrt{2} \approx 2.828$.

However, as $W = U^\dagger = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$, the eigenvalues of W are i and $-i$. Since W is normal, $\mathcal{W}(W)$ is the convex hull of the eigenvalues, i.e., the segment $[-i, i]$ on the imaginary axis. The minimum distance to the origin is $d_{\min} = 0$. Thus, the exact diamond norm is $2\sqrt{1 - 0^2} = 2$. ⋄

Example 3.63 (Qubit Phase Shift Channel)**.** We illustrate the computation using a single-qubit example. Consider the identity channel $\mathcal{I}$ ($U = I$) and the phase shift channel $\mathcal{P}_\theta$, defined by the unitary $V = P_\theta = \mathrm{diag}(1, e^{i\theta})$. We wish to compute $\|\mathcal{I} - \mathcal{P}_\theta\|_\diamond$.

We apply Proposition 3.61. We compute $W = U^\dagger V = P_\theta$. We need to determine the numerical range $\mathcal{W}(P_\theta)$. Since P_θ is a normal operator, its numerical range is the convex hull of its eigenvalues, $\{1, e^{i\theta}\}$. This is the line segment (chord) connecting 1 and $e^{i\theta}$ in the complex plane.

We seek the minimum distance $d_{\min}$ from the origin to this segment. Geometrically, this distance is the altitude of the isosceles triangle formed by the origin and the two eigenvalues.

The length of the base of the triangle (the chord) is $|1 - e^{i\theta}| = \sqrt{(1 - \cos\theta)^2 + \sin^2\theta} = \sqrt{2 - 2\cos\theta} = 2|\sin(\theta/2)|$. The area of the triangle is $\frac{1}{2}|\sin\theta|$. Let h be the altitude, which corresponds to $d_{\min}$. The area is also $\frac{1}{2} \cdot \text{base} \cdot h$.

$$d_{\min} = h = \frac{|\sin\theta|}{2|\sin(\theta/2)|} = \frac{2|\sin(\theta/2)\cos(\theta/2)|}{2|\sin(\theta/2)|} = |\cos(\theta/2)|. \tag{3.194}$$

Substituting this minimum value into Eq. (3.184):

$$\|\mathcal{I} - \mathcal{P}_\theta\|_\diamond = 2\sqrt{1-\cos^2(\theta/2)} = 2\sqrt{\sin^2(\theta/2)} = 2|\sin(\theta/2)|. \tag{3.195}$$

By Proposition 3.61, the induced trace norm is identical: $\|\mathcal{I} - \mathcal{P}_\theta\|_{1\to 1} = 2|\sin(\theta/2)|$.

For instance, if $\theta = \pi$, the channel is the Pauli-Z channel $\mathcal{Z}$. The diamond norm is $2|\sin(\pi/2)| = 2$. The minimum overlap is $d_{\min} = 0$. This is achieved by the input state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, since $\langle +| Z |+\rangle = 0$. ◇

The following example illustrates that the standard induced trace norm can drastically underestimate the "size" of a map that is not completely positive.

Example 3.64 (Transpose Map)**.** Let $\mathcal{T} : \mathbb{C}^{N\times N} \to \mathbb{C}^{N\times N}$ be the transpose map, $\mathcal{T}(X) = X^\top$, defined in a fixed basis. Since the transpose preserves the eigenvalues of Hermitian matrices and maps density matrices to density matrices, it preserves the 1-norm for positive inputs. It can be shown that $\|\mathcal{T}\|_{1\to 1} = 1$.

However, consider the action of $\mathcal{T} \otimes \mathcal{I}_N$ on the unnormalized maximally entangled state $|\Omega\rangle = \sum_{i=1}^N |i\rangle \otimes |i\rangle$. The corresponding density matrix is $\omega = \sum_{i,j} |i\rangle\langle j| \otimes |i\rangle\langle j|$. Applying the partial transpose yields

$$(\mathcal{T} \otimes \mathcal{I}_N)(\omega) = \sum_{i,j} |j\rangle\langle i| \otimes |i\rangle\langle j|, \tag{3.196}$$

which is the SWAP operator. The eigenvalues of the SWAP operator on $\mathbb{C}^N \otimes \mathbb{C}^N$ are $+1$ (on the symmetric subspace of dimension $N(N+1)/2$) and -1 (on the antisymmetric subspace of dimension $N(N-1)/2$). The trace norm is the sum of singular values (absolute values of eigenvalues):

$$\|(\mathcal{T} \otimes \mathcal{I}_N)(\omega)\|_1 = \frac{N(N+1)}{2} + \frac{N(N-1)}{2} = N^2. \tag{3.197}$$

Since $\|\omega\|_1 = \||\Omega\rangle\|^2 = N$, we find that for this specific state, the ratio of output norm to input norm is N. Thus $\|\mathcal{T}\|_\diamond \geq N$. ◇

3.6.3. Induced trace distance and diamond distance. The **induced trace distance** between two linear maps $\mathcal{Q}, \mathcal{R}$ is

$$D(\mathcal{Q}, \mathcal{R}) := \frac{1}{2}\|\mathcal{Q} - \mathcal{R}\|_{1\to 1}. \tag{3.198}$$

Example 3.65 (Trace distance for classical channel)**.** Given two transition matrices $Q, R \in \mathbb{R}^{N\times N}$, the corresponding classical channels are

$$\mathcal{Q}[\rho] = \sum_{i,j\in[N]} Q_{ij}|i\rangle\langle j|\rho|j\rangle\langle i|, \quad \mathcal{R}[\rho] = \sum_{i,j\in[N]} R_{ij}|i\rangle\langle j|\rho|j\rangle\langle i|. \tag{3.199}$$

Then

$$\begin{aligned} D(\mathcal{Q},\mathcal{R}) =& \frac{1}{2}\sup_{\|\rho\|_1=1}\|\mathcal{Q}[\rho]-\mathcal{R}[\rho]\|_1 \\ =& \frac{1}{2}\sup_{\|\rho\|_1=1}\sum_i\left|\sum_j (Q_{ij}-R_{ij})\rho_{jj}\right| \\ \leq& \frac{1}{2}\sup_{\|\rho\|_1=1}\left(\max_j\sum_i|Q_{ij}-R_{ij}|\right)\operatorname{Tr}|\rho| \\ \leq& \frac{1}{2}\sup_{\|\rho\|_1=1}\left(\max_j\sum_i|Q_{ij}-R_{ij}|\right)\|\rho\|_1 \\ =& D(Q,R), \end{aligned} \tag{3.200}$$

which is the induced total variation distance between the transition matrices Q, R. Here we have used Proposition 3.31 in the last inequality. On the other hand, choosing $\rho = |j'\rangle\langle j'|$ with $j' = \arg\max_j \sum_i |Q_{ij} - R_{ij}|$, we have $D(\mathcal{Q},\mathcal{R}) \geq D(Q,R)$. This proves that the induced trace distance is consistent with the induced total variation distance on classical channels:

$$D(\mathcal{Q},\mathcal{R}) = D(Q,R). \tag{3.201}$$

◇

The metric induced by the diamond norm is known as the diamond distance. The factor of $1/2$ normalizes the metric such that perfectly distinguishable channels have a distance of 1, analogous to the trace distance for quantum states.

Definition 3.66 (Diamond Distance)**.** *Let $\mathcal{Q},\mathcal{R} : \mathbb{C}^{N\times N} \to \mathbb{C}^{M\times M}$ be two linear maps. The **diamond distance** between them is defined as*

$$D_\diamond(\mathcal{Q},\mathcal{R}) := \frac{1}{2}\|\mathcal{Q}-\mathcal{R}\|_\diamond. \tag{3.202}$$

Quantum channels satisfy the linear error growth property with respect to the diamond distance. The proof is also very similar to Proposition 3.55.

Proposition 3.67. *Let $\{\mathcal{U}_i\}_{i=1}^K$ and $\{\widetilde{\mathcal{U}}_i\}_{i=1}^K$ be sequences of unitary channels generated by the unitary operators $\{U_i\}_{i=1}^K$ and $\{\widetilde{U}_i\}_{i=1}^K$, respectively. The diamond distance between the composite channels is bounded by*

$$D_\diamond(\mathcal{U}_K\cdots\mathcal{U}_1, \widetilde{\mathcal{U}}_K\cdots\widetilde{\mathcal{U}}_1) \leq \sum_{i=1}^K \left\|U_i-\widetilde{U}_i\right\|_\infty. \tag{3.203}$$

Proof. First, we observe that quantum channels satisfy a linear error growth property with respect to the diamond distance. The proof of this property relies on a telescoping sum argument, which is strictly analogous to the proof of Proposition 3.55 and is therefore omitted. This yields the bound

$$D_\diamond(\mathcal{U}_K\cdots\mathcal{U}_1, \widetilde{\mathcal{U}}_K\cdots\widetilde{\mathcal{U}}_1) \leq \frac{1}{2}\sum_{i=1}^K \left\|\widetilde{\mathcal{U}}_i-\mathcal{U}_i\right\|_\diamond. \tag{3.204}$$

It suffices to bound the diamond norm difference for a single step. Recalling Eq. (3.182), we have the general bound $\left\|\widetilde{\mathcal{U}}_i - \mathcal{U}_i\right\|_\diamond \leq 2\left\|U_i - \widetilde{U}_i\right\|$. Substituting this estimate into the linear error growth inequality completes the proof. □

Notes and further reading

The formalism of quantum channels rests on foundational results in operator theory. The operator-sum representation (Theorem 3.18) is due to Kraus [KBDW83], while the dilation theorem (Theorem 3.20) was established by Stinespring [Sti55]. The isomorphism characterizing completely positive maps via their action on entangled states is attributed to Choi [Cho75] and Jamiołkowski [Jam72].

The induced trace distance provides a useful way to compare two channels via their action on input states. It is worth noting that the contractivity properties of the trace distance used in this context rely on positivity and trace preservation, and do not require complete positivity. By contrast, complete positivity is required to ensure that a channel remains positive when extended by an identity map on an arbitrary ancillary register. This distinction becomes operationally visible in the channel discrimination task: for some pairs of channels, optimal discrimination is only possible when the input is entangled with an ancillary register. This motivates the use of stabilized distances such as the diamond norm (the completely bounded trace norm), which explicitly accounts for ancillary extensions. For distance measures, Helstrom [Hel69] provided the operational interpretation of the trace distance in terms of state discrimination. Fidelity was studied by Uhlmann [Uhl76] as transition probability. The tight relationship between these two measures (Theorem 3.46) was established by Fuchs and van de Graaf [FVDG02]. The diamond norm was introduced to quantum computing by Kitaev [Kit97] to quantify the accuracy of quantum gates in a manner robust to entanglement, and is closely related to the completely bounded norm in operator algebra. We refer readers to [Wat18, Chapter 3.3] for further discussion.

Most of the discussions in this book will be restricted to unitary channels, and these unitary channels are often applied to pure states. Nevertheless, the concept of a quantum channel is helpful for understanding the probabilistic nature of quantum algorithms. For a systematic treatment of density operators and quantum channels, we refer readers to [Wat18, Chapter 2] and [NC00, Section 2.4, 8.2]. We refer readers to [Wat18, Chapter 3] for properties of the norms and distances introduced here, and their applications in discrimination-type problems. For matrix analysis tools, such as Schatten norms, we refer to [Bha97].

CHAPTER 4

Universality of quantum circuits

CHAPTER 5

Quantum processing of classical information

Quantum algorithms often require classical data to be loaded, processed, and manipulated within a quantum circuit. This chapter explores how classical information can be encoded and operated on in a quantum computing framework. We begin with the reversible simulation of classical logic gates, a prerequisite for embedding classical computation into quantum circuits. We then discuss uncomputation, which is very useful for cleaning up intermediate states without disturbing the computation's outcome. The chapter proceeds to cover fixed-point number representation and quantum random access memory (QRAM). Finally, we present methods for implementing certain classical arithmetic operations within quantum circuits.

5.1. Reversible simulation of classical gates

How can we compare the computational power of quantum computers to that of classical computers? While it remains extremely difficult to prove that quantum computers are fundamentally more powerful than classical ones, it is well established that quantum computers are at least as powerful. More precisely, any classical circuit can be simulated asymptotically efficiently by a quantum circuit.

The key idea is that all classical gates can be simulated in a reversible way. Some classical logic gates, such as the NOT gate, are already reversible and can be directly implemented by the Pauli X gate. However, many commonly used gates, including AND, OR, and NAND, are not reversible and cannot be directly translated into unitary transformations.

Reversible computation, which predates quantum computing, was originally studied in the context of thermodynamics and the fundamental limits of energy dissipation [Lan61]. In this model of computation, each operation can be reversed, and information is preserved throughout the process. To simulate arbitrary classical circuits in a reversible form, it is sufficient to construct reversible versions of universal gates such as the NAND gate. Once a reversible version of a universal gate is available, the entire classical computation can be lifted into a reversible framework, which can then be embedded into a quantum circuit using unitary operations.

Example 5.1 (Toffoli is universal for classical computation)**.** All boolean logic can be implemented using only NAND gates. NAND and FANOUT (i.e., making a copy of a classical bit x) are together universal for classical computation. The Toffoli gate is a controlled-controlled-NOT gate, and with an ancilla initialized to $|0\rangle$ it computes x AND y into the target register. We can use the Toffoli gate to simulate NAND and FANOUT. Therefore the Toffoli gate is universal for classical computation.

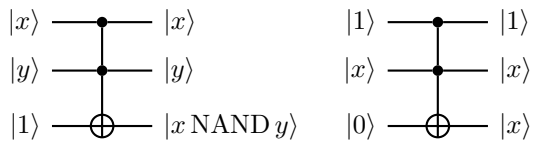


FIGURE 5.1. Using the Toffoli gate to implement NAND and FANOUT

◇

Exercise 5.1. Give explicit expressions for using Toffoli gates to implement AND, NOT, XOR, and OR.

A classical computation procedure can be expressed as the evaluation of a boolean map $f : \{0,1\}^n \to \{0,1\}^m$, which may be irreversible. However, it can be made into a reversible classical gate

$$(z, x) \mapsto (z \oplus f(x), x). \tag{5.1}$$

In particular, $(0^m, x) \mapsto (f(x), x)$ is a reversible map that can then be implemented using unitary operations. Efficient implementation of $x \mapsto f(x)$ on a classical computer means that the number of elementary classical gates (e.g., AND, NOT, NAND gates) is at most $\text{poly}(n)$, and the classical implementation of the map uses at most $\text{poly}(n)$ additional bits for storage. By converting each of the elementary classical gate into a reversible gate, we can implement

$$U_f : |0\rangle^{\otimes w} |0\rangle^{\otimes m} |x\rangle \mapsto |g(x)\rangle |f(x)\rangle |x\rangle . \tag{5.2}$$

Using $w = \text{poly}(n)$ ancilla qubits, the depth of the quantum circuit is $\text{poly}(n)$.

THEOREM 5.2. *Any irreversible classical computation using* $\text{poly}(n)$ *classical gates can be simulated on a quantum computer using* $\text{poly}(n)$ *simple quantum gates and* $\text{poly}(n)$ *qubits.*

Up to a polynomial slowdown, a quantum computer is at least as powerful as classical computers. It should be noted that such a procedure is likely to be extremely inefficient. Thus the construction used in Theorem 5.2 is not expected to be practically useful beyond the simplest scenario.

5.2. Uncomputation

Unlike classical bits, qubits can exist in superpositions of computational basis states, which enables interference effects in computation. However, qubits are also prone to interference and can easily lose their coherence, causing computational errors. When a quantum computer performs a computation, it can create a large number of ancilla qubits (also called working qubits, or garbage register) that are entangled with the qubits carrying the actual result of the computation. If these ancilla qubits are not properly reset back to their initial state (usually $|0\rangle^{\otimes a}$), they can interfere with subsequent computations and cause errors. This resetting process is called **uncomputation**. Other than avoiding interference, uncomputation is also important for the purpose of resource management. Quantum systems available today have a limited number of qubits. By uncomputing, we can reuse qubits more efficiently.

Uncomputation needs to be done in a very specific way to maintain the integrity of the quantum computation. Simply resetting qubits (for example, by measuring the ancilla qubits and resetting them to $|0\rangle$) is not sufficient, as it can destroy the superposition and entanglement of the other

qubits in the system. Furthermore, due to the no-deleting theorem, there is no generic unitary operator that can set a black-box state to $|0\rangle^{\otimes w}$.

Let us now consider how to perform uncomputation when implementing a classical mapping. In quantum computing, an **oracle** means a black box operation that for a given input provides an output, usually the result of evaluating a function on that input. With the help of a working register, we assume that the oracle implementing Eq. (5.2) is available.

In order to set the working register back to $|0\rangle^{\otimes w}$ while keeping the input and output state, we must use the information stored in U_f explicitly. We introduce yet another m-qubit ancilla register initialized at $|0\rangle^{\otimes m}$. Then we can use an m-qubit CNOT controlled on the output register and obtain

$$|0\rangle^{\otimes m}|g(x)\rangle|f(x)\rangle|x\rangle \mapsto \underbrace{|f(x)\rangle}_{\text{ancilla}}\underbrace{|g(x)\rangle}_{\text{working}}\underbrace{|f(x)\rangle}_{\text{output}}\underbrace{|x\rangle}_{\text{input}}. \tag{5.3}$$

It is important to remember that in the operation above, the multi-qubit CNOT gate only performs the classical copying operation in the computational basis, and does not violate the no-cloning theorem.

Recall that $U_f^{-1} = U_f^\dagger$, so

$$(I^{\otimes m} \otimes U_f^\dagger)|f(x)\rangle|g(x)\rangle|f(x)\rangle|x\rangle = |f(x)\rangle|0\rangle^{\otimes w}|0\rangle^{\otimes m}|x\rangle. \tag{5.4}$$

Finally we apply an m-qubit SWAP operator on the ancilla and output registers to obtain

$$|f(x)\rangle|0\rangle^{\otimes w}|0\rangle^{\otimes m}|x\rangle \mapsto |0\rangle^{\otimes m}|0\rangle^{\otimes w}|f(x)\rangle|x\rangle. \tag{5.5}$$

After this procedure, both the ancilla and the working register are set to the initial state. They are no longer entangled to the input or output register, and can be reused for other purposes. The circuit for this uncomputation step is shown in Fig. 5.2.

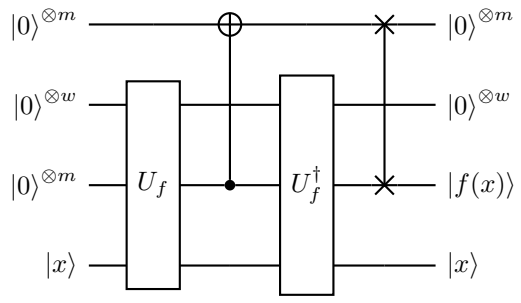


FIGURE 5.2. Circuit for uncomputation. The CNOT and SWAP operators indicate the multi-qubit copy and swap operations, respectively.

Remark 5.3 (Discarding working registers)**.** After the uncomputation as shown in Fig. 5.2, the first two registers are unchanged before and after the application of the circuit (though they are changed during the intermediate steps). Therefore Fig. 5.2 effectively implements a unitary

$$(I^{\otimes(m+w)} \otimes V_f)|0\rangle^{\otimes m}|0\rangle^{\otimes w}|0\rangle^{\otimes m}|x\rangle = |0\rangle^{\otimes m}|0\rangle^{\otimes w}|f(x)\rangle|x\rangle, \tag{5.6}$$

or equivalently

$$V_f \left|0\right\rangle^{\otimes m} \left|x\right\rangle = \left|f(x)\right\rangle \left|x\right\rangle . \tag{5.7}$$

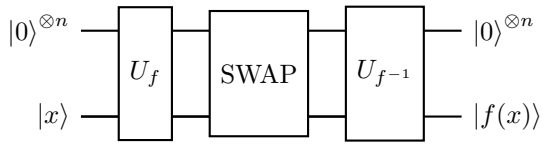
In the definition of V_f, all working registers have been discarded. This allows us to simplify the notation and focus on the essence of the quantum algorithms under study. Using the technique of uncomputation, if the map $x \mapsto f(x)$ can be efficiently implemented on a classical computer, then we can implement this map efficiently on a quantum computer as well with a controllable amount of quantum resources. ◇

Example 5.4. Given $f : \{0,1\}^n \to \{0,1\}^n$, in general, the transformation $\left|x\right\rangle \mapsto \left|f(x)\right\rangle$ is not unitary. However, when f is a bijection, and we have access to both f, f^{-1} as follows:

$$U_f : \left|z\right\rangle \left|x\right\rangle \mapsto \left|z \oplus f(x)\right\rangle \left|x\right\rangle , \quad U_{f^{-1}} : \left|z\right\rangle \left|x\right\rangle \mapsto \left|z \oplus f^{-1}(x)\right\rangle \left|x\right\rangle , \tag{5.8}$$

we can use them to construct the unitary transformation $U'_f : \left|x\right\rangle \mapsto \left|f(x)\right\rangle$.

To implement U'_f, we will use an ancilla register initialized in the $\left|0\right\rangle^{\otimes n}$ state to hold the result of applying f or f^{-1}. Apply U_f to the state $\left|0\right\rangle^{\otimes n} \left|x\right\rangle$ to get $\left|f(x)\right\rangle \left|x\right\rangle$. This setup now contains the desired mapping in the first register, but it is entangled with the input in the second register. Next apply SWAP to the two registers and the state becomes $\left|x\right\rangle \left|f(x)\right\rangle$. Apply $U_{f^{-1}}$ to the state $\left|x\right\rangle \left|f(x)\right\rangle$ to get $\left|x \oplus f^{-1}(f(x))\right\rangle \left|f(x)\right\rangle = \left|x \oplus x\right\rangle \left|f(x)\right\rangle = \left|0\right\rangle^{\otimes n} \left|f(x)\right\rangle$. The ancilla register is restored to $\left|0\right\rangle^{\otimes n}$ and can be discarded. This gives our desired U'_f. The circuit is as follows.



◇

Example 5.5. Another common usage of the uncomputation is to disentangle two registers. Consider the following sequence of operations

$$\begin{aligned} \sum_j c_j \left|v_j\right\rangle \left|0\right\rangle^{\otimes a} \left|0\right\rangle^{\otimes b} &\xrightarrow{U_a} \sum_j c_j \left|v_j\right\rangle \left|u_j\right\rangle \left|0\right\rangle^{\otimes b} \\ &\xrightarrow{U_b} \sum_j c_j \left|v_j\right\rangle \left|u_j\right\rangle (\beta_j \left|0\right\rangle^{\otimes b} + \sqrt{1 - \left|\beta_j\right|^2} \left|\perp_j\right\rangle). \end{aligned} \tag{5.9}$$

Here U_a only acts on the first and second register, U_b only acts on the second and third register, and $\left|\perp_j\right\rangle$ is a state that is orthogonal to $\left|0\right\rangle^{\otimes b}$. Our goal is to obtain a state proportional to

$$\sum_j c_j \beta_j \left|v_j\right\rangle \left|0\right\rangle^{\otimes a} \left|0\right\rangle^{\otimes b} . \tag{5.10}$$

This cannot be done by measuring the third register and check whether the outcome is 0^b, since it will lead to $\sum_j c_j \beta_j \left|v_j\right\rangle \left|u_j\right\rangle \left|0\right\rangle^{\otimes b}$, which entangles the first two registers. The correct procedure is to perform uncomputation by applying $U_a^\dagger$ to the first two registers, which gives a state

$$\sum_j c_j \left|v_j\right\rangle \left|0\right\rangle^{\otimes a} (\beta_j \left|0\right\rangle^{\otimes b} + \sqrt{1 - \left|\beta_j\right|^2} \left|\perp_j\right\rangle). \tag{5.11}$$

Then measuring the third register produces the desired state. ◇

5.3. Fixed point number representation and quantum random access memory

When we want to perform arithmetic operations on a quantum computer, such as addition, multiplication, or more complex functions, we need to encode the numbers we are working with into qubit states. On classical computers, floating point number representations are an efficient way to represent numbers with a wide numerical range. However, on quantum computers, it is often convenient to encode numbers into amplitudes or phases (e.g., via phase kickback). Therefore it is difficult in general to handle numbers that are too large or too small (e.g., $3.14 \times 10^{\pm 12}$). The standard practice is to use a binary fixed point representation of real numbers.

Any integer $k \in [N]$ where $N = 2^n$ can be expressed as an n-bit string as $k = (k_{n-1} \cdots k_0)$ with $k_i \in \{0, 1\}$. This is called the binary representation of the integer k. It should be interpreted as

$$k = \sum_{i=0}^{n-1} k_i 2^i. \tag{5.12}$$

The number k divided by 2^m $(0 \leq m \leq n)$ can be written as (note that the binary point is shifted to be after k_m):

$$a = \frac{k}{2^m} = \sum_{i=0}^{n-1} k_i 2^{i-m} =: (k_{n-1} \cdots k_m . k_{m-1} \cdots k_0). \tag{5.13}$$

The most common case is $m = n$, where

$$a = \frac{k}{2^n} = \sum_{i=0}^{n-1} k_i 2^{i-n} =: (0.k_{n-1} \cdots k_0) \equiv (.k_{n-1} \cdots k_0). \tag{5.14}$$

Sometimes we may also write $a = 0.k_1 \cdots k_n$, which is simply a relabeling of the digits. For a given real number $0 \leq a < 1$ written as

$$a = (0.k_1 \cdots k_n k_{n+1} \cdots), \tag{5.15}$$

the number $(0.k_1 \cdots k_n)$ is called the n-bit **fixed point representation** (or n-bit binary representation) of a. Therefore to represent a to additive precision ϵ, we will need $n = \lceil \log_2(1/\epsilon) \rceil$ bits of precision. If the sign of a is also important, we may reserve one extra bit $s \in \{0, 1\}$ to indicate its sign and interpret $(s.k_1 \cdots k_n)$ as $(-1)^s (0.k_1 \cdots k_n)$. A complex number z can be represented using two real numbers as $z = a + ib$, where $a, b \in \mathbb{R}$ are given in the fixed point number representation.

Definition 5.6. *For a length $N = 2^n$ classical data vector x, assume that each component x_i has a d-bit representation. Then the **quantum random access memory** (QRAM) is a unitary U_{QRAM} acting on $n + d$ qubits:*

$$U_{\mathrm{QRAM}} |i\rangle |y\rangle = |i\rangle |y \oplus x_i\rangle . \tag{5.16}$$

The implementation of U_{QRAM} often uses working registers, and such a dependence is hidden in Eq. (5.16) after the uncomputation step. Sometimes QRAM is called the quantum random access classical memory (QRACM). Ideally, the cost for implementing QRAM is poly(n), but this may not be possible if x represents an unstructured classical data set, and the cost for implementing QRAM may be as high as poly(N).

5.4. Classical arithmetic operations

Using the fixed point number representation and reversible computation, we can approximately implement classical arithmetic operations on quantum computers. The map $x \mapsto f(x)$ can be implemented as $U_f |\widetilde{x}\rangle |y\rangle = |\widetilde{x}\rangle |y \oplus \widetilde{f}(x)\rangle$ using e.g., a QRAM. Here $\widetilde{x}$ and $\widetilde{f}(x)$ are n-bit fixed point representation of $x, f(x)$ in the computational basis of the quantum register, respectively. However, it may be much more efficient to implement certain classical arithmetic operations on-the-fly on quantum computers without referring to a QRAM. For instance, $x \mapsto 2x$ can be implemented as a shift operation in the binary format that can be implemented via a sequence of SWAP gates. Other arithmetic mappings, such as $x \mapsto x^2$, as well as binary operations $(x, y) \mapsto x + y$, $(x, y) \mapsto xy$ are harder to implement. Furthermore, these operations can be implemented on quantum computers without going through the process of the reversible implementation of elementary classical gates. Some other classical functions, such as $x \mapsto \arccos(x)$ can be even more difficult to implement. In general, implementation of classical arithmetic operations on quantum computers will incur a significant overhead, both in terms of the number of ancilla qubits and the circuit depth.

Many arithmetic operations involve a procedure called the controlled rotation, which transforms the information stored in a register from a fixed point representation to the amplitude of the wavefunction.

Proposition 5.7 (Controlled rotation given rotation angles)**.** *Let $0 \leq \theta < 1$ have exact d-bit fixed point representation $\theta = (.\theta_{d-1} \cdots \theta_0)$. Then there is a $(d+1)$-qubit unitary U_θ such that*

$$U_\theta : |0\rangle|\theta\rangle \mapsto (\cos(\pi\theta)|0\rangle + \sin(\pi\theta)|1\rangle)|\theta\rangle. \tag{5.17}$$

Proof. First (by e.g. Taylor expansion)

$$\exp(-i\tau\sigma_y) = \begin{pmatrix} \cos(\tau) & -\sin(\tau) \\ \sin(\tau) & \cos(\tau) \end{pmatrix} =: R_y(2\tau). \tag{5.18}$$

Here $R_y(\cdot)$ performs a single-qubit rotation around the y-axis. For any $j \in [2^d]$ with its binary representation $j = j_{d-1} \cdots j_0$, we have

$$j/2^d = (.j_{d-1} \cdots j_0). \tag{5.19}$$

So choose $\tau = \pi(.j_{d-1} \cdots j_0)$, and define

$$U_\theta = \sum_{j \in [2^d]} \exp(-i\pi(.j_{d-1} \cdots j_0)\sigma_y) \otimes |j\rangle\langle j|. \tag{5.20}$$

Applying U_θ to $|0\rangle|\theta\rangle$ gives the desired results. This is a sequence of single-qubit rotations on the signal qubit, each controlled by a single qubit. □

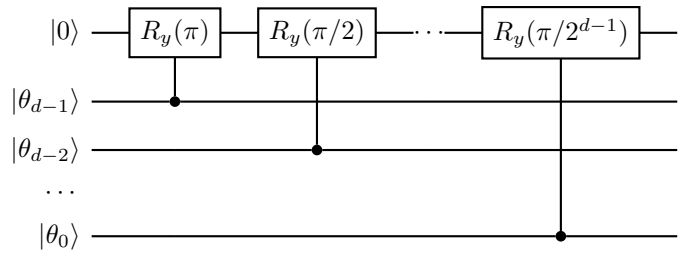


FIGURE 5.3. Quantum circuit for the controlled rotation operation U_θ.

Example 5.8 (Diagonal matrix multiplication using controlled rotation)**.** Let $0 \le a < 1$ be given by an d-bit fixed point representation using an d-qubit register, $f : \mathbb{R} \to \mathbb{R}$ be a function satisfying $|f(a)| \le 1$ for all $0 \le a < 1$. For simplicity assume $f(a) \ge 0$; the case of signed $f(a)$ can be handled by additionally computing the sign of $f(a)$ and applying a controlled phase flip on the $|1\rangle$ branch. We would like to construct a circuit that approximately implements

$$|a\rangle \to f(a)\,|a\rangle\,. \tag{5.21}$$

More generally, the state $|\psi\rangle = \sum_a \psi_a\,|a\rangle$ is mapped to $\sum_a \psi_a f(a)\,|a\rangle$. This can be viewed as multiplying a diagonal matrix $D = \mathrm{diag}\{f(a)\}$ to $|\psi\rangle$.

To implement such a mapping, we first define

$$\theta(a) = \frac{1}{\pi}\arccos f(a). \tag{5.22}$$

Note that even though a is exactly given by d-bits, $\theta(a)$ may not be. So we assume that it can be *rounded* to an d'-bit number $\widetilde{\theta}(a)$. For simplicity we assume d' is large enough so that the error of the fixed point representation is negligible in this step. To implement the mapping $a \mapsto \widetilde{\theta}(a)$, we can construct a classical arithmetics circuit

$$U_{\mathrm{angle}}\,|0^{d'}\rangle\,|a\rangle = |\widetilde{\theta}(a)\rangle\,|a\rangle\,, \tag{5.23}$$

whose construction may require $\mathrm{poly}(\max\{d, d'\})$ gates and an additional working register of $\mathrm{poly}(\max\{d, d'\})$ qubits, which are not displayed here. Therefore, the entire controlled rotation operation needed is given by the circuit in Fig. 5.4.

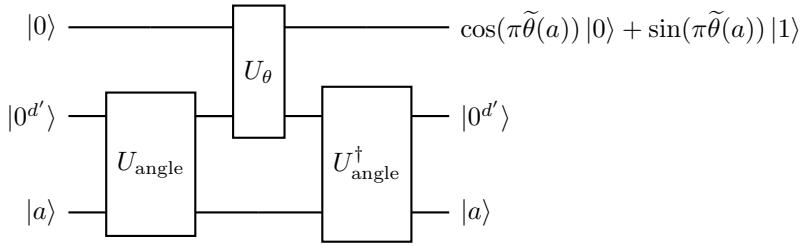


FIGURE 5.4. Circuit for using controlled rotation to implement the multiplication of a diagonal matrix (not including additional working register for classical arithmetic operations).

Note that through the uncomputation $U_{\text{angle}}^\dagger$, the d' ancilla qubits also become a working register. After uncomputation, the ancillas are returned to $|0^{d'}\rangle$ (together with any additional workspace used in U_{angle}), so they may be reused. We obtain a unitary U_{CR} satisfying

$$U_{\text{CR}}\,|0\rangle\,|a\rangle = \Big(\cos(\pi\widetilde{\theta}(a))\,|0\rangle + \sin(\pi\widetilde{\theta}(a))\,|1\rangle\Big)\,|a\rangle \approx \Big(f(a)\,|0\rangle + \sqrt{1-f(a)^2}\,|1\rangle\Big)\,|a\rangle\,. \tag{5.24}$$

Measure the single ancilla qubit. If the result is 0, the data register is projected onto a state proportional to $\sum_a \psi_a f(a)\,|a\rangle$, i.e., the mapping in Eq. (5.21) up to renormalization. If the input state is $|\psi\rangle = \sum_a \psi_a\,|a\rangle$, the probability of obtaining 0 after measuring the ancilla qubit is

$$\mathbb{P}(0) \approx \sum_a |\psi_a|^2\,|f(a)|^2\,. \tag{5.25}$$

◇

Example 5.9 (Use of arithmetic operations in the HHL algorithm)**.** The last step of the Harrow–Hassidim–Lloyd (HHL) algorithm for solving a linear system of equations $Ax = b$ with a Hermitian matrix A involves the following arithmetic operations. For simplicity assume λ_j (eigenvalues of A) are given exactly in a d-bit fixed point number representation, and $\lambda_j \in [\delta, 1]$ for some $\delta > 0$. Start from a linear combination of states $|\psi\rangle = \sum_j \beta_j\,|0\rangle\,|\lambda_j\rangle\,|v_j\rangle$, we would like to construct a state

$$|\psi'\rangle = \sum_j \frac{C\beta_j}{\lambda_j}\,|1\rangle\,|\lambda_j\rangle\,|v_j\rangle + |0\rangle\,|\perp\rangle\,. \tag{5.26}$$

Here C is a normalization constant chosen so that $|C/\lambda_j| < 1$ for all $\lambda_j \in [\delta, 1]$, and $|\perp\rangle$ is an irrelevant unnormalized state. Viewing this as a diagonal matrix multiplication problem, the function of interest is

$$f(a) = \frac{C}{a}, \quad a \in [\delta, 1]. \tag{5.27}$$

The implementation involves the classical arithmetic circuit for computing

$$\theta(a) = \frac{1}{\pi}\arcsin f(a) = \frac{1}{\pi}\arcsin(C/a) \tag{5.28}$$

using d' bits ($d' > d$).

Once $|\psi'\rangle$ is prepared, we can uncompute $|\lambda_j\rangle$ to obtain a state

$$\sum_j \frac{C\beta_j}{\lambda_j} |1\rangle |0^d\rangle |v_j\rangle + |0\rangle |\perp'\rangle \tag{5.29}$$

to disentangle the λ_j register from the v_j register. If we measure the first ancilla register and obtain 1, we obtain the desired form of the solution in the HHL algorithm. ◇

Notes and further reading

Reversible computation predates quantum computing and has both physical and algorithmic motivations. For background on reversible embeddings of classical circuits into unitary dynamics, see [NC00, Section 3.2.5]. For fixed-point encodings and reversible arithmetic (addition, multiplication, and function evaluation), a detailed treatment is given in [RP11, Chapter 6]. For standard universal classical gate constructions and decompositions into elementary quantum gates, see [BBC$^+$95]. There is also opportunity to optimize the cost of the uncomputation stage. An example is Gidney's construction [Gid18] of the quantum adder circuit. The QRAM model [GLM08] should be interpreted as an assumption about data access rather than an automatic feature of a fault-tolerant architecture.

CHAPTER 6

Query complexity and quantum complexity theory

CHAPTER 7

Perturbation theory

Numerical computation is inexact: errors in the input data, discretization error, and roundoff errors can all affect the final output. Analogous error sources in quantum algorithms include finite-precision state preparation, approximate arithmetic, and imperfect implementations of primitives such as block encodings or Hamiltonian simulation. This chapter provides tools that allow us to isolate the part of the error that is intrinsic to the mathematical task itself, by asking how much the output can change under small perturbations of the input.

Perturbation theory provides a streamlined way to separate the error from data from the error from algorithms. Once the sensitivity of the underlying problem is quantified, one can treat the available input as exact and focus on how a chosen algorithm propagates or amplifies perturbations. This viewpoint is closely aligned with backward error analysis in classical computation.

We first introduce basic concepts such as forward error, backward error and condition number, and then apply the perturbation analysis to two prototypical problems: linear systems of equations, and eigenvalue problems. Some perturbation tools will also be used later in quantum phase estimation, Hamiltonian simulation, quantum singular value transformation, and related primitives.

7.1. Forward and backward error

For simplicity, consider a smooth function $f : \mathbb{R}^d \to \mathbb{R}$. Let $a \in \mathbb{R}^d$ be the input, let $f(a)$ denote the exact value, and let $\mathrm{alg}(f)(a)$ denote the output produced by an algorithm intended to approximate $f(a)$. The quantity $|\mathrm{alg}(f)(a) - f(a)|$ is called the **forward error**. It is the end-to-end discrepancy between the quantity we want and the quantity we actually compute, so it is the most direct measure of accuracy. However, forward error alone does not identify the source of the discrepancy. When it is large, it may be unclear whether the problem itself is highly sensitive to perturbations of the input or whether the algorithm has introduced substantial additional error.

This ambiguity is the reason to consider backward error. If there exists a perturbation δa such that $\mathrm{alg}(f)(a) = f(a+\delta a)$, then δa is called a **backward perturbation**. When such a representation exists, one defines the **backward error** as the (typically minimal) size $\|\delta a\|$. Backward error asks whether the computed output is the exact answer to a nearby problem. If $\|\delta a\|$ is small, then the algorithm has changed the data only slightly, and a large forward error should be attributed primarily to the sensitivity of f. If every such perturbation must be large, then the algorithm itself is significantly distorting the problem, and the loss of accuracy should be blamed on the algorithm.

Thus forward error typically mixes two effects: sensitivity of the problem and error introduced by the algorithm, whereas backward error helps separate them. The following examples illustrate this distinction. Suppose $d = 1$ and the goal is to compute $f(a) = e^a$ at $a = 100$ to some absolute precision. Then an input perturbation δa produces $|f(a + \delta a) - f(a)| \approx e^{100} |\delta a|$ for small δa, regardless of the algorithm. A more complex example is the prediction of trajectories in chaotic dynamics, where a tiny perturbation of the initial state a can lead to a macroscopically different

trajectory $f(a)$. In such cases, a large forward error may be unavoidable because it reflects the sensitivity of the problem itself rather than a defect of the algorithm.

Conversely, even for a well-posed problem, a poor algorithmic choice can produce a large forward error. Consider approximating $f(a) = g'(x)|_{x=a}$ for a smooth function g via a forward difference quotient:

$$\mathrm{alg}(f)(a) = \frac{(g(a+h)+u_1)-(g(a)+u_2)}{h} = g'(a) + \frac{1}{2}g''(a)h + \frac{u_1-u_2}{h} + \mathcal{O}(h^2). \tag{7.1}$$

Here h is the step size, and u_1, u_2 model the (absolute) evaluation errors in finite precision. The truncation term $\frac{1}{2}g''(a)h+\mathcal{O}(h^2)$ decreases with h, whereas the roundoff term $(u_1-u_2)/h$ is amplified when h is small. If h is chosen too small relative to the evaluation errors, then the roundoff term dominates and the forward error can become $\mathcal{O}(1)$ even when g is smooth. Since h is an algorithmic parameter, this is an algorithm-induced loss of accuracy.

When $g''(a) \neq 0$, this instability can also be expressed in terms of a backward perturbation by solving for δa such that

$$g'(a+\delta a) = g'(a) + \frac{1}{2}g''(a)h + \frac{u_1-u_2}{h}, \tag{7.2}$$

which yields to leading order:

$$\delta a \approx \frac{h}{2} + \frac{u_1-u_2}{g''(a)h}. \tag{7.3}$$

This makes explicit that taking h too small can force $\|\delta a\|$ to be large, so in this case the failure is attributable to the algorithm rather than to unusual sensitivity of the underlying problem.

More generally, suppose the input data itself is inexact. Let $\widetilde{a}$ denote the available input with $\|\widetilde{a}-a\| \leq \epsilon'$. Then the total error admits the decomposition

$$|\mathrm{alg}(f)(\widetilde{a}) - f(a)| \leq |\mathrm{alg}(f)(\widetilde{a}) - f(\widetilde{a})| + |f(\widetilde{a}) - f(a)|\,. \tag{7.4}$$

The first term is the algorithmic error for the (putative) exact input $\widetilde{a}$, while the second term is the error induced by data uncertainty. By the mean value theorem,

$$|f(\widetilde{a}) - f(a)| \leq \sup_{0\leq\theta\leq 1} \left\|\nabla f\big(a+\theta(\widetilde{a}-a)\big)\right\| \, \|\widetilde{a}-a\|\,. \tag{7.5}$$

For small ϵ', this is often approximated by $\|\nabla f(a)\|\,\epsilon'$. In this scalar-output setting, $\|\nabla f(a)\|$ is a natural local **condition number**: it measures the sensitivity of $f(a)$ to perturbations of the input.

This decomposition provides several insights. First, when $\|\nabla f(a)\|$ is large, even a small input uncertainty can produce a large forward error, indicating that the problem is locally ill-conditioned. Second, when the input data has inherent uncertainty, there is limited benefit in refining the algorithm beyond the scale of that uncertainty: if $|\mathrm{alg}(f)(\widetilde{a}) - f(\widetilde{a})|\,|f(\widetilde{a}) - f(a)|$, then further reductions in algorithmic error yield diminishing returns in the total error. This perspective is especially relevant in quantum algorithms, where the input (for example, a matrix provided via block encoding or a Hamiltonian given through a simulation procedure) is itself constructed approximately.

To relate this to backward error analysis, suppose that for a given input a one can write

$$\mathrm{alg}(f)(a) = f(a+\delta a), \tag{7.6}$$

for some backward perturbation δa. Then

$$|\mathrm{alg}(f)(a) - f(a)| = |f(a+\delta a) - f(a)| \approx \|\nabla f(a)\|\,\|\delta a\|\,. \tag{7.7}$$

Combining this with Eq. (7.4), we see that there is typically no practical benefit in guaranteeing a backward error $\|\delta a\|$ that is far smaller than the uncertainty already present in the input data. An algorithm is called **backward stable** if it admits backward perturbations that remain small for all admissible inputs.

Throughout this book, we will not generally prove backward stability explicitly. Instead, once perturbation theory confirms that the problem is not inherently ill-conditioned, we will focus on proving that the forward error of the quantum algorithm is small.

Example 7.1 (s-sparse matrices)**.** In matrix computations, let A denote the exact input matrix and $\widetilde{A} = A + E$ its numerical approximation. The perturbation E may arise from various sources, such as roundoff errors introduced during the loading of matrix entries into QRAM. If A is unstructured, the resulting norm $\|E\|$ may scale with the dimension of A, and thus become prohibitively large for high-dimensional systems.

Now consider the case where A is an s-sparse matrix, and assume that the perturbation E is also s-sparse with entrywise bound $\|E\|_{\max} \leq \epsilon$. Then, as shown in Lemma 2.46, the operator norm of the perturbation satisfies $\|E\| \leq s\epsilon$, which is independent of the overall dimension of A.

Perturbation results in matrix analysis often take the form of a condition number multiplied by $\|E\|$. This simple estimate demonstrates that such results remain meaningful in the quantum setting, even for matrices of exponentially large size 2^n, provided the matrix is sparse. ◇

Example 7.2. Let A, B be $N \times N$ matrices, and assume A, B commute with their commutator, i.e., $[A, [A, B]] = [B, [A, B]] = 0$. We would like to evaluate the matrix exponential $f(A, B) = e^{(A+B)t}$ for some small t. The algorithm we are using is

$$\mathrm{alg}(f)(A, B) = e^{At} e^{Bt}, \tag{7.8}$$

which introduces an error. The Baker–Campbell–Hausdorff formula (BCH) states that

$$\mathrm{alg}(f)(A, B) = e^{At} e^{Bt} = e^{(A+B)t + \frac{t^2}{2}[A,B]} = f\left(A + \frac{t}{2}[A, B], B\right). \tag{7.9}$$

Thus the algorithmic output coincides exactly with the true value of f evaluated on a perturbed input $(A + \delta A, B + \delta B)$, for instance with $\delta A = \frac{t}{2}[A, B]$ and $\delta B = 0$. In this sense the backward perturbation has size $\mathcal{O}(t)$ (more precisely, $\|\delta A\| = \frac{t}{2}\|[A, B]\|$ in operator norm). The backward perturbation is not unique: one may equally write $\mathrm{alg}(f)(A, B) = f(A, B + \frac{t}{2}[A, B])$, and thereby attribute the perturbation to B instead. ◇

Remark 7.3. Backward error analysis might not always be applicable. For instance, in computing the outer product $M = aa^\top$, the exact output has rank 1, whereas an algorithm may produce an approximation $\widetilde{M} = \mathrm{alg}(M)$ of rank larger than 1. In this case there need not exist any δa such that $\widetilde{M} = (a + \delta a)(a + \delta a)^\top$. ◇

7.2. Perturbation theory for linear systems of equations

Consider the linear systems $Ax = b$ and $(A + \delta A)\widetilde{x} = b + \delta b$. We would like to bound the solution error $\delta x := \widetilde{x} - x$ in terms of the perturbations δA and δb. Subtracting the two equations gives

$$A\delta x = -(\delta A)\widetilde{x} + \delta b. \tag{7.10}$$

Assume that A is invertible and that $\|\delta A\|$ and $\|\delta b\|$ are small. In this linear regime, we discard all second-order terms in δA and δb and obtain

$$\delta x = -A^{-1}(\delta A)x + A^{-1}\delta b. \tag{7.11}$$

Therefore,

$$\|\delta x\| \leq \left\|A^{-1}\right\| \left(\|\delta A\| \, \|x\| + \|\delta b\|\right), \tag{7.12}$$

and hence

$$\frac{\|\delta x\|}{\|x\|} \leq \left\|A^{-1}\right\| \|A\| \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\| \, \|x\|}\right) \leq \left\|A^{-1}\right\| \|A\| \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right). \tag{7.13}$$

Here we used $\|A\| \, \|x\| \geq \|Ax\| = \|b\|$. The quantity $\kappa(A) := \left\|A^{-1}\right\| \|A\|$ is called the condition number of A. It controls the amplification of **relative error** in the solution by relative error in the input data.

The analysis above omits second-order contributions in δA and δb. For larger perturbations, a similar argument can be made rigorous, but it requires controlling these higher-order terms. First, we rewrite Eq. (7.10) as

$$(A + \delta A)\delta x = -(\delta A)x + \delta b. \tag{7.14}$$

This gives

$$\frac{\|\delta x\|}{\|x\|} \leq \left\|(A + \delta A)^{-1}\right\| \|A\| \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right). \tag{7.15}$$

Exercise 7.1. For any square matrix A satisfying $\|A\| < 1$, prove that the matrix $I - A$ is invertible, and

$$\left\|(I - A)^{-1}\right\| \leq (1 - \|A\|)^{-1}. \tag{7.16}$$

Now apply Eq. (7.16). Assume $\left\|A^{-1}\right\| \|\delta A\| < 1$. Then

$$\left\|(A + \delta A)^{-1}\right\| \leq \left\|(I + A^{-1}\delta A)^{-1}\right\| \left\|A^{-1}\right\| \leq \frac{1}{1 - \|A^{-1}\delta A\|} \left\|A^{-1}\right\| \leq \frac{1}{1 - \|A^{-1}\| \, \|A\| \, \frac{\|\delta A\|}{\|A\|}} \left\|A^{-1}\right\|. \tag{7.17}$$

Therefore

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right). \tag{7.18}$$

There is an interesting generalization of the perturbation analysis above to the quantum setting. When solving linear systems on a quantum computer, because a quantum state has to be normalized, we cannot directly obtain an approximation to the solution vector x. Instead we can only aim at preparing the normalized state $|x\rangle := x/\|x\|$, and likewise $|\widetilde{x}\rangle := \widetilde{x}/\|\widetilde{x}\|$. According to the relation

??, the trace distance is upper bounded by the error of the normalized solution:

$$\begin{aligned} D(|x\rangle\langle x|, |\widetilde{x}\rangle\langle\widetilde{x}|) =& \sin\theta(|x\rangle\langle x|, |\widetilde{x}\rangle\langle\widetilde{x}|) \\ \leq& D_p(|x\rangle, |\widetilde{x}\rangle) \leq \||x\rangle - |\widetilde{x}\rangle\| \\ =& \left\| \frac{x}{\|x\|} - \frac{\widetilde{x}}{\|\widetilde{x}\|} \right\| \\ \leq& \left\| \frac{x}{\|x\|} - \frac{\widetilde{x}}{\|x\|} \right\| + \left\| \frac{\widetilde{x}}{\|x\|} - \frac{\widetilde{x}}{\|\widetilde{x}\|} \right\| \\ =& \frac{\|\delta x\|}{\|x\|} + \left| \frac{\|\widetilde{x}\|}{\|x\|} - 1 \right| \\ \leq& 2\frac{\|\delta x\|}{\|x\|} \\ \leq& \frac{2\kappa(A)}{1 - \kappa(A)\frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right). \end{aligned} \tag{7.19}$$

In the last inequality we used Eq. (7.18). So compared to the standard problem of solving $Ax = b$, the relative error for computing the normalized solution is amplified by at most a factor of 2.

7.3. Perturbation theory for Hermitian eigenvalue problems

Consider the eigenvalue problem $A|x\rangle = \lambda|x\rangle$ for a Hermitian matrix A. Then the eigenvalue λ is real. We would like to estimate the perturbations of the eigenvalue and eigenvector in the perturbed problem $(A + \delta A)|\widetilde{x}\rangle = \widetilde{\lambda}|\widetilde{x}\rangle$. For simplicity, assume that λ and $\widetilde{\lambda}$ are simple eigenvalues of A and $A + \delta A$, respectively. Let $\widetilde{\lambda} = \lambda + \delta\lambda$ and $|\widetilde{x}\rangle = |x\rangle + \delta x$. The use of the ket notation implies that $|x\rangle$ and $|\widetilde{x}\rangle$ are normalized, but δx need not be. In the linear regime, assume that $\|\delta A\|$ is small and drop second and higher order terms. Then

$$A\delta x + \delta A|x\rangle = \lambda\delta x + \delta\lambda|x\rangle. \tag{7.20}$$

Applying $\langle x|$ to both sides of the equation, we have

$$\langle x|A\delta x + \langle x|\delta A|x\rangle = \lambda\langle x|\delta x + \delta\lambda\langle x|x\rangle. \tag{7.21}$$

Since $\langle x|A = \lambda\langle x|$, we obtain the perturbation of the eigenvalue

$$\delta\lambda = \langle x|\delta A|x\rangle = \mathrm{Tr}[\delta A|x\rangle\langle x|]. \tag{7.22}$$

In particular,

$$|\delta\lambda| \leq \|\delta A\|. \tag{7.23}$$

For example, if $A(\alpha)$ depends smoothly on a parameter α, then

$$\frac{\mathrm{d}\lambda}{\mathrm{d}\alpha} = \langle x|A'(\alpha)|x\rangle. \tag{7.24}$$

This is an example of the Hellmann–Feynman theorem.

The perturbation of the eigenvector satisfies the equation

$$(\lambda I - A)\delta x = (\delta A - \delta\lambda)|x\rangle. \tag{7.25}$$

Since $\lambda I - A$ is not invertible, the solution to this linear system of equations is not unique. The unique solution can be obtained by minimizing the global-phase-invariant distance between $|x\rangle$ and $|\widetilde{x}\rangle$, which fixes the phase and leads to the constraint $\langle x|\,\delta x = 0$. We obtain

$$\delta x = Q(\lambda I - A)^{-1} Q(\delta A - \delta\lambda)\,|x\rangle = Q(\lambda I - A)^{-1} Q\delta A\,|x\rangle\,. \tag{7.26}$$

where $Q = I - |x\rangle\langle x|$ is a projection operator. Therefore in the linear regime,

$$D(|x\rangle\langle x|, |\widetilde{x}\rangle\langle\widetilde{x}|) = \sin\theta(|x\rangle\langle x|, |\widetilde{x}\rangle\langle\widetilde{x}|) \le \||x\rangle - |\widetilde{x}\rangle\| = \|\delta x\| \le \left\|Q(\lambda I - A)^{-1}Q\right\| \|\delta A\| \le \frac{\|\delta A\|}{\Delta} = \frac{\|A\|}{\Delta}\cdot\frac{\|\delta A\|}{\|A\|}. \tag{7.27}$$

Here Δ is the minimal distance between λ and the rest of the eigenvalues of A, and is often called the (absolute) spectral gap. The ratio $\Delta/\|A\|$ is called the relative spectral gap, which describes the spectral gap when A is normalized. Here the inverse of the relative gap, i.e., $\|A\|/\Delta$, plays the role of the condition number for the eigenvector x.

Exercise 7.2. Let $A\,|x_k\rangle = \lambda_k\,|x_k\rangle$ and $\lambda = \lambda_l$ be a simple eigenvalue of A. Show that Eq. (7.26) can be as

$$\delta x = \sum_{k\neq l} |x_k\rangle\,\frac{1}{\lambda_l - \lambda_k}\,\langle x_k|\delta A|x_l\rangle\,. \tag{7.28}$$

So far we have assumed that λ is a simple eigenvalue and the perturbation δA is infinitesimally small. The rigorous version of the bound for eigenvalues is given by **Weyl's inequality**, which holds for any A and δA. The following statement is a simplified version of Weyl's inequality [HJ91, Theorem 4.3.1].

THEOREM 7.4 (Weyl). *Let A and E be $n\times n$ Hermitian matrices. Denote the ordered eigenvalues of $A, A+E$ by $\lambda_i(A), \lambda_i(A+E)$ respectively, where $i = 1, 2, \ldots, n$, and the eigenvalues are arranged in non-increasing order. Then*

$$|\lambda_i(A+E) - \lambda_i(A)| \le \|E\|\,, \quad i = 1, \ldots, n. \tag{7.29}$$

PROOF. Given A and E are $n \times n$ Hermitian matrices, and $\lambda_i(A)$ and $\lambda_i(A+E)$ are their ordered eigenvalues. By the **Courant-Fischer min-max principle** [Bha97, Theorem 3.1.2] the i-th eigenvalue of a Hermitian matrix M can be characterized as follows:

$$\lambda_i(M) = \max_{\substack{S\subseteq\mathbb{C}^n \\ \dim(S)=i}} \min_{\substack{x\in S \\ \|x\|=1}} x^\dagger M x = \min_{\substack{S\subseteq\mathbb{C}^n \\ \dim(S)=n-i+1}} \max_{\substack{x\in S \\ \|x\|=1}} x^\dagger M x. \tag{7.30}$$

Let $S \subseteq \mathbb{C}^n$ be the subspace with $\dim(S) = n - i + 1$ corresponding to $\lambda_i(A)$. Then

$$\lambda_i(A+E) \le \max_{\substack{x\in S \\ \|x\|=1}} x^\dagger(A+E)x \le \max_{\substack{x\in S \\ \|x\|=1}} x^\dagger A x + \|E\| = \lambda_i(A) + \|E\|\,. \tag{7.31}$$

Similarly, let $S \subseteq \mathbb{C}^n$ be the subspace with $\dim(S) = n - i + 1$ corresponding to $\lambda_i(A+E)$. Then the same argument shows

$$\lambda_i(A) \le \lambda_i(A+E) + \|E\|\,. \tag{7.32}$$

This completes the proof. □

What about the perturbation of eigenvectors? A rigorous version of Eq. (7.27) is called the **Davis-Kahan** $\sin\theta$ **theorem**. We state a version without proof, which is a direct consequence of a more general result in [Bha97, Theorem VII.3.1].

THEOREM 7.5 (Davis-Kahan $\sin\theta$ theorem). *Let A and $A+\delta A$ be $N\times N$ Hermitian matrices. Let P be a spectral projector of A associated with eigenvalues in an interval $[a,b]$, and $\widetilde{P}$ be a spectral projector of $A+\delta A$ associated with eigenvalues in an interval $[a-\Delta, b+\Delta]$, respectively. Then*

$$\left\|P(I-\widetilde{P})\right\| \leq \frac{\|\delta A\|}{\Delta}. \tag{7.33}$$

To see how $\sin\theta$ appears, let us revisit the case of a single isolated eigenvalue, where $P = |x\rangle\langle x|, \widetilde{P} = |\widetilde{x}\rangle\langle\widetilde{x}|$ are spectral projectors associated with the eigenvalue $\lambda, \widetilde{\lambda}$ of $A, A+\delta A$, respectively. Choose the phase factor so that $\langle x|\widetilde{x}\rangle = \cos\theta > 0$. Then $P\widetilde{P} = |x\rangle\langle x|\widetilde{x}\rangle\langle\widetilde{x}| = |x\rangle\cos(\theta)\langle\widetilde{x}|$, and

$$\left\|P(I-\widetilde{P})\right\| = \||x\rangle - \cos\theta\,|\widetilde{x}\rangle\| = \sqrt{1-\cos^2\theta} = \sin\theta = D(P,\widetilde{P}), \tag{7.34}$$

which is the trace distance between the pure states $P, \widetilde{P}$. We refer to [Bha97, Chapter VII.1] for a more general connection between $P(I-\widetilde{P})$ and the principal angle of the two associated subspaces.

Example 7.6. An interesting connection between the linear system solver and a Hermitian eigenvalue problem in the quantum setting is as follows. For simplicity, let A be Hermitian positive definite with $A \succeq \Delta I$ for some $\Delta > 0$, and assume $\|b\| = 1$. Write $|b\rangle := b$. Then an alternative formulation of finding $|x\rangle = x/\|x\|$ such that $Ax = b$ can be stated as an eigenvalue problem:

$$\mathcal{A}\begin{pmatrix}|x\rangle\\0\end{pmatrix} = 0, \quad \mathcal{A} = \begin{pmatrix}0 & AQ_b\\ Q_bA & 0\end{pmatrix}, \quad Q_b = I - |b\rangle\langle b|. \tag{7.35}$$

Now consider a perturbed problem

$$\widetilde{\mathcal{A}} := \begin{pmatrix}0 & (A+\delta A)Q_b\\ Q_b(A+\delta A) & 0\end{pmatrix} \tag{7.36}$$

and suppose that

$$\widetilde{\mathcal{A}}\begin{pmatrix}|\widetilde{x}\rangle\\0\end{pmatrix} = 0. \tag{7.37}$$

Let $\langle x|\widetilde{x}\rangle = \cos\theta > 0$. All nonzero eigenvalues of $\mathcal{A}$ have magnitude at least Δ. Apply Theorem 7.5 and we find

$$D(|x\rangle\langle x|, |\widetilde{x}\rangle\langle\widetilde{x}|) = \sin\theta \leq \frac{\|\delta A\|}{\Delta} = \frac{\|A\|}{\Delta}\frac{\|\delta A\|}{\|A\|}. \tag{7.38}$$

Note that $\|A\|/\Delta = \|A\|\left\|A^{-1}\right\| = \kappa(A)$ is precisely the condition number of A. ◇

7.4. Additional perturbation theorems

Definition 7.7 (Optimal matching distance)**.** *The **optimal matching distance** between $x, y \in \mathbb{C}^N$ is*

$$D_m(x,y) = \min_{\sigma\in S_N}\max_i \left|x_i - y_{\sigma(i)}\right|, \tag{7.39}$$

where S_N denotes the symmetric group on N elements.

Exercise 7.3. If $x, y \in \mathbb{R}^N$, and $x_1 \leq x_2 \leq \cdots \leq x_N$, $y_1 \leq y_2 \leq \cdots \leq y_N$, prove that the optimal matching distance is equal to $\max_i |x_i - y_i|$.

Using Exercise 7.3, Weyl's inequality can also be stated as

$$D_m(\lambda(A), \lambda(A+E)) \leq \|E\|, \tag{7.40}$$

where $\lambda(A)$ denotes the set of eigenvalues of A.

There is a nontrivial generalization of Weyl's inequality to unitary matrices. The following theorem is a restatement of [Bha97, Theorem VI.3.11].

THEOREM 7.8 (Perturbation of eigenvalues for unitary matrices). *For any two unitary matrices $U, V \in \mathbb{C}^{N\times N}$,*

$$D_m(\lambda(U), \lambda(V)) \leq \|U - V\|, \tag{7.41}$$

where $\lambda(A)$ denotes the set of eigenvalues of A.

This is useful, for example, in the context of phase estimation (see Chapter 16), to bound the phase errors in terms of the error in the input unitary. To control perturbations of eigenvectors, we will use the following Davis–Kahan type bound, restated from [Bha97, Theorem VII.3.3] for unitary matrices.

THEOREM 7.9. *Let $U, V \in \mathrm{U}(N)$. Let P be a spectral projector of U associated with eigenvalues in a subset $S \subset \mathbb{C}$, and let $\widetilde{Q}$ be a spectral projector of V associated with eigenvalues in a subset $\widetilde{S} \subset \mathbb{C}$. Assume that* $\operatorname{dist}(S, \widetilde{S}) = \Delta > 0$. *Then there exists a universal constant $C < 2.91$ such that*

$$\left\| P\widetilde{Q} \right\| \leq \frac{C\,\|U - V\|}{\Delta}. \tag{7.42}$$

If U has a simple eigenvalue $e^{i\lambda}$ with a spectral projector P, and V has a simple eigenvalue in the disk $B(e^{i\lambda}, \Delta)$ in the complex plane with a spectral projector $\widetilde{P}$, then

$$D(P, \widetilde{P}) = \left\| P(I - \widetilde{P}) \right\| \leq \frac{C\,\|U - V\|}{\Delta}. \tag{7.43}$$

There is a counterpart of Weyl's theorem for singular values. The following result, restated from [Bha97, Problem III.6.13], will be useful when we analyze singular value transformations in Chapter 12.

THEOREM 7.10 (Perturbation of singular values). *For any two matrices $A, B \in \mathbb{C}^{N\times N}$,*

$$D_m(\Sigma(A), \Sigma(B)) \leq \|A - B\|, \tag{7.44}$$

where $\Sigma(A)$ denotes the set of singular values of A.

For general matrices, the **Gershgorin circle theorem** provides a simple and efficient way to estimate the range in which all eigenvalues must lie.

THEOREM 7.11 (Gershgorin circle theorem, see e.g. [GVL13, Theorem 7.2.1]). *Let $A \in \mathbb{C}^{N\times N}$ with entries a_{ij}. For each $i = 1, \ldots, N$, define*

$$R_i = \sum_{j \neq i} |a_{ij}|. \tag{7.45}$$

Let

$$D_i = \{z \in \mathbb{C} : |z - a_{ii}| \leq R_i\}, \tag{7.46}$$

be a closed disc centered at a_{ii} with radius R_i, called a Gershgorin disc. Then every eigenvalue of A lies within at least one of the Gershgorin discs D_i.

Notes and further reading

Many of the perturbation bounds used in quantum algorithm design are inherited directly from classical numerical linear algebra and matrix analysis. General discussions of forward and backward error, condition numbers, and stability can be found in standard texts on numerical computation and linear algebra; see, for instance, [HJ91, TB97, Dem97, Hig02, GVL13]. For a systematic treatment of matrix inequalities and perturbation theorems for eigenvalues, singular values, spectral subspaces, and matrix functions, see Bhatia [Bha97]; for a focused account of eigenvalue and singular value sensitivity, see Stewart and Sun [SS90]. For a more general operator-theoretic perspective, including unbounded and infinite-dimensional settings, see Kato [Kat76].

CHAPTER 8

Statistical estimates

Part III

Algorithm

CHAPTER 9

Block encoding

This chapter introduces block encoding as an input model for matrix problems on a quantum computer. The basic difficulty is that many tasks in scientific computation are naturally phrased in terms of non-unitary linear maps, whereas the native operations available to quantum hardware are unitary. Block encoding addresses this mismatch by representing a target matrix A (up to a subnormalization factor and a prescribed error tolerance) as a submatrix block of a larger unitary U_A, so that applying U_A and post-selecting on ancilla qubits effectively applies A to a state.

The possibility of constructing an efficiently implementable U_A depends strongly on the structure of A and on the assumed access model. For a dense matrix without additional structure, any reasonable input model is typically prohibitive, since the input description may itself require exponential resources. We therefore focus on a few concrete settings in which block encodings can be constructed efficiently under suitable oracle access assumptions.

The true power of block encoding does not come directly from the ability to represent arbitrary matrices within blocks of a larger unitary. Rather, it stems from the ability to compose block encodings to block encode more complicated matrices and functions of matrices. We then describe how block encodings can be combined to obtain encodings of matrix additions and multiplications, while tracking the corresponding subnormalization factors and errors. Linear combinations of unitaries provide a flexible mechanism for such constructions. In this way, block encoding serves as an interface between matrix-oriented problem statements and unitary circuit realizations used throughout subsequent chapters.

9.1. Block encoding

The simplest example of block encoding is the following: assume we can find a $(n+1)$-qubit unitary matrix $U_A \in \mathrm{U}(2N)$ (where $N = 2^n$) such that

$$U_A = \begin{pmatrix} A & * \\ * & * \end{pmatrix}$$

where $*$ means that the corresponding matrix entries are irrelevant, then for any n-qubit quantum state $|b\rangle$, we can consider the state

$$|0, b\rangle = |0\rangle |b\rangle = \begin{pmatrix} b \\ 0 \end{pmatrix}, \tag{9.1}$$

and

$$U_A |0, b\rangle = \begin{pmatrix} Ab \\ * \end{pmatrix} =: |0\rangle A |b\rangle + |\perp\rangle . \tag{9.2}$$

Here the (unnormalized) state $|\perp\rangle$ can be written as $|1\rangle|\psi\rangle$ for some (unnormalized) state $|\psi\rangle$ that is irrelevant to the computation of $A|b\rangle$. In particular, it satisfies the orthogonality relation.

$$(|0\rangle\langle 0| \otimes I_N)|\perp\rangle = 0. \tag{9.3}$$

In order to obtain $A|b\rangle$, we measure the ancilla qubit and postselect on the outcome 0. This can be summarized into the following quantum circuit:

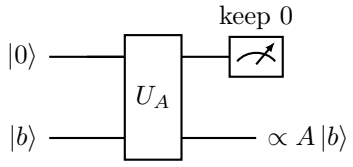


FIGURE 9.1. Circuit for block encoding of A using one ancilla qubit. By measuring the ancilla qubit and postselecting on the outcome 0, the state in the system register is a normalized state proportional to $A|b\rangle$.

Note that the output state is normalized after the measurement takes place. The success probability of obtaining 0 from the measurement can be computed as

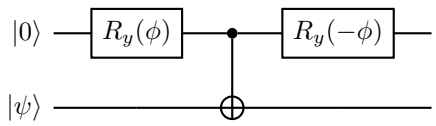
$$p(0) = \|A|b\rangle\|^2 = \langle b|A^\dagger A|b\rangle\,. \tag{9.4}$$

So the missing information of the norm $\|A|b\rangle\|$ can be recovered via the success probability $p(0)$ if needed. We find that the success probability is only determined by $A, |b\rangle$, and is independent of other irrelevant components of U_A.

Example 9.1. Consider the 2×2 matrix

$$A = \frac{3}{4}I + \frac{1}{4}X = \begin{pmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix}. \tag{9.5}$$

Consider the following circuit ($\phi = \frac{1}{3}\pi$)



Here

$$R_y(\theta) := \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} = e^{-i\theta Y/2} \tag{9.6}$$

is the Y-rotation matrix. One may directly verify that U_A is an exact block encoding of A using one ancilla qubit. ◇

Note that we may not need to restrict the matrix U_A to be an $(n+1)$-qubit matrix. If we can find any $(n+m)$-qubit unitary matrix U_A so that

$$U_A = \begin{pmatrix} A & * & \cdots & * \\ * & * & \cdots & * \\ \vdots & & \vdots & \\ * & * & \cdots & * \end{pmatrix} \tag{9.7}$$

Here each $*$ stands for an n-qubit matrix, and there are 2^m block rows / columns in U_A. Using the partial application of operators in Definition 2.25, the relation above can be written compactly using the braket notation as

$$A = \langle 0^m | U_A | 0^m \rangle . \tag{9.8}$$

Exercise 9.1. Given a unitary matrix U and any submatrix block A, prove that $\|A\| \leq 1$.

In order to find such a block encoding U_A, Exercise 9.1 shows that a necessary condition for the existence of U_A is that $\|A\| \leq 1$. However, if we can find sufficiently large α and U_A so that

$$A/\alpha = \langle 0^m | U_A | 0^m \rangle . \tag{9.9}$$

By measuring the m ancilla qubits and postselecting on the outcome 0^m, we still obtain the normalized state $\frac{A|b\rangle}{\|A|b\rangle\|}$. The number α is hidden in the success probability:

$$p(0^m) = \frac{1}{\alpha^2} \|A|b\rangle\|^2 = \frac{1}{\alpha^2} \langle b|A^\dagger A|b\rangle . \tag{9.10}$$

So if α is chosen to be too large, the probability of obtaining all 0's from the measurement can be vanishingly small.

Finally, it can be difficult to find U_A to block encode A exactly. This is not a problem, since it is sufficient if we can find U_A to block encode A up to some error ϵ. We are now ready to give the definition of block encoding in Definition 9.2.

Definition 9.2 (**Block encoding**). *Given an n-qubit matrix A, if we can find $\alpha, \epsilon \in \mathbb{R}_+$, and an $(m+n)$-qubit unitary matrix U_A so that*

$$\|A - \alpha \langle 0^m | U_A | 0^m \rangle\| \leq \epsilon, \tag{9.11}$$

then U_A is called an (α, m, ϵ)-block-encoding of A. When the block encoding is exact with $\epsilon = 0$, U_A is called an (α, m)-block-encoding of A. The set of all (α, m, ϵ)-block-encodings of A is denoted by $\mathrm{BE}_{\alpha,m}(A, \epsilon)$. The parameter α is referred to as the block encoding factor, or the subnormalization factor.

When discussing block encodings, we often ignore certain errors such as the error in the finite precision number representation. We define a shorthand notation $\mathrm{BE}_{\alpha,m}(A) = \mathrm{BE}_{\alpha,m}(A, 0)$. Assume we know each matrix element of the n-qubit matrix A_{ij}, and we are given an $(n+m)$-qubit unitary U_A. In order to verify that $U_A \in \mathrm{BE}_{1,m}(A)$, we only need to verify that

$$\langle 0^m, i | U_A | 0^m, j \rangle = A_{ij}, \tag{9.12}$$

and U_A applied to any vector $|0^m, b\rangle$ can be obtained via the superposition principle.

Therefore we may first evaluate the state $U_A |0^m, j\rangle$, perform an inner product with $|0^m, i\rangle$, and verify the resulting inner product is A_{ij}. We will also use the following technique frequently. Assume $U_A = U_B U_C$, and then

$$\langle 0^m, i | U_A | 0^m, j \rangle = \langle 0^m, i | U_B U_C | 0^m, j \rangle = (U_B^\dagger |0^m, i\rangle)^\dagger (U_C |0^m, j\rangle). \tag{9.13}$$

So we can evaluate the states $U_B^\dagger |0^m, i\rangle, U_C |0^m, j\rangle$ independently, and then verify the inner product is A_{ij}. Such a calculation amounts to running the circuit Fig. 9.2, and if the ancilla qubits are measured to be 0^m, the system qubits return the normalized state $\sum_i A_{ij} |i\rangle / \|\sum_i A_{ij} |i\rangle\|$.

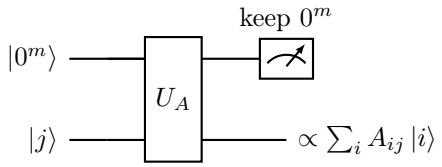


FIGURE 9.2. Circuit for general block encoding of A.

Example 9.3. For any n-qubit matrix A with $\|A\| \leq 1$ with singular value decomposition $A = W\Sigma V^\dagger$, all singular values in the diagonal matrix Σ are in $[0, 1]$. Then we may construct an $(n+1)$-qubit unitary matrix ($N = 2^n$)

$$\begin{aligned} U_A :=& \begin{pmatrix} W & 0 \\ 0 & I_N \end{pmatrix} \begin{pmatrix} \Sigma & \sqrt{I_N - \Sigma^2} \\ \sqrt{I_N - \Sigma^2} & -\Sigma \end{pmatrix} \begin{pmatrix} V^\dagger & 0 \\ 0 & I_N \end{pmatrix} \\ =& \begin{pmatrix} A & W\sqrt{I_N - \Sigma^2} \\ \sqrt{I_N - \Sigma^2} V^\dagger & -\Sigma \end{pmatrix} \end{aligned} \tag{9.14}$$

which is a $(1, 1)$-block-encoding of A. ◇

Example 9.3 shows that in principle, any matrix A with $\|A\| \leq 1$ can be accessed via a $(1, 1, 0)$-block-encoding. However, this construction does not state how to construct A using simple one and two qubit gates.

Example 9.4 (Random circuit block encoded matrix)**.** How can we construct a pseudo-random non-unitary matrix on a quantum computer? A naive approach would be to generate a dense pseudo-random matrix A classically and then encode it into a quantum circuit. However, this is highly inefficient in practice, particularly for large matrices, due to the exponential overhead in loading dense classical data into a quantum system.

Instead, we seek to work with matrices that are inherently easy to generate within a quantum circuit model. This motivates the **random circuit based block-encoded matrix** (RACBEM) model. Rather than first constructing a matrix A and then searching for a block-encoding unitary U_A, the RACBEM model reverses the thought process: we begin by constructing a unitary U_A that is easy to implement on a quantum computer, typically using random quantum circuits, and then extract A as a subblock of U_A. This provides a practical and scalable way to generate structured pseudo-random non-unitary matrices compatible with quantum algorithm design. Similar to the LINPACK benchmark, which is used to rank classical supercomputers in the TOP500 list by solving $Ax = b$ for pseudorandom matrices A, such block-encoded pseudorandom matrices can serve as a useful tool for benchmarking scientific computing applications on quantum computers. ◇

9.2. Linear combination of unitaries

The **linear combination of unitaries** (LCU) is an important quantum primitive, which allows quantum algorithms to be implemented as a superposition of unitary matrices rather than

attempting to find a single unitary that accomplishes a desired task. This often simplifies the design and analysis of quantum algorithms. LCU can also be viewed as a special way for constructing block encoding. Combined with a technique called qubitization, which will be discussed in detail in Chapter 10, LCU can be used to implement a large class of matrix functions (eigenvalue transformations) and generalized matrix functions (singular value transformations).

Let $T = \sum_{i=0}^{K-1} \alpha_i U_i$ be a linear combination of unitary matrices U_i. For simplicity let $K = 2^a$. Then

$$U_{\mathrm{SEL}} := \sum_{i\in[K]} |i\rangle\langle i| \otimes U_i, \tag{9.15}$$

implements the selection of U_i conditioned on the value of the a-qubit ancilla register (also called the control register). U_{SEL} is called a **select oracle**.

If all linear combination coefficients $\alpha_i \geq 0$, we can let V_{PREP} be a unitary operation satisfying

$$V_{\mathrm{PREP}} |0^a\rangle = \frac{1}{\sqrt{\|\alpha\|_1}} \sum_{i\in[K]} \sqrt{\alpha_i} |i\rangle , \tag{9.16}$$

which is called a **prepare oracle**. The 1-norm of the coefficients is given by

$$\|\alpha\|_1 = \sum_i |\alpha_i| . \tag{9.17}$$

In matrix form,

$$V_{\mathrm{PREP}} = \frac{1}{\sqrt{\|\alpha\|_1}} \begin{pmatrix} \sqrt{\alpha_0} & * & \cdots & * \\ \vdots & * & \ddots & \vdots \\ \sqrt{\alpha_{K-1}} & * & \cdots & * \end{pmatrix}. \tag{9.18}$$

where the first column is $V_{\mathrm{PREP}} |0^a\rangle$, and all other columns are orthogonal to it. Then

$$V_{\mathrm{PREP}}^\dagger = \frac{1}{\sqrt{\|\alpha\|_1}} \begin{pmatrix} \sqrt{\alpha_0} & \cdots & \sqrt{\alpha_{K-1}} \\ * & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & * \end{pmatrix}. \tag{9.19}$$

More generally, we can arbitrarily decompose $\alpha_i = \beta_i \gamma_i$, so that

$$V_{\mathrm{PREP}} = \frac{1}{\|\beta\|_2} \begin{pmatrix} \beta_0 & * & \cdots & * \\ \vdots & * & \ddots & \vdots \\ \beta_{K-1} & * & \cdots & * \end{pmatrix}, \quad \widetilde{V}_{\mathrm{PREP}} = \frac{1}{\|\gamma\|_2} \begin{pmatrix} \gamma_0 & \cdots & \gamma_{K-1} \\ * & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & * \end{pmatrix} \tag{9.20}$$

are unitaries and can be efficiently implemented. When $\alpha_i \geq 0$, we can choose $\beta_i = \gamma_i = \sqrt{\alpha_i}$ which gives $\widetilde{V}_{\mathrm{PREP}} = V_{\mathrm{PREP}}^\dagger$. Then T can be implemented using the unitary given in Lemma 9.5.

Lemma 9.5 (Linear combination of unitaries)**.** *For*

$$T = \sum_{i=0}^{K-1} \alpha_i U_i, \quad \alpha_i = \beta_i \gamma_i, \quad K = 2^a, \quad U_i \in \mathrm{U}(2^n), \tag{9.21}$$

let $U_{\mathrm{SEL}}, V_{\mathrm{PREP}}, \widetilde{V}_{\mathrm{PREP}}$ *be given in Eqs.* (9.15) *and* (9.20)*, respectively. Define*

$$W = (\widetilde{V}_{\mathrm{PREP}} \otimes I_n) U_{\mathrm{SEL}} (V_{\mathrm{PREP}} \otimes I_n) \tag{9.22}$$

as implemented in Fig. 9.3. Then $W \in \mathrm{BE}_{\|\beta\|_2\|\gamma\|_2,a}(T)$. *The smallest subnormalization factor is obtained by setting*

$$|\beta_i| = |\gamma_i| = \sqrt{|\alpha_i|}, \quad i \in [K], \tag{9.23}$$

and $W \in \mathrm{BE}_{\|\alpha\|_1,a}(T)$.

Proof. For any n-qubit state $|\psi\rangle$,

$$U_{\mathrm{SEL}}(V_{\mathrm{PREP}} \otimes I_n) |0^a\rangle |\psi\rangle = U_{\mathrm{SEL}} \frac{1}{\|\beta\|_2} \sum_i \beta_i |i\rangle |\psi\rangle = \frac{1}{\|\beta\|_2} \sum_i \beta_i |i\rangle U_i |\psi\rangle . \tag{9.24}$$

Let the state $|\widetilde{\perp}\rangle$ collect all the states marked by $*$ orthogonal to $|0^a\rangle$, and use $\beta_i\gamma_i = \alpha_i$,

$$(\widetilde{V}_{\mathrm{PREP}}\otimes I_n)U_{\mathrm{SEL}}(V_{\mathrm{PREP}}\otimes I_n) |0^a\rangle |\psi\rangle = \frac{1}{\|\beta\|_2 \|\gamma\|_2} |0^a\rangle \sum_i \alpha_i U_i |\psi\rangle + |\widetilde{\perp}\rangle = \frac{1}{\|\beta\|_2 \|\gamma\|_2} |0^a\rangle T |\psi\rangle + |\widetilde{\perp}\rangle . \tag{9.25}$$

Use Cauchy-Schwarz

$$\|\alpha\|_1 = \sum_i |\alpha_i| = \sum_i |\beta_i\gamma_i| \le \|\beta\|_2 \|\gamma\|_2 , \tag{9.26}$$

we find that the optimal prepare oracle should satisfy $|\beta_i| = |\gamma_i| = \sqrt{|\alpha_i|}, \forall i$. □

The LCU Lemma states that the number of ancilla qubits needed only depends logarithmically on K, the number of terms in the linear combination. Hence it is possible to implement the linear combination of a very large number of terms efficiently. From a practical perspective, the select and prepare oracles use multi-qubit controls, and may be difficult to implement themselves. Furthermore, if the select and prepare oracles are implemented directly, the number of multi-qubit controls again depends linearly on K and is not desirable. Therefore an efficient implementation using LCU (in terms of the gate complexity) also requires additional structure in these oracles.

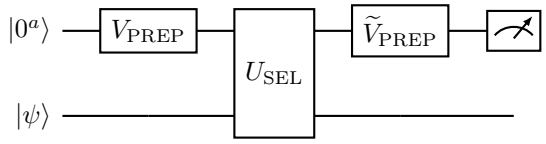


Figure 9.3. Circuit for linear combination of unitaries. When all coefficients are nonnegative, we may set $\widetilde{V}_{\mathrm{PREP}} = V_{\mathrm{PREP}}^\dagger$.

An important application of LCU is that if A, B can be accessed via their block encodings, then we can construct a block encoding of the matrix addition $A + B$.

Example 9.6 (Linear combination of two block encoded matrices)**.** Let U_A, U_B be two n-qubit unitaries, and we would like to construct a block encoding of $T = U_A + U_B$.

There are two terms in total, so one ancilla qubit is needed. The prepare oracle needs to implement

$$V_{\mathrm{PREP}} |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \tag{9.27}$$

so this is the Hadamard gate. The circuit is given by Fig. 9.4, which constructs $W \in \mathrm{BE}_{2,1}(T)$.

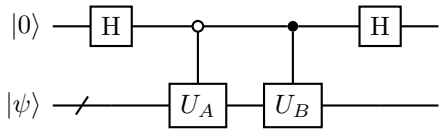


FIGURE 9.4. Circuit for linear combination of two unitaries.

◇

Exercise 9.2. Let A, B be two n-qubit matrices encoded by $U_A \in \mathrm{BE}_{1,m}(A), U_B \in \mathrm{BE}_{1,m}(B)$. Construct a circuit to block encode $C = A + B$. What about $U_A \in \mathrm{BE}_{\alpha_A,m}(A), U_B \in \mathrm{BE}_{\alpha_B,m}(B)$?

Exercise 9.3. Consider a system described by the linear combination $T = X + Y + 2Z$, where X, Y, Z are the Pauli matrices. Construct a select oracle U for this system, and describe how to use the LCU technique to construct a block encoding of T.

Example 9.7. Consider the following TFIM model with periodic boundary conditions ($Z_n = Z_0$), and $n = 2^{\mathfrak{n}}$,

$$\hat{H} = -\sum_{i\in[n]} Z_i Z_{i+1} - \sum_{i\in[n]} X_i. \tag{9.28}$$

In order to use LCU, we need $(\mathfrak{n}+1)$ ancilla qubits. In this case, the prepare oracle can be simply constructed from the Hadamard gate

$$V_{\mathrm{PREP}} = \mathrm{H}^{\otimes(\mathfrak{n}+1)}, \tag{9.29}$$

and the select oracle implements

$$U_{\mathrm{SEL}} = \sum_{i\in[n]} |i\rangle\langle i| \otimes (-Z_i Z_{i+1}) + \sum_{i\in[n]} |i+n\rangle\langle i+n| \otimes (-X_i). \tag{9.30}$$

The corresponding $W \in \mathrm{BE}_{2n,\mathfrak{n}+1}(\hat{H})$. ◇

Example 9.8 (Highly oscillatory integral)**.** Consider evaluating the matrix integral $\int_0^1 A(s)\,ds$, where $A(s) \in \mathbb{C}^{2^n\times 2^n}$, $A(0) = A(1)$ and $\sup_{s\in[0,1]} \|A(s)\| \le 1$. Given that the entries of $A(s)$ exhibit significant oscillations as a function of s, in general there is no known efficient method (classical or quantum) to compute this integral without using a sufficiently fine grid and numerical quadrature. For simplicity, we adopt a uniform grid defined by $\{s_k = \frac{k}{M}\}_{k=0}^{M}$, where M is sufficiently large, to implement the quadrature method.

$$\int_0^1 A(s)\,\mathrm{d}s = \frac{1}{M}\sum_{k=0}^{M-1} A(k/M) + E, \quad \|E\| \le \epsilon. \tag{9.31}$$

For each s, assume that $A(s)$ has a $(1, a, 0)$-block encoding denoted by $U_{A(s)}$, and the s-dependence can be implemented coherently using e.g., classical arithmetic operations. In a discretized setting, let $M = 2^m$, this means that the following select oracle defined on a register with $m + a + n$ qubits:

$$U_{\mathrm{SEL}} = \sum_{k=0}^{M-1} |k\rangle\langle k| \otimes U_{A(k/M)}, \tag{9.32}$$

which we assume can be efficiently implemented with cost $\mathrm{poly}(mn)$. The prepare oracle is simply the m-qubit Hadamard gate $\mathrm{H}^{\otimes m}$. Then the circuit $(\mathrm{H}^{\otimes m} \otimes I_{a+n}) U_{\mathrm{SEL}} (\mathrm{H}^{\otimes m} \otimes I_{a+n})$ is a $(1, a+m, \epsilon)$-block encoding of the matrix-valued integral $\int_0^1 A(s)\,\mathrm{d}s$. It uses m ancilla qubits, and the gate complexity is dominated by that of the select oracle and is $\mathrm{poly}(mn)$. This is an exponential improvement in the parameter M for constructing such a block encoding, compared to a direct classical quadrature implementation whose cost is at least linear in M. ◇

Example 9.9 (Fourier transformation and eigenvalue transformation)**.** With a subroutine performing Hamiltonian simulation, we can combine it with LCU to implement matrix functions expressed as a matrix Fourier series. Let H be an n-qubit Hermitian matrix. Consider $f(x) \in \mathbb{R}$ given by its Fourier expansion (up to a normalization factor)

$$f(x) = \int \hat{f}(k) e^{ikx}\,\mathrm{d}k, \tag{9.33}$$

and we are interested in computing the matrix function via numerical quadrature

$$f(H) = \int \hat{f}(k) e^{ikH}\,\mathrm{d}k \approx \Delta k \sum_{k \in \mathcal{K}} \hat{f}(k) e^{ikH}. \tag{9.34}$$

Here $\mathcal{K}$ is a uniform grid discretizing the interval $[-L, L]$ using $|\mathcal{K}| = 2^{\mathfrak{k}}$ grid points, and the grid spacing is $\Delta k = 2L/|\mathcal{K}|$. The prepare oracle is given by the coefficients $c_k = \Delta k \hat{f}(k)$, and the corresponding subnormalization factor is

$$\|c\|_1 = \sum_{k \in \mathcal{K}} \Delta k \left|\hat{f}(k)\right| \approx \int \left|\hat{f}(k)\right| \mathrm{d}k. \tag{9.35}$$

The select oracle is

$$U_{\mathrm{SELECT}} = \sum_{k \in \mathcal{K}} |k\rangle\langle k| \otimes e^{ikH}. \tag{9.36}$$

This can be efficiently implemented using the controlled matrix powers as in Fig. 16.2, where the basic unit is the short time Hamiltonian simulation $e^{i\Delta k H}$. This can be used to block encode a large class of matrix functions. ◇

9.3. Block encodings of matrix additions and multiplications

We now record basic composition rules for block encodings that will be used throughout the book.

The linear combination of unitaries (LCU) construction from Section 9.2 immediately yields a block encoding of a sum of block-encoded matrices. For simplicity, we state the result for $M = 2^m$ summands.

Proposition 9.10 (Sum of M block-encoded matrices)**.** *Let $M = 2^m$ and let $A_0, \ldots, A_{M-1}$ be matrices of the same dimension. Assume that for each $j \in [M]$ we are given a block encoding*

$$U_{A_j} \in \mathrm{BE}_{\alpha_j, a}(A_j, \epsilon_j), \qquad \alpha_j \geq 0. \tag{9.37}$$

Set $\gamma := \sum_{j=0}^{M-1} \alpha_j > 0$. Let $U_{\mathrm{SEL}} := \sum_{j \in [M]} |j\rangle\langle j| \otimes U_{A_j}$ be the select oracle acting on an m-qubit control register, the a-qubit ancilla register, and the system register. Let V_{PREP} be any unitary on

the m-qubit control register satisfying

$$V_{\mathrm{PREP}}\,|0^m\rangle = \frac{1}{\sqrt{\gamma}} \sum_{j=0}^{M-1} \sqrt{\alpha_j}\,|j\rangle\,. \tag{9.38}$$

Define

$$W := (V_{\mathrm{PREP}}^{\dagger} \otimes I)\, U_{\mathrm{SEL}}\, (V_{\mathrm{PREP}} \otimes I). \tag{9.39}$$

Then

$$W \in \mathrm{BE}_{\gamma,a+m}\left(\sum_{j=0}^{M-1} A_j,\ \sum_{j=0}^{M-1} \epsilon_j \right). \tag{9.40}$$

Proof. Write $B_j := \langle 0^a | U_{A_j} | 0^a \rangle$, so that $\|A_j - \alpha_j B_j\| \le \epsilon_j$ and $\|B_j\| \le 1$. By direct computation of the $(|0^m, 0^a\rangle)$ block,

$$\langle 0^m, 0^a | W | 0^m, 0^a \rangle = \sum_{j=0}^{M-1} \frac{\alpha_j}{\gamma}\, B_j. \tag{9.41}$$

Therefore

$$\left\| \sum_{j=0}^{M-1} A_j - \gamma\, \langle 0^m, 0^a | W | 0^m, 0^a \rangle \right\| \le \sum_{j=0}^{M-1} \|A_j - \alpha_j B_j\| \le \sum_{j=0}^{M-1} \epsilon_j, \tag{9.42}$$

which is the claimed block-encoding statement. □

Example 9.11 (Multiplication of block encoded matrices)**.** If A, B are given by their block encodings $U_A \in \mathrm{BE}_{\alpha,a}(A), U_B \in \mathrm{BE}_{\beta,b}(B)$, then the product AB can also be block encoded (see Fig. 9.5), which uses $a+b$ ancilla qubits. This is because $AB/(\alpha\beta) = \langle 0^{a+b} | (U_A \otimes I_b)(I_a \otimes U_B) | 0^{a+b} \rangle$. Hence $(U_A \otimes I_b)(I_a \otimes U_B) \in \mathrm{BE}_{\alpha\beta,a+b}(AB)$.

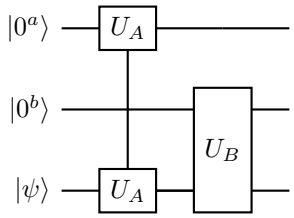


Figure 9.5. Quantum circuit for block encoding the product of matrices using $a + b$ ancilla qubits.

◇

We next record a simple (though not always ancilla-optimal) rule for block encoding a product.

Proposition 9.12 (Product of M block-encoded matrices)**.** *Let $A_0, \dots, A_{M-1}$ be matrices with compatible dimensions. Assume that for each $j \in [M]$ we are given*

$$U_{A_j} \in \mathrm{BE}_{\alpha_j, a_j}(A_j, \epsilon_j). \tag{9.43}$$

Let U be the unitary obtained by applying $U_{A_0}, U_{A_1}, \dots, U_{A_{M-1}}$ sequentially on disjoint ancilla registers (of sizes $a_0, \dots, a_{M-1}$) and a common system register. Then

$$U \in \mathrm{BE}_{\prod_{j=0}^{M-1}\alpha_j,\, \sum_{j=0}^{M-1} a_j}\Big(A_{M-1}\cdots A_0,\ \prod_{j=0}^{M-1}(\alpha_j+\epsilon_j) - \prod_{j=0}^{M-1}\alpha_j\Big). \tag{9.44}$$

Proof. For each j, define $B_j := \langle 0^{a_j}|U_{A_j}|0^{a_j}\rangle$ so that $\|A_j - \alpha_j B_j\| \le \epsilon_j$ and $\|B_j\| \le 1$. Since the ancilla registers are disjoint, we have

$$\langle 0^{a_0+\cdots+a_{M-1}}|U|0^{a_0+\cdots+a_{M-1}}\rangle = B_{M-1}\cdots B_0. \tag{9.45}$$

It remains to bound

$$\left\| A_{M-1}\cdots A_0 - \Big(\prod_{j=0}^{M-1}\alpha_j\Big) B_{M-1}\cdots B_0 \right\|. \tag{9.46}$$

We prove by induction on M the inequality

$$\left\| \prod_{j=0}^{M-1} A_j - \prod_{j=0}^{M-1}(\alpha_j B_j) \right\| \le \prod_{j=0}^{M-1}(\alpha_j+\epsilon_j) - \prod_{j=0}^{M-1}\alpha_j. \tag{9.47}$$

The case $M = 1$ is immediate. For the induction step, write $P := \prod_{j=0}^{M-2} A_j$ and $\widetilde{P} := \prod_{j=0}^{M-2}(\alpha_j B_j)$. Then

$$\begin{aligned} \left\| A_{M-1}P - (\alpha_{M-1}B_{M-1})\widetilde{P} \right\| &\le \|(A_{M-1} - \alpha_{M-1}B_{M-1})P\| + \left\| \alpha_{M-1}B_{M-1}(P-\widetilde{P}) \right\| \\ &\le \epsilon_{M-1}\|P\| + \alpha_{M-1}\left\| P - \widetilde{P} \right\|. \end{aligned} \tag{9.48}$$

Using $\|A_j\| \le \alpha_j + \epsilon_j$ (by $\|A_j\| \le \|A_j - \alpha_j B_j\| + \alpha_j \|B_j\|$), we have $\|P\| \le \prod_{j=0}^{M-2}(\alpha_j+\epsilon_j)$. Applying the induction hypothesis to $\left\|P - \widetilde{P}\right\|$ yields

$$\begin{aligned} \left\| A_{M-1}P - (\alpha_{M-1}B_{M-1})\widetilde{P} \right\| &\le \epsilon_{M-1}\prod_{j=0}^{M-2}(\alpha_j+\epsilon_j) + \alpha_{M-1}\Big(\prod_{j=0}^{M-2}(\alpha_j+\epsilon_j) - \prod_{j=0}^{M-2}\alpha_j\Big) \\ &= \prod_{j=0}^{M-1}(\alpha_j+\epsilon_j) - \prod_{j=0}^{M-1}\alpha_j, \end{aligned} \tag{9.49}$$

completing the induction. □

The procedure in Example 9.11 is not the most efficient way for block encoding the product of matrices. In Example 11.9, we have demonstrated that using deferred measurement, we only need one extra ancilla qubit to record whether the ancilla register is in the all 0 state. Specifically, assume $a = b$ for simplicity; Fig. 9.6 is a schematic circuit (the control denotes a check of the ancilla register being in $|0^a\rangle$) that constructs a unitary in $\mathrm{BE}_{\alpha\beta, a+1}(AB)$.

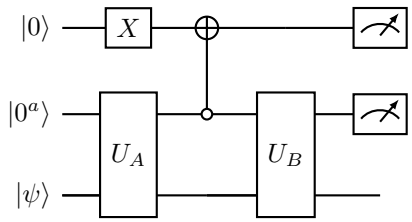


FIGURE 9.6. Quantum circuit for block encoding the product of matrices using $a+1$ ancilla qubits (assuming $a=b$).

Following this strategy, when multiplying L matrices A_i each given by $U_{A_i} \in \mathrm{BE}_{\alpha_i,a}(A_i)$, we can introduce $L-1$ ancilla qubits to obtain a unitary in $\mathrm{BE}_{\prod_{i=1}^{L}\alpha_i,a+L-1}(A_L\cdots A_1)$. Even more efficiently, using the compression gadget in Example 11.10, the number of ancilla qubits can be reduced to $a+\lceil\log_2(L+1)\rceil$.

Note that the matrix power A^L is a special case of multiplying L matrices. However, the method in Example 9.11 for encoding A^L can be highly inefficient. To see this, consider a matrix A with spectral radius

$$\rho(A) = \max\{\,|\lambda| \mid \lambda \in \mathrm{Spec}(A)\,\}, \tag{9.50}$$

where $\mathrm{Spec}(A)$ denotes the set of eigenvalues of A. Suppose that $\rho(A)<1$. Then there exists a constant C such that $\sup_{L\in\mathbb{N}}\left\|A^L\right\| \le C$. However, it is still possible that $\|A\|>1$, which means that the block encoding subnormalization factor of A must satisfy $\alpha \ge \|A\| > 1$. As a result, the subnormalization factor for encoding A^L using the method in Example 9.11 would scale as α^L, growing exponentially with L. This discrepancy in computing matrix powers is closely related to the challenges of solving linear differential equations. This is a topic that will be discussed in Chapter 20.

9.4. Example: implementing generalized measurements

9.5. Example: Quantum error correction as block encoding

9.6. Query models for matrix entries

Throughout the discussion we assume A is an n-qubit, square matrix, and the max norm of A (see Definition 2.45) satisfies $\|A\|_{\max}<1$.

To query the entries of a matrix, one of the most convenient form is to encode the information of the matrix as the amplitude of a known vector, e.g.,

$$O_A|0\rangle|i\rangle|j\rangle = \left(A_{ij}|0\rangle + \sqrt{1-|A_{ij}|^2}\,|1\rangle\right)|i\rangle|j\rangle\,. \tag{9.51}$$

In other words, given $i,j\in[N]$, O_A performs a controlled rotation (controlling on i,j) on the ancilla qubit, which encodes the information in terms of amplitude of $|0\rangle$. We refer to Eq. (9.51) as the **amplitude oracle**.

Example 9.13 (Construction of amplitude oracle)**.** Assume $\|A\|_{\max}<1$ and $A_{ij}\in\mathbb{R}$ for all i,j, and that we have access to a **bit oracle**

$$\widetilde{O}_A|0^{a'}\rangle|i\rangle|j\rangle = |\widetilde{A}_{ij}\rangle|i\rangle|j\rangle\,. \tag{9.52}$$

Here $\widetilde{A}_{ij}$ is a d'-bit fixed point representation of A_{ij}, and the value of $\widetilde{A}_{ij}$ is either computed on-the-fly with a quantum computer, or obtained through an external database using e.g., QRAM in Definition 5.6. Using the classical arithmetic operations (see Section 5.4), we can first convert this oracle into an oracle

$$O'_A |0^d\rangle |i\rangle |j\rangle = |\widetilde{\theta}_{ij}\rangle |i\rangle |j\rangle , \tag{9.53}$$

where $0 \le \widetilde{\theta}_{ij} < 1$, and $\widetilde{\theta}_{ij}$ is a d-bit representation of $\theta_{ij} = \arccos(A_{ij})/\pi$, and with some abuse of notation we redefine $\widetilde{A}_{ij} = \cos(\pi\widetilde{\theta}_{ij})$. This step may require some additional work registers not shown here.

Now using the controlled rotation in Proposition 5.7 and Fig. 5.3, the information of $\widetilde{\theta}_{ij}$ can now be transferred to the amplitude of the ancilla qubit. We should then perform uncomputation and free the work register storing such intermediate information $\widetilde{\theta}_{ij}$. The procedure is as follows

$$\begin{aligned} |0\rangle \underbrace{|0^d\rangle}_{\text{work register}} |i\rangle |j\rangle &\xrightarrow{I_1\otimes O'_A} |0\rangle |\widetilde{\theta}_{ij}\rangle |i\rangle |j\rangle \\ &\xrightarrow{\text{CR}} \left(\widetilde{A}_{ij} |0\rangle + \sqrt{1 - \left|\widetilde{A}_{ij}\right|^2} |1\rangle\right) |\widetilde{\theta}_{ij}\rangle |i\rangle |j\rangle \\ &\xrightarrow{I_1\otimes (O'_A)^{-1}} \left(\widetilde{A}_{ij} |0\rangle + \sqrt{1 - \left|\widetilde{A}_{ij}\right|^2} |1\rangle\right) |0^d\rangle |i\rangle |j\rangle \end{aligned} \tag{9.54}$$

After the uncomputation, the d-bit working register can be discarded, and we obtain the desired amplitude oracle of the input matrix A. ◇

Exercise 9.4. Construct a query oracle O_A similar to that in Eq. (9.54), when $A_{ij} \in \mathbb{C}$ with $\|A\|_{\max} < 1$.

9.7. Block encoding of s-sparse matrices

Example 9.14 (Block encoding of a diagonal matrix)**.** As a special case, let us consider the block encoding of a diagonal matrix, which is also a 1-sparse matrix. Since the row and column indices are the same, we may simplify the oracle Eq. (9.51) into

$$O_A |0\rangle |i\rangle = \left(A_{ii} |0\rangle + \sqrt{1 - |A_{ii}|^2} |1\rangle\right) |i\rangle . \tag{9.55}$$

Let $U_A = O_A$. Direct calculation shows that for any $i, j \in [N]$,

$$\langle 0| \langle i| U_A |0\rangle |j\rangle = A_{ii}\delta_{ij}. \tag{9.56}$$

This proves that $U_A \in \mathrm{BE}_{1,1}(A)$, i.e., U_A is a $(1,1)$-block-encoding of the diagonal matrix A. ◇

Example 9.15 (General 1-sparse matrices)**.** In a 1-sparse matrices, there is only one nonzero entry in each row and each column of the matrix. This means that for each $j \in [N]$, there is a unique $c(j) \in [N]$ such that $A_{c(j),j} \neq 0$, and the mapping c is a permutation. Assume that there exists a unitary O_c satisfying that

$$O_c |j\rangle = |c(j)\rangle , \quad O_c^\dagger |c(j)\rangle = |j\rangle . \tag{9.57}$$

The implementation of O_c may require the usage of some work registers that are omitted here.

We assume the matrix entry $A_{c(j),j}$ can be queried via

$$O_A |0\rangle |j\rangle = \left(A_{c(j),j} |0\rangle + \sqrt{1 - \left|A_{c(j),j}\right|^2} |1\rangle \right) |j\rangle . \tag{9.58}$$

Now we construct $U_A = (I \otimes O_c)O_A$, and compute the matrix element

$$\langle 0| \langle i| U_A |0\rangle |j\rangle = \langle 0| \langle i| \left(A_{c(j),j} |0\rangle + \sqrt{1 - \left|A_{c(j),j}\right|^2} |1\rangle \right) |c(j)\rangle = A_{c(j),j}\delta_{i,c(j)}. \tag{9.59}$$

This proves that $U_A \in \mathrm{BE}_{1,1}(A)$. ◇

For a general s-sparse matrix, we have $\|A\| \le s \|A\|_{\max}$ according to Lemma 2.46, and the explicit construction of a block encoding below requires us to choose the subnormalization factor to be $\alpha = s \|A\|_{\max}$. Without loss of generality, we may assume each row and each column has exactly s designated entries by padding with zeros. For simplicity, let $s = 2^{\mathfrak{s}}$. Also we assume $\|A\|_{\max} = 1$ and will set $\alpha = s$.

Let us consider the construction of a block encoding for an s-sparse matrix by decomposing A into a sum of 1-sparse matrices. View the sparsity pattern of A as a bipartite graph with left vertices labeled by row indices and right vertices labeled by column indices, where an edge (i, j) is present if $A_{ij} \neq 0$. After padding with zero entries so that each row and each column has exactly s incident edges, the resulting bipartite graph is s-regular. By König's line-coloring theorem (see e.g., [Die25, Proposition 5.3.1]), the edges of an s-regular bipartite graph can be decomposed into s disjoint perfect matchings. Fix such a decomposition and index the matchings by $\ell \in [s]$. For each ℓ and each column j, let $c(j, \ell)$ denote the unique row index matched to j in the ℓ-th matching. Then, for each fixed ℓ, the mapping $j \mapsto c(j, \ell)$ is a permutation of $[N]$.

For each $\ell \in [s]$, define $A^{(\ell)}$ by setting $A^{(\ell)}_{c(j,\ell),j} = A_{c(j,\ell),j}$ and all other entries of $A^{(\ell)}$ to zero. Then each $A^{(\ell)}$ is 1-sparse, and

$$A = \sum_{\ell \in [s]} A^{(\ell)}. \tag{9.60}$$

According to Example 9.15, we may access each permutation $j \mapsto c(j, \ell)$ via a unitary oracle. Packaging all ℓ together, we assume access to a unitary O_c such that

$$O_c |\ell\rangle |j\rangle = |\ell\rangle |c(j, \ell)\rangle . \tag{9.61}$$

Similarly, we assume that the normalized matrix entries can be queried via

$$O_A |0\rangle |\ell\rangle |j\rangle = \left(A_{c(j,\ell),j} |0\rangle + \sqrt{1 - \left|A_{c(j,\ell),j}\right|^2} |1\rangle \right) |\ell\rangle |j\rangle . \tag{9.62}$$

We then define $D = \mathrm{H}^{\otimes \mathfrak{s}}$ (the $\mathfrak{s}$-qubit Hadamard transform) satisfying

$$D |0^{\mathfrak{s}}\rangle = \frac{1}{\sqrt{s}} \sum_{\ell \in [s]} |\ell\rangle . \tag{9.63}$$

With these ingredients, the circuit in Fig. 9.7 can be interpreted as an LCU-type construction over $\{A^{(\ell)}\}_{\ell \in [s]}$, yielding a block encoding with subnormalization factor $\alpha = s$.

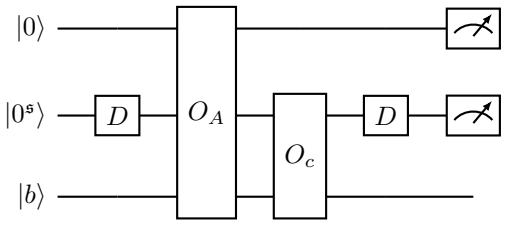


FIGURE 9.7. Quantum circuit for block encoding a s-sparse matrix using linear combination of unitaries. The measurement means that to obtain a state $\propto A|b\rangle$, the ancilla register should all return the value 0.

We now show that the circuit in Fig. 9.7 defines a unitary $U_A \in \mathrm{BE}_{s,\mathfrak{s}+1}(A)$ by direct calculation.

Proposition 9.16. *The circuit in Fig. 9.7 defines $U_A \in \mathrm{BE}_{s,\mathfrak{s}+1}(A)$.*

PROOF. We may write

$$U_A = (I \otimes D \otimes I)(I \otimes O_c)O_A(I \otimes D \otimes I). \tag{9.64}$$

In order to compute the inner product $\langle 0|\langle 0^{\mathfrak{s}}|\langle i|U_A|0\rangle|0^{\mathfrak{s}}\rangle|j\rangle$, we apply D, O_A, O_c to $|0\rangle|0^{\mathfrak{s}}\rangle|j\rangle$ successively as

$$\begin{aligned} |0\rangle|0^{\mathfrak{s}}\rangle|j\rangle &\xrightarrow{D} \frac{1}{\sqrt{s}}\sum_{\ell\in[s]}|0\rangle|\ell\rangle|j\rangle \\ &\xrightarrow{O_A} \frac{1}{\sqrt{s}}\sum_{\ell\in[s]}\left(A_{c(j,\ell),j}|0\rangle + \sqrt{1-\left|A_{c(j,\ell),j}\right|^2}|1\rangle\right)|\ell\rangle|j\rangle \\ &\xrightarrow{O_c} \frac{1}{\sqrt{s}}\sum_{\ell\in[s]}\left(A_{c(j,\ell),j}|0\rangle + \sqrt{1-\left|A_{c(j,\ell),j}\right|^2}|1\rangle\right)|\ell\rangle|c(j,\ell)\rangle\,. \end{aligned} \tag{9.65}$$

Instead of multiplying the leftmost factor $I \otimes D \otimes I$ to the last line, we apply it to $|0\rangle|0^{\mathfrak{s}}\rangle|i\rangle$ first to obtain (note that D is Hermitian)

$$|0\rangle|0^{\mathfrak{s}}\rangle|i\rangle \xrightarrow{D} \frac{1}{\sqrt{s}}\sum_{\ell'\in[s]}|0\rangle|\ell'\rangle|i\rangle\,. \tag{9.66}$$

Finally, taking the inner product yields

$$\langle 0|\langle 0^{\mathfrak{s}}|\langle i|U_A|0\rangle|0^{\mathfrak{s}}\rangle|j\rangle = \frac{1}{s}\sum_{\ell}A_{c(j,\ell),j}\delta_{i,c(j,\ell)} = \frac{1}{s}A_{ij}. \tag{9.67}$$

□

Example 9.17. Let us use the circuit in Fig. 9.7 to construct a block encoding of

$$A = \begin{bmatrix}\alpha_1 & \alpha_2\\ \alpha_2 & \alpha_1\end{bmatrix}, \quad 0 \le \alpha_i \le 1, \quad i = 1, 2. \tag{9.68}$$

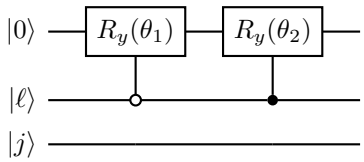
This matrix satisfies $\|A\|_{\max} = 1$, and can be viewed as a 2-sparse matrix. We can simply use CNOT as the O_c circuit by examining the truth table

ℓ	j
0	0
0	1
1	0
1	1

$\rightarrow$

ℓ	$c(j,\ell)$
0	0
0	1
1	1
1	0

Meanwhile O_A can be implemented using controlled $R_y(\theta_i), \theta_i = 2\arccos(\alpha_i), i = 1, 2$.



For example, when $\alpha_1 = 1, \alpha_2 = 0.5$, The resulting matrix is

$$U_A = \begin{pmatrix} 0.500 & 0.250 & 0.500 & -0.250 & 0.0 & -0.433 & 0.0 & 0.433 \\ 0.250 & 0.500 & -0.250 & 0.500 & -0.433 & 0.0 & 0.433 & 0.0 \\ 0.500 & -0.250 & 0.500 & 0.250 & 0.0 & 0.433 & 0.0 & -0.433 \\ -0.250 & 0.500 & 0.250 & 0.500 & 0.433 & 0.0 & -0.433 & 0.0 \\ 0.0 & 0.433 & 0.0 & -0.433 & 0.500 & 0.250 & 0.500 & -0.250 \\ 0.433 & 0.0 & -0.433 & 0.0 & 0.250 & 0.500 & -0.250 & 0.500 \\ 0.0 & -0.433 & 0.0 & 0.433 & 0.500 & -0.250 & 0.500 & 0.250 \\ -0.433 & 0.0 & 0.433 & 0.0 & -0.250 & 0.500 & 0.250 & 0.500 \end{pmatrix}. \tag{9.69}$$

This is a $(2, 2)$-block-encoding of A. ◇

Example 9.18 (Banded matrix)**.** A banded matrix of bandwidth s can be defined to have the sparsity pattern

$$c(j, \ell) = j + \ell - \ell_0 \pmod N, \tag{9.70}$$

for some shift $\ell_0 \in \mathbb{Z}$. The O_c circuit in Eq. (9.61) can be constructed using an adder circuit to perform the addition operation. ◇

Let us consider a different input model to construct the block encoding of a general s-sparse matrices. We assume access to the following two $(2n)$-qubit oracles

$$\begin{aligned} O_r |\ell\rangle |i\rangle &= |r(i, \ell)\rangle |i\rangle , \\ O_c |\ell\rangle |j\rangle &= |c(j, \ell)\rangle |j\rangle . \end{aligned} \tag{9.71}$$

Here $r(i, \ell), c(j, \ell)$ gives the ℓ-th nonzero entry in the i-th row and j-th column, respectively. It should be noted that although the index $\ell \in [s]$, we should expand it into an n-qubit state (e.g. let ℓ take the last $\mathfrak{s}$ qubits of the n-qubit register following the binary representation of integers).

We assume that the matrix entries are queried using the following oracle using controlled rotations

$$O_A |0\rangle |i\rangle |j\rangle = \left(A_{ij} |0\rangle + \sqrt{1 - |A_{ij}|^2} |1\rangle \right) |i\rangle |j\rangle , \tag{9.72}$$

where the rotation is controlled by both row and column indices. However, if $A_{ij} = 0$ for some i, j, the rotation can be arbitrary, as there will be no contribution due to the usage of O_r, O_c.

Proposition 9.19. *Fig. 9.8 defines* $U_A \in \mathrm{BE}_{s,n+1}(A)$.

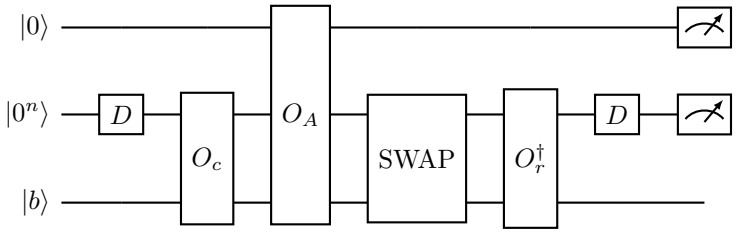


FIGURE 9.8. Quantum circuit for block encoding of general sparse matrices. The measurement means that to obtain a state $\propto A|b\rangle$, the ancilla register should all return the value 0.

PROOF. We apply the first four gate sets to the source state

$$\begin{aligned} &|0\rangle|0^n\rangle|j\rangle \\ \xrightarrow{D,O_c,O_A}&\frac{1}{\sqrt{s}}\sum_{\ell\in[s]}\left(A_{c(j,\ell),j}|0\rangle+\sqrt{1-\left|A_{c(j,\ell),j}\right|^2}|1\rangle\right)|c(j,\ell)\rangle|j\rangle \\ \xrightarrow{\text{SWAP}}&\frac{1}{\sqrt{s}}\sum_{\ell\in[s]}\left(A_{c(j,\ell),j}|0\rangle+\sqrt{1-\left|A_{c(j,\ell),j}\right|^2}|1\rangle\right)|j\rangle|c(j,\ell)\rangle. \end{aligned} \tag{9.73}$$

We then apply D and O_r to the target state

$$|0\rangle|0^n\rangle|i\rangle \xrightarrow{D,O_r} \frac{1}{\sqrt{s}}\sum_{\ell'\in[s]}|0\rangle|r(i,\ell')\rangle|i\rangle. \tag{9.74}$$

Then the inner product gives

$$\begin{aligned} \langle 0|\langle 0^n|\langle i|U_A|0\rangle|0^n\rangle|j\rangle =&\frac{1}{s}\sum_{\ell,\ell'}A_{c(j,\ell),j}\delta_{i,c(j,\ell)}\delta_{r(i,\ell'),j} \\ =&\frac{1}{s}\sum_{\ell}A_{c(j,\ell),j}\delta_{i,c(j,\ell)}=\frac{1}{s}A_{ij}. \end{aligned} \tag{9.75}$$

If $A_{ij}\neq 0$, then there exists a unique ℓ such that $i=c(j,\ell)$ and a unique ℓ' such that $j=r(i,\ell')$; if $A_{ij}=0$, then the same computation gives $\langle 0|\langle 0^n|\langle i|U_A|0\rangle|0^n\rangle|j\rangle=0$. □

9.8. Hermitian block encoding

So far we have considered general s-sparse matrices. Note that if A is a Hermitian matrix, its (α,m,ϵ)-block-encoding U_A does not need to be Hermitian. Even if $\epsilon=0$, we only have that the upper-left n-qubit block of U_A is Hermitian. For instance, even the block encoding of a Hermitian, diagonal matrix in Example 9.14 may not be Hermitian. On the other hand, there are cases when $U_A=U_A^\dagger$ is a Hermitian matrix, and hence the definition:

Definition 9.20 (Hermitian block encoding)**.** *Let U_A be an (α,m,ϵ)-block-encoding of A. If U_A is also Hermitian, then it is called an (α,m,ϵ)-Hermitian-block-encoding of A. When $\epsilon=0$, it is called an (α,m)-Hermitian-block-encoding. The set of all (α,m,ϵ)-Hermitian-block-encodings of A is denoted by $\mathrm{HBE}_{\alpha,m}(A,\epsilon)$, and we define $\mathrm{HBE}_{\alpha,m}(A)=\mathrm{HBE}_{\alpha,m}(A,0)$.*

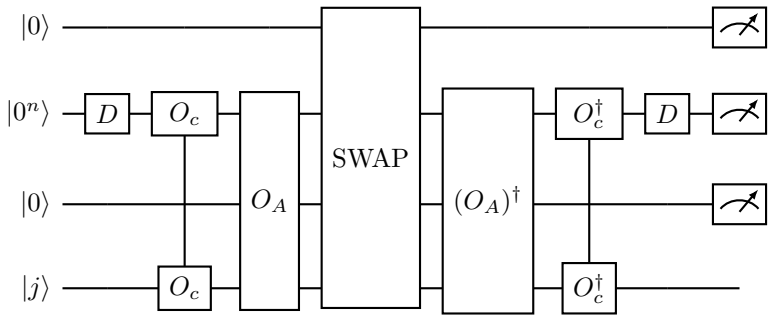


FIGURE 9.9. Quantum circuit for Hermitian block encoding of a general Hermitian matrix

The Hermitian block encoding provides the simplest scenario of the qubitization process in Section 10.2.1.

Next we consider the Hermitian block encoding of an s-sparse Hermitian matrix. Since A is Hermitian, we only need one oracle to query the location of the nonzero entries

$$O_c |\ell\rangle |j\rangle = |c(j,\ell)\rangle |j\rangle . \tag{9.76}$$

Here $c(j,\ell)$ gives the ℓ-th nonzero entry in the j-th column. It can also be interpreted as the ℓ-th nonzero entry in the j-th row. Again the first register needs to be interpreted as an n-qubit register. The operator D is the same as in **??**.

Unlike all discussions before, we introduce two control qubits, and a quantum state in the computational basis takes the form $|a\rangle |i\rangle |b\rangle |j\rangle$, where $a, b \in \{0,1\}, i, j \in [N]$. In other words, we may view $|a\rangle |i\rangle$ as the first register, and $|b\rangle |j\rangle$ as the second register. The $(n+1)$-qubit SWAP gate is defined as

$$\mathrm{SWAP} |a\rangle |i\rangle |b\rangle |j\rangle = |b\rangle |j\rangle |a\rangle |i\rangle . \tag{9.77}$$

To query matrix entries, we need access to the square root of A_{ij} as (note that act on the second single-qubit register)

$$O_A |i\rangle |0\rangle |j\rangle = |i\rangle \left(\sqrt{A_{ij}} |0\rangle + \sqrt{1 - |A_{ij}|} |1\rangle \right) |j\rangle . \tag{9.78}$$

Throughout we assume $\|A\|_{\max} \leq 1$, so that the right-hand side is normalized. The square root operation is well defined if $A_{ij} \geq 0$ for all entries. If A has negative (or complex) entries, we first write $A_{ij} = |A_{ij}| e^{i\theta_{ij}}, \theta_{ij} \in [0, 2\pi)$, and the square root is uniquely defined as $\sqrt{A_{ij}} = \sqrt{|A_{ij}|} e^{i\theta_{ij}/2}$.

Proposition 9.21. *Fig. 9.9 defines* $U_A \in \mathrm{HBE}_{s,n+2}(A)$.

PROOF. Apply the first four gate sets to the source state gives

$$\begin{aligned}
&|0\rangle|0^n\rangle|0\rangle|j\rangle \xrightarrow{D}\xrightarrow{O_c}\\
&\xrightarrow{O_A}\frac{1}{\sqrt{s}}\sum_{\ell\in[s]}|0\rangle|c(j,\ell)\rangle\left(\sqrt{A_{c(j,\ell),j}}|0\rangle+\sqrt{1-|A_{c(j,\ell),j}|}|1\rangle\right)|j\rangle\\
&\xrightarrow{\text{SWAP}}\frac{1}{\sqrt{s}}\sum_{\ell\in[s]}\left(\sqrt{A_{c(j,\ell),j}}|0\rangle+\sqrt{1-|A_{c(j,\ell),j}|}|1\rangle\right)|j\rangle|0\rangle|c(j,\ell)\rangle
\end{aligned}\tag{9.79}$$

Apply the last three gate sets to the target state

$$\begin{aligned}
&|0\rangle|0^n\rangle|0\rangle|i\rangle \xrightarrow{D}\xrightarrow{O_c}\\
&\xrightarrow{O_A}\frac{1}{\sqrt{s}}\sum_{\ell'\in[s]}|0\rangle|c(i,\ell')\rangle\left(\sqrt{A_{c(i,\ell'),i}}|0\rangle+\sqrt{1-|A_{c(i,\ell'),i}|}|1\rangle\right)|i\rangle
\end{aligned}\tag{9.80}$$

Finally, take the inner product as

$$\begin{aligned}
&\langle 0|\langle 0^n|\langle 0|\langle i|U_A|0\rangle|0^n\rangle|0\rangle|j\rangle\\
=&\frac{1}{s}\sum_{\ell,\ell'}\sqrt{A_{c(j,\ell),j}}\sqrt{A^*_{c(i,\ell'),i}}\delta_{i,c(j,\ell)}\delta_{c(i,\ell'),j}\\
=&\frac{1}{s}\sqrt{A_{ij}}\sqrt{A^*_{ji}}=\frac{1}{s}\left(\sqrt{A_{ij}}\right)^2=\frac{1}{s}A_{ij}.
\end{aligned}\tag{9.81}$$

In this equality, we have used that A is Hermitian: $A_{ij}=A^*_{ji}$, and there exists a unique ℓ such that $i=c(j,\ell)$, as well as a unique ℓ' such that $j=c(i,\ell')$ when A_{ij} is nonzero. □

Exercise 9.5. Let $A\in\mathbb{C}^{N\times N}$ ($N=2^n$) be a Hermitian matrix with entries on the complex unit circle $A_{ij}=e^{i\theta_{ij}}$, $\theta_{ij}\in[0,2\pi)$, which can be accessed via a $2n$ qubit unitary $V\in\mathbb{C}^{N^2\times N^2}$ such that

$$V|0^n\rangle|j\rangle=\frac{1}{\sqrt{N}}\sum_{i\in[N]}e^{i\theta_{ij}/2}|i\rangle|j\rangle,\quad j\in[N].$$

Use V to implement a block encoding U of A with n ancilla qubits. What is the subnormalization factor α for this block encoding?

9.9. Block encoding beyond the computational basis

So far we have assumed that a matrix $A\in\mathbb{C}^{N\times N}$ is accessed through the upper left $N\times N$ block of a unitary $U_A\in\mathrm{U}(MN)$ in the computational basis with $M=2^m$, i.e., via the projector $\Pi_{0^m}:=|0^m\rangle\langle 0^m|\otimes I$ on m ancillas. However, the notion of block encoding is more general: one may encode a (possibly rectangular) matrix $A\in\mathbb{C}^{N'\times N}$ between two subspaces of the ambient space.

Choose an orthonormal basis of $\mathbb{C}^{MN}$ expressed in terms of the columns of a unitary $\Xi\in\mathrm{U}(MN)$ as

$$\mathcal{B}=\{|\varphi_0\rangle,\ldots,|\varphi_{N-1}\rangle,|v_N\rangle,\ldots,|v_{MN-1}\rangle\},\tag{9.82}$$

where the vectors $|\varphi_0\rangle,\ldots,|\varphi_{N-1}\rangle$ are the first N columns of Ξ and span the range of a projector Π (equivalently, $\Pi=\sum_{j=0}^{N-1}|\varphi_j\rangle\langle\varphi_j|$). Similarly, choose another orthonormal basis of $\mathbb{C}^{MN}$ expressed in terms of the columns of another unitary $\Xi'\in\mathrm{U}(MN)$ as

$$\mathcal{B}'=\{|\psi_0\rangle,\ldots,|\psi_{N'-1}\rangle,|w_{N'}\rangle,\ldots,|w_{MN-1}\rangle\},\tag{9.83}$$

where the vectors $|\psi_0\rangle, \ldots, |\psi_{N'-1}\rangle$ are the first N' columns of Ξ' and span the range of a projector Π' (equivalently, $\Pi' = \sum_{i=0}^{N'-1} |\psi_i\rangle\langle\psi_i|$). We naturally assume $N' \le MN$.

Now fix two projectors Π, Π' of rank N and N', where $\{|\varphi_j\rangle\}_{j\in[N]}$ is an orthonormal basis for range(Π), and $\{|\psi_i\rangle\}_{i\in[N']}$ is an orthonormal basis for range(Π'). Let $\mathscr{U}_A \in \mathrm{U}(MN)$ be a unitary with

$$\Pi' \mathscr{U}_A \Pi = \sum_{i\in[N'],\, j\in[N]} |\psi_i\rangle\, A_{ij}\, \langle\varphi_j| \,. \tag{9.84}$$

This is the same block-encoding idea as in the computational basis, except that the input and output subspaces are no longer required to be computational-basis subspaces.

To relate this formulation to the usual block matrix picture, set

$$U_A := (\Xi')^\dagger \, \mathscr{U}_A \, \Xi. \tag{9.85}$$

Then U_A is the matrix representation of $\mathscr{U}_A$ with respect to the bases $\mathcal{B}, \mathcal{B}'$, and

$$[\mathscr{U}_A]_{\mathcal{B}}^{\mathcal{B}'} = U_A = \begin{pmatrix} A & * \\ * & * \end{pmatrix}. \tag{9.86}$$

The preceding discussion was exact. In applications we also allow a subnormalization factor and approximation error, as in Definition 9.2, but now relative to the chosen subspaces.

Definition 9.22. *Let $A \in \mathbb{C}^{N'\times N}$. Fix $\alpha, \epsilon \in \mathbb{R}_+$, and an ambient $(m+n)$-qubit space of dimension MN. Let Π, Π' be orthogonal projectors on $\mathbb{C}^{MN}$ with* $\mathrm{rank}(\Pi) = N$ *and* $\mathrm{rank}(\Pi') = N'$. *Choose unitaries $\Xi, \Xi' \in \mathrm{U}(MN)$ whose first N and N' columns form orthonormal bases for* range(Π) *and* range(Π')*, respectively, as in Eqs.* (9.82) *and* (9.83). *Let $\Xi_N \in \mathbb{C}^{MN\times N}$ and $(\Xi')_{N'} \in \mathbb{C}^{MN\times N'}$ denote the matrices consisting of these first N and N' columns. Given a unitary $\mathscr{U}_A \in \mathrm{U}(MN)$, define the induced block*

$$\widetilde{A} := ((\Xi')_{N'})^\dagger \, \mathscr{U}_A \, \Xi_N \in \mathbb{C}^{N'\times N}. \tag{9.87}$$

We say that $\mathscr{U}_A$ is an (α, m, ϵ)-block-encoding of A with respect to (Π, Π') if

$$\left\| A - \alpha \widetilde{A} \right\| \le \epsilon. \tag{9.88}$$

In the special case $\Pi = \Pi' = \Pi_{0^m}$ and $N' = N$, this reduces to the usual computational-basis definition.

Notes and further reading

The mathematical idea underlying block encodings is a form of unitary dilation: linear maps that are not themselves unitary can often be realized as a sub-block of a larger unitary acting on an extended space. In quantum information, this viewpoint is closely related to dilation theorems for completely positive maps. In quantum algorithms, the block-encoding terminology (together with explicit bookkeeping of the subnormalization factor and approximation error) was systematized as part of the modern polynomial-transformation framework; see [GSLW19].

The linear combination of unitaries (LCU) primitive used here originates in the Hamiltonian simulation algorithm [CW12, BCC$^+$14]. In particular, the sparse-matrix block-encoding constructions in this chapter are closely aligned with the query models developed for sparse Hamiltonian simulation (see, e.g., [BACS07]) and with the block-encoding-based linear-systems framework (see, e.g., [CKS17], which can be directly connected to the quantum circuit for Hermitian block encoding in Fig. 9.9). The connection between block encodings and quantum walks is mediated by the fact

that many walk operators are themselves natural block encodings; see Szegedy's quantization of Markov chains [Sze04] for an early and influential formulation, which will be discussed in detail in Chapter 17. The RACBEM input model for pseudorandom nonunitary matrices was introduced in [DL21]. The quantum circuit in Fig. 9.8 is essentially the construction in [GSLW18, Lemma 48], which gives a $(s, n+3)$-block-encoding. The construction in Fig. 9.8 slightly simplifies the procedure and saves two extra qubits (used to mark whether $\ell \geq s$).

CHAPTER 10

Qubitization

Block encodings provide a unified interface for accessing matrices within quantum circuits, but simply iterating the encoding unitary is often insufficient to transform the underlying matrix. For example, if a block encoding U_A is Hermitian, its powers merely alternate between U_A and the identity. Qubitization addresses this limitation by constructing a unitary iterate whose action preserves two-dimensional subspaces associated with the eigenstructure of the encoded matrix. Within these subspaces, the iterate acts as a rotation, so that its powers implement Chebyshev polynomial transformations of the spectrum.

In this chapter, we progressively introduce the construction of qubitization, starting with Hermitian matrices encoded by Hermitian block encodings, then extending to general matrices. We demonstrate that the qubitization iterate naturally implements the singular value transformation using Chebyshev polynomials. We then show that the iterate can be interpreted using the cosine–sine decomposition in linear algebra. Finally, we combine qubitization with linear combination of unitaries to construct arbitrary polynomial transformations of definite parity.

10.1. Eigenvalue transformation and singular value transformation

Consider a Hermitian matrix $A \in \mathbb{C}^{N\times N}$. Then A has the eigenvalue decomposition

$$A = V\Lambda V^\dagger. \tag{10.1}$$

Here $\Lambda = \mathrm{diag}(\{\lambda_i\})$ is a diagonal matrix, and $\lambda_0 \le \cdots \le \lambda_{N-1}$. Let the scalar function f be well defined on all λ_i's. We first recall the definition of matrix function restricted to Hermitian matrices.

Definition 10.1 (Matrix function of Hermitian matrices, or eigenvalue transformation)**.** *Let $A \in \mathbb{C}^{N\times N}$ be a Hermitian matrix with eigenvalue decomposition Eq.* (10.1)*. Let $f : \mathbb{R} \to \mathbb{C}$ be a scalar function such that $f(\lambda_i)$ is defined for all $i \in [N]$. The matrix function, or eigenvalue transformation of A is defined as*

$$f(A) := Vf(\Lambda)V^\dagger, \tag{10.2}$$

where

$$f(\Lambda) = \mathrm{diag}\left(f(\lambda_0), f(\lambda_1), \ldots, f(\lambda_{N-1})\right). \tag{10.3}$$

For any square matrix $A \in \mathbb{C}^{N\times N}$, the singular value decomposition (SVD) of A reads

$$A = W\Sigma V^\dagger, \tag{10.4}$$

or equivalently

$$A|v_i\rangle = \sigma_i|w_i\rangle, \quad A^\dagger|w_i\rangle = \sigma_i|v_i\rangle, \quad \sigma_i \ge 0, \quad i \in [N]. \tag{10.5}$$

The columns of W, V are called the left and right singular vectors of A, respectively. When A is given by its block encoding $U_A \in \mathrm{BE}_{1,m}(A)$, the singular values of A are in $[0, 1]$.

We may apply a function $f(\cdot)$ on its singular values and define generalized matrix functions below. Unlike matrix functions of Hermitian matrices, we can define three types of generalized matrix functions depending on how we choose the left and right singular vectors.

Definition 10.2 (Generalized matrix functions)**.** *Given $A \in \mathbb{C}^{N\times N}$ with singular value decomposition Eq.* (10.4)*, and let $f : \mathbb{R}_+ \to \mathbb{C}$ be a scalar function such that $f(\sigma_i)$ is defined for all $i \in [N]$. The* ***balanced generalized matrix function*** *is defined as*

$$f^{\diamond}(A) := Wf(\Sigma)V^\dagger, \tag{10.6}$$

where

$$f(\Sigma) = \operatorname{diag}\left(f(\sigma_0), f(\sigma_1), \ldots, f(\sigma_{N-1})\right). \tag{10.7}$$

The ***left generalized matrix function*** *and* ***right generalized matrix function*** *are defined in terms of the left and right singular vectors respectively as*

$$f^{\triangleleft}(A) := Wf(\Sigma)W^\dagger, \quad f^{\triangleright}(A) := Vf(\Sigma)V^\dagger. \tag{10.8}$$

Proposition 10.3. *The following relations hold:*

$$f^{\diamond}(A^\dagger) = (f^{\diamond}(A))^\dagger, \quad f^{\triangleright}(A) = f^{\triangleleft}(A^\dagger), \tag{10.9}$$

and

$$f^{\triangleright}(A) = f^{\diamond}(\sqrt{A^\dagger A}) = f(\sqrt{A^\dagger A}), \quad f^{\triangleleft}(A) = f^{\diamond}(\sqrt{AA^\dagger}) = f(\sqrt{AA^\dagger}). \tag{10.10}$$

Proof. Just note that $A^\dagger A = V\Sigma^2 V^\dagger$, we have $\sqrt{A^\dagger A} = V\Sigma V^\dagger$. So the eigenvalue and singular value decomposition coincide for both $\sqrt{A^\dagger A}$ and $\sqrt{AA^\dagger}$. □

For technical reasons that will become clear later, the definition of singular value transformation in quantum algorithms depends on the parity of f.

Definition 10.4 (Singular value transformation for functions with definite parity)**.** *Given $A \in \mathbb{C}^{N\times N}$ with singular value decomposition Eq.* (10.4)*, let $f : \mathbb{R} \to \mathbb{C}$ be a scalar function such that $f(\pm\sigma_i)$ is defined for all $i \in [N]$. The* ***singular value transformation*** *of A is defined as*

$$f^{\mathrm{SV}}(A) = \begin{cases} f^{\diamond}(A), & f \text{ is odd,} \\ f^{\triangleright}(A), & f \text{ is even.} \end{cases} \tag{10.11}$$

We are often interested in a polynomial f. For a scalar x, the set of all real polynomials of finite degree forms the **real polynomial ring**, denoted by $\mathbb{R}[x]$. Similarly, the set of all complex polynomials of finite degree forms the **complex polynomial ring**, denoted by $\mathbb{C}[x]$.

When A is a Hermitian matrix and $A \succeq 0$, its eigenvalue decomposition and singular value decomposition coincide, so are its eigenvalue and singular value transformations of A.

When A is an indefinite Hermitian matrix, its eigenvalue decomposition is $A = VDV^\dagger$, and its singular value decomposition can be written as $A = W\Sigma V^\dagger$ with $W = V\operatorname{sign}(D), \Sigma = |D|$.

(1) If f is an odd function, then

$$f^{\mathrm{SV}}(A) = f^{\diamond}(A) = Wf(\Sigma)V^\dagger = Vf(\operatorname{sign}(D)\Sigma)V^\dagger = Vf(D)V^\dagger = f(A). \tag{10.12}$$

(2) If f is an even function, then

$$f^{\mathrm{SV}}(A) = f^{\triangleright}(A) = Vf(\Sigma)V^\dagger = Vf(D)V^\dagger = f(A). \tag{10.13}$$

Therefore as long as f has definite parity, the eigenvalue and singular value transformation of a Hermitian matrix A are the same.

For a general matrix $A \in \mathbb{C}^{N\times N}$, we can define a dilated Hermitian matrix using one ancilla qubit:

$$\widetilde{A} = \begin{bmatrix} 0 & A^\dagger \\ A & 0 \end{bmatrix}. \tag{10.14}$$

When A is given by its block encoding $U_A \in \mathrm{BE}_{1,m}(A)$, the dilated Hermitian matrix $\widetilde{A}$ can be obtained with one ancilla qubit through $U_{\widetilde{A}} = |0\rangle\langle 1| \otimes U_A^\dagger + |1\rangle\langle 0| \otimes U_A$, i.e., $U_{\widetilde{A}} \in \mathrm{BE}_{1,m}(\widetilde{A})$. Note that this requires the controlled version of $U_A, U_A^\dagger$.

From the SVD in Eq. (10.5), we can construct

$$|z_i^\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle|v_i\rangle \pm |1\rangle|w_i\rangle). \tag{10.15}$$

Direct calculation shows

$$\widetilde{A}|z_i^\pm\rangle = \pm\sigma_i|z_i^\pm\rangle, \tag{10.16}$$

i.e., $\{|z_i^\pm\rangle\}$ are all the eigenvectors of $\widetilde{A}$.

For an arbitrary polynomial $f \in \mathbb{C}[x]$, the matrix function $f(\widetilde{A})$ takes the form

$$\begin{aligned} f(\widetilde{A}) &= \sum_i |z_i^+\rangle f(\sigma_i)\langle z_i^+| + |z_i^-\rangle f(-\sigma_i)\langle z_i^-| \\ &= \sum_i \begin{pmatrix} |v_i\rangle f_{\mathrm{even}}(\sigma_i)\langle v_i| & |v_i\rangle f_{\mathrm{odd}}(\sigma_i)\langle w_i| \\ |w_i\rangle f_{\mathrm{odd}}(\sigma_i)\langle v_i| & |w_i\rangle f_{\mathrm{even}}(\sigma_i)\langle w_i| \end{pmatrix} \\ &= \begin{pmatrix} f_{\mathrm{even}}^{\triangleright}(A) & f_{\mathrm{odd}}^{\diamond}(A^\dagger) \\ f_{\mathrm{odd}}^{\diamond}(A) & f_{\mathrm{even}}^{\triangleleft}(A) \end{pmatrix}. \end{aligned} \tag{10.17}$$

Here

$$f_{\mathrm{even}}(x) = \frac{1}{2}(f(x)+f(-x)), \quad f_{\mathrm{odd}}(x) = \frac{1}{2}(f(x)-f(-x)). \tag{10.18}$$

Therefore applying the eigenvalue transformation of a dilated matrix $\widetilde{A}$ automatically implements singular value transformation of A using polynomials of even and odd parities.

In particular, if f is an even function, then

$$f(\widetilde{A})|0\rangle|\psi\rangle = |0\rangle f^{\triangleright}(A)|\psi\rangle. \tag{10.19}$$

In other words, by measuring the ancilla qubit we obtain 0 with certainty, and the state in the system register is $f_{\mathrm{even}}^{\triangleright}(A)|\psi\rangle$. Similarly, if f is odd, then

$$f(\widetilde{A})|0\rangle|\psi\rangle = |1\rangle f^{\diamond}(A)|\psi\rangle, \tag{10.20}$$

i.e., by measuring the ancilla qubit we obtain the output 1 with certainty.

In summary, when the function of interest is of definite parity, the singular value transformation and the eigenvalue transformation applied to a dilated Hermitian matrix are two sides of the same coin.

On the other hand, not all eigenvalue transformations can be expressed as singular value transformations. Consider the matrix power A^k as an example. Assume that A is a general non-Hermitian matrix that can be diagonalized as $A = VDV^{-1}$ and has the singular value decomposition $A = W\Sigma V^\dagger$. The matrix power is then given by $A^k = VD^kV^{-1}$. However, this expression

cannot be directly written using the singular value decomposition. To see this, consider the case where $k = 2$. The squared matrix is $A^2 = (W\Sigma V^\dagger)(W\Sigma V^\dagger) = W\Sigma V^\dagger W\Sigma V^\dagger$. Here, the unitary matrix product $V^\dagger W$ does not generally have a simple expression, preventing a straightforward formulation of A^k in terms of singular values.

Many other matrix functions, such as matrix exponential e^A, matrix logarithm $\log A$ etc, cannot be expressed using singular value transformations either for general matrices. One notable exception is the matrix inverse: if A is invertible and $A = W\Sigma V^\dagger$, then $A^{-1} = V\Sigma^{-1}W^\dagger$. Since $f(x) = x^{-1}$ is odd, we find that $f^{\mathrm{SV}}(A^\dagger) = f^\diamond(A^\dagger) = A^{-1}$. Indeed, this will be the basis for using the quantum singular value transformation for computing matrix inverses.

10.2. Qubitization of Hermitian matrices and Chebyshev eigenvalue transformation

Let $A \in \mathbb{C}^{N\times N}$ be a Hermitian matrix with eigenvalue decomposition Eq. (10.1) with $\|A\| \le 1$. The matrix Chebyshev polynomial $T_k(A)$ is a matrix function defined by the Chebyshev polynomial of the first kind:

$$T_k(x) = \cos(k \arccos(x)), \quad x \in [-1, 1], \quad k \in \mathbb{N}. \tag{10.21}$$

Qubitization provides an explicit quantum circuit to implement eigenvalue transformation with Chebyshev polynomials.

10.2.1. Qubitization of Hermitian matrices with Hermitian block encoding. We first introduce some heuristic idea behind qubitization. For any $-1 \le \lambda \le 1$, we can consider a 2×2 rotation matrix,

$$O(\lambda) = \begin{pmatrix} \lambda & -\sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & \lambda \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{10.22}$$

where we have performed the change of variable $\lambda = \cos\theta$ with $0 \le \theta \le \pi$.

Now direct computation shows

$$O^k(\lambda) = \begin{pmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{pmatrix}. \tag{10.23}$$

Using the definition of Chebyshev polynomials (of first and second kinds, respectively)

$$T_k(\lambda) = \cos(k\theta) = \cos(k \arccos\lambda), \quad U_{k-1}(\lambda) = \frac{\sin(k\theta)}{\sin\theta} = \frac{\sin(k\arccos\lambda)}{\sqrt{1-\lambda^2}}, \tag{10.24}$$

we have

$$O^k(\lambda) = \begin{pmatrix} T_k(\lambda) & -\sqrt{1-\lambda^2}U_{k-1}(\lambda) \\ \sqrt{1-\lambda^2}U_{k-1}(\lambda) & T_k(\lambda) \end{pmatrix}. \tag{10.25}$$

Note that if we can somehow replace λ by A, we immediately obtain a $(1, 1)$-block-encoding for the Chebyshev polynomial $T_k(A)$! This is precisely what qubitization aims at achieving, though there are some small twists.

In the simplest scenario, we assume that $U_A \in \mathrm{HBE}_{1,m}(A)$. Start from the spectral decomposition

$$A = \sum_i \lambda_i |v_i\rangle\langle v_i|, \tag{10.26}$$

we have that for each eigenstate $|v_i\rangle$,

$$U_A |0^m\rangle |v_i\rangle = |0^m\rangle A |v_i\rangle + |\widetilde{\perp}_i\rangle = \lambda_i |0^m\rangle |v_i\rangle + |\widetilde{\perp}_i\rangle . \tag{10.27}$$

Here $|\widetilde{\perp_i}\rangle$ is an unnormalized state that is orthogonal to all states of the form $|0^m\rangle|\psi\rangle$, i.e.,

$$\Pi|\widetilde{\perp_i}\rangle = 0. \tag{10.28}$$

where

$$\Pi = |0^m\rangle\langle 0^m| \otimes I \tag{10.29}$$

is a projection operator.

Since the right hand side of Eq. (10.27) is a normalized state, we may also write

$$|\widetilde{\perp_i}\rangle = \sqrt{1-\lambda_i^2}\,|\perp_i\rangle\,, \tag{10.30}$$

where $|\perp_i\rangle$ is a normalized state.

Now if $\lambda_i = \pm 1$, then $\mathcal{H}_i = \text{span}\{|0^m\rangle|v_i\rangle\}$ is already an invariant subspace of U_A, and $|\perp_i\rangle$ can be any state. Otherwise, use the fact that $U_A = U_A^\dagger$, we can apply U_A again to both sides of Eq. (10.27) and obtain

$$U_A|\perp_i\rangle = \sqrt{1-\lambda_i^2}\,|0^m\rangle|v_i\rangle - \lambda_i|\perp_i\rangle\,. \tag{10.31}$$

Therefore $\mathcal{H}_i = \text{span}\{|0^m\rangle|v_i\rangle, |\perp_i\rangle\}$ is an invariant subspace of U_A. Furthermore, the matrix representation of U_A with respect to the basis $\mathcal{B}_i = \{|0^m\rangle|v_i\rangle, |\perp_i\rangle\}$ is

$$[U_A]_{\mathcal{B}_i} = \begin{pmatrix} \lambda_i & \sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & -\lambda_i \end{pmatrix}, \tag{10.32}$$

i.e., U_A restricted to $\mathcal{H}_i$ is a reflection operator. This also leads to the name "qubitization", which means that each eigenvector $|v_i\rangle$ is "qubitized" into a two-dimensional space $\mathcal{H}_i$.

In order to construct a block encoding for $T_k(A)$, we need to turn U_A into a rotation. For this note that $\mathcal{H}_i$ is also an invariant subspace for the projection operator $\Pi = |0^m\rangle\langle 0^m|$:

$$[\Pi]_{\mathcal{B}_i} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{10.33}$$

Similarly define $Z_\Pi = 2\Pi - I$, since

$$[Z_\Pi]_{\mathcal{B}_i} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{10.34}$$

Z_Π acts as a reflection operator restricted to each subspace $\mathcal{H}_i$. Then $\mathcal{H}_i$ is an invariant subspace for the following matrix called the **iterate**

$$O = U_A Z_\Pi \tag{10.35}$$

and

$$[O]_{\mathcal{B}_i} = \begin{pmatrix} \lambda_i & -\sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & \lambda_i \end{pmatrix} \tag{10.36}$$

is the desired rotation matrix. Therefore

$$[O^k]_{\mathcal{B}_i} = [(U_A Z_\Pi)^k]_{\mathcal{B}_i} = \begin{pmatrix} T_k(\lambda_i) & -\sqrt{1-\lambda_i^2}U_{k-1}(\lambda_i) \\ \sqrt{1-\lambda_i^2}U_{k-1}(\lambda_i) & T_k(\lambda_i) \end{pmatrix}. \tag{10.37}$$

Since $\{|0^m\rangle|v_i\rangle\}$ spans the range of Π, we have

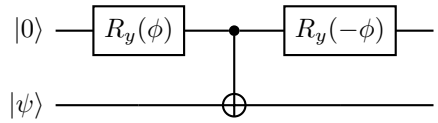
$$O^k = \begin{pmatrix} T_k(A) & * \\ * & * \end{pmatrix} \tag{10.38}$$

i.e., $O^k = (U_A Z_\Pi)^k$ is a $(1, m)$-block-encoding of the Chebyshev polynomial $T_k(A)$.

Example 10.5. Recall the 2×2 Hermitian matrix in Example 9.1,

$$A = \frac{3}{4} I + \frac{1}{4} X = \begin{pmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix}. \tag{10.39}$$

To illustrate the Hermitian-block-encoding setting of this subsection, we use the circuit from Example 9.1 with $\phi = \frac{\pi}{3}$:



This circuit implements the unitary

$$U_A = \begin{pmatrix} A & -\sqrt{I - A^2} \\ -\sqrt{I - A^2} & \frac{1}{4} I + \frac{3}{4} X \end{pmatrix} \in \mathrm{HBE}_{1,1}(A). \tag{10.40}$$

In this example,

$$\sqrt{I - A^2} = \frac{\sqrt{3}}{4}(I - X), \qquad I - 2A^2 = -\frac{1}{4} I - \frac{3}{4} X. \tag{10.41}$$

Since $m = 1$, we have $Z_\Pi = Z \otimes I$. The qubitization iterate is $O = U_A Z_\Pi$. Let us verify for $k = 2$. First,

$$O = U_A Z_\Pi = \begin{pmatrix} A & -\sqrt{I - A^2} \\ -\sqrt{I - A^2} & \frac{1}{4} I + \frac{3}{4} X \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix} = \begin{pmatrix} A & \sqrt{I - A^2} \\ -\sqrt{I - A^2} & -\frac{1}{4} I - \frac{3}{4} X \end{pmatrix}.$$

This particular matrix satisfies a number of identities such as $2A\sqrt{I - A^2} = \sqrt{I - A^2}$. Direct calculation shows

$$O^2 = \begin{pmatrix} 2A^2 - I & 2A\sqrt{I - A^2} \\ -2A\sqrt{I - A^2} & 2A^2 - I \end{pmatrix}.$$

The top-left block of O^2 is

$$T_2(A) = 2A^2 - I = \frac{1}{4} I + \frac{3}{4} X = \begin{pmatrix} 0.25 & 0.75 \\ 0.75 & 0.25 \end{pmatrix}.$$

◇

In order to implement Z_Π, note that if $m = 1$, then Z_Π is just the Pauli Z gate. When $m > 1$, the circuit in Fig. 10.1 maps $|1\rangle |b\rangle$ to $|1\rangle |b\rangle$ if $b = 0^m$, and to $-|1\rangle |b\rangle$ if $b \neq 0^m$. So this precisely implements Z_Π where the signal qubit $|1\rangle$ is used as a work register. We may also discard the signal qubit, and resulting unitary is denoted by Z_Π.

Therefore the circuit in Fig. 10.1 implements the operator O. Repeating the circuit k times gives a $(1, m)$-block-encoding of $T_k(A)$.

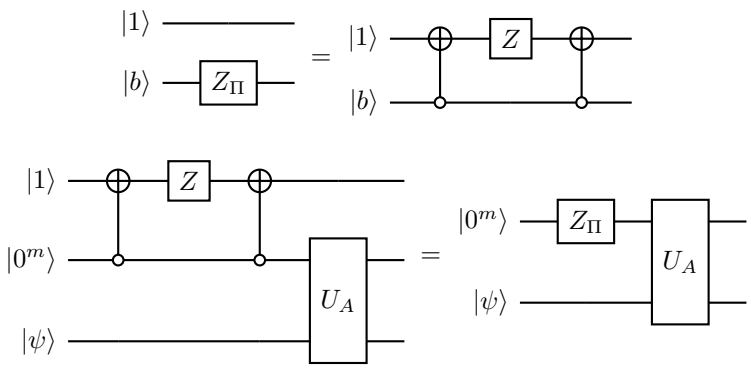


FIGURE 10.1. Circuit implementing one step of qubitization with a Hermitian block encoding of a Hermitian matrix. Here $U_A \in \mathrm{HBE}_{1,m}(A)$.

10.2.2. Qubitization of Hermitian matrices with general block encoding. In Section 10.2.1 we assume that $U_A = U_A^\dagger$ to block encode a Hermitian matrix A. For instance, s-sparse Hermitian matrices, such Hermitian block encodings can be constructed following the construction in Fig. 9.9. However, this can come at the expense of requiring additional structures and oracles. In general, the block encoding of a Hermitian matrix may not be Hermitian itself. In this section we demonstrate that the strategy of qubitization can be modified to accommodate general block encodings.

Again start from the eigendecomposition Eq. (10.26), we apply U_A to $|0^m\rangle|v_i\rangle$ and obtain

$$U_A|0^m\rangle|v_i\rangle = \lambda_i|0^m\rangle|v_i\rangle + \sqrt{1-\lambda_i^2}\,|\perp_i'\rangle\,, \tag{10.42}$$

where $|\perp_i'\rangle$ is a normalized state satisfying $\Pi|\perp_i'\rangle = 0$.

Since U_A block-encodes a Hermitian matrix A, we have

$$U_A^\dagger = \begin{pmatrix} A & * \\ * & * \end{pmatrix}, \tag{10.43}$$

which implies that there exists another normalized state $|\perp_i\rangle$ satisfying $\Pi|\perp_i\rangle = 0$ and

$$U_A^\dagger|0^m\rangle|v_i\rangle = \lambda_i|0^m\rangle|v_i\rangle + \sqrt{1-\lambda_i^2}\,|\perp_i\rangle\,. \tag{10.44}$$

Now apply U_A to both sides of Eq. (10.44), we obtain

$$|0^m\rangle|v_i\rangle = \lambda_i^2|0^m\rangle|v_i\rangle + \lambda_i\sqrt{1-\lambda_i^2}\,|\perp_i'\rangle + \sqrt{1-\lambda_i^2}U_A|\perp_i\rangle\,, \tag{10.45}$$

which gives

$$U_A|\perp_i\rangle = \sqrt{1-\lambda_i^2}\,|0^m\rangle|v_i\rangle - \lambda_i|\perp_i'\rangle\,. \tag{10.46}$$

Define

$$\mathcal{B}_i = \{|0^m\rangle|v_i\rangle, |\perp_i\rangle\}, \quad \mathcal{B}_i' = \{|0^m\rangle|v_i\rangle, |\perp_i'\rangle\}, \tag{10.47}$$

and the associated two-dimensional subspaces $\mathcal{H}_i = \operatorname{span}\mathcal{B}_i, \mathcal{H}_i' = \operatorname{span}\mathcal{B}_i'$, we find that U_A maps $\mathcal{H}_i$ to $\mathcal{H}_i'$. Correspondingly $U_A^\dagger$ maps $\mathcal{H}_i'$ to $\mathcal{H}_i$.

Then Eqs. (10.42) and (10.46) give the matrix representation

$$[U_A]^{\mathcal{B}_i'}_{\mathcal{B}_i} = \begin{pmatrix} \lambda_i & \sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & -\lambda_i \end{pmatrix}. \tag{10.48}$$

Similar calculation shows that

$$[U_A^\dagger]^{\mathcal{B}_i}_{\mathcal{B}_i'} = \begin{pmatrix} \lambda_i & \sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & -\lambda_i \end{pmatrix}. \tag{10.49}$$

Meanwhile both $\mathcal{H}_i$ and $\mathcal{H}_i'$ are the invariant subspaces of the projector Π, with matrix representation

$$[\Pi]_{\mathcal{B}_i} = [\Pi]_{\mathcal{B}_i'} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{10.50}$$

Therefore

$$[Z_\Pi]_{\mathcal{B}_i} = [Z_\Pi]_{\mathcal{B}_i'} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{10.51}$$

Hence $\mathcal{H}_i$ is an invariant subspace of $\widetilde{O} = U_A^\dagger Z_\Pi U_A Z_\Pi$, with matrix representation

$$[\widetilde{O}]_{\mathcal{B}_i} = \begin{pmatrix} \lambda_i & -\sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & \lambda_i \end{pmatrix}^2. \tag{10.52}$$

Repeating k times, we have

$$\begin{aligned} [\widetilde{O}^k]_{\mathcal{B}_i} =& (U_A^\dagger Z_\Pi U_A Z_\Pi)^k = \begin{pmatrix} \lambda_i & -\sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & \lambda_i \end{pmatrix}^{2k} \\ =& \begin{pmatrix} T_{2k}(\lambda_i) & -\sqrt{1-\lambda_i^2} U_{2k-1}(\lambda_i) \\ \sqrt{1-\lambda_i^2} U_{2k-1}(\lambda_i) & T_{2k}(\lambda_i) \end{pmatrix}. \end{aligned} \tag{10.53}$$

Since any vector $|0^m\rangle |\psi\rangle$ can be expanded in terms of the eigenvectors $|0^m\rangle |v_i\rangle$, we have

$$(U_A^\dagger Z_\Pi U_A Z_\Pi)^k = \begin{pmatrix} T_{2k}(A) & * \\ * & * \end{pmatrix}. \tag{10.54}$$

Therefore if we would like to construct an **even** order Chebyshev polynomial $T_{2k}(A)$, the circuit $(U_A^\dagger Z_\Pi U_A Z_\Pi)^k$ straightforwardly gives a $(1, m)$-block-encoding.

In order to construct the block-encoding of an **odd** polynomial $T_{2k+1}(A)$, we note that

$$[U_A Z_\Pi (U_A^\dagger Z_\Pi U_A Z_\Pi)^k]^{\mathcal{B}_i'}_{\mathcal{B}_i} = \begin{pmatrix} T_{2k+1}(\lambda_i) & -\sqrt{1-\lambda_i^2} U_{2k}(\lambda_i) \\ \sqrt{1-\lambda_i^2} U_{2k}(\lambda_i) & T_{2k+1}(\lambda_i) \end{pmatrix}. \tag{10.55}$$

Using the fact that $\mathcal{B}_i, \mathcal{B}_i'$ share the common basis $|0^m\rangle |v_i\rangle$, we still have the block-encoding

$$U_A Z_\Pi (U_A^\dagger Z_\Pi U_A Z_\Pi)^k = \begin{pmatrix} T_{2k+1}(A) & * \\ * & * \end{pmatrix}. \tag{10.56}$$

Therefore $U_A Z_\Pi (U_A^\dagger Z_\Pi U_A Z_\Pi)^k$ is a $(1, m)$-block-encoding of $T_{2k+1}(A)$.

In summary, the block-encoding of $T_l(A)$ is given by applying $U_A Z_\Pi$ and $U_A^\dagger Z_\Pi$ alternately. If $l = 2k$, then there are exactly k such pairs. Otherwise if $l = 2k + 1$, then there is an extra $U_A Z_\Pi$. The effect is to map each eigenvector $|0^m\rangle |v_i\rangle$ back and forth between the two-dimensional subspaces $\mathcal{H}_i$ and $\mathcal{H}_i'$. We summarize these results into the following theorem.

Proposition 10.6 (Chebyshev eigenvalue transformation)**.** *Let $A \in \mathbb{C}^{N\times N}$ be an n-qubit Hermitian matrix given by its block encoding $U_A \in \mathrm{BE}_{1,m}(A)$. Let $Z_\Pi = (2|0^m\rangle\langle 0^m| - I)\otimes I$. Then*

$$(U_A^\dagger Z_\Pi U_A Z_\Pi)^k \in \mathrm{BE}_{1,m}(T_{2k}(A)), \quad U_A Z_\Pi (U_A^\dagger Z_\Pi U_A Z_\Pi)^k \in \mathrm{BE}_{1,m}(T_{2k+1}(A)), \quad k\in\mathbb{N}. \tag{10.57}$$

10.3. Qubitization of general matrices and Chebyshev singular value transformation

In Section 10.2.2 we have observed that when A is a Hermitian matrix, the qubitization procedure introduces two different subspaces $\mathcal{H}_i$ and $\mathcal{H}_i'$ associated with each eigenvector $|v_i\rangle$. In particular, U_A maps $\mathcal{H}_i$ to $\mathcal{H}_i'$, and $U_A^\dagger$ maps $\mathcal{H}_i'$ to $\mathcal{H}_i$. Furthermore, both $\mathcal{H}_i$ and $\mathcal{H}_i'$ are the invariant subspaces of the projection operator Π. Therefore $\mathcal{H}_i$ is an invariant subspace of $U_A^\dagger f(\Pi) U_A$ for any function f.

For a general matrix A, the eigenvalues of A may not be on the real line. In fact, A may not be diagonalizable. Here we illustrate that the correct generalization for a general matrix A is singular value transformation defined as a generalized matrix function. The procedure below almost entirely parallels that of Section 10.2.2.

Starting from the SVD in Eq. (10.5), we apply U_A to $|0^m\rangle|v_i\rangle$ and obtain

$$U_A|0^m\rangle|v_i\rangle = \sigma_i|0^m\rangle|w_i\rangle + \sqrt{1-\sigma_i^2}\,|\perp_i'\rangle, \tag{10.58}$$

where $|\perp_i'\rangle$ is a normalized state satisfying $\Pi|\perp_i'\rangle = 0$.

Since U_A block encodes a matrix A, we have

$$U_A^\dagger = \begin{pmatrix} A^\dagger & * \\ * & * \end{pmatrix}, \tag{10.59}$$

which implies that there exists another normalized state $|\perp_i\rangle$ satisfying $\Pi|\perp_i\rangle = 0$ and

$$U_A^\dagger|0^m\rangle|w_i\rangle = \sigma_i|0^m\rangle|v_i\rangle + \sqrt{1-\sigma_i^2}\,|\perp_i\rangle. \tag{10.60}$$

Applying U_A to both sides of Eq. (10.60), we obtain

$$|0^m\rangle|w_i\rangle = \sigma_i^2|0^m\rangle|w_i\rangle + \sigma_i\sqrt{1-\sigma_i^2}\,|\perp_i'\rangle + \sqrt{1-\sigma_i^2}U_A|\perp_i\rangle, \tag{10.61}$$

which gives

$$U_A|\perp_i\rangle = \sqrt{1-\sigma_i^2}\,|0^m\rangle|w_i\rangle - \sigma_i|\perp_i'\rangle. \tag{10.62}$$

Define

$$\mathcal{B}_i = \{|0^m\rangle|v_i\rangle, |\perp_i\rangle\}, \quad \mathcal{B}_i' = \{|0^m\rangle|w_i\rangle, |\perp_i'\rangle\}, \tag{10.63}$$

and the associated two-dimensional subspaces $\mathcal{H}_i = \operatorname{span}\mathcal{B}_i, \mathcal{H}_i' = \operatorname{span}\mathcal{B}_i'$, we find that U_A maps $\mathcal{H}_i$ to $\mathcal{H}_i'$. Correspondingly $U_A^\dagger$ maps $\mathcal{H}_i'$ to $\mathcal{H}_i$.

Then Eqs. (10.58) and (10.62) give the matrix representation

$$[U_A]_{\mathcal{B}_i}^{\mathcal{B}_i'} = \begin{pmatrix} \sigma_i & \sqrt{1-\sigma_i^2} \\ \sqrt{1-\sigma_i^2} & -\sigma_i \end{pmatrix}. \tag{10.64}$$

Similar calculation shows that

$$[U_A^\dagger]_{\mathcal{B}_i'}^{\mathcal{B}_i} = \begin{pmatrix} \sigma_i & \sqrt{1-\sigma_i^2} \\ \sqrt{1-\sigma_i^2} & -\sigma_i \end{pmatrix}. \tag{10.65}$$

Meanwhile both $\mathcal{H}_i$ and $\mathcal{H}_i'$ are the invariant subspaces of the projector Π, with matrix representation

$$[\Pi]_{\mathcal{B}_i} = [\Pi]_{\mathcal{B}_i'} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{10.66}$$

Therefore

$$[Z_\Pi]_{\mathcal{B}_i} = [Z_\Pi]_{\mathcal{B}_i'} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{10.67}$$

Hence $\mathcal{H}_i$ is an invariant subspace of $\widetilde{O} = U_A^\dagger Z_\Pi U_A Z_\Pi$, with matrix representation

$$[\widetilde{O}]_{\mathcal{B}_i} = \begin{pmatrix} \sigma_i & -\sqrt{1-\sigma_i^2} \\ \sqrt{1-\sigma_i^2} & \sigma_i \end{pmatrix}^2. \tag{10.68}$$

The quantum circuit for each $\widetilde{O}$ is

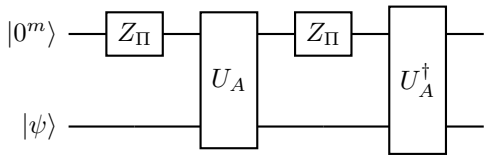


FIGURE 10.2. Circuit implementing one step of qubitization. This block encodes $T_2^{(SV)}(A)$. Here $U_A \in \mathrm{BE}_{1,m}(A)$. Note that the implementation of Z_Π requires a working qubit.

Repeating k times, we have

$$\begin{aligned} [\widetilde{O}^k]_{\mathcal{B}_i} =& (U_A^\dagger Z_\Pi U_A Z_\Pi)^k = \begin{pmatrix} \sigma_i & -\sqrt{1-\sigma_i^2} \\ \sqrt{1-\sigma_i^2} & \sigma_i \end{pmatrix}^{2k} \\ =& \begin{pmatrix} T_{2k}(\sigma_i) & -\sqrt{1-\sigma_i^2}U_{2k-1}(\sigma_i) \\ \sqrt{1-\sigma_i^2}U_{2k-1}(\sigma_i) & T_{2k}(\sigma_i) \end{pmatrix}. \end{aligned} \tag{10.69}$$

In other words,

$$\widetilde{O}^k = \begin{pmatrix} \sum_i v_i T_{2k}(\sigma_i) v_i^\dagger & * \\ * & * \end{pmatrix} = \begin{pmatrix} T_{2k}^{\triangleright}(A) & * \\ * & * \end{pmatrix}. \tag{10.70}$$

Therefore, the circuit $(U_A^\dagger Z_\Pi U_A Z_\Pi)^k$ yields a $(1,m)$-block-encoding of $T_{2k}^{\triangleright}(A)$.

Similarly,

$$[U_A Z_\Pi (U_A^\dagger Z_\Pi U_A Z_\Pi)^k]_{\mathcal{B}_i}^{\mathcal{B}_i'} = \begin{pmatrix} T_{2k+1}(\sigma_i) & -\sqrt{1-\sigma_i^2}U_{2k}(\sigma_i) \\ \sqrt{1-\sigma_i^2}U_{2k}(\sigma_i) & T_{2k+1}(\sigma_i) \end{pmatrix}. \tag{10.71}$$

In other words,

$$U_A Z_\Pi (U_A^\dagger Z_\Pi U_A Z_\Pi)^k = \begin{pmatrix} \sum_i w_i T_{2k+1}(\sigma_i) v_i^\dagger & * \\ * & * \end{pmatrix} = \begin{pmatrix} T_{2k+1}^{\diamond}(A) & * \\ * & * \end{pmatrix}. \tag{10.72}$$

Therefore, the circuit $U_A Z_\Pi (U_A^\dagger Z_\Pi U_A Z_\Pi)^k$ yields a $(1,m)$-block-encoding of $T_{2k+1}^{\diamond}(A)$.

Proposition 10.7 (Chebyshev singular value transformation)**.** *Let $A \in \mathbb{C}^{N\times N}$ be an n-qubit Hermitian matrix given by its block encoding $U_A \in \mathrm{BE}_{1,m}(A)$. Let $Z_\Pi = (2|0^m\rangle\langle 0^m| - I_m) \otimes I_n$. Then*

$$(U_A^\dagger Z_\Pi U_A Z_\Pi)^k \in \mathrm{BE}_{1,m}(T_{2k}^{\mathrm{SV}}(A)), \quad U_A Z_\Pi (U_A^\dagger Z_\Pi U_A Z_\Pi)^k \in \mathrm{BE}_{1,m}(T_{2k+1}^{\mathrm{SV}}(A)), \quad k \in \mathbb{N}. \tag{10.73}$$

Example 10.8. Consider the 2×2 nilpotent matrix

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}. \tag{10.74}$$

Its singular value decomposition $A = W\Sigma V^\dagger$ is given by

$$W = I, \quad \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad V = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{10.75}$$

Note that $T_2^{\triangleright}(A) = VT_2(\Sigma)V^\dagger = X\,\mathrm{diag}(1,-1)X = \mathrm{diag}(-1,1)$.

A $(1,1)$-block-encoding of A can be constructed as

$$U_A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{10.76}$$

Here the basis order is $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. The qubitization iterate $\widetilde{O} = U_A^\dagger Z_\Pi U_A Z_\Pi$ can be computed directly. With $Z_\Pi = Z \otimes I = \mathrm{diag}(1,1,-1,-1)$, we have

$$\widetilde{O} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{10.77}$$

The top-left block is $\mathrm{diag}(-1,1)$, which matches $T_2^{\triangleright}(A)$. ◇

10.4. Cosine–sine decomposition and qubitization

The fact that an arbitrarily large block encoding matrix U_A can be partially block diagonalized into N subblocks of size 2×2 may seem a rather peculiar algebraic structure. In this section we use the **cosine–sine (CS) decomposition** to provide a unified perspective of qubitization. Qubitization stems from the fact that all involved transformations act on direct sums of irreducible two-dimensional subspaces. The CS decomposition makes this observation manifest and further provides a representation for the unitary where these rotations can be expressed as a block matrix wherein the sub-blocks play an analogous role to cosine and sine functions.

THEOREM 10.9 (Cosine–sine decomposition of a unitary matrix). *Let $q \geq p$ and $U \in \mathbb{C}^{(p+q)\times(p+q)}$ be any unitary matrix. There exists a decomposition*

$$U = \begin{pmatrix} W_1 & 0 \\ 0 & W_2 \end{pmatrix} \begin{pmatrix} C & S & 0 \\ S & -C & 0 \\ 0 & 0 & I_{q-p} \end{pmatrix} \begin{pmatrix} V_1^\dagger & 0 \\ 0 & V_2^\dagger \end{pmatrix}. \tag{10.78}$$

Here, $W_1, V_1 \in \mathbb{C}^{p\times p}$, $W_2, V_2 \in \mathbb{C}^{q\times q}$ are unitary matrices and $C = \mathrm{diag}(c_1, \ldots, c_p)$, $S = \mathrm{diag}(s_1, \ldots, s_p)$ are real, non-negative diagonal matrices so that $C^2 + S^2 = I_p$.

PROOF. Let

$$U = \begin{pmatrix} U_{00} & U_{01} \\ U_{10} & U_{11} \end{pmatrix}, \tag{10.79}$$

where $U_{00} \in \mathbb{C}^{p\times p}$ and $U_{11} \in \mathbb{C}^{q\times q}$. The proof proceeds by considering singular value decompositions of each of the block matrices in the unitary. Using the SVD, U_{00} can be expressed for some unitary V_1 and $W_1^\dagger$ as

$$U_{00} = W_1 C V_1^\dagger. \tag{10.80}$$

As U_{00} is embedded in a unitary, we must have that its singular values C are in $[0, 1]$.

Next, consider the QR decomposition

$$U_{10}V_1 = W_2 R \tag{10.81}$$

for a unitary matrix $W_2 \in \mathbb{C}^{q\times q}$ and an upper triangular matrix $R \in \mathbb{C}^{q\times p}$ with non-negative diagonal elements. Using a similar argument we can see that there exists a unitary matrix $V_2 \in \mathbb{C}^{q\times q}$ and a lower triangular matrix $L \in \mathbb{C}^{p\times q}$ with non-negative diagonal elements such that

$$W_1^\dagger U_{01} = L V_2^\dagger. \tag{10.82}$$

This shows that we can write

$$U = \begin{pmatrix} W_1 & 0 \\ 0 & W_2 \end{pmatrix} \begin{pmatrix} C & L \\ R & W_2^\dagger U_{11} V_2 \end{pmatrix} \begin{pmatrix} V_1^\dagger & 0 \\ 0 & V_2^\dagger \end{pmatrix}. \tag{10.83}$$

Now let us argue about the structure of R, L. From the fact that the rows and columns of any unitary matrix must be orthonormal, and that R is upper triangular,

$$R = \begin{pmatrix} S \\ 0 \end{pmatrix}, \quad C^2 + S^2 = I_p. \tag{10.84}$$

By the same argument we have

$$L = \begin{pmatrix} S & 0 \end{pmatrix}, \quad C^2 + S^2 = I_p. \tag{10.85}$$

In other words, $R = L^\dagger$. Continuing the same reasoning,

$$W_2^\dagger U_{11} V_2 = \begin{pmatrix} -C & 0 \\ 0 & U_{22} \end{pmatrix}, \tag{10.86}$$

for some unitary U_{22}. If we absorb this unitary U_{22} into either W_2 or V_2, we obtain the desired factorization in Eq. (10.78). □

Given $U_A \in \mathrm{BE}_{1,m}(A)$ for an n-qubit matrix $A = W\Sigma V^\dagger$, Theorem 10.9 implies that there exists $W', V' \in \mathbb{C}^{N(M-1)\times N(M-1)}$ so that

$$U_A = \begin{pmatrix} W & 0 \\ 0 & W' \end{pmatrix} \begin{pmatrix} \Sigma & S & 0 \\ S & -\Sigma & 0 \\ 0 & 0 & I_{(M-2)N} \end{pmatrix} \begin{pmatrix} V^\dagger & 0 \\ 0 & V'^\dagger \end{pmatrix}. \tag{10.87}$$

Here, $S = \sqrt{I - \Sigma^2}$ is the supplementary diagonal matrix. For notation brevity, the large unitary matrices are denoted as

$$\widetilde{W} := \begin{pmatrix} W & 0 \\ 0 & W' \end{pmatrix}, \quad \widetilde{V} := \begin{pmatrix} V & 0 \\ 0 & V' \end{pmatrix}. \tag{10.88}$$

The matrix in the middle exhibits a special structure. Let $\mathscr{P}$ be a permutation matrix of size $2N \times 2N$ that permutes rows $\{0, 1 \ldots, N-1, N, \ldots, 2N-1\}$ to $\{0, N, 1, N+1, \ldots, N-1, 2N-1\}$. Then we may verify

$$\mathscr{P} \bigoplus_{i\in[N]} \begin{pmatrix} \sigma_i & \sqrt{1-\sigma_i^2} \\ \sqrt{1-\sigma_i^2} & -\sigma_i \end{pmatrix} \mathscr{P}^\dagger = \begin{pmatrix} \Sigma & S \\ S & -\Sigma \end{pmatrix}. \tag{10.89}$$

In other words,

$$\begin{pmatrix} \Sigma & S & 0 \\ S & -\Sigma & 0 \\ 0 & 0 & I_{(M-2)N} \end{pmatrix} = \left\{ \mathscr{P} \bigoplus_{i\in[N]} \begin{pmatrix} \sigma_i & \sqrt{1-\sigma_i^2} \\ \sqrt{1-\sigma_i^2} & -\sigma_i \end{pmatrix} \mathscr{P}^\dagger \right\} \bigoplus I_{(M-2)N} \tag{10.90}$$

is expressed as a direct sum of 2-by-2 blocks and an identity matrix. This is exactly the matrix representation used by qubitization as in Eq. (10.64).

THEOREM 10.10 (Qubitization from cosine–sine decomposition). *For any n-qubit matrix A encoded by $U_A \in \mathrm{BE}_{1,m}(A)$, there exists $(m+n)$-qubit unitary matrices $\widetilde{W}, \widetilde{V}$ and an $(n+1)$-qubit permutation matrix $\mathscr{P}$, such that*

$$U_A = \widetilde{W} \begin{pmatrix} \Sigma & S & 0 \\ S & -\Sigma & 0 \\ 0 & 0 & I_{(M-2)N} \end{pmatrix} \widetilde{V}^\dagger = \widetilde{W} \left\{ \mathscr{P} \bigoplus_{i\in[N]} \begin{pmatrix} \sigma_i & \sqrt{1-\sigma_i^2} \\ \sqrt{1-\sigma_i^2} & -\sigma_i \end{pmatrix} \mathscr{P}^\dagger \bigoplus I_{(M-2)N} \right\} \widetilde{V}^\dagger, \tag{10.91}$$

and

$$U_A^\dagger = \widetilde{V} \begin{pmatrix} \Sigma & S & 0 \\ S & -\Sigma & 0 \\ 0 & 0 & I_{(M-2)N} \end{pmatrix} \widetilde{W}^\dagger = \widetilde{V} \left\{ \mathscr{P} \bigoplus_{i\in[N]} \begin{pmatrix} \sigma_i & \sqrt{1-\sigma_i^2} \\ \sqrt{1-\sigma_i^2} & -\sigma_i \end{pmatrix} \mathscr{P}^\dagger \bigoplus I_{(M-2)N} \right\} \widetilde{W}^\dagger. \tag{10.92}$$

Following the same decomposition, Z_Π can be decomposed as

$$Z_\Pi = \left\{ \mathscr{P} \left(\bigoplus_{i\in[N]} Z \right) \mathscr{P}^\dagger \right\} \bigoplus (-I)_{(M-2)N}. \tag{10.93}$$

Thanks to the block diagonal structure, Z_Π commutes with $\widetilde{V}, \widetilde{W}$. So the definition of Z_Π does not explicitly refer to either $\widetilde{W}$ or $\widetilde{V}$. This would not be true if Z were replaced by Pauli X or Y matrices, and this is the key reason allowing us to choose a convenient phase matrix as in **??**. Also use the fact that $\mathscr{P}^\dagger \mathscr{P} = I$ for a permutation matrix, we have

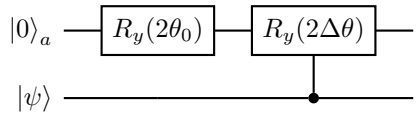
$$(U_A^\dagger Z_\Pi U_A Z_\Pi)^k = \widetilde{V} \left\{ \mathscr{P} \bigoplus_{i\in[N]} \begin{pmatrix} T_{2k}(\sigma_i) & -\sqrt{1-\sigma_i^2} U_{2k-1}(\sigma_i) \\ \sqrt{1-\sigma_i^2} U_{2k-1}(\sigma_i) & T_{2k}(\sigma_i) \end{pmatrix} \mathscr{P}^\dagger \bigoplus I_{(M-2)N} \right\} \widetilde{V}^\dagger. \tag{10.94}$$

Similarly

$$U_A Z_\Pi (U_A^\dagger Z_\Pi U_A Z_\Pi)^k = \widetilde{W} \left\{ \mathscr{P} \bigoplus_{i\in[N]} \begin{pmatrix} T_{2k+1}(\sigma_i) & -\sqrt{1-\sigma_i^2} U_{2k}(\sigma_i) \\ \sqrt{1-\sigma_i^2} U_{2k}(\sigma_i) & T_{2k+1}(\sigma_i) \end{pmatrix} \mathscr{P}^\dagger \bigoplus (-I)_{(M-2)N} \right\} \widetilde{V}^\dagger. \tag{10.95}$$

This result shows that we can decompose an arbitrary unitary into a direct sum of cosine or sine matrices that, in effect, carry out two dimensional rotations.

Example 10.11. Consider a single-qubit Hermitian matrix $A = \mathrm{diag}(0.8, 0.6)$ encoded using one ancilla qubit ($m = 1$). The singular values are $\sigma_0 = 0.8$ and $\sigma_1 = 0.6$, which are distinct. The corresponding complementary values are $s_0 = \sqrt{1 - 0.8^2} = 0.6$ and $s_1 = \sqrt{1 - 0.6^2} = 0.8$. We can implement a block encoding U_A using a controlled rotation circuit on the ancilla, controlled by the system qubit:



where $\theta_0 = \arccos(0.8)$ and $\Delta\theta = \arccos(0.6) - \arccos(0.8)$. The unitary U_A takes the form

$$U_A = \begin{pmatrix} 0.8 & 0 & -0.6 & 0 \\ 0 & 0.6 & 0 & -0.8 \\ 0.6 & 0 & 0.8 & 0 \\ 0 & 0.8 & 0 & 0.6 \end{pmatrix}.$$

Here, the top-left block is precisely A. The qubitization iterate $O = U_A Z_\Pi$, with $Z_\Pi = Z \otimes I$, flips the sign of the columns where the ancilla is $|1\rangle$:

$$O = U_A(Z \otimes I) = \begin{pmatrix} 0.8 & 0 & 0.6 & 0 \\ 0 & 0.6 & 0 & 0.8 \\ 0.6 & 0 & -0.8 & 0 \\ 0 & 0.8 & 0 & -0.6 \end{pmatrix}.$$

Applying the permutation $\mathscr{P}$ that reorders the basis to group by system index $\{|0\rangle_a |0\rangle, |1\rangle_a |0\rangle, |0\rangle_a |1\rangle, |1\rangle_a |1\rangle\}$ yields a block diagonal matrix:

$$\mathscr{P} O \mathscr{P}^\dagger = \begin{pmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{pmatrix} \bigoplus \begin{pmatrix} 0.6 & 0.8 \\ 0.8 & -0.6 \end{pmatrix}.$$

Thus the 4×4 operator decouples into two independent 2×2 reflections, each acting on the subspace associated with a singular vector. ◇

10.5. Linear combination of unitaries and qubitization

Let $f \in \mathbb{R}[x]$ be a polynomial of definite parity. For simplicity, assume f is an even polynomial of degree $2(K-1)$ and set $K = 2^a$. Let us combine LCU and qubitization to construct the block encoding of:

$$f^{\mathrm{SV}}(A) = \sum_{k \in [K]} \alpha_k T_{2k}^{\mathrm{SV}}(A). \tag{10.96}$$

Here A is given by its block encoding $U_A \in \mathrm{BE}_{1,m}(A)$. Due to the connection between eigenvalue and singular value transformation in Section 10.1, when A is Hermitian this becomes an eigenvalue transformation $f(A)$.

Using qubitization (Proposition 10.7), we have constructed $U_{2k} \in \mathrm{BE}_{1,m}(T_{2k}^{\mathrm{SV}}(A))$. The select oracle is given by

$$U_{\mathrm{SEL}} := \sum_{k \in [K]} |k\rangle\langle k| \otimes U_{2k}. \tag{10.97}$$

The problem with a direct implementation of the select oracle via multi-qubit controls is that the complexity is very high. The circuit U_{2k} to block encode $T_{2k}^{\mathrm{SV}}(A)$ makes $2k$ queries to U_A. Therefore

the total number of queries to construct the select oracle is $\mathcal{O}(K^2)$. This is highly inefficient: the circuit blocks $\widetilde{O} = U_A^\dagger Z_\Pi U_A Z_\Pi$ in each U_{2k} are implemented independently and not reused.

An efficient implementation of the select oracle makes use of a binary representation $k = (k_a \cdots k_0)$. Then direct calculation shows that the circuit in Fig. 10.3 correctly implements U_{SEL}. The total number of queries to U_A is $2(1 + 2 + \cdots + 2^{a-1}) = 2^{a+1} - 2 = 2K - 2$. This is equal to the query complexity for block encoding a single term $T_{2K-2}(A)$.

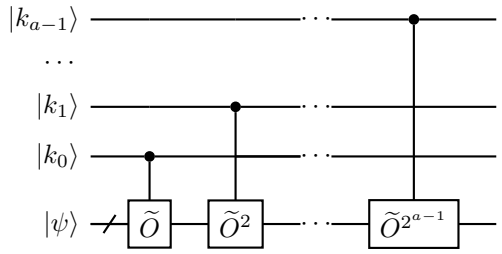


FIGURE 10.3. Circuit for select oracle for even matrix Chebyshev polynomials up to order $2^{a+1} - 2$. Here $\widetilde{O} = U_A^\dagger Z_\Pi U_A Z_\Pi$.

We also assume the availability of the prepare oracle

$$V_{\mathrm{PREP}} |0^a\rangle = \frac{1}{\|\beta\|_2} \sum_{k\in[K]} \beta_k |k\rangle, \quad \widetilde{V}_{\mathrm{PREP}} |0^a\rangle = \frac{1}{\|\gamma\|_2} \sum_{k\in[K]} \gamma_k |k\rangle \tag{10.98}$$

with β_i, γ_i described by Lemma 9.5. Then using LCU, we obtain a $(\|\alpha\|_1, m + a)$-block-encoding of $f^{\mathrm{SV}}(A)$. The gate complexity is

$$(2K - 2) \times \mathrm{gate}(U_A) + \mathrm{gate}(V_{\mathrm{PREP}}) + \mathrm{gate}(\widetilde{V}_{\mathrm{PREP}}). \tag{10.99}$$

Note that the gate complexity depends on the cost for the implementation of the prepare oracle, which may involve QRAM. The construction of the LCU for an odd polynomial is similar.

Exercise 10.1. Construct the circuit for efficient implementation of the select oracle

$$U = \sum_{k\in[K]} |k\rangle\langle k| \otimes U_{2k+1}. \tag{10.100}$$

THEOREM 10.12 (LCU based singular value transformation for polynomials of definite parity). *Let $A \in \mathbb{C}^{N\times N}$ be encoded by its $(1, m)$-block-encoding U_A. For a polynomial $f(x) \in \mathbb{C}[x]$ with degree d of parity $(d \mod 2)$*

$$f(x) = \sum_{k=0}^{d} \alpha_k T_k(x), \tag{10.101}$$

we can implement a $(\|\alpha\|_1, m + a)$-block encoding of the singular value transformation $f^{\mathrm{SV}}(A)$. It uses $a = \mathcal{O}(\log_2 d)$ additional ancilla qubits, and queries $U_A, U_A^\dagger$ for $\mathcal{O}(d)$ times.

If A is a Hermitian matrix, we may construct an eigenvalue transformation $f(A)$ for a general polynomial f. Note that

$$f_{\text{even}}(x) = \frac{1}{2}(f(x) + f(-x)), \quad f_{\text{odd}}(x) = \frac{1}{2}(f(x) - f(-x)), \tag{10.102}$$

we can implement a block encoding of $f_{\text{even}}(A)$ and $f_{\text{odd}}(A)$, respectively. Then we use one more ancilla qubit to implement a block encoding of $f_{\text{even}}(A) + f_{\text{odd}}(A) = f(A)$; this multiplies the subnormalization factor by 2.

10.6. Qubitization beyond the computational basis

Assume that $A \in \mathbb{C}^{N' \times N}$ is given via a subspace block-encoding as in Section 9.9 and Definition 9.22, i.e., by a unitary $\mathscr{U}_A \in \mathrm{U}(MN)$ and projectors Π, Π' with $\operatorname{rank}(\Pi) = N$ and $\operatorname{rank}(\Pi') = N'$. Define the associated reflections

$$Z_\Pi = 2\Pi - I, \qquad Z_{\Pi'} = 2\Pi' - I. \tag{10.103}$$

The qubitization iterate, defined as a product of reflections, is

$$\mathscr{U}_A^\dagger Z_{\Pi'} \mathscr{U}_A Z_\Pi. \tag{10.104}$$

When $N' = N$, this reduces to the usual qubitization iterate under a change of basis. When $N' \neq N$, the same iterate remains well defined and admits the same CS-decomposition-based analysis; this is the natural setting for QSVT with a basis change in Section 13.3.

Example 10.13. Let $m = n = 1$, so $M = N = 2$ and $MN = 4$, and take $\mathscr{U}_A = I \in \mathrm{U}(4)$. Fix

$$\Pi = |0\rangle\langle 0| \otimes I, \tag{10.105}$$

so that $Z_\Pi = 2\Pi - I = Z \otimes I$. Let $\theta \in (0, \pi/2)$ and define the real rotation

$$R(\theta) := \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \in \mathrm{U}(2). \tag{10.106}$$

Set $\Xi := I$ and $\Xi' := R(\theta) \otimes I$, and define $\Pi' := \Xi'\Pi(\Xi')^\dagger$. Then $Z_{\Pi'} = \Xi' Z_\Pi (\Xi')^\dagger$.

With respect to the bases $\{|00\rangle, |01\rangle\}$ for $\operatorname{range}(\Pi)$ and $\{\Xi'|00\rangle, \Xi'|01\rangle\}$ for $\operatorname{range}(\Pi')$, the encoded matrix is the overlap matrix

$$A_{ij} = \langle \psi_i | \varphi_j \rangle, \qquad |\varphi_j\rangle := |0j\rangle, \ |\psi_i\rangle := \Xi'|0i\rangle, \tag{10.107}$$

and a direct computation gives $A = \cos\theta\, I_2$.

The qubitization step (the product of reflections) is

$$\mathscr{U}_A^\dagger Z_{\Pi'} \mathscr{U}_A Z_\Pi = Z_{\Pi'} Z_\Pi = (R(\theta) Z R(\theta)^\dagger Z) \otimes I = R(2\theta) \otimes I. \tag{10.108}$$

Hence the iterate has eigenvalues $e^{\pm 2i\theta}$ (each with multiplicity 2), and the corresponding eigenphases $\pm 2\theta$ determine the singular value $\cos\theta$ of A. ◇

Notes and references

Background on generalized matrix functions can be found in [HBI73, ABF16], which correspond to the "balanced" generalized matrix function $f^\diamond(A)$, i.e., singular value transformation for odd functions.

The cosine–sine (CS) decomposition provides an algebraic explanation for the qubitization structure by making the underlying direct-sum decomposition into 2×2 blocks explicit; see [Don23, TT24]. The same two-dimensional reduction principle also appears as Jordan's lemma: the product

of two reflections acts as a rotation on an invariant two-dimensional subspace [Jor75], and this perspective is closely related to product decompositions in quantum signal processing [Haa19].

The viewpoint of block-encoding and qubitization beyond the computational basis connects directly to Grover's search algorithm, amplitude amplification, and Szegedy's quantum walk. The Chebyshev expansion combined with LCU already yields arbitrary polynomial singular value transformations (for functions of definite parity), but it uses an additional control register to implement the required linear combination with a logarithmic overhead. A more direct and elegant alternative is quantum signal processing and quantum singular value transformation, which implement polynomial transformations via a structured SU(2) recursion, with further connections to nonlinear Fourier analysis on SU(2) [AMT24, ALM$^+$26].

CHAPTER 11

Amplitude amplification based algorithms

Amplitude amplification is one of the most significant tools at the disposal of a quantum algorithm designer. The central idea behind these methods is that the amplitude (or probability) of a desired portion of a quantum state can be rotated between the success and failure branches of a quantum algorithm. Relative to statistical sampling, this quantum approach is fully unitary and requires quadratically fewer operations. This leads to quadratic advantages for sampling algorithms involving statistical sampling, such as Monte-Carlo integration, machine learning applications such as Boltzmann machine training and nearest neighbor classification. Further, any problem in NP can more generally be quadratically accelerated relative to a brute force classical approach and further much of our understanding of the limitations of quantum algorithms stems from amplitude amplification and in turn its earliest incarnation: the celebrated Grover search algorithm.

Grover's algorithm was one of the earliest quantum algorithms to demonstrate a provable speedup in the oracle model: while any classical strategy requires $\Theta(N)$ queries in the worst case, Grover succeeds with $\mathcal{O}(\sqrt{N})$ oracle queries. More importantly, its analysis isolates a mechanism that recurs throughout quantum algorithms: a small success probability can be encoded as an eigenphase of a two-dimensional rotation, and repeated two-reflection steps coherently drive that phase so that the success probability becomes bounded below by a constant. This abstraction leads to amplitude amplification, and further to oblivious amplitude amplification, where one reconstructs a target unitary using only a block encoding and reflections on ancillas, without access to the input state.

The common thread running through amplitude amplification algorithms (sometimes called Grover-type algorithms) in this chapter is that they admit an interpretation as Chebyshev polynomial transformations as seen in qubitization. In this viewpoint, iterating a fixed two-reflection unitary implements a Chebyshev polynomial on an underlying singular value, and "amplification" becomes the task of choosing a polynomial that maps the relevant parameter to ± 1. We also prove a matching lower bound that shows the limits of any such polynomial-based amplification strategy.

11.1. Unstructured search problem and Grover's algorithm

The simplest example of the amplitude amplification paradigm for algorithmic design is Grover's problem. This problem can be thought of as the following. Assume we have $N = 2^n$ boxes, and we are given the promise that only one of the boxes contains an orange, and each of the remaining boxes contains an apple. The goal is to find the box that contains the orange.

Mathematically, given a Boolean function $f : \{0,1\}^n \to \{0,1\}$ as either a bit oracle or a phase oracle and the promise that there exists a unique marked state x_0 such that $f(x_0) = 1$, we would like to find x_0. This is called an **unstructured search problem**. Classically, there is no simpler method than opening $(N-1)$ boxes in the worst case to determine x_0. This fact is demonstrated in the following proposition.

Proposition 11.1. *Let $f : \{0,1\}^n \to \{0,1\}$ be a function such that there exists a marked element x^* such that $f(x) = 1$ if and only if $x = x^*$. No unbiased classical algorithm exists that can identify x^* with probability greater than $1/2$ using fewer than $2^n - 1$ queries to f in the worst case scenario.*

PROOF. Let us assume, seeking a contradiction, that there exists an algorithm that can identify the marked element x^* using at most $2^n - 2$ queries in the worst case scenario. If the algorithm requires fewer than $2^n - 2$ queries then we can pad the list of queries that we make to size $2^n - 2$ as we know that because there is only one marked element any subsequent queries will have $f(x) = 0$ if x is not the marked element. Thus without loss of generality we can assume that precisely $2^n - 2$ queries are always used to find the marked element.

If our algorithm makes precisely $2^n - 2$ queries to identify the marked value of x^*. We can then denote the particular values used in the input to the oracle to be $x_1, \ldots, x_{2^n-2}$. This means that even if $x^* \notin \{x_1, \ldots, x_{2^n-2}\}$ then x^* must be uniquely identifiable by these values. There are two values of x not in the set of inputs and let us denote these as x^* and x'. As we are interested in an algorithm that can decide between these hypotheses in the worst case scenario, let us assume that the marked element happens to be one of these two values. Now we can see that if x' is the marked element then we would have that $f(x_1) = f(x_2) = \cdots = f(x_{2^n-2}) = 0$. Similarly, if x^* were the marked element then our evidence is exactly the same: $f(x_1) = f(x_2) = \cdots = f(x_{2^n-2}) = 0$. Thus if the algorithm returns an estimate $x = x^*$ with probability greater than $1/2$ then the estimator must be biased towards x^* because the evidence observed from both queries is equally likely from both possibilities. Thus by contradiction, the number of queries needed to identify x^* with probability greater than $1/2$ must be greater than or equal to $2^n - 1$. □

The above argument shows that it is not possible for a classical computer to solve the search problem, with high probability, using a number of queries to the boxes that is less than linear in the number of boxes for the search problem.

At first glance, it may seem that quantum algorithms may not be able to provide an advantage for the search problem due to the fact that each query made to the oracle in a classical setting provides nearly negligible information about the marked element. In fact, this was the intuition behind the pioneering work of Boyer et al [BBHT98] which initially sought to show that quantum computers cannot provide a substantial advantage for searching in terms of the queries made to V_f. This intuition though proved misleading. Their attempts to prove query lower bounds on the decision version of the search problem, wherein the user must decide if there are one or zero marked elements such that $f(x) = 1$, showed that at least $\Omega(\sqrt{2^n})$ queries to $f(x)$ are needed to identify the marked element. Any algorithm that can find the marked x, if it exists, can be used to determine if a marked element is present. Thus we can reduce from the decision search problem to see that $\Omega(\sqrt{2^n})$ queries are needed to identify the marked element as well (which can also be seen because the decision search problem is equivalent to computing the OR function of the bits held for $f(x)$ in the oracle, which requires $\Omega(\sqrt{N})$) queries from **??**. This opens up the possibility that a quadratic separation is possible.

Grover's algorithm algorithm settled any doubt about whether a quantum advantage exists for the search problem by revealing that $\mathcal{O}(\sqrt{N})$ queries are needed to find the marked element with high probability. This, combined with the aforementioned lower bound suggests that the optimal quantum scaling for search is $\Theta(\sqrt{2^n})$, which constitutes a quadratic advantage over the best possible classical algorithms. The uses the natural quantum generalization of the classical oracle for the marking function f and is represented as the following unitary bit oracle

$$V_f |x, y\rangle = |x, y \oplus f(x)\rangle, \quad x \in \{0,1\}^n, y \in \{0,1\}, \tag{11.1}$$

Algorithm 11.1 Grover's Algorithm for the case of 1 marked element

Require: Oracle V_f marking solutions, search space size $N = 2^n$ and assume that there is precisely one marked element x^* such that $f(x^*) = 1$

Ensure: A solution x such that $f(x) = 1$ (with high probability)

1: Initialize $|\psi\rangle \leftarrow |0\rangle^{\otimes n} |1\rangle$
2: Apply Hadamard transform to each qubit: $|\psi\rangle \leftarrow \mathrm{H}^{\otimes n+1} |\psi\rangle = |+\rangle^{\otimes n} |-\rangle$
3: **for** $k = 1$ to $\left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor$ **do**
4: Apply oracle: $|\psi\rangle \leftarrow V_f |\psi\rangle$
5: Apply diffusion operator: $|\psi\rangle \leftarrow R_{\psi_0} |\psi\rangle = (2 |\psi_0\rangle\langle\psi_0| \otimes I - 1) |\psi\rangle$
6: **end for**
7: Measure $|\psi\rangle$ to obtain x
8: **return** x

Grover's algorithm, given in Algorithm 11.1 allows us to and can find x^* with high probability using $\mathcal{O}(\sqrt{N})$ queries.

The origin of the quadratic speedup can be summarized as follows: while classical probabilistic algorithms work with probability densities, quantum algorithms work with wavefunction amplitudes, of which the square gives the probability densities. This means that a rotation through an angle θ in Hilbert space will lead to a growth in probability of $\mathcal{O}(\theta^2)$ in probability. This allows us to attain, through Born's rule, a quadratic advantage over classical computing. The following claim can be made specifically about Grover's algorithm.

THEOREM 11.2 (Quantum Search). *Assume $f : \{0,1\}^n \mapsto \{0,1\}$ is a boolean function such that there exists a unique $x^* \in \{0,1\}^n$ such that $f(x^*) = 1$. Algorithm 11.1 will return the value of x^* with probability greater than $2/3$ using $\mathcal{O}(\sqrt{N})$ queries to V_f.*

PROOF. The algorithm starts from a uniform superposition of all states as the initial state

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x \in [N]} |x\rangle . \tag{11.2}$$

This state can be prepared using Hadamard gates as

$$|\psi_0\rangle = \mathrm{H}^{\otimes n} |0^n\rangle . \tag{11.3}$$

We would like to **amplify** the desired amplitude corresponding to $|x_0\rangle$ from $1/\sqrt{N}$ to $\sqrt{p} = \Omega(1)$.

The first step of Grover's algorithm is to turn the bit oracle given in Eqn. (11.1) into a phase oracle that returns a phase of -1 for each marked element. The simplest way to achieve this is to use the fact that $X |-\rangle = - |-\rangle$ to see that a bit oracle with output on $|-\rangle$ will provide a sign flip, which is precisely the choice made in the second step of Algorithm 11.1. More specifically, note that for any $x \in \{0,1\}^n$,

$$V_f |x, -\rangle = \frac{1}{\sqrt{2}} (|x, f(x)\rangle - |x, 1 \oplus f(x)\rangle) = (-1)^{f(x)} |x, -\rangle . \tag{11.4}$$

Any quantum state $|\psi\rangle$ can be decomposed as

$$|\psi\rangle = \alpha |x_0\rangle + \beta |\varphi\rangle , \tag{11.5}$$

where $|\varphi\rangle$ is a unit vector orthogonal to $|x_0\rangle$, i.e., $\langle\varphi|x_0\rangle = 0$. We have

$$V_f |\psi\rangle \otimes |-\rangle = (-\alpha |x_0\rangle + \beta |\varphi\rangle) \otimes |-\rangle . \tag{11.6}$$

Here the minus sign is gained through the phase kickback. Discarding the $|-\rangle$ which is unchanged by applying V_f, we obtain an n-qubit unitary

$$R_{x_0}(\alpha\,|x_0\rangle + \beta\,|\varphi\rangle) = -\alpha\,|x_0\rangle + \beta\,|\varphi\rangle\,. \tag{11.7}$$

Therefore R_{x_0} is a reflection operator across the hyperplane orthogonal to $|x_0\rangle$ as

$$R_{x_0} = I - 2\,|x_0\rangle\,\langle x_0|\,. \tag{11.8}$$

The Grover iteration $V_f R_{x_0}$ can then be expressed as

$$W = (2|\psi_0\rangle\langle\psi_0| \otimes I - I)(I - 2|x^*\rangle\langle x^*|). \tag{11.9}$$

As the Grover iterate is a product of two reflections it can be seen from Jordan's lemma that this creates a rotation in two dimensional space spanned by $|\psi_0\rangle$ and $|x^*\rangle$. In effect the Grover iterate solves the search problem by rotating the initial state towards to the target state but to see this we will explicitly verify that this rotation operator acts as a rotation within this space. The geometric picture is in fact even clearer in Fig. 11.1 and the conclusion can be observed without explicit computation. This same argument will appear many times throughout the text including in amplitude amplification, qubitization and quantum walks and as such we will discuss it in detail here as well.

To see that W acts as a two-dimensional rotation on this space let us begin by hypothesizing that W has an invariant subspace of the form of $\mathrm{span}(|\psi_0\rangle\,, W\,|\psi_0\rangle)$. We will validate this assumption that subspace remains invariant under applications of W. First note that this basis for the subspace is not necessarily orthonormal. We can orthogonalize it using Gram-Schmidt orthogonalization, which allows us to define an orthogonal state $|\psi_0^\perp\rangle$ via

$$|\psi_0^\perp\rangle = \frac{W\,|\psi_0\rangle - |\psi_0\rangle\,\langle\psi_0|\,W\,|\psi_0\rangle}{\sqrt{1 - |\,\langle\psi_0|\,W\,|\psi_0\rangle\,|^2}} \tag{11.10}$$

We can then express the sub-matrix of W inside this space as

$$[W]_{\psi_0,\psi_0^\perp} = \begin{bmatrix} \langle\psi_0|\,W\,|\psi_0\rangle & \langle\psi_0|\,W\,|\psi_0^\perp\rangle \\ \langle\psi_0^\perp|\,W\,|\psi_0\rangle & \langle\psi_0^\perp|\,W\,|\psi_0^\perp\rangle \end{bmatrix}. \tag{11.11}$$

We can now proceed to compute the matrix elements of the submatrix. Specifically,

$$\begin{aligned} \langle\psi_0|\,W\,|\psi_0\rangle &= \langle\psi_0|\,(2|\psi_0\rangle\langle\psi_0| \otimes I - I)(I - 2|x^*\rangle\langle x^*|)\,|\psi_0\rangle \\ &= 1 - 2|\,\langle\psi_0|x^*\rangle\,|^2 \\ &:= 1 - 2\cos^2(\theta/2) = \cos(\theta) \end{aligned} \tag{11.12}$$

where $\theta = 2\arcsin(\langle\psi_0|x^*\rangle) = 2\arcsin(1/\sqrt{N})$ will turn out to be the rotation angle that the Grover iteration applies within this space.

Similarly, we see that the second matrix element in the first row can be written as

$$\begin{aligned} \langle\psi_0|\,W\,|\psi_0^\perp\rangle &= \frac{\langle\psi_0|\,W^2\,|\psi_0\rangle - |\,\langle\psi_0|\,W\,|\psi_0\rangle\,|^2}{\sin(\theta)} \\ &= \frac{\langle\psi_0|\,(2|\psi_0\rangle\langle\psi_0| - I)(I - 2|x^*\rangle\langle x^*|)(2|\psi_0\rangle\langle\psi_0| - I) - \cos^2(\theta)}{\sin(\theta)} \\ &= \frac{1 - 8\cos^2(\theta/2) + 8\cos^4(\theta/2) - \cos^2(\theta)}{\sin(\theta)} \\ &= -\sin(\theta) \end{aligned} \tag{11.13}$$

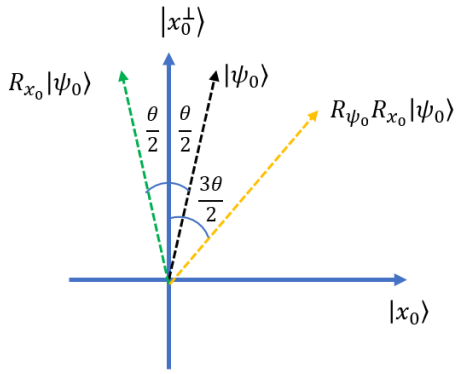


FIGURE 11.1. Geometric interpretation of one Grover iteration.

Repeating this reasoning to the two remaining elements yields the matrix

$$[W]_{\psi_0,\psi_0^\perp} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \equiv e^{-iY\theta} \tag{11.14}$$

As the submatrix is unitary and W is unitary, it follows that there cannot be any other non-zero matrix elements in the corresponding rows and columns of the sub-matrix. This means that the hypothesis that the dynamics is confined to an irreducible two-dimensional subspace is valid.

Applying the Grover operator k times, we obtain from the fact that $\left(e^{-i\theta Y}\right)^k = e^{-ik\theta Y}$

$$G^k |\psi_0\rangle = \sin((2k+1)\theta/2)\,|x_0\rangle + \cos((2k+1)\theta/2)\,|x_0^\perp\rangle\,. \tag{11.15}$$

The probability of failure then satisfies, given that $(2k+1)\theta/2 \leq \pi/2$ for our choice of k

$$\begin{aligned} \mathbb{P}(fail) &= \cos^2((2k+1)\theta/2) = \cos^2\left(\left\lfloor \frac{\pi}{4}\sqrt{N}\right\rfloor \theta + \frac{\theta}{2}\right) \\ &\leq \cos^2\left(\frac{\pi}{4}\sqrt{N}\theta - \frac{\theta}{2}\right) \leq \cos^2\left(\frac{\pi}{2} - \frac{1}{\sqrt{N}}\right) \end{aligned} \tag{11.16}$$

Then by the mean value theorem there exists a value of $x^* \in [0, 1/\sqrt{N}]$ such that

$$\begin{aligned} \cos^2\left(\frac{\pi}{2} - \frac{1}{\sqrt{N}}\right) &= \cos^2(\pi/2) + \frac{1}{\sqrt{N}}\frac{\partial}{\partial x}\cos^2\left(\frac{\pi}{2} - x\right)\Big|_{x=x^*} \\ &= \frac{2}{\sqrt{N}}\sin(2x^*) \leq \frac{4}{N}. \end{aligned} \tag{11.17}$$

In practice, $\sqrt{N}\pi/4$ may not be an integer, and we may not have $\sin((2k+1)\theta/2) \approx 1$. As mentioned in Theorem 11.2, this can lead to a probability of failure as large as $4/N$. This means that Grover's algorithm may fail, although the probability of such failures is small compared to the cost. In particular, as the output of Grover's algorithm can be classically checked using the marking function $f(x)$ we can see if a failure happened and then try again. The number of trials needed before a success is achieved is then geometrically distributed with a success probability of

at least $\mathbb{P}(succ) \geq 1 - 4/N$. Using the fact that the mean number of such trials is $1/\mathbb{P}(succ)$ and each trial requires $\mathcal{O}(\sqrt{N})$ queries to V_f it follows that the expected query complexity of the search algorithm is

$$\mathcal{O}(\sqrt{N}/\mathbb{P}(succ)) \in \mathcal{O}\left(\frac{\sqrt{N}}{1-4/N}\right) \in \mathcal{O}\left(\sqrt{N} + 4/\sqrt{N}\right) = \mathcal{O}(\sqrt{N}) \tag{11.18}$$

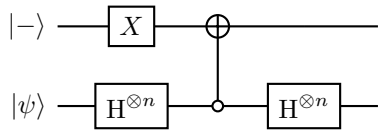
Thus if the expected number of iterations needed in order to find a success is $\mathcal{O}(\sqrt{N})$ it follows from Markov's inequality that with probability greater than $2/3$ we will also only require $\mathcal{O}(\sqrt{N})$ queries to the marking oracle to find the marked state, which proves our claim. □

Interestingly, even if we do not know how to check x_0, statistical amplification as in **??** implies that applying the majority voting can also robustly produce x_0 with probability of failure δ using $\mathcal{O}(\log(1/\delta))$ repetition of the circuit provided that the probability of success is greater than $1/2$.

To draw the quantum circuit of Grover's algorithm, we need an implementation of R_{ψ_0}. Note that

$$R_{\psi_0} = \mathrm{H}^{\otimes n}(2\,|0^n\rangle\,\langle 0^n| - I)\,\mathrm{H}^{\otimes n}\,. \tag{11.19}$$

This can be implemented via the following circuit using one ancilla qubit:



Here the controlled-NOT gate is an n-qubit controlled-X gate, and is only active if the system qubits are in the 0^n state. Discarding the signal qubit, we obtain an implementation of R_{ψ_0}. Since the signal qubit $|-\rangle$ only changes up to a sign, it can be reused for both R_{ψ_0} and R_{x_0}.

Exercise 11.1. Construct a circuit to show that the reflector R_{ψ_0} can also be implemented without using any ancilla qubits.

The reasoning behind Grover's search for a single marked state can be generalized straight forwardly to M marked states as we will see below. The primary difference is that the number of Grover iterations that are needed to achieve a marked state are, as expected, reduced if there are more potential marked states that we can measure. However, as with the previous discussion, the number of marked states needs to be known in order for a **direct** application of the reasoning behind Grover's algorithm to work in this setting. We discuss this situation in the following example.

Example 11.3 (Generalization of Grover's algorithm to multiple marked states)**.** Suppose that there are $M \geq 1$ marked states among $N = 2^n$ basis states, i.e., $f(x) = 1$ for M values of $x \in \{0,1\}^n$ and $f(x) = 0$ otherwise. As in the single-marked case, it is convenient to collect the marked subspace into a single normalized state

$$|w\rangle = \frac{1}{\sqrt{M}} \sum_{x:f(x)=1} |x\rangle\,, \tag{11.20}$$

and similarly the (normalized) superposition of unmarked states

$$|w^\perp\rangle = \frac{1}{\sqrt{N-M}} \sum_{x:f(x)=0} |x\rangle\,. \tag{11.21}$$

Then the uniform superposition can be written as

$$|\psi_0\rangle = \sqrt{\frac{M}{N}}\,|w\rangle + \sqrt{\frac{N-M}{N}}\,|w^\perp\rangle = \sin(\theta/2)\,|w\rangle + \cos(\theta/2)\,|w^\perp\rangle\,, \tag{11.22}$$

where $\sin(\theta/2) = \sqrt{M/N}$.

Using phase kickback, the phase oracle implements the reflection

$$R_w\,|x\rangle = (-1)^{f(x)}\,|x\rangle\,. \tag{11.23}$$

Restricted to the invariant subspace $\mathrm{span}\{|w\rangle\,, |w^\perp\rangle\}$, this operator acts as $R_w = I - 2|w\rangle\langle w|$. As before, we choose

$$R_{\psi_0} = -(I - 2|\psi_0\rangle\langle\psi_0|) = 2|\psi_0\rangle\langle\psi_0| - I, \tag{11.24}$$

Let $\mathcal{B} = \{|w\rangle\,, |w^\perp\rangle\}$. The same calculation as above shows that the Grover iterate $G = R_{\psi_0}R_w$ satisfies

$$G^k\,|\psi_0\rangle = \sin\Big(\frac{(2k+1)\theta}{2}\Big)\,|w\rangle + \cos\Big(\frac{(2k+1)\theta}{2}\Big)\,|w^\perp\rangle\,. \tag{11.25}$$

Thus, the success probability of measuring a marked state equals $\sin^2\big(\frac{(2k+1)\theta}{2}\big)$. Choosing $k \approx \frac{\pi}{2\theta} - \frac{1}{2} \approx \frac{\pi}{4}\sqrt{\frac{N}{M}}$ makes this probability bounded below by a constant. ◇

Example 11.4 (Grover's algorithm as Chebyshev singular value transformation)**.** Let us view Grover's algorithm from the perspective of qubitization beyond the computational basis as in Section 10.6. Let $|x_0\rangle$ be the marked state, and $|\psi_0\rangle$ be the uniform superposition of states. We define an orthonormal basis set $\mathcal{B} = \{|\psi_0\rangle\,, |v_1\rangle\,, \ldots, |v_{N-1}\rangle\}$, where all states $|v_i\rangle$ are orthogonal to $|\psi_0\rangle$. Similarly define an orthonormal basis set $\mathcal{B}' = \{|x_0\rangle\,, |w_1\rangle\,, \ldots, |w_{N-1}\rangle\}$, where all states $|w_i\rangle$ are orthogonal to $|x_0\rangle$. Then the matrix of reflection operator R_{ψ_0} with respect to $\mathcal{B}, \mathcal{B}'$ is (let $a = 1/\sqrt{N} = \sin(\theta/2)$)

$$[R_{\psi_0}]^{\mathcal{B}'}_{\mathcal{B}} = \begin{pmatrix} a & * \\ * & * \end{pmatrix}, \tag{11.26}$$

Therefore $\mathscr{U}_A = R_{\psi_0}$ serves as the block encoding of a 1×1 scalar $A = a = 1/\sqrt{N}$. The projectors $\Pi = |\psi_0\rangle\langle\psi_0|$ and $\Pi' = |x_0\rangle\langle x_0|$ are implicitly defined via the provided reflection operator $Z_\Pi = R_{\psi_0}, Z_{\Pi'} = -R_{x_0}$, respectively. This also means $\mathscr{U}_A Z_\Pi = R_{\psi_0}^2 = I$.

The qubitization invoking $\mathscr{U}_A$ for $(2k+1)$ times gives

$$\Pi'\mathscr{U}_A Z_\Pi(\mathscr{U}_A^\dagger Z_{\Pi'}\mathscr{U}_A Z_\Pi)^k\Pi = \Pi'(R_{\psi_0}(-R_{x_0}))^k\Pi = T_{2k+1}(a)|x_0\rangle\langle\psi_0| = (-1)^k \sin\left(\frac{(2k+1)\theta}{2}\right)|x_0\rangle\langle\psi_0|. \tag{11.27}$$

Here we have used the fact that $T_{2k+1}(a) = (-1)^k \sin((2k+1)\arcsin(a))$ for $a \in [0,1]$. Letting $\sin((2k+1)\theta/2) \approx 1$, we recover the same query complexity as Grover's algorithm. Note that $\mathscr{U}_A^\dagger Z_{\Pi'}\mathscr{U}_A Z_\Pi = R_{\psi_0}(-R_{x_0}) = -G$, this procedure differs from the original Grover's algorithm just by a global phase factor $(-1)^k$. ◇

11.2. Amplitude amplification

The idea behind Grover's algorithm is not restricted to the problem of unstructured search. It can be extended beyond this setting to a broader concept that is called **amplitude amplification**

(AA) [BHMT02], which is used ubiquitously as a subroutine to achieve quadratic speedups. Further, amplitude amplification has also proven invaluable as a technique for converting information stored in an amplitude to information stored in a phase. This is the key insight behind amplitude estimation, which allows us to learn a probability quadratically faster than statistical sampling allows.

Just as Grover's problem has one, or multiple, marked states we assume for amplitude amplification that there is a set of states that we consider to be "good", and assume in our context that our algorithm is successful if any of the states in the good space is found. For example, for Grover's problem the good states correspond to the marked states. A more complicated example is given below involving binary satisfiability.

Example 11.5. Amplitude amplification is actually quite natural to apply to any problem in the complexity class NP. This is because decision problems in NP can (if one has a "yes" instance) be verified in polynomial time. For example we can consider the case of 3-SAT which is an NP-complete problem, which means that all problems in the complexity class NP can be solved by an oracle that solves any instance of 3-SAT. Specifically, the 3-SAT problem is specified by a list of literals $x_1, \dots, x_n$ and a conjuncative normal form formula of the formula. ◇

Grover's algorithm in fact exists as a special case of a larger algorithmic design paradigm known as amplitude amplification. Amplitude amplification follows the exact same structure as Grover's search but instead of marking a set of bit strings, amplitude amplification marks instead a subspace of valid states. Amplitude amplification can then be used to rotate probability between states in this "good" subspace and the "bad" subspace which is its compliment.

Amplitude amplification involves a decomposition of the Hilbert space into two subspaces which we define by the projector Π_{good} and its compliment $I - \Pi_{\text{good}} = \Pi_{\text{bad}}$ (where recall that a projector is a Hermitian operator that squares to itself).

$$|\psi_0\rangle = (\Pi_{\text{good}} + \Pi_{\text{bad}})\,|\psi_0\rangle = \sqrt{p}\,|\psi_{\text{good}}\rangle + \sqrt{1-p}\,|\psi_{\text{bad}}\rangle\,, \tag{11.28}$$

where $p = \mathbb{P}(\text{good})1$. Here $|\psi_{\text{good}}\rangle$ and $|\psi_{\text{bad}}\rangle$ are orthonormal. We do not have direct access to $|\psi_{\text{good}}\rangle$, but would like to obtain a state that has a large overlap with $|\psi_{\text{good}}\rangle$. Amplitude amplification gives a way of providing a quadratic advantage for such problems and as a result is used ubiquitously in quantum algorithms in place of statistical sampling or brute force searches.

The following proposition gives our main claims about the performance of amplitude amplification. This result actually subsumes Grover's algorithm, but as we will see the slightly more abstract definition of amplitude amplification will make some applications more obvious.

Proposition 11.6 (Amplitude Amplification)**.** *Let $\Pi_{\text{good}} \in L(\mathbb{C}^N)$ be a projection operator (i.e. $\Pi_{\text{good}}^2 = \Pi_{\text{good}}$) further let $R_{\psi_0} \in L(\mathbb{C}^N)$ be a reflection operator such that $R_{\psi_0} = (2|\psi_0\rangle\langle\psi_0| - I)$ for initial state $|\psi_0\rangle \in \mathbb{C}^N$ and assume that we are provided oracles (and their inverses) $R_{\text{good}} := (I - 2P_{\text{good}})$ with $\sqrt{p} = |\Pi_{\text{good}}\,|\psi_0\rangle\,| := \sin(\theta/2)$ abd $U_{\psi_0} : |0\rangle \mapsto |\psi_0\rangle$. Let $W = R_{\psi_0} R_{\text{good}}$ has the following properties*

1. *For any integer $k \geq 0$, $|\Pi_{\text{good}} W^k\,|\psi_0\rangle\,| = \sin((2k+1)\theta/2)$.*
2. *The dynamics of $|\psi_0\rangle$ within the subspace $\text{span}(|\psi_0\rangle, |\psi_0^\perp\rangle)$ takes the form*

$$[W]_{\psi_0,\psi_0^\perp} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

where $|\psi_0^\perp\rangle = (W\,|\psi_0\rangle - \cos(\theta)\,|\psi_0\rangle)/\sin(\theta)$.

(3) *The eigenvectors of* $[W]_{\psi_0,\psi_0^\perp}$ *are* $(|\psi_0\rangle - i\,|\psi_0^\perp\rangle)/\sqrt{2}$ *and* $(|\psi_0\rangle - i\,|\psi_0^\perp\rangle)/\sqrt{2}$*. The corresponding eigenvalues of* $[W]_{\psi_0,\psi_0^\perp}$ *are* $e^{-i\theta}, e^{i\theta}$ *which can be expressed as*

$$e^{\pm i \arcsin(\sqrt{\langle\psi_0|\Pi_{\text{good}}|\psi_0\rangle})}$$

PROOF. The derivation of amplitude amplification is very similar to the analysis of Grover's algorithm. We first need to show that the dynamics of W keeps the initial state $|\psi_0\rangle$ within an irreducible two dimensional space. As with the discussion in Theorem 11.2 we note that if our proposed two dimensional space is $\text{span}(|\psi_0\rangle, |\psi_0^\perp\rangle)$. Then under this assumption we can express the amplitude amplification operator, W, as

$$[W]_{\psi_0,\psi_0^\perp} = \begin{bmatrix} \langle\psi_0|\,W\,|\psi_0\rangle & \langle\psi_0|\,W\,|\psi_0^\perp\rangle \\ \langle\psi_0^\perp|\,W\,|\psi_0\rangle & \langle\psi_0^\perp|\,W\,|\psi_0^\perp\rangle \end{bmatrix}. \tag{11.29}$$

This is exactly the same form as we saw for Grover's problem and so we can carry out the same calculation to find the matrix elements. Explicitly, we can compute matrix elements via

$$\begin{aligned} \langle\psi_0|\,W\,|\psi_0\rangle &= \langle\psi_0|\,R_{\psi_0}(I - 2\Pi_{\text{good}})\,|\psi_0\rangle \\ &= \langle\psi_0|\,(I - 2\Pi_{\text{good}}\,|\psi_0\rangle \\ &= 1 - 2p = 1 - 2\sin^2(\theta/2) \\ &= \cos(\theta). \end{aligned} \tag{11.30}$$

This yields exactly the same qualitative result that we saw from Grover's search. The remainder of the matrix elements can be computed in exactly the same manner leading to

$$[W]_{\psi_0,\psi_0^\perp} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{11.31}$$

As the sub-matrix of W is unitary it follows that this is an irreducible subspace because if W had any other non-zero matrix elements in the same row or columns as ψ_0 and $\psi_0^\perp$ then the norm of the corresponding vector would be greater than 1 and in turn W could not be unitary as the norms of every row and column in a unitary matrix must be 1.

The eigenvalues and eigenvectors of W inside this irreducible subspace can be found by noting that

$$[W]_{\psi_0,\psi_0^\perp} = e^{-iY\theta} = e^{-iY\arcsin(\sqrt{p})} = e^{-iY\arcsin\left(\sqrt{\langle\psi_0|\Pi_{\text{good}}|\psi_0\rangle}\right)} \tag{11.32}$$

and the eigenvectors of this operator are $(|\psi_0\rangle - i\,|\psi_0^\perp\rangle)/\sqrt{2}$ and $(|\psi_0\rangle + i\,|\psi_0^\perp\rangle)/\sqrt{2}$ which correspond to the eigenvectors of Y.

Finally from this definition we see that repeated operations of W simply apply a rotation multiple times. As rotation angles add under composition we have that

$$W^k\,|\psi_0\rangle \equiv \begin{bmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \equiv \cos(k\theta)\,|\psi_0\rangle + \sin(k\theta)\,|\psi_0^\perp\rangle \tag{11.33}$$

In order to find the probability of finding a good state we need to now use the fact that $\Pi_{\text{good}}\,|\psi_0\rangle := \sin(\theta/2)\,|\psi_{\text{good}}\rangle$. Similarly, by orthogonality and the fact that $|\psi_0\rangle = \sin(\theta/2)\,|\psi_{\text{good}}\rangle + \cos(\theta/2)\,|\psi_{\text{bad}}\rangle$ we must have from orthogonality that

$$\langle\psi_0^\perp|\psi_0\rangle = \langle\psi_0^\perp|\psi_{\text{good}}\rangle\sin(\theta/2) + \langle\psi_0^\perp|\psi_{\text{bad}}\rangle\cos(\theta/2) = 0 \tag{11.34}$$

we then have that if we choose $\langle\psi_0^\perp|\psi_0\rangle$ to be positive (which we can do using a choice of the irrelevant global phase) then

$$\begin{aligned} \Pi_{\text{good}}\,|\psi_0^\perp\rangle &= \cos(\theta/2)\,|\psi_{\text{good}}\rangle \\ \Pi_{\text{bad}}\,|\psi_0^\perp\rangle &= -\sin(\theta/2)\,|\psi_{\text{bad}}\rangle \end{aligned} \tag{11.35}$$

Then we have that

$$\begin{aligned} \Pi_{\text{good}}W^k\,|\psi_0\rangle &= \cos(k\theta)\Pi_{\text{good}}\,|\psi_0\rangle + \sin(k\theta)\Pi_{\text{good}}\,|\psi_0^\perp\rangle \\ &= \cos(k\theta)\sin(\theta/2)\,|\psi_{\text{good}}\rangle + \sin(k\theta)\cos(\theta/2)\,|\psi_{\text{good}}\rangle \\ &= \cos((2k+1)\theta/2)\,|\psi_{\text{good}}\rangle\,. \end{aligned} \tag{11.36}$$

This demonstrates the final claim about the success probability of measuring a good state after k applications of the amplitude amplification "walk" operator. □

Example 11.7 (Reflection with respect to signal qubits)**.** One common scenario is that the implementation of U_{ψ_0} requires m ancilla qubits (also called signal qubits), i.e.,

$$U_{\psi_0}\,|0^m\rangle\,|0^n\rangle = \sqrt{p}\,|0^m\rangle\,|\psi\rangle + \sqrt{1-p}\,|\perp\rangle\,, \tag{11.37}$$

where $|\perp\rangle$ is some orthogonal state satisfying

$$(\Pi\otimes I)\,|\perp\rangle = 0, \quad \Pi = |0^m\rangle\langle 0^m|. \tag{11.38}$$

Therefore

$$|\psi_{\text{good}}\rangle = |0^m\rangle\,|\psi\rangle\,, \quad |\psi_{\text{bad}}\rangle = |\perp\rangle\,. \tag{11.39}$$

This setting is special since the "good" state can be verified by measuring the ancilla qubits after applying U_{ψ_0} in Eq. (11.37), and post-select the outcome 0^m. In particular, the expected number of measurements needed to obtain $|\psi_{\text{good}}\rangle$ is $1/p$.

In order to employ the amplitude amplification procedure, we first note that the reflection operator can be simplified as

$$R_{\text{good}} = (I - 2|0^m\rangle\langle 0^m|)\otimes I. \tag{11.40}$$

This is because $|\psi_{\text{good}}\rangle$ can be entirely identified by measuring the ancilla qubits. Meanwhile

$$R_{\psi_0} = U_{\psi_0}(2|0^{m+n}\rangle\langle 0^{m+n}| - I)U_{\psi_0}^\dagger. \tag{11.41}$$

Let $G = R_{\psi_0}R_{\text{good}}$. For a suitable integer $k = \mathcal{O}(1/\sqrt{p})$, applying G^k to $U_{\psi_0}\,|0^{m+n}\rangle$ yields a state with constant overlap with $|\psi_{\text{good}}\rangle$. This achieves the desired quadratic speedup. ◇

Corollary 11.8. *Under the assumptions of Proposition 11.6 let us further assume that we are provided with the true value of p and that we have access to controlled oracle $cR_{\text{good}} = |1\rangle\langle 1|\otimes R_{\text{good}} + |0\rangle\langle 0|\otimes I$ then there exists a quantum algorithm that can yield a state of the form $|0\rangle\,|\psi_{\text{good}}\rangle$ such that $\Pi_{\text{good}}\,|\psi_{\text{good}}\rangle = |\psi_{\text{good}}\rangle$ that uses $\mathcal{O}(1/\sqrt{p})$ queries to controlled R_{good}.*

Proof. The core observation behind this corollary is that there exist specific success probabilities such that after k applications of W the success probability is 100%. We then can, if the success probability is known, purposefully lower the success probability to one of these special probabilities and then boost the success probability to 100% using a modified amplitude amplification circuit for this lowered success probability. This is perhaps counter intuitive as it means that we aim to purposefully lower the probability in order to then boost it to 100%.

From Proposition 11.6 we have that the probability of success of amplitude amplification after k iterations is

$$P_{succ} = \sin^2((2k+1)\theta/2) \tag{11.42}$$

The success probability is then 100% for integer k

$$\theta/2 = \frac{\pi}{2(2k+1)} = \arcsin(\sqrt{p}) \;\Rightarrow\; p = \sin^2\left(\frac{\pi}{2(2k+1)}\right). \tag{11.43}$$

Note that if $p = 1/4$ then $k = 1$ is a solution. Otherwise numerical approximations to the first several special success probabilities for $k = 0, 1, 2, 3, 4$

$$p \approx [1, 1/4, 0.09549, 0.04951, 0.03015, \ldots] \tag{11.44}$$

Our aim will be to reduce p to one such probability, specifically let p^* be the target success probability

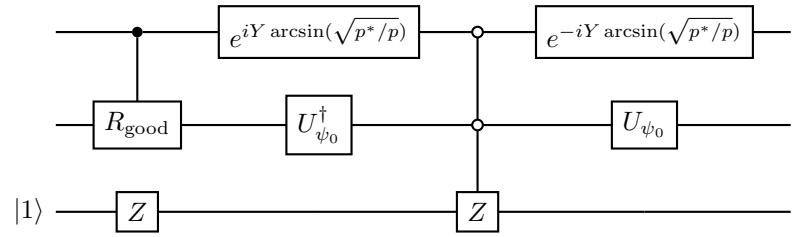
$$p^* = \sin^2\left(\frac{\pi}{2\left(2\left\lceil \frac{\arccos(\sqrt{p})}{2\arcsin(\sqrt{p})}\right\rceil + 1\right)}\right) := \sin^2\left(\frac{\pi}{2\,(2k^*+1)}\right) \tag{11.45}$$

We have then that

$$(|1\rangle\langle 1| \otimes \Pi_{\text{good}}) e^{-iY \arcsin(\sqrt{p^*/p})} |0\rangle |\psi_0\rangle = \sqrt{p^*}\, |1\rangle |\psi_{\text{good}}\rangle . \tag{11.46}$$

Let us then define $|\psi_0'\rangle := e^{-iY \arccos(\sqrt{p^*/p})} |0\rangle |\psi_0\rangle$ We then aim to perform an iteration of amplitude amplification about $|\psi_0'\rangle$ with our good space now defined by the projector $\Pi'_{\text{good}} := |1\rangle\langle 1| \otimes \Pi_{\text{good}}$. We can combine these two together to form a single amplitude amplification operation for this initial and marked space and implement it via the following circuit

$W' :=$ (11.47)



We then see from this that each query to W' requires a single query to controlled R_{good} which can be in turn implemented using two queries to U_{good} (including an inverse query).

As the probability of success of the modified experiment is p^* which is constructed such that k^* iterations of the operator W' will transform

$$|e^{-iY \arcsin(\sqrt{p^*/p})}\rangle |\psi_0\rangle |1\rangle \mapsto |1\rangle |\psi_{\text{good}}\rangle |1\rangle \tag{11.48}$$

Which can be reverted to the form required by our claim by applying NOT gates to the ancillary qubits.

As $\mathcal{O}(1)$ queries to controlled R_{good} and U_{ψ_0} (and their inverses) are made per query to W' this implies that the number of queries needed to apply W'^{k^*} is $\mathcal{O}(k^*) = \mathcal{O}(1/\sqrt{p})$ as claimed.

□

This shows that we can use amplitude amplification to rotate the state deterministically to a good state, if we know the success probability before starting. This is significant because in some applications, such as Grover's problem, we often know the success probability apriori. We also know it for problems such as preparing the state $\frac{1}{\sqrt{L}}\sum_{j=0}^{L-1}|j\rangle$ from $(\mathrm{H}|0\rangle)^{\otimes\lceil\log_2(L)\rceil}$. However, it is important to note that doing so also requires that we have access to a controllable (and invertible) version of our oracles. This is often, in practice, a very reasonable assumption but is important to consider in settings such as in learning theory wherein the assumptions made by models such as the quantum PAC model of learning implicitly forbid the use of these oracles [ADW17].

11.3. Applications of Amplitude Amplification

We will now provide some examples of how amplitude amplification can be used to solve particular problems. These examples represent just a small fraction of the scope of amplitude amplification as an algorithmic design primative.

Example 11.9. Consider an algorithm that incorporates multiple intermediate measurements, where success is determined by measuring all ancilla qubits to yield 0. One example is the multiplication of block encodings in Section 9.3. Let the probability of success at the i-th stage (conditioned on the previous stages being successful) be denoted as p_i. Therefore, the cumulative probability of achieving all 0s across stages is $p = p_1 \times \cdots \times p_L$, and the number of repetitions needed for success is $\mathcal{O}(1/p)$. By the principle of deferred measurement, one can transform this into a coherent process by using $L-1$ additional ancilla qubits. This enables the construction of a reflection operator acting on the L signal qubits. Applying amplitude amplification then reduces the number of repetitions to $\mathcal{O}(1/\sqrt{p})$.

◇

Example 11.10 (Compression gadget)**.** Fig. 11.2 uses $L-1$ ancilla qubits to coherently implement an algorithm that incorporates multiple intermediate measurements. It turns out that the number of ancilla qubits can be significantly reduced to $\mathcal{O}(\log L)$.

To do this we introduce a counter register to count how many intermediate measurements are successful. This counter register contains $m = \lceil\log_2(L+1)\rceil$ qubits, so it represents integers modulo $M := 2^m$. We introduce a unitary operator ADD on this register defined as

$$\mathrm{ADD}\,|c\rangle = |(c+1) \bmod M\rangle\,. \tag{11.49}$$

This operator can be implemented as a classical arithmetic circuit, and performs addition modulo M. Correspondingly $\mathrm{ADD}^\dagger$ performs subtraction. We first add L to the counter register, subtract by 1 each time a measurement result is successful. In the end, if all steps are successful the counter register will be in the state $|0^m\rangle$. The circuit construction for the above coherent procedure is described in Fig. 11.3. Then we may apply amplitude amplification to enhance the success probability using $\mathcal{O}(1/\sqrt{p})$ repetitions of the coherent circuit.

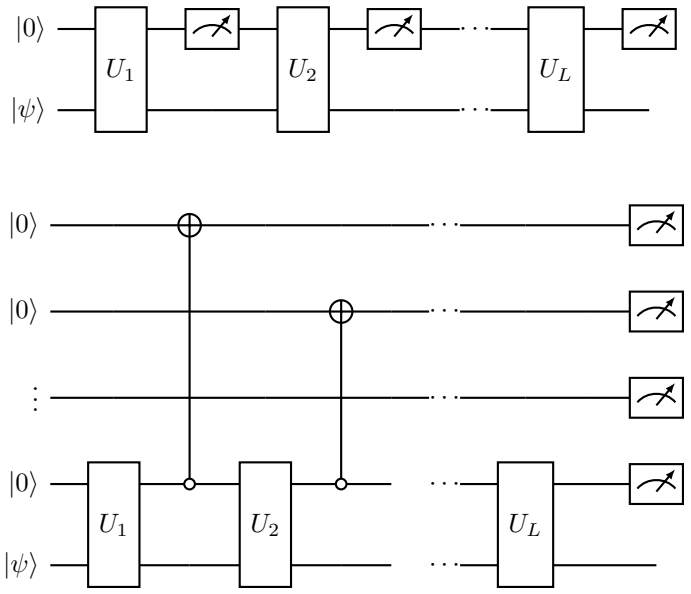


FIGURE 11.2. Using deferred measurement to coherently implement an algorithm that incorporates multiple intermediate measurements. This allows us to use the amplitude amplification procedure to reduce the number of repetitions.

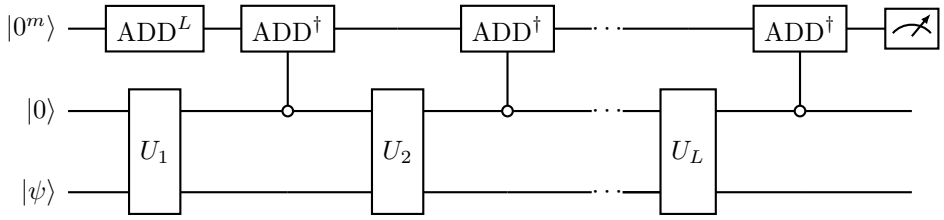


FIGURE 11.3. Circuit for compression gadget to coherently implement an algorithm that incorporates multiple intermediate measurements. The counter register, containing $m = \lceil \log_2(L+1) \rceil$ qubits, is used for keeping track of whether each measurement is successful. The ADD circuit implements addition by 1 modulo the smallest power of 2 that is larger than or equal to $L + 1$.

◇

11.4. Oblivious amplitude amplification

Example 11.11 (Oblivious amplitude amplification)**.** Assume that we have access to a block encoding $V \in \mathrm{BE}_{\gamma,a}(U)$, where $U \in \mathrm{U}(N)$. Then V serves as a $(1, a)$-block encoding of $A = U/\gamma$. When V is applied to a state vector $|\psi\rangle$, the postselected vector is $U|\psi\rangle/\gamma$. If we would like to apply amplitude amplification to boost the success probability, it requires access to a reflection operator with respect to the initial state $|\psi\rangle$. However, since U is unitary, we can achieve this

reconstruction without relying on the initial state (thus the name "oblivious"). This means U can be reconstructed only using multiple applications of V and $V^\dagger$.

The key observations are (1) the set of singular values of A is a single point $\{\gamma^{-1}\}$, and (2) for any unitary matrix, $U(U^\dagger U) = U$. In particular, the singular value transformation of A using any odd polynomial f satisfies

$$f^\diamond(A) = f(\gamma^{-1})U. \tag{11.50}$$

If we can choose an odd polynomial such that $f(\gamma^{-1}) = \pm 1$, then $f^\diamond(A) = \pm U$. This process only uses reflections with respect to the a ancilla qubits used to block encode A, and is oblivious with respect to the initial state.

Consider the case $\gamma = 2$ first. Note that the third order Chebyshev polynomial $T_3(x) = 4x^3 - 3x$ satisfies $T_3(\frac{1}{2}) = -1$. So up to a phase we only need to implement $T_3^\diamond(A)$, which can be performed using qubitization without invoking LCU, see Fig. 11.4.

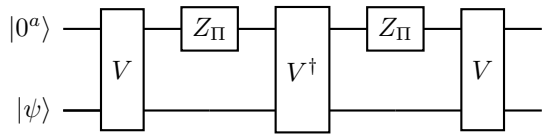


FIGURE 11.4. Circuit for implementing the oblivious amplitude amplification with $V \in \mathrm{BE}_{2,a}(U)$ for a unitary matrix U. This implements $T_3^\diamond(U/2) = -U$.

What about a more general γ? Focusing on odd polynomials, and using the fact that $T_{2k+1}(x) = \cos((2k+1)\arccos(x))$, if γ satisfies, for some $k \in \mathbb{N}$,

$$\gamma^{-1} = \sin\frac{\pi}{2(2k+1)}. \tag{11.51}$$

Then

$$T_{2k+1}(\gamma^{-1}) = \cos\left((2k+1)\arccos(\gamma^{-1})\right) = \cos\left((2k+1)\left(\frac{\pi}{2} - \frac{\pi}{2(2k+1)}\right)\right) = \cos(k\pi) = (-1)^k. \tag{11.52}$$

Therefore oblivious amplitude amplification can be achieved using qubitization, which implements $T_{2k+1}^\diamond(U/\gamma) = (-1)^k U$. This uses $k+1$ queries to V and k queries to $V^\dagger$. In particular, when $k = 1$, we have $\gamma^{-1} = \sin\frac{\pi}{6} = \frac{1}{2}$. Fig. 11.5 plots the polynomials and choice of γ^{-1} for $T_{2k+1}(x)$ for $k = 1, 2, 3$.

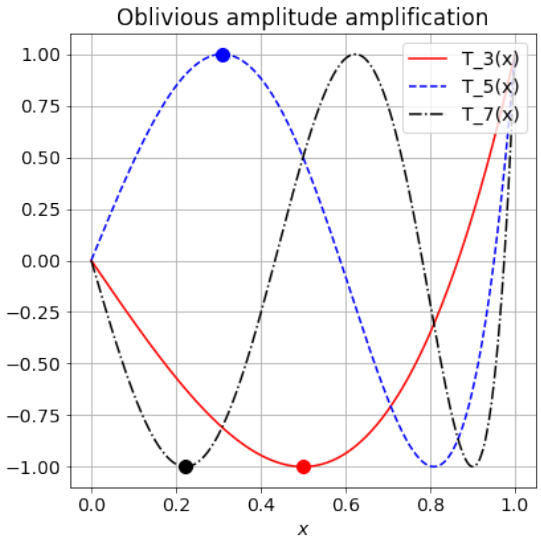


FIGURE 11.5. Chebyshev polynomials (lines) and associated choice of γ^{-1} (filled dots) used for oblivious amplitude amplification.

◇

11.5. Oblivious amplitude amplification of quantum channels

We follow the Kraus and Stinespring formalisms introduced in Section 3.2.

Let $\mathcal{Q} : L(\mathbb{C}^N) \to L(\mathbb{C}^N)$ be a quantum channel. Fix a Kraus representation

$$\mathcal{Q}(\rho) = \sum_{m\in[M]} K_m \rho K_m^\dagger, \tag{11.53}$$

where $K_m \in \mathbb{C}^{N\times N}$ and $\sum_{m\in[M]} K_m^\dagger K_m = I$. From Theorem 3.20, the associated Stinespring isometry $V : \mathbb{C}^N \to \mathbb{C}^M \otimes \mathbb{C}^N$ may be written as

$$V = \sum_{m\in[M]} |m\rangle \otimes K_m, \tag{11.54}$$

where the $|m\rangle$ register is the environment $A \cong \mathbb{C}^M$, so that $\mathcal{Q}(\rho) = \mathrm{Tr}_A[V\rho V^\dagger]$.

Assume without loss of generality that $M = 2^a$ (otherwise pad the Kraus family with zero operators), and that we have a coherent "select" oracle that applies a block encoding of the chosen Kraus operator. Concretely, suppose we can implement

$$U_{\mathrm{SELECT}} = \sum_m |m\rangle\langle m| \otimes U_{K_m}, \quad U_{K_m} \in \mathrm{BE}_{\alpha,b}(K_m), \tag{11.55}$$

so that, on input $|\psi\rangle\,|0^b\rangle$ and upon postselecting the b ancillas back to $|0^b\rangle$, U_{K_m} implements K_m/α on $|\psi\rangle$. Then we can construct a block encoding of the Stinespring isometry V as

$$W = U_{\mathrm{SELECT}} \left(\mathrm{H}^{\otimes a} \otimes I_{b+n}\right). \tag{11.56}$$

More precisely, for any n-qubit state $|\psi\rangle$, if the a-qubit register is initialized to $|0^a\rangle$ and we postselect the b ancillas to $|0^b\rangle$, then

$$(\langle 0^b| \otimes I)(W)(|0^a\rangle \otimes |\psi\rangle \otimes |0^b\rangle) = \frac{1}{\alpha\sqrt{M}} V |\psi\rangle . \tag{11.57}$$

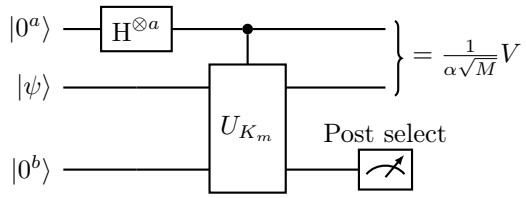


FIGURE 11.6. Circuit for implementing (a scaled version of) the Stinespring isometry of a quantum channel given the block encoding of its Kraus operators.

Recall the construction of the oblivious amplitude amplification in Section 11.4. If we can choose $\alpha\sqrt{M}$ to satisfy the condition

$$(\alpha\sqrt{M})^{-1} = \sin \frac{\pi}{2(2k+1)}, \tag{11.58}$$

for some $k \in \mathbb{N}$, then we can apply the same Chebyshev-based construction (interleaving $W, W^\dagger$ with the reflections Z_Π as in Section 11.4) to amplify the success probability. Although V is not unitary, this construction applies because V is an isometry, so all its nonzero singular values equal 1 and hence the nonzero singular values of $V/(\alpha\sqrt{M})$ are all equal to $(\alpha\sqrt{M})^{-1}$. With the choice above, the resulting transformation maps this singular value to ± 1, yielding a unitary $\mathcal{U}$ (up to an overall phase) such that, on input $|0^a\rangle |\psi\rangle |0^{b+1}\rangle$, the $b+1$ ancillas are returned to $|0^{b+1}\rangle$ and the induced map on the remaining registers is $V|\psi\rangle$. In particular, the postselection on the b ancillas is eliminated.

As a result, we obtain an efficient implementation of the quantum channel

$$\mathcal{Q}(\rho) = \mathrm{Tr}_{a'} \left[\mathcal{U}(|0^{a'}\rangle\langle 0^{a'}| \otimes \rho)\mathcal{U}^\dagger \right], \tag{11.59}$$

where the partial trace is over the $a' = a + b + 1$ ancillas. This construction is called the oblivious amplitude amplification of quantum channels.

Remark 11.12 (Implication for amplitude amplification)**.** Due to the close relation between unstructured search and amplitude amplification, it means that given a state $|\psi\rangle$ of which the probability of obtaining the "good" component is $p_0 1$, no quantum algorithms can amplify the amplitude to $\Omega(1)$ using $o\left(p_0^{-\frac{1}{2}}\right)$ queries to the reflection operators. ◇

Example 11.13 (Amplitude damping)**.** Assuming access to an oracle in Eq. (11.37), where p_0 is large, we can easily dampen the amplitude to any number $|\alpha| \le \sqrt{p_0}$.

We introduce an additional signal qubit. Then Eq. (11.37) becomes

$$(I \otimes U_{\psi_0}) |0\rangle |0\rangle |0^n\rangle = |0\rangle \left(\sqrt{p_0} |0\rangle |\psi_0\rangle + \sqrt{1-p_0} |\perp\rangle \right). \tag{11.60}$$

Define a single qubit rotation operation as

$$R_\theta |0\rangle = \cos\theta |0\rangle + \sin\theta |1\rangle , \tag{11.61}$$

and we have

$$\begin{aligned} &(R_\theta \otimes I_{m+n})(I \otimes U_{\psi_0}) |0\rangle |0^m\rangle |0^n\rangle \\ &= \cos\theta |0\rangle (\sqrt{p_0} |0^m\rangle |\psi_0\rangle + \sqrt{1-p_0} |\perp\rangle) + \sin\theta |1\rangle (\sqrt{p_0} |0^m\rangle |\psi_0\rangle + \sqrt{1-p_0} |\perp\rangle) \\ &:= \sqrt{p_0}\cos\theta |0\rangle |0^m\rangle |\psi_0\rangle + \sqrt{1-p_0\cos^2\theta} |\perp'\rangle . \end{aligned} \tag{11.62}$$

Here $(|0^{m+1}\rangle \langle 0^{m+1}| \otimes I_n) |\perp'\rangle = 0$. We only need to choose $\sqrt{p_0}\cos\theta = \alpha$. ◇

Notes and further reading

Grover's algorithm [Gro96] achieves a quadratic speedup in query complexity for searching unsorted databases in the black-box model. However, the implications for gate complexity are more nuanced: whether a speedup in query complexity translates into an overall gate-level advantage depends on the cost of implementing the oracle, as well as on the overhead of state preparation and reflection operations. This dependence is model-specific and often function-dependent. A universally recognized instance where Grover-type search yields a clear asymptotic advantage in gate complexity over the best classical approach has not been established so far.

The lower bound in **??** is a hybrid argument showing that $\Omega(\sqrt{N})$ queries are necessary, matching Grover's $\mathcal{O}(\sqrt{N})$ upper bound up to constants. Alternative lower-bound techniques include the polynomial method [BBC$^+$01], which relates query complexity to the degree of a polynomial representing (or approximating) the acceptance probability and connects, for symmetric functions, to classical approximation theory such as [Pat92]. The adversary method provides another approach and admits several equivalent formulations; see [SS04, HLS07].

CHAPTER 12

Quantum signal processing

The linear combination of unitaries and qubitization allow us to express a wide range of matrix computation tasks as matrix polynomial transformations on quantum computers. However, these constructions can incur nontrivial circuit costs, particularly in implementing the prepare and select oracles. Quantum signal processing (QSP) and quantum singular value transformation (QSVT) provide an alternative route by realizing polynomial transformations through a structured product of SU(2) rotations specified by a sequence of phases.

This chapter introduces the QSP part, i.e., a structured product of SU(2) rotations. Along with the representation power of QSP for target polynomials, there are two associated computational tasks. First, given a QSP-admissible target polynomial, one must synthesize the corresponding phase factors. Second, when the goal is to approximate a function on a subinterval of $[0, 1]$, one must design a polynomial that both approximates the target and satisfies the admissibility constraints required by QSP. We describe simple numerical procedures addressing these tasks, including a fixed-point iteration algorithm for finding phase factors, and a convex-optimization approach for constructing near-optimal admissible polynomials. We also illustrate the connection between QSP and the nonlinear Fourier transform on SU(2), and generalization of QSP beyond representing polynomial functions.

12.1. Quantum signal processing

Let $x \in [-1, 1]$ be a scalar with a one-qubit Hermitian block encoding

$$U(x) = \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix}. \tag{12.1}$$

Then

$$O(x) = U(x)Z = \begin{pmatrix} x & -\sqrt{1-x^2} \\ \sqrt{1-x^2} & x \end{pmatrix} \tag{12.2}$$

is a rotation matrix.

Consider the following parameterized expression:

$$U_\Phi(x) = e^{i\phi_0 Z} O(x) e^{i\phi_1 Z} O(x) \cdots e^{i\phi_{d-1} Z} O(x) e^{i\phi_d Z}. \tag{12.3}$$

By setting $\phi_0 = \cdots = \phi_d = 0$, we immediately obtain the block encoding of the Chebyshev polynomial $T_d(x)$. The representation power of this formulation is characterized by Theorem 12.1, which is based on slight modification of [GSLW19, Theorem 4]. In the following discussion, even functions have parity 0 and odd functions have parity 1.

THEOREM 12.1 (Quantum signal processing). *There exists a set of* ***phase factors*** $\Phi := (\phi_0, \cdots, \phi_d) \in \mathbb{R}^{d+1}$ *such that*

$$U_\Phi(x) = e^{i\phi_0 Z} \prod_{j=1}^{d} \left[O(x) e^{i\phi_j Z} \right] = \begin{pmatrix} P(x) & -Q(x)\sqrt{1-x^2} \\ \overline{Q(x)}\sqrt{1-x^2} & \overline{P(x)} \end{pmatrix}, \tag{12.4}$$

if and only if $P, Q \in \mathbb{C}[x]$ *satisfy*

(1) $\deg(P) \le d, \deg(Q) \le d-1$,
(2) P *has parity* $d \bmod 2$ *and* Q *has parity* $d-1 \bmod 2$, *and*
(3) $|P(x)|^2 + (1-x^2)|Q(x)|^2 = 1$ *for all* $x \in [-1,1]$.

Here $\deg Q = -1$ *means* $Q = 0$.

PROOF OF THEOREM 12.1. This theorem is proved by direct computation.

$\Rightarrow$:

Since both $e^{i\phi Z}$ and $O(x)$ are unitary, the matrix $U_\Phi(x)$ is always a unitary matrix, which (together with the structure in Eq. (12.4)) immediately implies the condition (3). Below we only need to verify the conditions (1), (2).

When $d = 0$, $U_\Phi(x) = e^{i\phi_0 Z}$, which gives $P(x) = e^{i\phi_0}$ and $Q = 0$, satisfying all three conditions. For induction, suppose $U_{(\phi_0,\cdots,\phi_{d-1})}(x)$ takes the form in Eq. (12.4) with degree $d-1$, then for any $\phi \in \mathbb{R}$, we have

$$\begin{aligned} U_{(\phi_0,\cdots,\phi_{d-1},\phi)}(x) &= \begin{pmatrix} P(x) & -Q(x)\sqrt{1-x^2} \\ \overline{Q(x)}\sqrt{1-x^2} & \overline{P(x)} \end{pmatrix} \begin{pmatrix} x & -\sqrt{1-x^2} \\ \sqrt{1-x^2} & x \end{pmatrix} \begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{pmatrix} \\ &= \begin{pmatrix} xP(x) - (1-x^2)Q(x) & -\sqrt{1-x^2}(P(x) + xQ(x)) \\ \sqrt{1-x^2}(\overline{P(x)} + x\overline{Q(x)}) & x\overline{P(x)} - (1-x^2)\overline{Q(x)} \end{pmatrix} \begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{pmatrix} \\ &= \begin{pmatrix} e^{i\phi}(xP(x) - (1-x^2)Q(x)) & -e^{-i\phi}\sqrt{1-x^2}(P(x) + xQ(x)) \\ e^{i\phi}\sqrt{1-x^2}(\overline{P(x)} + x\overline{Q(x)}) & e^{-i\phi}(x\overline{P(x)} - (1-x^2)\overline{Q(x)}) \end{pmatrix}. \end{aligned} \tag{12.5}$$

Therefore $U_{(\phi_0,\cdots,\phi_{d-1},\phi)}(x)$ satisfies conditions (1),(2).

$\Leftarrow$:

When $d = 0$, the only possibility is $P(x) = e^{i\phi_0}$ and $Q = 0$, which satisfies Eq. (12.4).

For $d > 0$, when d is even we may first consider the special case $\deg P = 0$, i.e., $P(x) = e^{i\phi_0}$ and $Q = 0$. In this case, note that

$$O^{-1}(x) = O(x)^\dagger = \begin{pmatrix} x & \sqrt{1-x^2} \\ -\sqrt{1-x^2} & x \end{pmatrix} = e^{-i\frac{\pi}{2}Z} O(x) e^{+i\frac{\pi}{2}Z}, \tag{12.6}$$

we may set $\phi_j = (-1)^j \frac{\pi}{2}$, $j = 1, \ldots, d$, and

$$e^{i\phi_0 Z} \prod_{j=1}^{d} \left[O(x) e^{i\phi_j Z} \right] = e^{i\phi_0 Z} \prod_{k=1}^{\frac{d}{2}} \left[O(x) e^{-i\frac{\pi}{2}Z} O(x) e^{+i\frac{\pi}{2}Z} \right] = e^{i\phi_0 Z} \prod_{k=1}^{\frac{d}{2}} \left[O(x) O(x)^\dagger \right] = e^{i\phi_0 Z}. \tag{12.7}$$

Thus the statement holds.

Now given P, Q satisfying conditions (1)–(3), with $\deg P = \ell > 0$, and $\ell \equiv d \pmod 2$. Then $\deg(|P(x)|^2) = 2\ell > 0$, and according to the condition (3) we must have $\deg(Q) = \ell - 1$. Let P, Q

be expanded as

$$P(x) = \sum_{k=0}^{\ell} \alpha_k x^k, \quad Q(x) = \sum_{k=0}^{\ell-1} \beta_k x^k, \tag{12.8}$$

then the leading term of $|P(x)|^2 + (1-x^2)|Q(x)|^2$ is

$$|\alpha_\ell|^2 x^{2\ell} - x^2 |\beta_{\ell-1}|^2 x^{2\ell-2} = (|\alpha_\ell|^2 - |\beta_{\ell-1}|^2) x^{2\ell} = 0, \tag{12.9}$$

which implies $|\alpha_\ell| = |\beta_{\ell-1}|$, and $\alpha_\ell/\beta_{\ell-1}$ is a complex phase.

For any $\phi \in \mathbb{R}$, we have

$$\begin{aligned}
&\begin{pmatrix} P(x) & -Q(x)\sqrt{1-x^2} \\ \overline{Q(x)}\sqrt{1-x^2} & \overline{P(x)} \end{pmatrix} e^{-i\phi Z} O(x)^\dagger \\
=&\begin{pmatrix} P(x) & -Q(x)\sqrt{1-x^2} \\ \overline{Q(x)}\sqrt{1-x^2} & \overline{P(x)} \end{pmatrix} \begin{pmatrix} e^{-i\phi} & 0 \\ 0 & e^{i\phi} \end{pmatrix} \begin{pmatrix} x & \sqrt{1-x^2} \\ -\sqrt{1-x^2} & x \end{pmatrix} \\
=&\begin{pmatrix} e^{-i\phi}xP(x) + (1-x^2)Q(x)e^{i\phi} & -\sqrt{1-x^2}(-e^{-i\phi}P(x) + xQ(x)e^{i\phi}) \\ \sqrt{1-x^2}(-e^{i\phi}\overline{P(x)} + x\overline{Q(x)}e^{-i\phi}) & e^{i\phi}x\overline{P(x)} + (1-x^2)\overline{Q(x)}e^{-i\phi} \end{pmatrix} \\
=:&\begin{pmatrix} \widetilde{P}(x) & -\widetilde{Q}(x)\sqrt{1-x^2} \\ \overline{\widetilde{Q}(x)}\sqrt{1-x^2} & \overline{\widetilde{P}(x)} \end{pmatrix}.
\end{aligned} \tag{12.10}$$

It may appear that $\deg \widetilde{P} = \ell + 1$. However, by properly choosing ϕ we may obtain $\deg \widetilde{P} = \ell - 1$. Let $e^{2i\phi} = \alpha_\ell/\beta_{\ell-1}$. Then the coefficient of the $x^{\ell+1}$ term in $\widetilde{P}$ is

$$e^{-i\phi}\alpha_\ell - e^{i\phi}\beta_{\ell-1} = 0. \tag{12.11}$$

Similarly, the coefficient of the x^ℓ term in $\widetilde{Q}$ is

$$-e^{-i\phi}\alpha_\ell + e^{i\phi}\beta_{\ell-1} = 0. \tag{12.12}$$

The coefficient of the x^ℓ term in $\widetilde{P}$, and the coefficient of the $x^{\ell-1}$ term in $\widetilde{Q}$ are both 0 by the parity condition. So we have

(1) $\deg(\widetilde{P}) \le \ell - 1 \le d - 1, \deg(\widetilde{Q}) \le \ell - 2 \le d - 2$,
(2) $\widetilde{P}$ has parity $d - 1 \bmod 2$ and $\widetilde{Q}$ has parity $d - 2 \bmod 2$, and
(3) $|\widetilde{P}(x)|^2 + (1-x^2)|\widetilde{Q}(x)|^2 = 1, \forall x \in [-1, 1]$.

Here the condition (3) is automatically satisfied due to unitarity. The induction follows until $\ell = 0$, and apply the argument in Eq. (12.7) to represent the remaining constant phase factor if needed. □

Note that the normalization condition (3) in Theorem 12.1 imposes very strong constraints on the coefficients of $P, Q \in \mathbb{C}[x]$. If we are only interested in QSP for real polynomials, the conditions can be significantly relaxed. The following result is a variant of [GSLW19, Corollary 5].

THEOREM 12.2 (Quantum signal processing for real polynomials of definite parity). *Given a real polynomial $F(x) \in \mathbb{R}[x]$ of degree $d > 0$ satisfying*

(1) *F has parity $d \bmod 2$,*
(2) *$|F(x)| \le 1$ for all $x \in [-1, 1]$,*

then there exists polynomials $P(x), Q(x) \in \mathbb{C}[x]$ with $F = \operatorname{Im} P$ and a set of phase factors $\Phi := (\phi_0, \cdots, \phi_d) \in \mathbb{R}^{d+1}$ such that the QSP representation Eq. (12.4) holds.

Compared to Theorem 12.1, the conditions in Theorem 12.2 are much easier to satisfy: given any polynomial $F(x) \in \mathbb{R}[x]$ satisfying condition (1) on parity, we can always scale F to satisfy the condition (2) on its magnitude. Also note that

$$\mathrm{Re}[U_\Phi(x)]_{1,1} = \mathrm{Im}[e^{i\frac{\pi}{4}Z} U_\Phi(x) e^{i\frac{\pi}{4}Z}]_{1,1}. \tag{12.13}$$

Thus $\mathrm{Re}\, P(x) = \mathrm{Re}[U_\Phi(x)]_{1,1}$ can be recovered from the imaginary part by adding $\frac{\pi}{4}$ to both ϕ_0 and ϕ_d. Consequently, the conclusion of Theorem 12.2 also holds if we replace $F = \mathrm{Im}\, P$ by $F = \mathrm{Re}\, P$.

12.2. Symmetric quantum signal processing

There are multiple equivalent ways of stating the QSP parameterization commonly seen in the literature. Let us refer to the convention used in Theorem 12.1 as the **O-convention**, i.e.,

$$U_\Phi(x) = e^{i\phi_0 Z} \prod_{j=1}^{d} \left[O(x) e^{i\phi_j Z} \right], \quad O(x) = \begin{pmatrix} x & -\sqrt{1-x^2} \\ \sqrt{1-x^2} & x \end{pmatrix}. \tag{12.14}$$

We may also use the X-rotation

$$W(x) = e^{-i\frac{\pi}{4}Z} O(x) e^{+i\frac{\pi}{4}Z} = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} = e^{i \arccos(x) X}, \tag{12.15}$$

and express the QSP parameterization as

$$U_{\Phi W}(x) = e^{i\phi_0^W Z} \prod_{j=1}^{d} \left[W(x) e^{i\phi_j^W Z} \right] = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ i\overline{Q(x)}\sqrt{1-x^2} & \overline{P(x)} \end{pmatrix}. \tag{12.16}$$

This is referred to as the **W-convention**. There is a simple relation connecting between the phase factors using the O and W

$$\phi_j = \begin{cases} \phi_0^W - \frac{\pi}{4}, & j = 0, \\ \phi_j^W, & j = 1, \dots, d-1, \\ \phi_d^W + \frac{\pi}{4}, & j = d. \end{cases} \tag{12.17}$$

If we are interested in a real polynomial $F(x) \in \mathbb{R}[x]$ of degree d, due to the parity constraint, F can be expanded in the Chebyshev basis as

$$F(x) = \begin{cases} \sum_{j=0}^{\widetilde{d}-1} c_j T_{2j}(x), & F \text{ is even}, \\ \sum_{j=0}^{\widetilde{d}-1} c_j T_{2j+1}(x), & F \text{ is odd}. \end{cases} \tag{12.18}$$

Here $\widetilde{d} = \lceil \frac{d+1}{2} \rceil$, and $c = (c_0, c_1, \dots, c_{\widetilde{d}-1}) \in \mathbb{R}^{\widetilde{d}}$ is the Chebyshev coefficient vector. Thus the number of effective degrees of freedom in F is only $\widetilde{d}$. How to reconcile this with the $d+1$ degrees of freedom in the phase factors?

The QSP sequence in the W-convention offers a clue to this question. Since W is a complex symmetric matrix, i.e., $W = W^\top$, from Eq. (12.16) we have

$$U_{\Phi W}^\top(x) = e^{i\phi_d^W Z} \prod_{j=1}^{d} \left[W(x) e^{i\phi_{d-j}^W Z} \right], \tag{12.19}$$

i.e., the transpose of $U_{\Phi W}(x)$ can be obtained by reversing the order of the phase factors. In QSP applications we often do not care about Q, so if we choose Q to be a real polynomial, then the

matrix $U_{\Phi w}(x)$ is complex symmetric, i.e., $U_{\Phi w}(x) = U_{\Phi w}(x)^\top$. This means that the phase factors may also be chosen symmetrically:

$$\phi_j^W = \phi_{d-j}^W, \quad \forall j = 0, 1, \cdots, d. \tag{12.20}$$

With the symmetry condition, the number of effective degrees of freedom in phase factors is equal to $\widetilde{d} = \lceil \frac{d+1}{2} \rceil$, and matches that in F.

Example 12.3. Consider the all-zero phase factors in the W-convention $\Phi^W = (0, \ldots, 0) \in \mathbb{R}^{d+1}$. Direct calculation shows the upper-left entry of $U_{\Phi w}(x)$ is the Chebyshev polynomial $P(x) = T_d(x)$. The corresponding O-convention is $\Phi = (-\pi/4, 0, \ldots, 0, \pi/4)$. Now let $\Phi^W = (\pi/4, 0, \ldots, 0, \pi/4) \in \mathbb{R}^{d+1}$. The upper-left entry of $U_{\Phi w}(x)$ becomes $P(x) = iT_d(x)$. The corresponding O-convention is $\Phi = (0, 0, \ldots, 0, \pi/2)$. ◇

Example 12.4. A linear combination of Chebyshev polynomials $F(x) = 0.2T_1(x) + 0.4T_3(x)$ can be encoded using phase factors $\Phi^W = (1.3622, -0.1132, -0.1132, 1.3622)$, so that $F(x) = \mathrm{Im}[U_{\Phi w}(x)]_{1,1}$. ◇

The results below guarantee that, under the assumption that Q is a real polynomial, the symmetry restriction is without loss of generality and yields a unique phase vector in a canonical domain.

THEOREM 12.5 (Existence and uniqueness of symmetric phase factors, W-convention). *Consider any $P \in \mathbb{C}[x]$ and $Q \in \mathbb{R}[x]$ satisfying the following conditions.*

1. $\deg(P) = d$ *and* $\deg(Q) = d - 1$.
2. P *has parity* $(d \bmod 2)$ *and* Q *has parity* $(d - 1 \bmod 2)$.
3. $|P(x)|^2 + (1 - x^2)|Q(x)|^2 = 1$ *for all* $x \in [-1, 1]$.
4. *If* d *is odd, then the leading coefficient of* Q *is positive.*

Then there exists a ***unique*** *set of symmetric phase factors*

$$\Phi^W := (\phi_0^W, \phi_1^W, \cdots, \phi_d^W) \in D_d, \tag{12.21}$$

where Φ^W *is symmetric in the sense of Eq.* (12.20) *and*

$$D_d := \begin{cases} [-\frac{\pi}{2}, \frac{\pi}{2})^{\frac{d}{2}} \times [-\pi, \pi) \times [-\frac{\pi}{2}, \frac{\pi}{2})^{\frac{d}{2}}, & d \text{ is even}, \\ [-\frac{\pi}{2}, \frac{\pi}{2})^{d+1}, & d \text{ is odd}, \end{cases} \tag{12.22}$$

such that

$$U_{\Phi w}(x) = e^{i\phi_0^W Z} \prod_{j=1}^{d} \left[W(x) e^{i\phi_j^W Z} \right] = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ(x)\sqrt{1-x^2} & \overline{P(x)} \end{pmatrix}. \tag{12.23}$$

For real polynomials, the symmetric version of the theorem that is parallel to Theorem 12.2 is as follows.

Corollary 12.6 (Symmetric quantum signal processing for real polynomials, W-convention)**.** *Given a real polynomial* $F(x) \in \mathbb{R}[x]$ *of degree* d *satisfying*

1. F *has parity* $d \bmod 2$,
2. $|F(x)| \leq 1$ *for all* $x \in [-1, 1]$,

then there exists polynomials $G(x), Q(x) \in \mathbb{R}[x]$ *and a set of symmetric phase factors* $\Phi^W := (\phi_0^W, \phi_1^W, \cdots, \phi_d^W) \in \mathbb{R}^{d+1}$ *satisfying Eq.* (12.20) *such that the following QSP representation holds:*

$$U_{\Phi^W}(x) = e^{i\phi_0^W Z} \prod_{j=1}^{d} \left[W(x) e^{i\phi_j^W Z} \right] = \begin{pmatrix} G(x) + iF(x) & iQ(x)\sqrt{1-x^2} \\ iQ(x)\sqrt{1-x^2} & G(x) - iF(x) \end{pmatrix}. \tag{12.24}$$

12.3. Fixed-point iteration algorithm for finding phase factors

According to Eq. (12.18), a target polynomial for quantum signal processing $F \in \mathbb{R}[x]$ of degree d only has $\widetilde{d} = \lceil \frac{d+1}{2} \rceil$ effective degrees of freedom characterized by the Chebyshev coefficient vector $c = (c_0, c_1, \ldots, c_{\widetilde{d}-1}) \in \mathbb{R}^{\widetilde{d}}$.

Given the discussion in Section 12.2, we focus on finding symmetric phase factors Φ^W for a target polynomial of definite parity. We can define the associated **reduced phase factors** as

$$\Phi^R = (\phi_0^R, \phi_1^R, \ldots, \phi_{\widetilde{d}-1}^R) := \begin{cases} (\frac{1}{2}\phi_{\widetilde{d}-1}^W, \phi_{\widetilde{d}}^W, \cdots, \phi_d^W), & d \text{ is even}, \\ (\phi_{\widetilde{d}}^W, \phi_{\widetilde{d}+1}^W, \cdots, \phi_d^W), & d \text{ is odd}. \end{cases} \tag{12.25}$$

Under the symmetry condition Eq. (12.20) (and the canonical domain in Theorem 12.5), Φ^R uniquely determines Φ^W. The reason why the reduced phase factors start from the middle is that the phase factors tend to be large in the middle and decay to zero towards the ends (see Fig. 12.1).

Let $\mathscr{F} : \mathbb{R}^{\widetilde{d}} \to \mathbb{R}^{\widetilde{d}}$ denote the mapping from reduced phase factors $\Phi^R \in \mathbb{R}^{\widetilde{d}}$ (equivalently, from the associated symmetric phase factors Φ^W) to the Chebyshev coefficient vector $c \in \mathbb{R}^{\widetilde{d}}$ of the target polynomial $F(x)$ defined by $F(x) = \mathrm{Im}[U_{\Phi^W}(x)]_{1,1}$. The problem of finding phase factors in QSP is to solve the inverse problem: given $c \in \mathbb{R}^{\widetilde{d}}$ such that the associated real polynomial $F(x)$ satisfies the norm constraint in Corollary 12.6, find Φ^R such that $\mathscr{F}(\Phi^R) = c$.

There are many algorithms for finding phase factors. Here we present perhaps the simplest algorithm, called the **fixed-point iteration algorithm** (FPI). We start from a trivial reduced phase factors

$$\Phi^{R,(0)} = (0, 0, \ldots, 0) \in \mathbb{R}^{\widetilde{d}}. \tag{12.26}$$

The findings of Example 12.3 can be stated as that

$$\mathscr{F}(\Phi^{R,(0)}) = \mathbf{0} \in \mathbb{R}^{\widetilde{d}}, \tag{12.27}$$

i.e., $\mathscr{F}$ maps $\Phi^{R,(0)}$ to the all zero Chebyshev coefficients. Furthermore, the factor $\frac{1}{2}$ in Eq. (12.25) ensures that the Jacobian matrix of $\mathscr{F}$ at $\Phi^{R,(0)}$ is the scaled identity matrix, i.e., $\nabla\mathscr{F}(\Phi^{R,(0)}) = 2I_{\widetilde{d}}$.

Then given $c \in \mathbb{R}^{\widetilde{d}}$, starting from $\Phi^{R,(0)}$, the FPI algorithm is given in Algorithm 12.1. Conceptually, it only involves one line:

$$\Phi^{R,(\ell+1)} = \Phi^{R,(\ell)} - \left(\nabla\mathscr{F}(\Phi^{R,(0)}) \right)^{-1} \left(\mathscr{F}\left(\Phi^{R,(\ell)}\right) - c \right) = \Phi^{R,(\ell)} - \frac{1}{2}\left(\mathscr{F}\left(\Phi^{R,(\ell)}\right) - c \right), \quad \ell \in \mathbb{N}. \tag{12.28}$$

Ref. [DLNW24a] proves that the FPI method converges exponentially to one solution (called the maximal solution) when $\|c\|_1 \leq 0.861$.

Example 12.7. Consider $F(x) = \frac{1}{2}\cos(10x)$. We can first approximate the function by an even polynomial $p(x)$ using Chebyshev interpolation, and then use the fixed-point iteration (FPI) algorithm in Algorithm 12.1 to find symmetric phase factors Φ^W such that $\mathrm{Im}[U_{\Phi^W}(x)]_{1,1} = p(x)$.

Algorithm 12.1 Fixed-point iteration algorithm for finding reduced phase factors for a real polynomial with definite parity

Input: Chebyshev-coefficient vector c of a target polynomial, and stopping criteria.
Initialize $\Phi^{R,(0)} = (0, \ldots, 0)$, $\ell = 0$.
while stopping criterion is not satisfied **do**
  Update $\Phi^{R,(\ell+1)} \leftarrow \Phi^{R,(\ell)} - \frac{1}{2}\left(\mathscr{F}\left(\Phi^{R,(\ell)}\right) - c\right)$;
  Set $\ell \leftarrow \ell + 1$.
end while
Output: Reduced phase factors Φ^R.

Fig. 12.1 shows one such polynomial of degree 60. The QSP error (defined as the difference between the QSP representation and $p(x)$) approaches machine precision. The phase factors are symmetric with respect to the center of the interval and decay rapidly away from the center.

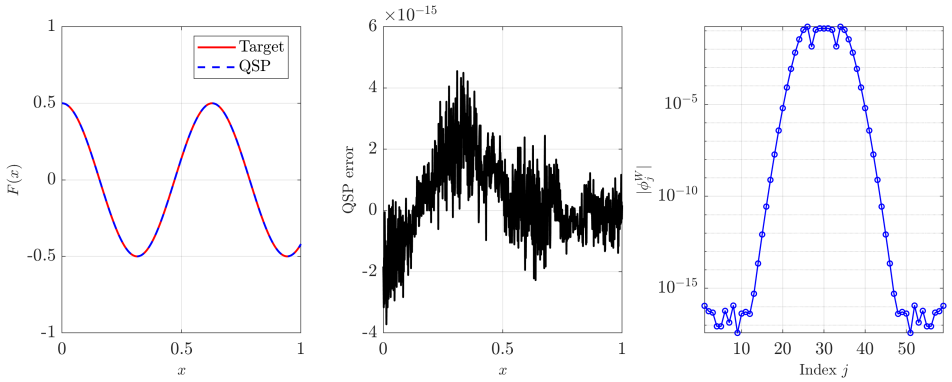


FIGURE 12.1. QSP representation of $F(x) = \frac{1}{2}\cos(10x)$ using an even polynomial $p(x)$ of degree 60. Left: the target function and the QSP representation of $p(x)$. Middle: Error between $p(x)$ and its QSP representation. Right: phase factors plotted on a log scale.

◇

12.4. Convex optimization-based method for constructing approximate polynomials

The fixed-point iteration method in Section 12.3 converts a QSP-admissible target polynomial into a phase sequence. We now discuss a complementary task: how to construct, for a given function g, an approximately optimal polynomial F that both approximates g on a prescribed set and satisfies the QSP admissibility constraint $|F(x)| \leq 1$.

Many applications of QSP (and, more generally, QSVT) aim to approximate a real function $g(x)$ of definite parity on a set $\mathcal{I} \subseteq [0, 1]$. Fix a polynomial degree d and define $\widetilde{d} := \lceil \frac{d+1}{2} \rceil$. According to Eq. (12.18), a real polynomial $F \in \mathbb{R}[x]$ of degree at most d with parity $d \bmod 2$ is

specified by its Chebyshev coefficient vector $c = (c_0, c_1, \ldots, c_{\widetilde{d}-1}) \in \mathbb{R}^{\widetilde{d}}$ via

$$F(x) = \sum_{j=0}^{\widetilde{d}-1} A_j(x) c_j, \tag{12.29}$$

where

$$A_j(x) = \begin{cases} T_{2j}(x), & d \text{ is even}, \\ T_{2j+1}(x), & d \text{ is odd}. \end{cases} \tag{12.30}$$

Conceptually, the coefficient vector c can be obtained by solving the following min-max optimization problem,

$$\begin{aligned} \min_{c \in \mathbb{R}^{\widetilde{d}}} \quad & \max_{x \in \mathcal{I}} |F(x) - g(x)| \\ \text{s.t.} \quad & F(x) = \sum_{j=0}^{\widetilde{d}-1} A_j(x) c_j, \quad |F(x)| \le 1, \quad \forall x \in [0,1]. \end{aligned} \tag{12.31}$$

Since F has definite parity, the constraint $|F(x)| \le 1$ on $[0,1]$ is equivalent to the corresponding constraint on $[-1,1]$. The objective function is convex in c and the constraint is linear in c. So this is a convex optimization problem. In fact, after introducing a slack variable denoted by z, the optimization problem becomes

$$\begin{aligned} \min_{c \in \mathbb{R}^{\widetilde{d}}} \quad & z \\ \text{s.t.} \quad & F(x) = \sum_{j=0}^{\widetilde{d}-1} A_j(x) c_j, \quad \forall x \in [0,1], \\ & F(x) \le 1, \quad \forall x \in [0,1], \\ & -F(x) \le 1, \quad \forall x \in [0,1], \\ & F(x) - g(x) \le z, \quad \forall x \in \mathcal{I}, \\ & g(x) - F(x) \le z, \quad \forall x \in \mathcal{I}. \end{aligned} \tag{12.32}$$

This is a linear programming problem. After discretization, it can be efficiently solved on a classical computer using standard convex optimization solvers.

In practice, we discretize Eq. (12.31) as follows. Without loss of generality, $\mathcal{I}$ takes the form of a union of intervals $\bigcup_{\ell=1}^{L} [a_\ell, b_\ell]$ with $0 \le a_1 < b_1 < a_2 < b_2 < \cdots < a_L < b_L \le 1$. Let $\mathcal{K}_{\mathcal{I}}$ be a set of grid points in $\mathcal{I}$, and let $\mathcal{K}_{\overline{\mathcal{I}}}$ be a set of grid points in the complement $\overline{\mathcal{I}} := [0,1] \backslash \mathcal{I}$ (which may be empty if $\mathcal{I} = [0,1]$). Since the bound constraint is enforced only on a finite grid, we generally have

$$\max_{x \in [0,1]} |F(x)| \ge \max_{x \in \mathcal{K}_{\mathcal{I}} \cup \mathcal{K}_{\overline{\mathcal{I}}}} |F(x)| \,. \tag{12.33}$$

To avoid overshooting, we assume $|g(x)| \le 1 - \eta$ for all $x \in \mathcal{I}$ and enforce $|F(x)| \le 1 - \eta$ on $\mathcal{K}_{\mathcal{I}} \cup \mathcal{K}_{\overline{\mathcal{I}}}$.

The discretized min-max optimization problem becomes

$$\begin{aligned} &\min_{c\in\mathbb{R}^{\widetilde{d}}} \quad \max_{x\in\mathcal{K}_{\mathcal{I}}} |F(x)-g(x)| \\ &\text{s.t.} \quad F(x)=\sum_{j=0}^{\widetilde{d}-1} A_j(x)c_j, \quad |F(x)|\le 1-\eta, \quad \forall x\in\mathcal{K}_{\mathcal{I}}\cup\mathcal{K}_{\overline{\mathcal{I}}}. \end{aligned} \tag{12.34}$$

For instance, using the CVXPY software package [GB14, DB16], the optimization problem can be solved with a few lines. Here the matrix A_{ij} evaluates $A_j(x_i)$ for $x_i \in \mathcal{K}_{\mathcal{I}} \cup \mathcal{K}_{\overline{\mathcal{I}}}$, and `I` is an index set referring to the set of grid points in $\mathcal{K}_{\mathcal{I}}$.

```
c = cp.Variable(n_degree)
f = cp.Variable(n_grid)
objective = cp.Minimize(cp.norm(f[I] - g[I], inf))
constraints = [f == A @ c,  f >= -(1-eta), f <= (1-eta)]
problem = cp.Problem(objective, constraints)
problem.solve()
```

12.5. Examples of quantum signal processing

In this section we consider two examples illustrating the workflow above. They will be used in Section 13.4.

12.5.1. Approximate sign function. We would like to approximate the sign function $\mathrm{sgn}(x)$ on $[-1,-\delta]\cup[\delta,1]$ by an odd polynomial, while keeping $|p(x)|\le 1$ on $[-1,1]$. We may use the following analytic construction (see [LC17a, Corollary 6] and [GSLW18, Lemma 25]). The proof of Lemma 12.8 is constructive.

Lemma 12.8 (Polynomial approximation to the sign function $\mathrm{sgn}(x)$)**.** *For any $\delta\in(0,1]$ and $\epsilon\in(0,\frac{1}{2})$, there exists an odd polynomial $p\in\mathbb{R}[x]$ of degree $d=\mathcal{O}\big(\frac{1}{\delta}\log(1/\epsilon)\big)$ that satisfies*

$$\sup_{x\in[\delta,1]} |p(x)-\mathrm{sgn}(x)|\le\epsilon, \qquad \sup_{x\in[-1,1]}|p(x)|\le 1. \tag{12.35}$$

Proof sketch. We first use the error function $\mathrm{erf}(kx)$ with $k=\mathcal{O}\big(\delta^{-1}\sqrt{\log(1/\epsilon)}\big)$ to approximate $\mathrm{sgn}(x)$ on $[-1,-\delta]\cup[\delta,1]$ to precision ϵ. We then estimate the accuracy of the Chebyshev approximation to this error function and multiply by a scaling factor to satisfy the bound constraint. □

From the convex optimization viewpoint, we may seek an odd polynomial F that approximates

$$g(x)=1, \quad x\in\mathcal{I}=[\delta,1]. \tag{12.36}$$

Fig. 12.2 compares the approximation error for the task of sign function approximation with $\delta=0.2$. The L^∞ errors measured on $[\delta,1]$ are comparable between the polynomial approximation derived using an analytic formula (through the constructive proof of Lemma 12.8) with degree 51, and the polynomial approximation from the convex optimization method with degree 31. We find that the convex optimization method can achieve a smaller error with a lower polynomial degree, which is not surprising since the optimization method directly minimizes the approximation error.

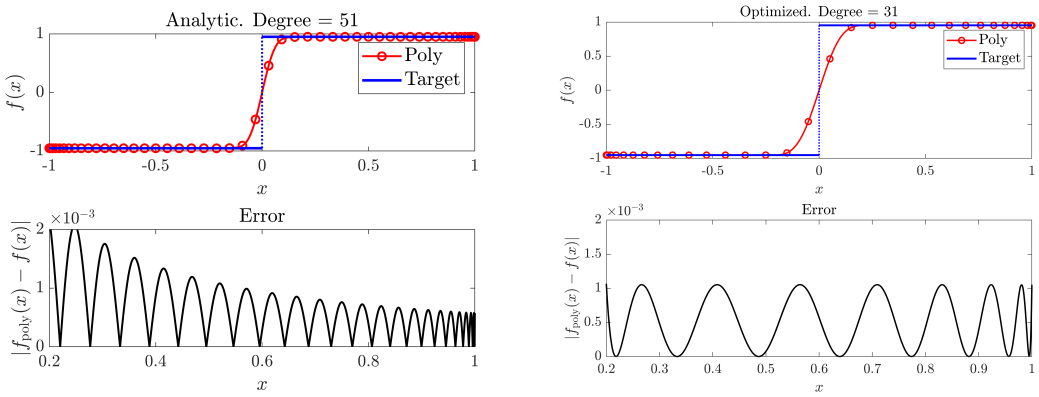


FIGURE 12.2. Comparison between polynomial approximations to the sign function on $[\delta, 1]$ using the analytic formula and convex optimization. Here $\delta = 0.2$.

12.5.2. Approximate truncated linear function. We consider an odd polynomial approximation to

$$g(x) = \alpha x, \quad x \in \mathcal{I} = [0, \alpha^{-1}]. \tag{12.37}$$

At first sight, this task seems trivial: the first-order polynomial $F(x) = \alpha x$ is odd and equals $g(x)$ exactly. However, αx violates the bound constraint on $[\alpha^{-1}, 1]$. It turns out that the polynomial must be more complicated in order to approximate a linear function as closely as possible on a subinterval, while satisfying the bound constraint on a larger interval. The proof of Lemma 12.9 is constructive, and we give a proof sketch here. We refer readers to [GSLW18, Lemma 29] and [LC17a, Corollary 8] for details.

Lemma 12.9 (Polynomial approximation to the truncated linear function)**.** *For any $\alpha \in (1, \infty)$, $\delta \in (0, 1]$, and $\epsilon \in (0, 1/(2\alpha)]$, there exists an odd polynomial $p \in \mathbb{R}[x]$ of degree $d = \mathcal{O}\big(\frac{\alpha}{\delta} \log(\alpha/\epsilon)\big)$ such that*

$$\sup_{x \in [0, \alpha^{-1}]} |(1+\delta)p(x) - \alpha x| \le \epsilon, \qquad \sup_{x \in [-1,1]} |p(x)| \le 1. \tag{12.38}$$

PROOF SKETCH. Using Lemma 12.8, we can construct a real polynomial $q(x)$ that approximates $\operatorname{sgn}(x)$ away from a small neighborhood of the origin at scale $\alpha^{-1}\delta$. Then the **even** polynomial

$$r(x) = \frac{q(x + \alpha^{-1}(1+\delta/2)) + q(\alpha^{-1}(1+\delta/2) - x)}{2}, \tag{12.39}$$

approximates a rectangular function supported on an interval of length $\mathcal{O}(\alpha^{-1})$,

$$g_{\mathrm{R},\alpha,\delta}(x) = \begin{cases} 1, & x \in [-\alpha^{-1}(1+\delta/2), \alpha^{-1}(1+\delta/2)], \\ 0, & \text{otherwise}, \end{cases} \tag{12.40}$$

on $\mathcal{I} = [0, \alpha^{-1}] \cup [\alpha^{-1}(1+\delta), 1]$. The target precision is $\epsilon' := \epsilon/\alpha$, and the polynomial degree is $d = \mathcal{O}\big(\frac{\alpha}{\delta} \log(\alpha/\epsilon)\big)$. The desired polynomial is obtained by setting $p(x) = \frac{\alpha x}{1+\delta} r(x)$. Therefore $(1+\delta)p(x)$ approximates αx on $[0, \alpha^{-1}]$ to precision ϵ. To satisfy the bound constraint $\sup_{x \in [-1,1]} |p(x)| \le 1$, one may include an additional scaling (at the level of $1 + \mathcal{O}(\epsilon)$) to avoid overshooting. □

Fig. 12.3 compares the approximation error with $\alpha = 5, \delta = 0.05$. The L^∞ error measured on $[0, \alpha^{-1}]$ are comparable between the polynomial approximation derived using an analytic formula (through the constructive proof of Lemma 12.9) with degree 1001, and the polynomial approximation from the convex optimization method with degree 51. We find that the convex optimization method can achieve a smaller error with a much lower polynomial degree. Furthermore, we find the polynomial approximation from the convex optimization method is much more oscillatory on $[\alpha^{-1}, 1]$ than the one from the analytic formula, which is perfectly acceptable since the approximation error is only measured on $[0, \alpha^{-1}]$.

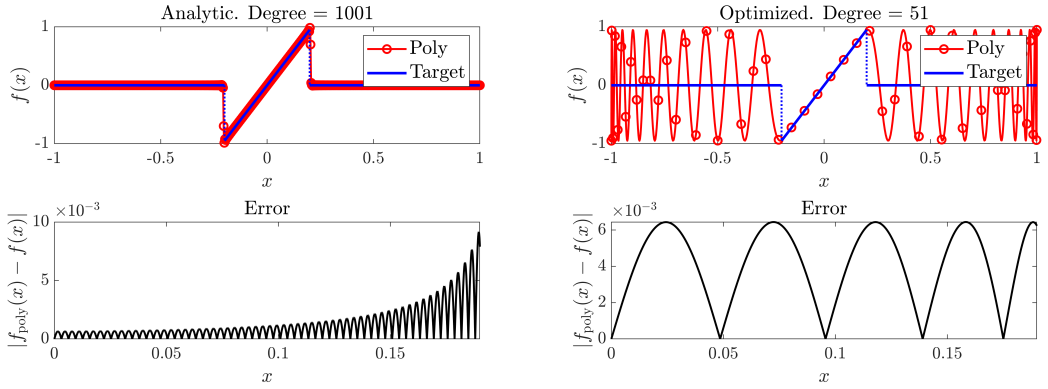


FIGURE 12.3. Comparison between polynomial approximations to x/a on $[0, a]$ using the analytic formula and convex optimization. Here $a = \alpha^{-1} = 0.2$, $\delta = 0.05$

12.6. Quantum signal processing and nonlinear Fourier transform on SU(2)

The key idea of QSP is that polynomial expressions, which are naturally built from addition and multiplication, can be represented through a structured product of matrices. This can be viewed as a special case of the **nonlinear Fourier transform** (NLFT), which replaces the addition operation in the linear Fourier transform with matrix multiplication.

In this section we introduce the NLFT on SU(2) in its simplest form. Given two integers $m < n$, let $\boldsymbol{\gamma} = (\gamma_m, \ldots, \gamma_n)$ be a complex-valued sequence. Its entries are called the nonlinear Fourier coefficients. The nonlinear Fourier transform of $\boldsymbol{\gamma}$ is defined as a finite product of matrix-valued functions,

$$\widehat{\boldsymbol{\gamma}}(z) := \prod_{k=m}^{n} \left[\frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & \gamma_k z^k \\ -\overline{\gamma_k} z^{-k} & 1 \end{pmatrix} \right] = \begin{pmatrix} a(z) & b(z) \\ * & * \end{pmatrix}, \quad z \in \mathbb{C}. \tag{12.41}$$

The upper left entry $a(z)$ and upper right entry $b(z)$ are in general not polynomials but Laurent polynomials, i.e., they can contain both positive and negative powers of z.

Taking the determinant of the matrix factors appearing in Eq. (12.41), we see that the determinant of each factor, and hence also of $\widehat{\boldsymbol{\gamma}}(z)$, is 1 everywhere. Moreover, when z is restricted to the unit circle denoted by $\mathbb{T}$, the matrix factors are elements of SU(2), and thus so is $\widehat{\boldsymbol{\gamma}}(z)$.

When $\sum_{k=m}^{n} |\gamma_k|$ is small, the NLFT of $\boldsymbol{\gamma}$ can be approximated by its linear approximation,

$$\widehat{\boldsymbol{\gamma}}(z) \approx \begin{pmatrix} 1 & \sum_{k=m}^{n} \gamma_k z^k \\ -\sum_{k=m}^{n} \overline{\gamma_k} z^{-k} & 1 \end{pmatrix}. \tag{12.42}$$

Therefore, the standard Fourier series can be viewed as the leading order contribution to the upper-right entry of $\widehat{\boldsymbol{\gamma}}(z)$, where the coefficients γ_k are sufficiently small that higher-order terms in the product expansion can be neglected. When the coefficients γ_k are not small, the difference between the two quantities can become significant. The transformation from $\boldsymbol{\gamma}$ to $\widehat{\boldsymbol{\gamma}}$ is called the forward NLFT, and the mapping from $\widehat{\boldsymbol{\gamma}}$ back to $\boldsymbol{\gamma}$ is called the inverse NLFT.

Proposition 12.10 (Connection between QSP and NLFT)**.** *For any $d \in \mathbb{N}$ and $\Phi^W := (\phi_0^W, \phi_1^W, \cdots, \phi_d^W)$ in the W-convention with $\phi_k^W \in (-\pi/2, \pi/2)$, define the sequence $\boldsymbol{\gamma} = (\gamma_k)_{k=0}^d$ with $\gamma_k := \tan \phi_k^W$. Then for all $\theta \in [0, \pi]$ we have*

$$S\,\mathrm{H}\; U_{\Phi W}(\cos\theta)\; \mathrm{H}\, S^\dagger = \widehat{\boldsymbol{\gamma}}(e^{2i\theta}) \begin{pmatrix} e^{id\theta} & 0 \\ 0 & e^{-id\theta} \end{pmatrix}, \tag{12.43}$$

where $U_{\Phi W}$ is defined in Eq. (12.16) *with $x = \cos\theta$,* H *is the Hadamard gate, and S is the phase gate.*

Furthermore, $\mathrm{Im}[P(\cos\theta)] = \mathrm{Re}\big(b(e^{2i\theta})e^{-id\theta}\big)$*, where P is the upper-left entry of $U_{\Phi W}$ (as in Eq.* (12.16)*), and b is the upper-right entry of $\widehat{\boldsymbol{\gamma}}$ as in Eq.* (12.41)*.*

Proof. Recall that $\mathrm{H}\,Z\,\mathrm{H} = X$ and $\mathrm{H}\,X\,\mathrm{H} = Z$. Set $x = \cos\theta$. Define $\widetilde{W}(x) := \mathrm{H}\,W(x)\,\mathrm{H}$. Since $\cos\theta = x$, we have

$$\widetilde{W}(x) = \mathrm{H}\,W(x)\,\mathrm{H} = \mathrm{H}\,e^{i\theta X}\,\mathrm{H} = e^{i\theta Z}. \tag{12.44}$$

Since S commutes with Z, it follows that $S\widetilde{W}(x)S^\dagger = \widetilde{W}(x)$. Together with the definition of γ_k, this implies

$$S\,\mathrm{H}\,e^{i\phi_k^W Z}\,\mathrm{H}\,S^\dagger = Se^{i\phi_k^W X}S^\dagger = e^{i\phi_k^W Y} = \frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & \gamma_k \\ -\overline{\gamma_k} & 1 \end{pmatrix}, \;\; k = 0, \ldots, d. \tag{12.45}$$

Next, the left-hand sides of Eq. (12.43) equate to $S\,\mathrm{H}\,U_{\Phi W}(\cos\theta)\,\mathrm{H}\,S^\dagger$, which we can also express in the following ordered product form using Eqs. (12.44) and (12.45):

$$S\,\mathrm{H}\,U_{\Phi W}(\cos\theta)\,\mathrm{H}\,S^\dagger = \frac{1}{\sqrt{1+|\gamma_0|^2}} \begin{pmatrix} 1 & \gamma_0 \\ -\overline{\gamma_0} & 1 \end{pmatrix} \prod_{k=1}^{d} \left[e^{i\theta Z} \frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & \gamma_k \\ -\overline{\gamma_k} & 1 \end{pmatrix} \right]. \tag{12.46}$$

Finally, notice the following identity, for any $t \in \mathbb{R}$:

$$\begin{pmatrix} e^{i\theta t} & 0 \\ 0 & e^{-i\theta t} \end{pmatrix} \begin{pmatrix} 1 & \gamma_k \\ -\overline{\gamma_k} & 1 \end{pmatrix} = \begin{pmatrix} 1 & \gamma_k e^{2i\theta t} \\ -\overline{\gamma_k} e^{-2i\theta t} & 1 \end{pmatrix} \begin{pmatrix} e^{i\theta t} & 0 \\ 0 & e^{-i\theta t} \end{pmatrix}, \tag{12.47}$$

which allows us to simplify the right-hand side of Eq. (12.46) to obtain

$$\begin{aligned} S\,\mathrm{H}\,U_{\Phi W}(\cos\theta)\,\mathrm{H}\,S^\dagger &= \left(\prod_{k=0}^{d} \left[\frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & \gamma_k e^{2ik\theta} \\ -\overline{\gamma_k} e^{-2ik\theta} & 1 \end{pmatrix} \right] \right) \begin{pmatrix} e^{id\theta} & 0 \\ 0 & e^{-id\theta} \end{pmatrix} \\ &= \widehat{\boldsymbol{\gamma}}(e^{2i\theta}) \begin{pmatrix} e^{id\theta} & 0 \\ 0 & e^{-id\theta} \end{pmatrix}. \end{aligned} \tag{12.48}$$

This is exactly Eq. (12.43). Furthermore, writing $\sin\theta = \sqrt{1-x^2}$ and using Eq. (12.16), a direct computation gives

$$S\,\mathrm{H}\,U_{\Phi w}(\cos\theta)\,\mathrm{H}\,S^\dagger = \begin{pmatrix} * & \mathrm{Im}[P(\cos\theta)] - i\,\mathrm{Im}[Q(\cos\theta)]\sin\theta \\ * & * \end{pmatrix}. \tag{12.49}$$

While the right-hand side of Eq. (12.43) is

$$\begin{pmatrix} * & b(e^{2i\theta})e^{-id\theta} \\ * & * \end{pmatrix}.$$

By comparing the real part of the upper right element and using Eq. (12.49), we conclude that $\mathrm{Im}[P(\cos\theta)] = \mathrm{Re}\big(b(e^{2i\theta})e^{-id\theta}\big)$. □

Based on Proposition 12.10, we see that computing the phase factors can be reduced to an inverse NLFT problem. In particular, if we are interested in solving QSP phase factors for $F(x) = \mathrm{Im}[P(x)]$, one may seek an SU(2)-valued function $\overbrace{\boldsymbol{\gamma}}(z)$ on $\mathbb{T}$ whose upper-right entry $b(z)$ satisfies $F(\cos\theta) = \mathrm{Re}\big(b(e^{2i\theta})e^{-id\theta}\big)$, recover the nonlinear Fourier coefficients $\boldsymbol{\gamma}$ via the inverse NLFT, and then obtain the phase factors by $\phi_k^W = \arctan(\gamma_k)$.

12.7. Infinite quantum signal processing

So far the discussion of QSP has been focused on polynomial functions. The problem of **infinite quantum signal processing** (iQSP) asks whether the QSP representation can be extended to non-polynomial functions f through a product of countably many unitary matrices. NLFT provides a satisfactory answer to this question. For simplicity, we focus on the case of real-valued even functions defined on $[0,1]$.

The convention for the integral on the unit circle $\mathbb{T}$ is

$$\int_{\mathbb{T}} g := \frac{1}{2\pi}\int_0^{2\pi} g(e^{i\theta})\,\mathrm{d}\theta. \tag{12.50}$$

If a real-valued measurable even function $f : [0,1] \to \mathbb{R}$ can be expressed as

$$f(\cos\theta) = g(e^{i\theta}), \quad \forall\theta \in [0, 2\pi), \tag{12.51}$$

for some function g defined on $\mathbb{T}$, then the **Szegő norm** of f, denoted $\|f\|_{\mathbf{S}}$, can be defined as the L^2 norm of g on $\mathbb{T}$:

$$\int_{\mathbb{T}} |g|^2 := \frac{1}{2\pi}\int_0^{2\pi} |f(\cos\theta)|^2\,\mathrm{d}\theta = \frac{2}{\pi}\int_0^1 |f(x)|^2\,\frac{\mathrm{d}x}{\sqrt{1-x^2}} := \|f\|_{\mathbf{S}}^2\,. \tag{12.52}$$

A real-valued measurable even function $f : [0,1] \to [-1,1]$ is called a **Szegő function** if it satisfies the following Szegő-type condition:

$$\int_0^1 \log|1 - f(x)^2|\frac{\mathrm{d}x}{\sqrt{1-x^2}} > -\infty\,. \tag{12.53}$$

We use $\mathbf{S}$ to denote the set of all Szegő functions. Since $y \le -\log(1-y)$ for all $y \in [0,1)$, the Szegő condition in Eq. (12.53) implies that $\|f\|_{\mathbf{S}} < \infty$.

In standard L^2 theory of Fourier analysis, the Plancherel identity plays a fundamental role, namely for $f(x) = \sum_{k=0}^\infty c_k T_k(x)$, we have

$$\int_{-1}^1 |f(x)|^2\,\frac{\mathrm{d}x}{\sqrt{1-x^2}} = \pi\,|c_0|^2 + \frac{\pi}{2}\sum_{k=1}^\infty |c_k|^2\,. \tag{12.54}$$

Let $\mathbf{P}$ denote the space of infinite sequences $\Phi = (\phi_k)_{k\in\mathbb{N}}$ with $\phi_k \in [-\pi/2, \pi/2]$. Given any $\Phi \in \mathbf{P}$ and $x \in [0,1]$, one can define a sequence of unitary matrices using the following recursive relation:

$$\begin{aligned} V_0(x,\Phi) &= e^{i\phi_0 Z} \\ V_d(x,\Phi) &= e^{i\phi_d Z} W(x) V_{d-1}(x,\Phi) W(x) e^{i\phi_d Z}. \end{aligned} \tag{12.55}$$

A direct check shows that this corresponds to symmetric phase factors Φ^W of the form

$$\Phi^W = (\phi_d, \phi_{d-1}, \ldots, \phi_1, \phi_0, \phi_1, \ldots, \phi_{d-1}, \phi_d) \in \mathbb{R}^{2d+1}, \tag{12.56}$$

such that $V_d(x,\Phi) = U_{\Phi^W}(x)$. So Φ can be viewed as the reduced phase factors in the infinite dimensional case. Let $P_d(x,\Phi) = [V_d(x,\Phi)]_{1,1}$. NLFT provides a L^2-theory for infinite QSP for all Szegő functions, together with a nonlinear generalization of the Plancherel identity [ALM+26, Theorem 1].

THEOREM 12.11. *For each $f \in \mathbf{S}$ satisfying $\|f\|_\infty < 1$, there exists a unique sequence $\Phi = (\phi_k)_{k\in\mathbb{N}} \in \mathbf{P}$ such that*

(a) $\operatorname{Im} P_d(\cdot,\Phi)$ *converges to f in the L^2 sense:*

$$\lim_{d\to\infty} \|\operatorname{Im} P_d(\cdot,\Phi) - f\|_{\mathbf{S}} = 0, \tag{12.57}$$

(b) *the following* ***nonlinear Plancherel identity*** *holds:*

$$-\frac{2}{\pi}\int_0^1 \log|1-f(x)^2| \frac{\mathrm{d}x}{\sqrt{1-x^2}} = \sum_{k\in\mathbb{Z}} \log(1+\tan^2\phi_{|k|}). \tag{12.58}$$

Notes and further reading

Quantum signal processing (QSP) grew out of single-qubit composite gate design, where structured products of SU(2) rotations are used to synthesize prescribed response functions; see [LYC16]. The formulation was subsequently developed into a general primitive for implementing polynomial transformations, achieving optimal query complexity for Hamiltonian simulation [LC17b]. The extension from this scalar SU(2) representation to matrix singular value transformation, together with the block-encoding viewpoint and its algorithmic consequences, was established in [GSLW19]; see also reviews in [MRTC21, Lin25]. The symmetric choice of phase factors was first suggested in [DMWL21], and Theorem 12.5 and Corollary 12.6 were subsequently developed in [WDL22].

From the algorithmic perspective, the proof of [GSLW19, Corollary 5] gives a constructive synthesis for real polynomials by first computing complementary polynomials satisfying the constraints of Theorem 12.1, and then recovering phase factors via a recursive "layer stripping" procedure. As analyzed in [Haa19], layer stripping is numerically unstable in general and may require $\mathcal{O}(d\log(d/\epsilon))$ bits of working precision, where d is the degree and ϵ is the target approximation error. Recent work has substantially improved the stability and practical performance of algorithms [CDG+20, DMWL21, Yin22, WDL22, DLNW24a, DLNW24b, BS24, AMT24, ALM+26, MW24, NY24, NSYL25]. The fixed-point iteration algorithm in Algorithm 12.1, together with many other numerical methods for finding phase factors, is implemented in QSPPACK[1]. We used QSPPACK throughout the book for QSP related examples.

Ref. [DLNW24a] provided the first infinite QSP construction. The connection between QSP and the SU(2) nonlinear Fourier transform (NLFT), as formalized in Proposition 12.10, was established

[1] `https://qsppack.gitbook.io/qsppack/`

in [AMT24]. Theorem 12.11 was first established in [AMT24] for Szegő" functions satisfying $\|f\|_\infty < \frac{1}{\sqrt{2}}$, and extended to all Szegő" functions in [ALM$^+$26]. Furthermore, NLFT provides a unified description of QSP and its generalization, called generalized quantum signal processing [MW24].

CHAPTER 13

Quantum singular value transformation

Quantum signal processing (QSP) provides a systematic way to implement scalar polynomial transformations by composing a sequence of single-qubit rotations with a fixed signal oracle. Quantum singular value transformation (QSVT) "lifts" the QSP construction from scalars to matrices via qubitization.

We begin by deriving QSVT directly from the qubitization structure given by the cosine–sine decomposition. We then discuss circuit-level refinements such as efficient controlled implementations and applications of QSVT including fixed-point amplitude amplification, uniform singular value amplification, and Gibbs state preparation via polynomial approximation and purification. We also use perturbation theory to explain the robustness of singular value transformations to approximate input oracles.

13.1. Derivation from cosine–sine decomposition

We begin with the cosine–sine decomposition form of qubitization (Theorem 10.10). For an n-qubit matrix A encoded by $U_A \in \mathrm{BE}_{1,m}(A)$, let $N = 2^n$ and $M = 2^m$. Then there exist $(m+n)$-qubit unitary matrices $\widetilde{W}, \widetilde{V}$, and an $(n+1)$-qubit permutation matrix $\mathscr{P}$ that permutes rows $\{0, 1, \ldots, N-1, N, \ldots, 2N-1\}$ to $\{0, N, 1, N+1, \ldots, N-1, 2N-1\}$, such that

(13.1)
$$U_A = \widetilde{W} \begin{pmatrix} \Sigma & S & 0 \\ S & -\Sigma & 0 \\ 0 & 0 & I_{(M-2)N} \end{pmatrix} \widetilde{V}^\dagger = \widetilde{W} \left\{ \mathscr{P} \bigoplus_{i \in [N]} \begin{pmatrix} \sigma_i & \sqrt{1-\sigma_i^2} \\ \sqrt{1-\sigma_i^2} & -\sigma_i \end{pmatrix} \mathscr{P}^\dagger \bigoplus I_{(M-2)N} \right\} \widetilde{V}^\dagger,$$

and

(13.2)
$$U_A^\dagger = \widetilde{V} \begin{pmatrix} \Sigma & S & 0 \\ S & -\Sigma & 0 \\ 0 & 0 & I_{(M-2)N} \end{pmatrix} \widetilde{W}^\dagger = \widetilde{V} \left\{ \mathscr{P} \bigoplus_{i \in [N]} \begin{pmatrix} \sigma_i & \sqrt{1-\sigma_i^2} \\ \sqrt{1-\sigma_i^2} & -\sigma_i \end{pmatrix} \mathscr{P}^\dagger \bigoplus I_{(M-2)N} \right\} \widetilde{W}^\dagger.$$

Following the same decomposition, Z_Π can be written as

(13.3)
$$Z_\Pi = \left\{ \mathscr{P} \left(\bigoplus_{i \in [N]} Z \right) \mathscr{P}^\dagger \right\} \bigoplus (-I)_{(M-2)N}.$$

It is important that each 2×2 block $\begin{pmatrix} \sigma_i & \sqrt{1-\sigma_i^2} \\ \sqrt{1-\sigma_i^2} & -\sigma_i \end{pmatrix}$ is a reflection; in particular, this block is unchanged when passing from U_A to $U_A^\dagger$.

Let $U_\Phi(x)$ denote the SU(2) matrix in the QSP representation (for instance, Eq. (12.4) in the O-convention) evaluated at $x \in [-1,1]$:

$$U_\Phi(x) = e^{i\phi_0 Z} \prod_{j=1}^{d} \left[O(x) e^{i\phi_j Z}\right] = \begin{pmatrix} P(x) & -Q(x)\sqrt{1-x^2} \\ \overline{Q(x)}\sqrt{1-x^2} & \overline{P(x)} \end{pmatrix} \tag{13.4}$$

Here P is a complex polynomial of degree at most d with parity $d \bmod 2$. We would like to lift this scalar identity to a unitary (still denoted by U_Φ) that implements a block-encoding of the singular value transformation $P^{\mathrm{SV}}(A)$.

Assume first that d is even and consider the following circuit:

$$U_\Phi = e^{i\phi_0 Z_\Pi} \prod_{j=1}^{d/2} \left[(U_A^\dagger Z_\Pi) e^{i\phi_{2j-1} Z_\Pi} (U_A Z_\Pi) e^{i\phi_{2j} Z_\Pi}\right]. \tag{13.5}$$

Then

$$\begin{aligned} U_\Phi =& \widetilde{V} \left\{ \mathscr{P} \bigoplus_{i\in[N]} e^{i\phi_0 Z} \prod_{j=1}^{d} \left[O(\sigma_i) e^{i\phi_j Z}\right] \mathscr{P}^\dagger \bigoplus I_{(M-2)N} \right\} \widetilde{V}^\dagger \\ =& \widetilde{V} \left\{ \mathscr{P} \bigoplus_{i\in[N]} U_\Phi(\sigma_i) \mathscr{P}^\dagger \bigoplus I_{(M-2)N} \right\} \widetilde{V}^\dagger. \end{aligned} \tag{13.6}$$

Therefore U_Φ is a $(1, m+1)$-block-encoding of $P^{\triangleright}(A)$, where $P \in \mathbb{C}[x]$ is the polynomial appearing as the $(1,1)$ entry in Eq. (12.4). Since d is even, P is even, and hence $P^{\mathrm{SV}}(A) = P^{\triangleright}(A)$.

Similarly, when d is odd, consider the circuit:

$$U_\Phi = e^{i\phi_0 Z_\Pi} (U_A Z_\Pi e^{i\phi_1 Z_\Pi}) \prod_{j=1}^{(d-1)/2} \left[(U_A^\dagger Z_\Pi) e^{i\phi_{2j} Z_\Pi} (U_A Z_\Pi) e^{i\phi_{2j+1} Z_\Pi}\right]. \tag{13.7}$$

Then

$$\begin{aligned} U_\Phi =& \widetilde{W} \left\{ \mathscr{P} \bigoplus_{i\in[N]} e^{i\phi_0 Z} \prod_{j=1}^{d} \left[O(\sigma_i) e^{i\phi_j Z}\right] \mathscr{P}^\dagger \bigoplus I_{(M-2)N} \right\} \widetilde{V}^\dagger \\ =& \widetilde{W} \left\{ \mathscr{P} \bigoplus_{i\in[N]} U_\Phi(\sigma_i) \mathscr{P}^\dagger \bigoplus I_{(M-2)N} \right\} \widetilde{V}^\dagger. \end{aligned} \tag{13.8}$$

Therefore U_Φ is a $(1, m+1)$-block-encoding of $P^{\diamond}(A)$. Since d is odd, P is odd, and hence $P^{\mathrm{SV}}(A) = P^{\diamond}(A)$.

The unitaries in Eqs. (13.5) and (13.7) are called **quantum singular value transformation** (QSVT).

Note that these expressions involve interleaved factors of Z_Π and also require implementing $e^{i\phi Z_\Pi}$. Below we show that these can be implemented using multi-qubit controlled-NOT gates, single-qubit rotations, and a simple modification of the phase factors.

To implement $e^{i\phi Z_\Pi}$, we note that the circuit denoted by $\mathrm{CR}(\phi)$ in Fig. 13.1 maps $|0\rangle |0^m\rangle$ to $e^{i\phi} |0\rangle |0^m\rangle$, and maps $|0\rangle |b\rangle$ to $e^{-i\phi} |0\rangle |b\rangle$ for $b \neq 0^m$. The first (signal) qubit is returned to $|0\rangle$,

so ignoring it, the induced operation on the m-qubit register is exactly $e^{i\phi Z_\Pi}$. Moreover, we do not need a separate implementation of Z_Π. Using $Z = -ie^{i\frac{\pi}{2}Z} = ie^{-i\frac{\pi}{2}Z}$, we have

$$Ze^{i\phi Z} = (-i)e^{i(\phi+\frac{\pi}{2})Z} = ie^{i(\phi-\frac{\pi}{2})Z}, \tag{13.9}$$

so by adding and subtracting $\pi/2$ to the phase factors in an alternating pattern we can absorb interleaved Z factors without introducing an additional global phase. This is particularly useful for controlled implementations of QSVT.

When d is even, let

$$\phi_j^C = \begin{cases} \phi_0, & j = 0, \\ \phi_j + \frac{\pi}{2}, & j \in \{1,\dots,d\} \text{ and } j \text{ is odd}, \\ \phi_j - \frac{\pi}{2}, & j \in \{1,\dots,d\} \text{ and } j \text{ is even}. \end{cases} \tag{13.10}$$

Then the overall global phase due to the resulting powers of i is canceled and

$$e^{i\phi_0^C Z}\prod_{j=1}^{d}\left[\begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} e^{i\phi_j^C Z}\right] = e^{i\phi_0 Z}\prod_{j=1}^{d}\left[O(x)e^{i\phi_j Z}\right] = U_\Phi(x) = \begin{pmatrix} P(x) & -Q(x)\sqrt{1-x^2} \\ \overline{Q(x)}\sqrt{1-x^2} & \overline{P(x)} \end{pmatrix}. \tag{13.11}$$

When d is odd, we can choose

$$\phi_j^C = \begin{cases} \phi_0 - \frac{\pi}{2}, & j = 0, \\ \phi_j + \frac{\pi}{2}, & j \in \{1,\dots,d\} \text{ and } j \text{ is odd}, \\ \phi_j - \frac{\pi}{2}, & j \in \{1,\dots,d\} \text{ and } j \text{ is even}. \end{cases} \tag{13.12}$$

This cancels all the global phase factors, but there is an additional Z factor

$$e^{i\phi_0^C Z}\prod_{j=1}^{d}\left[\begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} e^{i\phi_j^C Z}\right] = Ze^{i\phi_0 Z}\prod_{j=1}^{d}\left[O(x)e^{i\phi_j Z}\right] = ZU_\Phi(x) = \begin{pmatrix} P(x) & -Q(x)\sqrt{1-x^2} \\ -\overline{Q(x)}\sqrt{1-x^2} & -\overline{P(x)} \end{pmatrix}. \tag{13.13}$$

Despite the additional Z factor in Eq. (13.13), both sides are block-encodings of the same polynomial $P(x)$ in the $(1,1)$ entry.

The phase factors $\Phi^C = (\phi_0^C, \cdots, \phi_d^C)$ are called the **C-convention** (or circuit convention) of phase factors associated with Φ (the O-convention). The modification rule can be summarized as follows.

$$\phi_j^C = \begin{cases} \phi_0 - \frac{\pi}{2}, & j = 0 \text{ and } d \text{ is odd}, \\ \phi_0, & j = 0 \text{ and } d \text{ is even}, \\ \phi_j + \frac{\pi}{2}, & j \in \{1,\dots,d\} \text{ and } j \text{ is odd}, \\ \phi_j - \frac{\pi}{2}, & j \in \{1,\dots,d\} \text{ and } j \text{ is even}. \end{cases} \tag{13.14}$$

The modification rule of phase factors starting from the W-convention is similar and is omitted here.

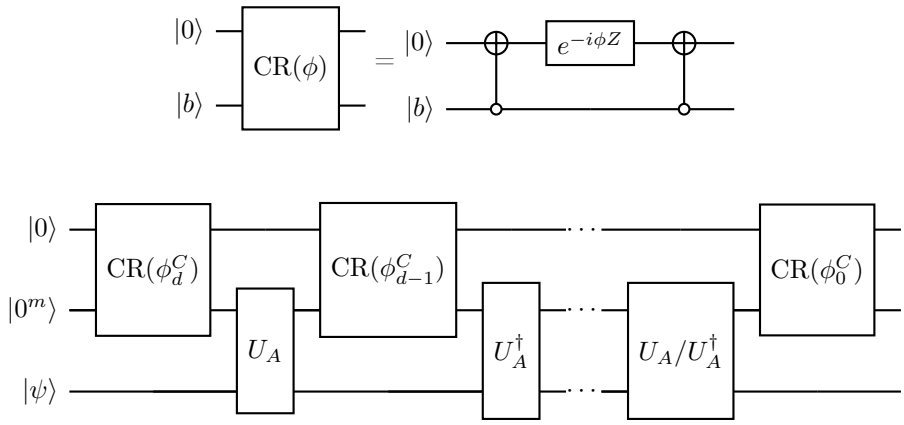


FIGURE 13.1. Circuit of quantum singular value transformation to construct $U_{P^{\mathrm{SV}}(A)} \in \mathrm{BE}_{1,m+1}(P^{\mathrm{SV}}(A))$, using $U_A \in \mathrm{BE}_{1,m}(A)$, and $U_A, U_A^\dagger$ are applied in an alternating order. The last gate is U_A when the degree d is odd, and is $U_A^\dagger$ if d is even. One possible way to express the phase factors $\{\phi_j^C\}$ in the circuit in terms of $\{\phi_j\}$ in Theorem 12.1 is given in Eq. (13.14).

THEOREM 13.1 (Quantum singular value transformation with complex polynomials of definite parity). *Let $A \in \mathbb{C}^{N\times N}$ be encoded by its $(1,m)$-block-encoding U_A. For a complex polynomial $P(x) \in \mathbb{C}[x]$ with degree d and parity given by $d \mod 2$ satisfying the conditions in Theorem 12.1, we can find a sequence of phase factors $\Phi \in \mathbb{R}^{d+1}$ according to Theorem 12.1, and a corresponding sequence of C-convention phase factors Φ^C according to Eq.* (13.14)*. With this sequence Φ^C, the circuit in Fig. 13.1 implements a $(1, m+1)$-block-encoding of $P^{\mathrm{SV}}(A)$, using $U_A, U_A^\dagger$, m-qubit controlled-NOT, and single-qubit rotation gates a total of $\mathcal{O}(d)$ times.*

13.2. Real polynomial singular value transformation

Instead of $P^{\mathrm{SV}}(A)$, in many applications we are only interested in a block-encoding of

$$F^{\mathrm{SV}}(A) = \frac{1}{2}(P^{\mathrm{SV}}(A) + \overline{P}^{\mathrm{SV}}(A)), \quad F(x) = \mathrm{Re}\, P(x). \tag{13.15}$$

This can be done by using Theorem 13.1 to construct a block-encoding of $P^{\mathrm{SV}}(A), \overline{P}^{\mathrm{SV}}(A)$, respectively, and use LCU with one ancilla qubit to construct the linear combination. However, due to the special structure of QSP, we demonstrate an elegant way to solve this problem without introducing any additional ancilla qubit.

Given $\Phi = (\phi_0, \cdots, \phi_d) \in \mathbb{R}^{d+1}$, define $-\Phi := (-\phi_0, \cdots, -\phi_d) \in \mathbb{R}^{d+1}$. Taking entrywise complex conjugation of $U_\Phi(x)$ in Eq. (12.4) gives

$$(U_\Phi(x))^* = U_{-\Phi}(x) = e^{-i\phi_0 Z} \prod_{j=1}^{d} \left[O(x) e^{-i\phi_j Z}\right] = \begin{pmatrix} \overline{P(x)} & -\overline{Q(x)}\sqrt{1-x^2} \\ Q(x)\sqrt{1-x^2} & P(x) \end{pmatrix}. \tag{13.16}$$

As a result, from qubitization, when d is even,

$$U_{-\Phi} = \widetilde{V}\left\{\mathscr{P}\bigoplus_{i\in[N]} U_{-\Phi}(\sigma_i)\mathscr{P}^\dagger \bigoplus I_{(M-2)N}\right\}\widetilde{V}^\dagger. \tag{13.17}$$

is a $(1, m+1)$-block-encoding of $\overline{P}^{\triangleright}(A)$ for an even polynomial P. From Eq. (13.10),

$$e^{-i\phi_0^C Z}\prod_{j=1}^{d}\left[\begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} e^{-i\phi_j^C Z}\right] = e^{-i\phi_0 Z}\prod_{j=1}^{d}\left[O(x)e^{-i\phi_j Z}\right] = U_{-\Phi}(x). \tag{13.18}$$

Thus $U_{-\Phi}$ can be implemented by negating each phase factor ϕ_j^C.

When d is odd,

$$U_{-\Phi} = \widetilde{W}\left\{\mathscr{P}\bigoplus_{i\in[N]} U_{-\Phi}(\sigma_i)\mathscr{P}^\dagger \bigoplus I_{(M-2)N}\right\}\widetilde{V}^\dagger \tag{13.19}$$

is a $(1, m+1)$-block-encoding of $\overline{P}^{\diamond}(A)$ for an odd polynomial P. From Eq. (13.12),

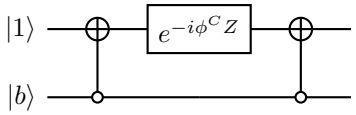
$$e^{-i\phi_0^C Z}\prod_{j=1}^{d}\left[\begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} e^{-i\phi_j^C Z}\right] = Ze^{-i\phi_0 Z}\prod_{j=1}^{d}\left[O(x)e^{-i\phi_j Z}\right] = ZU_{-\Phi}(x). \tag{13.20}$$

Thus negating each phase factor ϕ_j^C implements the unitary

$$\widetilde{W}\left\{\mathscr{P}\bigoplus_{i\in[N]} ZU_{-\Phi}(\sigma_i)\mathscr{P}^\dagger \bigoplus I_{(M-2)N}\right\}\widetilde{V}^\dagger \tag{13.21}$$

which is a block-encoding of $\overline{P}^{\diamond}(A)$ (the additional Z only affects other blocks).

In summary, it suffices to negate all phase factors in Φ^C. In order to implement $\mathrm{CR}(-\phi^C)$, we do not actually need to implement a new circuit. Instead we may simply initialize the signal qubit in $|1\rangle$:



which outputs $e^{-i\phi^C}|1\rangle|0^m\rangle$ if $b = 0^m$, and $e^{i\phi^C}|1\rangle|b\rangle$ if $b \neq 0^m$. Equivalently, relative to the convention of $\mathrm{CR}(\phi^C)$, this realizes $\mathrm{CR}(-\phi^C)$. In other words, the circuits for $U_{\overline{P}^{\mathrm{SV}}(A)}$ and $U_{P^{\mathrm{SV}}(A)}$ are exactly the same except that the state of the signal qubit is changed from $|0\rangle$ to $|1\rangle$.

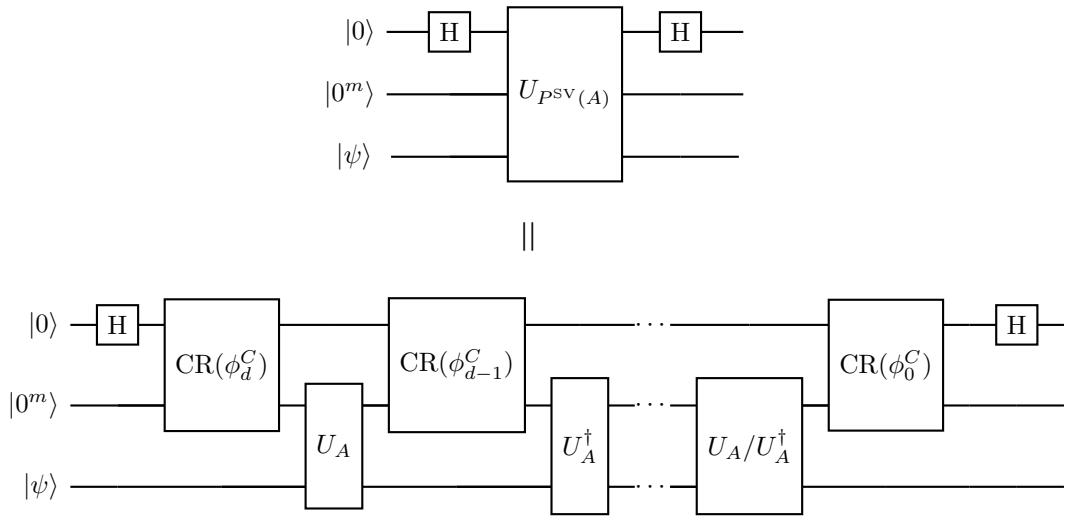


FIGURE 13.2. Circuit of quantum singular value transformation to construct $U_{F^{\mathrm{SV}}(A)} \in \mathrm{BE}_{1,m+1}(F^{\mathrm{SV}}(A))$, using $U_A \in \mathrm{BE}_{1,m}(A)$, and $U_A, U_A^\dagger$ are applied in an alternating order. Here $F(x) = \mathrm{Re}\, P(x)$. The last gate is U_A when the degree d is odd, and is $U_A^\dagger$ if d is even. One possible way to express the phase factors $\{\phi_j^C\}$ in the circuit in terms of $\{\phi_j\}$ in Theorem 12.1 is given in Eq. (13.14). Conceptually, this is a circuit using a linear combination of block-encodings for $P^{\mathrm{SV}}(A)$ and $\overline{P}^{\mathrm{SV}}(A)$.

Now we claim the circuit in Fig. 13.2 implements a block-encoding of $F^{\mathrm{SV}}(A)$ via a linear combination of unitaries. Direct calculation shows

$$
\begin{aligned}
& |0\rangle |0^m\rangle |\psi\rangle \\
\xrightarrow{\mathrm{H}\otimes I} & \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) |0^m\rangle |\psi\rangle \\
\xrightarrow{U_{P^{\mathrm{SV}}(A)}} & \frac{1}{\sqrt{2}} |0\rangle (|0^m\rangle P^{\mathrm{SV}}(A) |\psi\rangle + |\perp\rangle) + \frac{1}{\sqrt{2}} |1\rangle (|0^m\rangle \overline{P}^{\mathrm{SV}}(A) |\psi\rangle + |\perp'\rangle) \\
\xrightarrow{\mathrm{H}\otimes I} & |0\rangle \left(|0^m\rangle \frac{P^{\mathrm{SV}}(A) + \overline{P}^{\mathrm{SV}}(A)}{2} |\psi\rangle \right) + |1\rangle \left(|0^m\rangle \frac{P^{\mathrm{SV}}(A) - \overline{P}^{\mathrm{SV}}(A)}{2} |\psi\rangle \right) + |\widetilde{\perp}\rangle \\
= & |0\rangle |0^m\rangle F^{\mathrm{SV}}(A) |\psi\rangle + |\widetilde{\perp}\rangle
\end{aligned}
\tag{13.22}
$$

Here $|\perp\rangle, |\perp'\rangle$ are two $(m+n)$-qubit state orthogonal to any state $|0^m\rangle |x\rangle$, while $|\widetilde{\perp}\rangle$ is a $(m+n+1)$-qubit state orthogonal to any state of the form $|0\rangle |0^m\rangle |x\rangle$. In other words, upon measuring the $(m+1)$ ancilla qubits and obtaining $|0^{m+1}\rangle$, the corresponding (unnormalized) state in the system register is $F^{\mathrm{SV}}(A) |\psi\rangle = (\mathrm{Re}\, P)^{\mathrm{SV}}(A) |\psi\rangle$. As a byproduct, if we obtain $|1, 0^m\rangle$ in the ancilla register, then we obtain $i(\mathrm{Im}\, P)^{\mathrm{SV}}(A) |\psi\rangle$ in the system register.

Corollary 13.2 (Quantum singular value transformation with real polynomials of definite parity)**.** *Let $A \in \mathbb{C}^{N\times N}$ be encoded by its $(1,m)$-block-encoding U_A. For a real polynomial $F(x) \in \mathbb{R}[x]$*

with degree d and parity given by $d \mod 2$ satisfying the conditions in Theorem 12.2, we can find a sequence of phase factors $\Phi \in \mathbb{R}^{d+1}$ according to Theorem 12.2, and a corresponding sequence of C-convention phase factors Φ^C according to Eq. (13.14). With this sequence Φ^C, the circuit in Fig. 13.2 implements a $(1, m+1)$-block-encoding of $F^{\mathrm{SV}}(A)$, using $U_A, U_A^\dagger$, m-qubit controlled-NOT, and single-qubit rotations a total of $\mathcal{O}(d)$ times.

Example 13.3 (Controlled implementation of the QSVT circuit)**.** When implementing a controlled QSVT circuit U_Φ, one possibility is to implement a controlled version of every gate. This means that d controlled applications of U_A or $U_A^\dagger$ are required.

Observe that (1) a controlled implementation of the controlled rotation $\mathrm{CR}(\phi^C)$ can be implemented by controlling the single-qubit rotation gate; (2) $U_A^\dagger U_A = I$. We find that when d is even, the controlled QSVT circuit can be implemented by controlled single-qubit rotations without any controlled U_A or $U_A^\dagger$. When d is odd, there is one extra U_A that cannot be cancelled, so a single controlled U_A is needed (see Fig. 13.3).

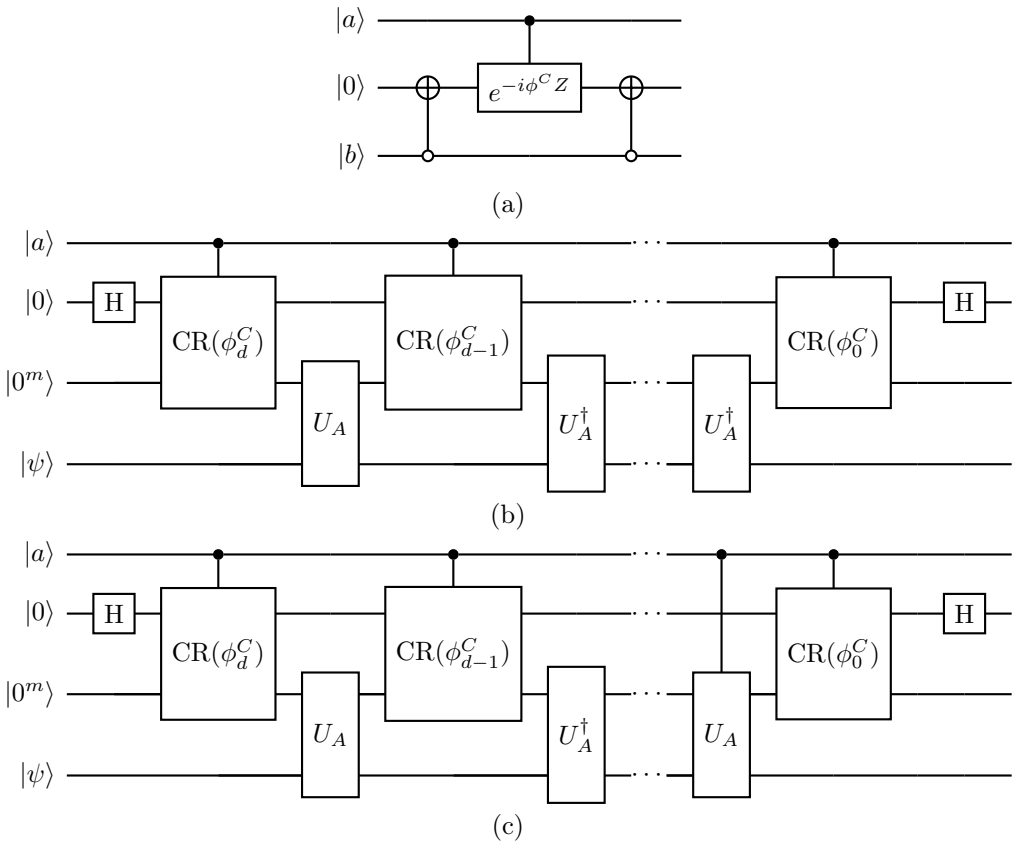


FIGURE 13.3. (a) Controlled implementation of the controlled rotation. (b) Circuit for controlled implementation of the QSVT circuit for even polynomial, which does not use controlled U_A or $U_A^\dagger$. (c) Circuit for controlled implementation of the QSVT circuit for odd polynomial, which uses the controlled U_A circuit once.

◇

For singular value transformations, we may extend the function of interest from $[0,1]$ to $[-1,1]$ as either an even or an odd function. In many applications, this choice is dictated by whether the final expression needs access to both left and right singular vectors, or only one of them. For eigenvalue transformations, there are additional degrees of freedom.

First, if the polynomial of interest $F(x) \in \mathbb{R}[x]$ does not have a definite parity, we can use the expression

$$F(x) = F_{\text{even}}(x) + F_{\text{odd}}(x), \tag{13.23}$$

where $F_{\text{even}}(x) = \frac{1}{2}(F(x) + F(-x)), F_{\text{odd}}(x) = \frac{1}{2}(F(x) - F(-x))$. If $|F(x)| \leq 1$ on $[-1,1]$, then $|F_{\text{even}}(x)|, |F_{\text{odd}}(x)| \leq 1$ on $[-1,1]$, and $F_{\text{even}}(x)$ and $F_{\text{odd}}(x)$ can each be constructed using QSVT in Corollary 13.2. Introducing another ancilla qubit and using the LCU technique, we obtain a $(2, m+2)$-block-encoding of $F(A) = F_{\text{even}}(A) + F_{\text{odd}}(A)$. Note that unlike the case of the block-encoding of $(\operatorname{Re} P)(A)$, we lose a subnormalization factor of 2 here. Using the construction in Fig. 13.3, the implementation of the select oracle in LCU only requires a single controlled implementation of U_A when implementing a controlled block-encoding for $F_{\text{odd}}(A)$.

Following the same principle, suppose the polynomial of interest $F(x) \in \mathbb{C}[x]$ satisfies $|F(x)| \leq 1$ for all $x \in [-1,1]$ (otherwise rescale F and account for the resulting subnormalization). We can write

$$F(x) = G(x) + iH(x) \tag{13.24}$$

with $G, H \in \mathbb{R}[x]$. Then $|G(x)| \leq 1$ and $|H(x)| \leq 1$ for all $x \in [-1,1]$. If G, H have definite parity, then Corollary 13.2 gives $U_{G(A)} \in \mathrm{BE}_{1,m+1}(G(A)), U_{H(A)} \in \mathrm{BE}_{1,m+1}(H(A))$. Applying LCU, we obtain $U_{F(A)} \in \mathrm{BE}_{2,m+2}(F(A))$.

If G, H do not have definite parity, then by the argument above we can construct $U_{G(A)} \in \mathrm{BE}_{2,m+2}(G(A))$ and $U_{H(A)} \in \mathrm{BE}_{2,m+2}(H(A))$. Applying another round of LCU then yields $U_{F(A)} \in \mathrm{BE}_{4,m+3}(F(A))$.

13.3. Quantum singular value transformation beyond the computational basis

So far we have assumed that a matrix $A \in \mathbb{C}^{N \times N}$ is accessed through the upper left $N \times N$ block of a unitary $U_A \in \mathrm{U}(MN)$ in the computational basis, i.e., via the projector $\Pi_{0^m} := |0^m\rangle\langle 0^m| \otimes I$ on m ancillas with $M = 2^m$. As explained in Section 9.9 and Definition 9.22, the notion of block-encoding is more general: one may replace Π_{0^m} by other projectors, and one may also encode a rectangular matrix $A \in \mathbb{C}^{N' \times N}$ between two subspaces.

Fix projectors Π, Π' on $\mathbb{C}^{MN}$ with $\operatorname{rank}(\Pi) = N$ and $\operatorname{rank}(\Pi') =: N'$ and a unitary $\mathscr{U}_A \in \mathrm{U}(MN)$ such that $\mathscr{U}_A$ is a block-encoding of A with respect to (Π, Π'). Equivalently, choose unitaries $\Xi, \Xi' \in \mathrm{U}(MN)$ defining bases $\mathcal{B}, \mathcal{B}'$ as in Eqs. (9.82) and (9.83), so that the matrix representation $U_A := (\Xi')^\dagger \mathscr{U}_A \Xi$ has the block form in Eq. (9.86).

The projectors Π, Π' can be accessed through the reflection operators

$$Z_\Pi = 2\Pi - I, \qquad Z_{\Pi'} = 2\Pi' - I. \tag{13.25}$$

With these reflections, the qubitization iterate becomes $\mathscr{U}_A^\dagger Z_{\Pi'} \mathscr{U}_A Z_\Pi$, and the QSVT construction carries over verbatim.

Let $A = W\Sigma V^\dagger$ be a singular value decomposition of A (with Σ possibly rectangular), and let the expanded singular vectors $\widetilde{W}, \widetilde{V}$ be given by the CS decomposition in Eq. (10.88). If the

polynomial degree d is even, define

$$\mathscr{U}_\Phi = e^{i\phi_0 Z_\Pi} \prod_{j=1}^{d/2} \left[(\mathscr{U}_A^\dagger Z_{\Pi'}) e^{i\phi_{2j-1} Z_{\Pi'}} (\mathscr{U}_A Z_\Pi) e^{i\phi_{2j} Z_\Pi} \right] \tag{13.26}$$

and if d is odd, define

$$\mathscr{U}_\Phi = e^{i\phi_0 Z_{\Pi'}} \mathscr{U}_A Z_\Pi \prod_{j=1}^{(d-1)/2} \left[(\mathscr{U}_A^\dagger Z_{\Pi'}) e^{i\phi_{2j-1} Z_{\Pi'}} (\mathscr{U}_A Z_\Pi) e^{i\phi_{2j} Z_\Pi} \right] \tag{13.27}$$

In a suitable orthonormal basis adapted to Π, Π' (equivalently, after the basis change described in Section 9.9), the unitary $\mathscr{U}_\Phi$ decomposes into a direct sum of 2×2 blocks $U_\Phi(\sigma_i)$ associated with the singular values σ_i of A, together with additional 1×1 blocks when $N' \neq N$.

Let us introduce the controlled rotation operator $\mathrm{CR}_\Pi(\phi)$ acting on an additional qubit and the $(n+m)$-qubit register, defined by

$$\mathrm{CR}_\Pi(\phi) := (\mathrm{C}_\Pi \mathrm{NOT})(e^{-i\phi Z} \otimes I)(\mathrm{C}_\Pi \mathrm{NOT}). \tag{13.28}$$

When the additional qubit is initialized in $|0\rangle$, the induced action on the $(n+m)$-qubit register is $e^{i\phi Z_\Pi}$. As discussed before, one may absorb the explicit reflections $Z_\Pi, Z_{\Pi'}$ into $\mathrm{CR}_\Pi, \mathrm{CR}_{\Pi'}$ by modifying the phase factors accordingly. To implement the rotation $\mathrm{CR}_\Pi(\phi)$ efficiently, we need access to

$$\begin{aligned} \mathrm{C}_\Pi \mathrm{NOT} :=& X \otimes \Pi + I_1 \otimes (I - \Pi) \\ =& X \otimes \frac{I + Z_\Pi}{2} + I_1 \otimes \frac{I - Z_\Pi}{2} \\ =& \frac{I_1 + X}{2} \otimes I + \frac{X - I_1}{2} \otimes Z_\Pi \\ =& |+\rangle\langle +| \otimes I + |-\rangle\langle -| \otimes (-Z_\Pi) \\ =& (\mathrm{H} \otimes I)(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes (-Z_\Pi))(\mathrm{H} \otimes I). \end{aligned} \tag{13.29}$$

Therefore assuming access to Z_Π, the $\mathrm{C}_\Pi \mathrm{NOT}$ gate can be implemented using the circuit in Fig. 13.4.

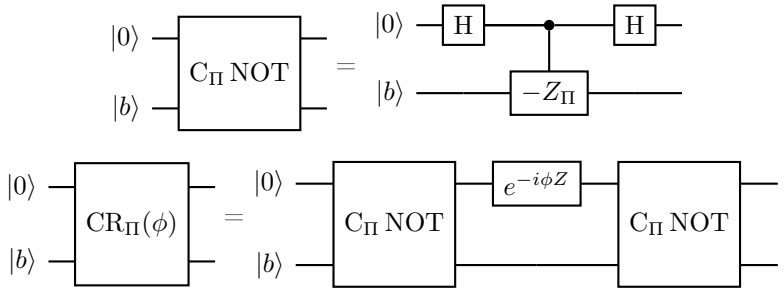


FIGURE 13.4. Circuit for implementing $\mathrm{C}_\Pi \mathrm{NOT}$ using a controlled reflection operator $-Z_\Pi$, and for $\mathrm{CR}_\Pi(\phi)$ that implements $e^{i\phi Z_\Pi}$ on the $(n+m)$-qubit register when the additional qubit is initialized in $|0\rangle$. The circuit can be further simplified using matrix identities.

This setup also covers the rectangular case $N' \neq N$. In summary, we obtain the following result.

Corollary 13.4 (Quantum singular value transformation with a basis change)**.** *Given a unitary $\mathscr{U}_A \in \mathrm{U}(MN)$ and two projectors Π, Π' on $\mathbb{C}^{MN}$ that can be accessed via reflection operators $Z_\Pi, Z_{\Pi'}$, with $\mathrm{rank}(\Pi) = N$ and $\mathrm{rank}(\Pi') =: N'$. Let $\mathcal{B}, \mathcal{B}'$ be orthonormal bases for $\mathrm{range}(\Pi)$ and $\mathrm{range}(\Pi')$, respectively. Then $[\mathscr{U}_A]^{\mathcal{B}'}_{\mathcal{B}}$ provides a $(1, m)$-block-encoding of A. Given a polynomial $F(x) \in \mathbb{R}[x]$ of degree d satisfying the conditions in Theorem 12.2, we can find a sequence of phase factors $\Phi \in \mathbb{R}^{d+1}$ according to Theorem 12.2, and a corresponding sequence of C-convention phase factors Φ^C according to Eq. (13.14). With this sequence Φ^C, we can implement a unitary $\mathscr{U}_\Phi$, so that when d is even, $[\mathscr{U}_\Phi]^{\mathcal{B}}_{\mathcal{B}}$ is a $(1, m+1)$-block-encoding of $F^{\triangleright}(A)$, and when d is odd, $[\mathscr{U}_\Phi]^{\mathcal{B}'}_{\mathcal{B}}$ is a $(1, m+1)$-block-encoding of $F^{\diamond}(A)$.*

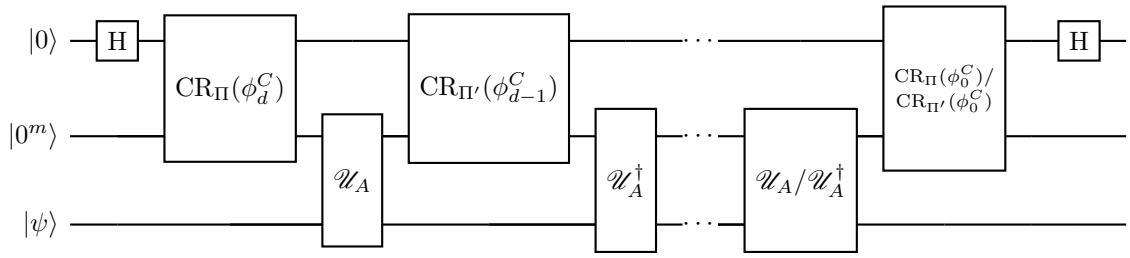


FIGURE 13.5. Circuit of quantum singular value transformation with real polynomials of definite parity and a basis change.

13.4. Example: Fixed-point amplitude amplification and uniform singular value amplification

Recall that Grover's search algorithm and amplitude amplification can overshoot the target state. This is not robust when the number of iterations is not chosen carefully. Moreover, for amplitude amplification the optimal iteration count depends on the initial overlap, which is typically unknown. Fixed-point amplitude amplification resolves these issues. Implemented via QSVT, it only requires a known lower bound on the overlap and avoids overshooting. The construction is closely related to oblivious amplitude amplification and singular value transformation.

Proposition 13.5 (**Fixed-point amplitude amplification**)**.** *Let $\mathscr{U}_A$ be an n-qubit unitary and Π' be an n-qubit orthogonal projector with $\mathrm{rank}(\Pi') \geq 1$ such that*

$$\Pi' \mathscr{U}_A |\varphi_0\rangle = a |\psi\rangle, \quad a \geq \delta > 0. \tag{13.30}$$

Then there is a $(n+1)$-qubit unitary circuit $\mathscr{U}_\Phi$ such that

$$D_p(|0\rangle |\psi\rangle, \mathscr{U}_\Phi |0\rangle |\varphi_0\rangle) \leq \epsilon, \tag{13.31}$$

which uses $\mathscr{U}_A, \mathscr{U}_A^\dagger, \mathrm{C}_{\Pi'}\mathrm{NOT}, \mathrm{C}_{|\varphi_0\rangle\langle\varphi_0|}\mathrm{NOT}$ and single-qubit rotation gates $\mathcal{O}(\log(1/\epsilon)\delta^{-1})$ times. Here $D_p(\cdot,\cdot)$ is the global phase invariant distance between two state vectors.

PROOF. Let $N = 2^n$. Construct an orthonormal basis $\mathcal{B} = \{|\varphi_0\rangle, |v_1\rangle, \ldots, |v_{N-1}\rangle\}$, where each $|v_i\rangle$ is orthogonal to $|\varphi_0\rangle$. Similarly, let $\mathcal{B}' = \{|\psi\rangle, |w_1\rangle, \ldots, |w_{N-1}\rangle\}$ be an orthonormal basis,

where each $|w_i\rangle$ is orthogonal to $|\psi\rangle$. Since $|\psi\rangle$ belongs to the range of Π',

$$\langle\psi|\mathscr{U}_A|\varphi_0\rangle = \langle\psi|\Pi'\mathscr{U}_A|\varphi_0\rangle = a, \tag{13.32}$$

i.e.,

$$[\mathscr{U}_A]_{\mathcal{B}}^{\mathcal{B}'} = \begin{pmatrix} a & * \\ * & * \end{pmatrix}. \tag{13.33}$$

Choose an odd real polynomial $F(x)$ satisfying

$$|F(x) - 1| \leq \epsilon^2/2, \quad \forall x \in [\delta, 1]. \tag{13.34}$$

In addition, to apply QSVT we require $|F(x)| \leq 1$ for all $x \in [-1, 1]$. Using Lemma 12.8, we can achieve this by approximating the sign function, and the polynomial degree is $\deg(F) = \mathcal{O}(\log(1/\epsilon)\delta^{-1})$. The corresponding QSVT circuit $\mathscr{U}_\Phi$ uses one ancilla qubit and implements a block encoding of $F(a)|\psi\rangle\langle\varphi_0|$. The overlap between $|0\rangle|\psi\rangle$ and $\mathscr{U}_\Phi|0\rangle|\varphi_0\rangle$ is $|F(a)|$. According to Eq. (3.68), the global phase invariant distance is

$$D_p(\mathscr{U}_\Phi|0\rangle|\varphi_0\rangle, |0\rangle|\psi\rangle) = \sqrt{2(1 - |F(a)|)} \leq \epsilon. \tag{13.35}$$

□

Example 13.6. Let us apply the fixed-point amplitude amplification to the unstructured search problem. Let $|x_0\rangle$ be the marked state, and let $|\psi_0\rangle$ be the uniform superposition. Let Π' be the projector onto $|x_0\rangle$. Then $\mathrm{C}_{\Pi'}\mathrm{NOT}$ and $\mathrm{C}_{|\psi_0\rangle\langle\psi_0|}\mathrm{NOT}$ are implemented by the reflection operators R_{x_0} and R_{ψ_0}, respectively. Furthermore,

$$\Pi' R_{\psi_0}|\psi_0\rangle = \frac{1}{\sqrt{N}}|x_0\rangle. \tag{13.36}$$

So we can take $\mathscr{U}_A = R_{\psi_0}$. By choosing an odd, real polynomial $F(x)$ satisfying

$$\left|F(1/\sqrt{N}) - 1\right| \leq \epsilon^2/2, \tag{13.37}$$

using $\deg(F) = \mathcal{O}(\log(1/\epsilon)\sqrt{N})$, we can find the marked state to precision ϵ. ◇

Next we discuss another application of QSVT. Recall that oblivious amplitude amplification is only applicable to block encodings of unitary matrices; a key simplification there is that the singular values are all equal to a single scalar, so one only needs to amplify that scalar. Now consider a general matrix $A \in \mathbb{C}^{N\times N}$ and a block encoding $U_A \in \mathrm{BE}_{\alpha,m}(A)$. Can we construct a new block encoding of A whose subnormalization factor is close to the optimal value $\|A\|$? This task is called uniform singular value amplification.

Proposition 13.7 (**Uniform singular value amplification**). *From a block encoding $U_A \in \mathrm{BE}_{\alpha,m}(A)$, for any $\delta \in (0, 1], \epsilon \in (0, 1/(2\alpha)]$ and $\alpha > \|A\|$, we can construct a $U_\Phi \in \mathrm{BE}_{\|A\|(1+\delta),m+1}(A, \epsilon)$, using $\mathcal{O}\left(\frac{\alpha}{\delta\|A\|}\log\left(\frac{\alpha}{\|A\|\epsilon}\right)\right)$ applications of $U_A, U_A^\dagger$.*

Proof. Let $A = \sum_i \sigma_i|u_i\rangle\langle v_i|$ be a singular value decomposition, so $\sigma_i \in [0, \|A\|]$. The block encoding condition $U_A \in \mathrm{BE}_{\alpha,m}(A)$ means that, upon projecting the ancilla register onto $|0^m\rangle$, the induced operator is A/α. In particular, QSVT applied to U_A implements odd polynomial transformations of the singular values of A/α.

Define $\alpha' := \alpha/\|A\| > 1$. Then the singular values of A/α lie in $[0, \alpha'^{-1}]$. Choose

$$\epsilon' := \min\left\{\frac{\epsilon}{\|A\|}, \frac{1}{2\alpha'}\right\}. \tag{13.38}$$

By Lemma 12.9, there exists an odd polynomial $p \in \mathbb{R}[x]$ with

$$\sup_{x\in[0,\alpha'^{-1}]} |(1+\delta)p(x) - \alpha' x| \le \epsilon', \qquad \sup_{x\in[-1,1]} |p(x)| \le 1, \tag{13.39}$$

and degree $\deg(p) = \mathcal{O}\Big(\frac{\alpha'}{\delta}\log\Big(\frac{\alpha'}{\epsilon'}\Big)\Big) = \mathcal{O}\Big(\frac{\alpha}{\delta\|A\|}\log\Big(\frac{\alpha}{\|A\|\epsilon}\Big)\Big)$.

Applying QSVT with this polynomial produces a unitary $U_\Phi \in \mathrm{BE}_{1,m+1}(p^\diamond(A/\alpha))$. Moreover,

$$\begin{aligned} \big\|\|A\|\,(1+\delta)p^\diamond(A/\alpha) - A\big\| &= \left\|\sum_i \Big(\|A\|\,(1+\delta)p\Big(\frac{\sigma_i}{\alpha}\Big) - \sigma_i\Big)\,|u_i\rangle\langle v_i|\right\| \\ &\le \|A\| \sup_{x\in[0,\alpha'^{-1}]} |(1+\delta)p(x) - \alpha' x| \le \|A\|\,\epsilon' \le \epsilon. \end{aligned} \tag{13.40}$$

Therefore $U_\Phi \in \mathrm{BE}_{\|A\|(1+\delta),m+1}(A,\epsilon)$. The number of applications of U_A and $U_A^\dagger$ is $\mathcal{O}(\deg(p))$, which gives the stated query complexity. □

13.5. Quantum Gibbs state preparation

Given a Hamiltonian $H \in \mathbb{C}^{N\times N}$ (without loss of generality we assume $H \succeq 0$), the **quantum Gibbs state** at inverse temperature $\beta = 1/T$ is defined as

$$\sigma_\beta = \frac{e^{-\beta H}}{Z_\beta}, \quad Z_\beta = \mathrm{Tr}\big[e^{-\beta H}\big]. \tag{13.41}$$

where Z_β is known as the **partition function**.

Quantum Gibbs states can be prepared using QSVT and a technique called **purification**. Consider the purified Gibbs state

$$|\sigma_\beta\rangle = \sqrt{\frac{N}{Z_\beta}}\big(I \otimes e^{-\beta H/2}\big)\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle\,|j\rangle\right), \tag{13.42}$$

which satisfies $\langle\sigma_\beta|\sigma_\beta\rangle = 1$, since $H \succeq 0$ implies $Z_\beta = \mathrm{Tr}[e^{-\beta H}] \le \mathrm{Tr}[I] = N$. The Gibbs state σ_β is then obtained by tracing out the first (ancillary) register:

$$\mathrm{Tr}_A[|\sigma_\beta\rangle\langle\sigma_\beta|] = \frac{1}{Z_\beta}e^{-\beta H/2}\left(\sum_{j=0}^{N-1}|j\rangle\langle j|\right)e^{-\beta H/2} = \frac{e^{-\beta H}}{Z_\beta} = \sigma_\beta. \tag{13.43}$$

Thus, it suffices to construct a block-encoding of $e^{-\beta H/2}$ and apply it to the maximally entangled state $\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle\,|j\rangle$.

Since $f(x) = e^{-\beta x/2}$ is neither even nor odd, one way to use QSVT is to approximate its even and odd parts separately. This is problematic if one insists on a uniform approximation on a symmetric interval, since $f(-x) = e^{\beta x/2}$ grows exponentially with β. One may instead try to approximate f by an even function, but the symmetrized function $g(x) = e^{-\beta|x|/2}$ has a cusp at $x = 0$, and consequently the approximation error decays only polynomially with the degree.

We therefore assume a different access model, namely $V_H \in \mathrm{BE}_{1,m}(I - H/\alpha_H)$, where α_H is chosen so that $0 \preceq H \preceq \alpha_H I$. The spectrum of $I - H/\alpha_H$ is contained in $[0,1]$. Using the identity

$$e^{-\beta H/2} = e^{-\frac{\beta\alpha_H}{2}(I-(I-H/\alpha_H))}, \tag{13.44}$$

we can construct a block-encoding of $e^{-\beta H/2}$ using V_H. For polynomial approximations to $e^{-\gamma(1-x)}$ on $[-1,1]$, we have the following result from [GSLW18, Corollary 64].

Proposition 13.8. *Let $\gamma > 0$ and $\epsilon \in \left(0, \frac{1}{2}\right]$. Then there exists a real polynomial P with degree $\mathcal{O}\left(\sqrt{\max\left[\gamma, \log\left(\frac{1}{\epsilon}\right)\right] \log\left(\frac{1}{\epsilon}\right)}\right)$ such that*

$$\left\| e^{-\gamma(1-x)} - P(x) \right\|_{[-1,1]} \leq \epsilon. \tag{13.45}$$

The polynomial in Proposition 13.8 does not have definite parity. Therefore we implement its even and odd parts using QSVT, and combine them to obtain a block-encoding of $e^{-\beta H/2}$ using $\mathcal{O}\left(\sqrt{\beta\alpha_H}\log(1/\epsilon)\right)$ queries to V_H. Since

$$\left\| \left(I \otimes e^{-\beta H/2}\right) \left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle |j\rangle \right) \right\| = \sqrt{\frac{Z_\beta}{N}}, \tag{13.46}$$

amplitude amplification yields a total query complexity

$$\mathcal{O}\left(\sqrt{\frac{N}{Z_\beta}} \sqrt{\beta\alpha_H} \log^2(1/\epsilon) \right) \tag{13.47}$$

for preparing $|\sigma_\beta\rangle$.

Remark 13.9. The $\mathcal{O}(\sqrt{\beta})$ scaling (here $\gamma = \beta\alpha_H/2$) may seem surprising, given that $\sup_{x\in[0,1]} \left| \frac{\mathrm{d}}{\mathrm{d}x} e^{-\gamma(1-x)} \right| = \gamma$. This is because, after the transformation, the largest derivative occurs at the boundary $x = 1$, while the Chebyshev polynomial $T_k(x) = \cos(k \arccos(x))$ varies rapidly near $x = 1$. Specifically,

$$T_k'(1) = \lim_{\theta\to 0} \frac{k\sin(k\theta)}{\sin(\theta)} = k^2. \tag{13.48}$$

Thus, a polynomial of degree $\mathcal{O}(\sqrt{\beta})$ is sufficient to resolve the large derivative at the boundary.

It is worth noting that this $\mathcal{O}(\sqrt{\beta})$ dependence relies on having access to $V_H \in \mathrm{BE}_{1,m}(I - H/\alpha_H)$. If we only have access to $V_H \in \mathrm{BE}_{\eta,m}(I - H/\alpha_H)$ for some constant $\eta > 1$, then rewriting

$$e^{-\beta H/2} = e^{-\frac{\beta\alpha_H\eta}{2}(I/\eta - (I - H/\alpha_H)/\eta)}. \tag{13.49}$$

shows that one is naturally led to the function $x \mapsto e^{-\gamma(\eta^{-1}-x)}$ (with $\gamma = \beta\alpha_H\eta/2$). This function exceeds 1 for $x > \eta^{-1}$, and therefore no polynomial bounded on $[-1, 1]$ can approximate it uniformly on $[-1, 1]$ to small additive error. In particular, Proposition 13.8 does not apply in this form. ⋄

13.6. Quantum eigenvalue transformation with Hamiltonian evolution oracles

Given the Hamiltonian evolution oracle $U = e^{-iH}$ with $0 \preceq H \preceq \pi$ for simplicity. Can we construct a block encoding of a matrix function $f(H)$ using QSP? One possibility is to first construct a block encoding of H by implementing the matrix logarithm $H = i \log U$ using QSVT, followed by another layer of QSVT to implement $f(H)$. Here we show that this process can be made much simpler using a single layer of QSVT-like circuit with one ancilla qubit. This is called the **quantum eigenvalue transformation of unitary matrices** with real polynomials (QETU).

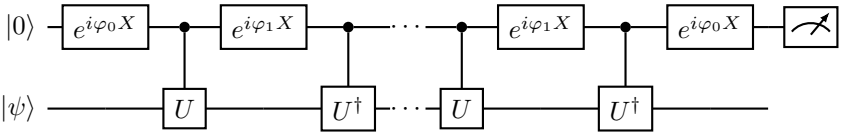


FIGURE 13.6. Circuit of quantum eigenvalue transformation of unitary matrices.

THEOREM 13.10 (Quantum eigenvalue transformation of unitary matrices). *Let $U = e^{-iH}$ with an n-qubit Hermitian matrix H. For any even real polynomial $F(x)$ of degree $2d$ satisfying $|F(x)| \leq 1, \forall x \in [-1,1]$, we can find a sequence of symmetric phase factors $\Phi := (\varphi_0, \varphi_1, \cdots, \varphi_1, \varphi_0) \in \mathbb{R}^{2d+1}$, such that the circuit in Fig. 13.6 denoted by $\mathcal{U}$ satisfies $(\langle 0| \otimes I_n)\mathcal{U}(|0\rangle \otimes I_n) = F\left(\cos \frac{H}{2}\right)$.*

Let the matrix function of interest be expressed as $f(H) = (f \circ g)(\cos \frac{H}{2})$, where $g(x) = 2\arccos(x)$. Therefore we can find an even polynomial approximation $F(x)$ so that

$$\sup_{x \in [\sigma_{\min}, \sigma_{\max}]} |(f \circ g)(x) - F(x)| \leq \epsilon. \tag{13.50}$$

Here $\sigma_{\min} = \cos \frac{\lambda_{\max}}{2}, \sigma_{\max} = \cos \frac{\lambda_{\min}}{2}$, respectively (note that $\cos(x/2)$ is a monotonically *decreasing* function on $[0, \pi]$). This ensures that the operator norm error satisfies

$$\|(\langle 0| \otimes I_n)\mathcal{U}(|0\rangle \otimes I_n) - f(H)\| \leq \epsilon. \tag{13.51}$$

Compared to the QSVT circuit for block encoding $f(H)$ with a real polynomial of definite parity, we find that instead of the Z rotation $e^{i\varphi Z}$, the circuit uses now the X rotation $e^{i\varphi X}$. For the proof of Theorem 13.10, we refer readers to [DLT22].

Exercise 13.1. Given access to a unitary $U = e^{-iH}$ where $\|H\| \leq \pi/2$. Use QSVT to design a quantum algorithm to approximately implement a block encoding of H, using controlled U and its inverses, as well as elementary quantum gates.

13.7. Perturbation theory of singular value transformations

So far we have assumed that $U \in \mathrm{BE}_{\alpha,m}(A)$ is an exact block encoding of A. What if we can only implement $\widetilde{U} \in \mathrm{BE}_{\alpha,m}(\widetilde{A})$ so that $\left\|\widetilde{A} - A\right\| \leq \epsilon$? Note that we cannot directly invoke the linear error growth property in Proposition 3.21, since we do not have access to the exact block encoding matrix U, and therefore cannot compute $\left\|U - \widetilde{U}\right\|$. As a result, it is desirable to develop a perturbation theory that can be used to directly quantify the error $\left\|f^{\mathrm{SV}}(\widetilde{A}) - f^{\mathrm{SV}}(A)\right\|$. We start by illustrating this is possible for the task of Hamiltonian simulation and oblivious amplitude amplification.

Example 13.11 (Perturbation analysis for Hamiltonian simulation)**.** Consider a block encoding $U_{\widetilde{H}} \in \mathrm{BE}_{1,m}(H, \epsilon)$. Then the Duhamel principle (Proposition 3.22) gives

$$\left\|e^{i\widetilde{H}} - e^{iH}\right\| = \left\|i \int_0^1 e^{iH(1-s)}(\widetilde{H} - H)e^{i\widetilde{H}s}\, \mathrm{d}s\right\| \leq \left\|\widetilde{H} - H\right\| \leq \epsilon. \tag{13.52}$$

This gives

$$\left\|\cos(\widetilde{H}) - \cos(H)\right\| = \left\|\frac{e^{i\widetilde{H}} + e^{-i\widetilde{H}}}{2} - \frac{e^{iH} + e^{-iH}}{2}\right\| \leq \frac{1}{2}\left\|e^{i\widetilde{H}} - e^{iH}\right\| + \frac{1}{2}\left\|e^{-i\widetilde{H}} - e^{-iH}\right\| \leq \epsilon. \tag{13.53}$$

Similarly

$$\left\|\sin(\widetilde{H}) - \sin(H)\right\| \leq \epsilon. \tag{13.54}$$

These bounds are independent of the polynomial degree used in constructing the approximation these functions. ◇

Example 13.12 (Refined perturbation analysis for oblivious amplitude amplification)**.** Let A be an approximate implementation of a unitary matrix U such that $\|A - U\| \leq \epsilon$. According to the oblivious amplitude amplification (see Example 11.11), if we choose

$$\gamma_k^{-1} = \sin\frac{\pi}{2(2k+1)}, \quad k \in \mathbb{N}_+, \tag{13.55}$$

then $T^{\diamond}_{2k+1}(U/\gamma_k) = (-1)^k U$. This means that if we have access to a block encoding $\mathcal{V} \in \mathrm{BE}_{\gamma_k,a}(A) = \mathrm{BE}_{\gamma_k,a}(U,\epsilon)$, then $T^{\diamond}_{2k+1}(A/\gamma_k)$ is an approximate implementation of $(-1)^k U$ using $k+1$ queries to $\mathcal{V}$ and k queries to $\mathcal{V}^\dagger$. We now bound the error $\left\|(-1)^k T^{\diamond}_{2k+1}(A/\gamma_k) - U\right\|$.

Start from the singular value decomposition $A = W\Sigma V^\dagger$, the perturbation theorem of singular values (Theorem 7.10) states that $\|\Sigma - I\| \leq \epsilon$. Then $T^{\diamond}_{2k+1}(A/\gamma_k) = W T_{2k+1}(\Sigma/\gamma_k) V^\dagger$ is an approximate implementation of $(-1)^k W V^\dagger$. Note that the Chebyshev polynomial at γ_k^{-1} satisfies

$$T_{2k+1}(\gamma_k^{-1}) = (-1)^k, \quad T'_{2k+1}(\gamma_k^{-1}) = 0, \quad T''_{2k+1}(\gamma_k^{-1}) = (-1)^{k+1}\frac{(2k+1)^2}{1-\gamma_k^{-2}}. \tag{13.56}$$

Then by Taylor's theorem and the continuity of T''_{2k+1}, for each k there exists some $\epsilon_k > 0$ such that for any $|x-1| \leq \epsilon \leq \epsilon_k$,

(13.57)

$$\begin{aligned}\left|(-1)^k T_{2k+1}(x/\gamma_k) - 1\right| =& \left|T_{2k+1}(x/\gamma_k) - T_{2k+1}(\gamma_k^{-1})\right| \leq 2\cdot\frac{1}{2}\cdot\left|T''_{2k+1}(\gamma_k^{-1})\right|\frac{\epsilon^2}{\gamma_k^2} = \frac{(2k+1)^2}{1-\gamma_k^{-2}}\cdot\frac{\epsilon^2}{\gamma_k^2} \\ =& \frac{(2k+1)^2\epsilon^2}{\gamma_k^2 - 1} = \left(\epsilon(2k+1)\tan\frac{\pi}{2(2k+1)}\right)^2 < \pi^2\epsilon^2.\end{aligned}$$

In the last inequality, we have used the fact that $a^{-1}\tan a < 2$ for any $a = \frac{\pi}{2(2k+1)} \in [0, \pi/6]$ to simplify the expression. This implies

$$\left\|(-1)^k T^{\diamond}_{2k+1}(A/\gamma_k) - WV^\dagger\right\| \leq \pi^2\epsilon^2. \tag{13.58}$$

Finally, if $\pi^2\epsilon < 1$, use the triangle inequality,

(13.59)

$$\left\|(-1)^k T^{\diamond}_{2k+1}(A/\gamma_k) - U\right\| \leq \left\|(-1)^k T^{\diamond}_{2k+1}(A/\gamma_k) - WV^\dagger\right\| + \left\|WV^\dagger - W\Sigma V^\dagger\right\| + \left\|W\Sigma V^\dagger - U\right\| = 3\epsilon.$$

This bound is independent of the degree of the polynomial degree used! Fig. 13.7 confirms the validity of this error bound. This bound agrees with the refined analysis of oblivious amplitude amplification in [GSLW19, Theorem 15].

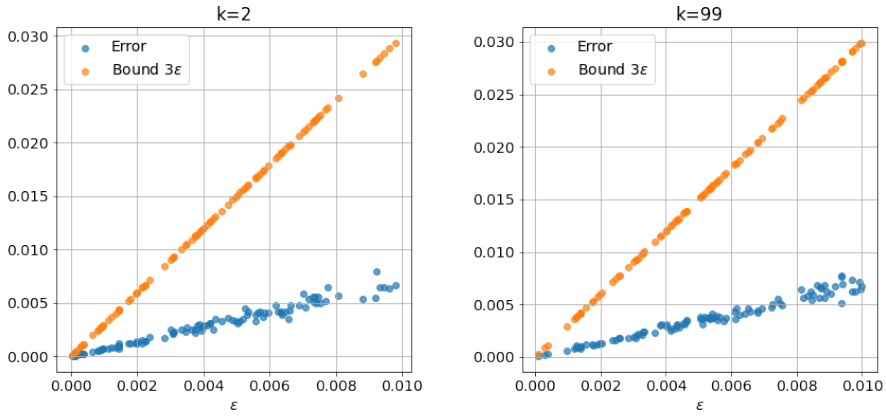


FIGURE 13.7. Error of oblivious amplitude amplification $\left\|(-1)^k T^{\diamond}_{2k+1}(A/\gamma_k) - U\right\|$ versus the error bound for 100 random matrices with $\|A-U\| = \epsilon$. Neither the computed error nor the error bound depends on k.

◇

Definition 13.13. *Given a function $\omega : [0,\infty) \to [0,\infty)$ and an interval $I \subset \mathbb{R}$, a function $f : I \to \mathbb{C}$ admits ω as a modulus of continuity if*

$$|f(x) - f(y)| \leq \omega(|x-y|), \quad \forall x, y \in I. \tag{13.60}$$

Results in approximation theory [FN09] can be used to characterize the robustness of eigenvalue transformation of Hermitian matrices.

THEOREM 13.14. *Let $f : [-1,1] \to \mathbb{C}$ be a function that admits a modulus of continuity $\omega : [0,2] \to [0,\infty)$. Then for all Hermitian matrices $A, \widetilde{A} \in \mathbb{C}^{N\times N}$ such that $\|A\|, \left\|\widetilde{A}\right\| \leq 1$, we have*

$$\left\|f(A) - f(\widetilde{A})\right\| \leq 4\left[\ln\left(\frac{2}{\|A-\widetilde{A}\|}+1\right)+1\right]^2 \omega(\|A-\widetilde{A}\|). \tag{13.61}$$

Using the close connection between the singular value transformation and the eigenvalue transformation of the dilated matrices discussed in Section 10.1, we can use Theorem 13.14 to derive the following perturbation result for the singular value transformation. We refer readers to [GSLW18, Section 3.3] for its proof and further applications and refinements of the result.

Corollary 13.15. *Let $f : [-1,1] \to \mathbb{C}$ be a function of definite parity that admits a modulus of continuity $\omega : [0,2] \to [0,\infty)$. Then for all matrices $A, \widetilde{A} \in \mathbb{C}^{\tilde{N}\times N}$ such that $\|A\|, \left\|\widetilde{A}\right\| \leq 1$, we have*

$$\left\|f^{\mathrm{SV}}(A) - f^{\mathrm{SV}}(\widetilde{A})\right\| \leq 4\left[\ln\left(\frac{2}{\|A-\widetilde{A}\|}+1\right)+1\right]^2 \omega(\|A-\widetilde{A}\|). \tag{13.62}$$

Notes and further reading

Many applications of QSVT can be found in the seminal paper [GSLW19]. When the input is Hermitian, the connection between singular value and eigenvalue transformation (cf. Section 10.1) implies that the same circuits implement eigenvalue transformations of A; this special case is sometimes called quantum eigenvalue transformation (QET). Recent usage often reserves "eigenvalue processing/transformation" for settings beyond Hermitian inputs, such as eigenvalue transformations associated with nonunitary or more general matrix dynamics; see, e.g., [ALL23, LS24]. A generalization of the QETU algorithm is given by generalized quantum signal processing [MW24], which can also be interpreted using the nonlinear Fourier transform [AMT24].

The $\mathcal{O}(\sqrt{\beta})$ dependence discussed in Section 13.5 can also be achieved under alternative access models for H, such as access to $\sqrt{H}$ [CS17, ACNR22]. The Gibbs sampler based on linear combination of Hamiltonian simulation [ALL23, ACL26] (see also **??**) applies to positive semidefinite Hamiltonians and requires only the ability to simulate the dynamics; its complexity is $\widetilde{\mathcal{O}}\left(\sqrt{\frac{N}{Z_\beta}}\beta\alpha_H \left(\log\left(\frac{1}{\epsilon}\right)\right)^{1/\gamma}\right)$ for any $\gamma \in (0,1)$. The factor $\sqrt{\frac{N}{Z_\beta}}$ becomes large at low temperature. This also means that efficient Gibbs preparation without additional structure is typically confined to sufficiently high-temperature regimes.

CHAPTER 14

Block encoding based Hamiltonian simulation

Simulation of one quantum Hamiltonian by another quantum system was also one of the motivations of Feynman's proposal for design of quantum computers [Fey82]. Hamiltonian simulation is the process of implementing the unitary operator $U(t) = e^{-itH}$, where H is the Hamiltonian of the system and t is the evolution time.

More generally, a quantum computer can be thought of as a quantum mechanical system wherein the system Hamiltonian is changed over time to implement the various gates that we wish to apply to our system. These control fields are actually not best modeled as a single fixed Hamiltonian, and instead is modeled by a time-dependent Hamiltonian. Such time-dependent Hamiltonians appear in a number of places. They are essential to understand the operations of quantum computers at a physical level and also are fundamental to approaches to understand adiabatic state preparation [WKAG09] and related adiabatic algorithms [FG98].

Our aim is to show how such quantum dynamics can be emulated on a quantum computer. We begin with simulation techniques for time-independent Hamiltonians. We then explain the time-ordering formalism for time-dependent Hamiltonians and then discuss how truncated Dyson series methods can be used to simulate the underlying dynamics. Finally we will use these results to present the interaction picture simulation method, which takes a time-independent Hamiltonian and transforms it into a related time-dependent Hamiltonian with the aim of reducing the computational complexity of quantum simulation.

14.1. Quantum signal processing and time-independent Hamiltonian simulation with optimal query complexity

Given a block encoding of a Hamiltonian H, the Hamiltonian simulation problem aims at constructing a block encoding of the unitary $U = e^{-iHt}$.

Since $e^{-iHt} = e^{-i(H/\alpha)(\alpha t)}$, the normalization factor α can be absorbed into the simulation time t, and we may assume $\alpha = 1$. If we start from a block encoding $U_{\widetilde{H}} \in \mathrm{BE}_{1,m}(\widetilde{H})$ with $\left\|\widetilde{H} - H\right\| \leq \epsilon'$, then the perturbation bound follows from Duhamel's formula (a.k.a. variation of constants):

$$\left\|e^{-i\widetilde{H}t} - e^{-iHt}\right\| = \left\|-i\int_0^t e^{-iH(t-s)}(\widetilde{H} - H)e^{-i\widetilde{H}s}\,\mathrm{d}s\right\| \leq \epsilon' t. \tag{14.1}$$

In order to approximate e^{-iHt} to precision ϵ, it is sufficient to choose $\epsilon' = \epsilon/(2t)$ and then approximate $e^{-i\widetilde{H}t}$ to precision $\epsilon/2$. Hence, without loss of generality, we assume the block encoding $U_H \in \mathrm{BE}_{1,m}(H)$ is error-free.

We use QSP/QSVT to construct a block encoding of $e^{-iHt} = \cos(Ht) - i\sin(Ht)$. Note that $\cos(tx)$ is real and even, whereas $\sin(tx)$ is real and odd. Accordingly, we construct block encodings for $\cos(Ht)$ and $\sin(Ht)$ separately using QSVT, and then combine them using LCU.

To implement this, we use the Fourier–Chebyshev series of the trigonometric functions on $[-1,1]$ (the Jacobi-Anger expansion):

$$\begin{aligned}\cos(tx) =& J_0(t) + 2\sum_{k=1}^{\infty}(-1)^k J_{2k}(t)T_{2k}(x),\\ \sin(tx) =& 2\sum_{k=0}^{\infty}(-1)^k J_{2k+1}(t)T_{2k+1}(x).\end{aligned} \tag{14.2}$$

Here $J_\nu(t)$ denotes Bessel functions of the first kind. This series converges very rapidly. We first prove a useful bound for the Lambert-W function.

Lemma 14.1. *Let $a > 0$ and $\epsilon \in (0, e^{-1})$, and set $L := \ln(1/\epsilon)$. Define*

$$K_*(a,\epsilon) := \frac{L}{W(L/a)}, \tag{14.3}$$

where W is the (principal branch of the) Lambert-W function. Then $K_*(a,\epsilon)$ is the unique positive solution of

$$\left(\frac{a}{K}\right)^K = \epsilon. \tag{14.4}$$

Moreover, $K_*(a,\epsilon)$ satisfies

$$K_*(a,\epsilon) = \mathcal{O}\left(a + \ln(1/\epsilon)\right). \tag{14.5}$$

There is also a more refined bound

$$K_*(a,\epsilon) = \mathcal{O}\left(a + \frac{\ln(1/\epsilon)}{\ln\left(e + \ln(1/\epsilon)/a\right)}\right). \tag{14.6}$$

Proof. Consider the function $f(K) := K\ln(K/a)$ for $K > a$. Since $f'(K) = \ln(K/a) + 1 > 0$, f is strictly increasing on (a,∞). The equation $(a/K)^K = \epsilon$ is equivalent to

$$f(K) = \ln(1/\epsilon) = L. \tag{14.7}$$

Writing $K = ay$ with $y > 1$ gives $ay\ln y = L$, or equivalently $y\ln y = L/a$. Setting $u := \ln y$ yields $ue^u = L/a$. Its solution defines the Lambert-W function as $u = W(L/a)$. Therefore

$$K = ay = ae^{W(L/a)} = \frac{L}{W(L/a)} = K_*(a,\epsilon). \tag{14.8}$$

The monotonicity of f implies that $K \geq K_*(a,\epsilon)$ is sufficient for $(a/K)^K \leq \epsilon$.

We first prove the crude bound Eq. (14.5). Take $K := a + L$. Using the inequality $\ln(1+x) \geq x/(1+x)$ for $x \geq 0$, we obtain

$$f(K) = K\ln\left(\frac{K}{a}\right) = (a+L)\ln\left(1+\frac{L}{a}\right) \geq (a+L)\cdot\frac{L/a}{1+L/a} = L. \tag{14.9}$$

Therefore $(a/K)^K = e^{-f(K)} \leq e^{-L} = \epsilon$, which implies $K_*(a,\epsilon) \leq a + L$.

To obtain the refined bound Eq. (14.6), we set $z := L/a$. If $z \leq 1$ (equivalently, $L \leq a$), then taking $K = 2a$ gives

$$\left(\frac{a}{K}\right)^K = \left(\frac{1}{2}\right)^{2a} = e^{-2a\ln 2} \leq e^{-a} \leq e^{-L} = \epsilon, \tag{14.10}$$

using $2\ln 2 > 1$. Hence $K = \mathcal{O}(a)$ suffices in this regime.

If $z \geq 1$, let $u := \frac{1}{2}\ln z \geq 0$. Since $e^u \geq u$ for $u \geq 0$, we have $\ln z = 2u \leq 2e^u = 2\sqrt{z}$, and therefore

$$ue^u = \tfrac{1}{2}(\ln z)\sqrt{z} \leq z. \tag{14.11}$$

By monotonicity of $w \mapsto we^w$ on $[0,\infty)$ and the defining relation $W(z)e^{W(z)} = z$, this implies $W(z) \geq u = (1/2)\ln z$. Consequently,

$$K_*(a,\epsilon) = \frac{L}{W(z)} \leq \frac{2L}{\ln z} = \mathcal{O}\left(\frac{\ln(1/\epsilon)}{\ln\left(e + \ln(1/\epsilon)/a\right)}\right). \tag{14.12}$$

Combining the two cases yields Eq. (14.6). □

Lemma 14.2 (Jacobi-Anger expansion)**.** *For any $t > 0$, $\epsilon \in (0, e^{-1})$, we can choose*

$$K = \Theta\left(t + \frac{\ln(1/\epsilon)}{\ln(e + \ln(1/\epsilon)/t)}\right) \tag{14.13}$$

such that

$$\sup_{x\in[-1,1]} \left|\cos(tx) - \left(J_0(t) + 2\sum_{k=1}^{K}(-1)^k J_{2k}(t)T_{2k}(x)\right)\right| \leq \epsilon, \tag{14.14}$$

and

$$\sup_{x\in[-1,1]} \left|\sin(tx) - 2\sum_{k=0}^{K}(-1)^k J_{2k+1}(t)T_{2k+1}(x)\right| \leq \epsilon. \tag{14.15}$$

Proof sketch. The Bessel function satisfies the tail bound

$$|J_m(t)| \leq \frac{1}{m!}\left|\frac{t}{2}\right|^m. \tag{14.16}$$

This means that the tail contribution is upper bounded by

$$2\sum_{k=2K+1}^{\infty}|J_k(t)| \leq 2\sum_{k=2K+1}^{\infty}\frac{1}{k!}\left|\frac{t}{2}\right|^k = \mathcal{O}\left(\frac{1}{(2K+1)!}\left(\frac{t}{2}\right)^{2K+1}e^{t/2}\right) = \mathcal{O}\left(\left(\frac{et}{2K+1}\right)^{2K+1}e^{t/2}\right). \tag{14.17}$$

In particular, one may view the choice of K as governed (up to constants) by a model inequality of the form $(a/K)^K \leq \epsilon$, whose solution involves the Lambert-W function (see Lemma 14.1). Applying the refined estimate Eq. (14.6) with $a = \Theta(t)$ yields Eq. (14.13). □

We now present the following algorithm for simulating time-independent Hamiltonians [LC17b], which matches the complexity lower bound for Hamiltonian simulation [BACS07] and thus achieves the optimal query complexity.

Theorem 14.3 (Time-independent Hamiltonian simulation with optimal query complexity). *From a block encoding $U_H \in \mathrm{BE}_{1,m}(H)$, for any $\epsilon \in (0,1), t > 0$, we can construct a unitary $U_\Phi \in \mathrm{BE}_{2(1+\epsilon),m+2}(e^{-iHt},\epsilon)$. It uses $\mathcal{O}\left(t + \frac{\ln(1/\epsilon)}{\ln(e+\ln(1/\epsilon)/t)}\right)$ applications of $U_H, U_H^\dagger$, and one application of controlled U_H.*

PROOF. Using Eq. (14.13), we choose K so that $\cos(tx)$ and $\sin(tx)$ are each approximated on $[-1,1]$ to precision $\epsilon/2$. Denote the resulting truncations by $C(x)$ and $S(x)$. Choose $\beta = 1+\epsilon$ so that $\left|\beta^{-1}C(x)\right| \leq 1$ and $\left|\beta^{-1}S(x)\right| \leq 1$ for all $x \in [-1,1]$. Moreover,

$$\max_{x\in[-1,1]} \left|(C(x) - iS(x)) - e^{-itx}\right| \leq \sqrt{\left(\frac{\epsilon}{2}\right)^2 + \left(\frac{\epsilon}{2}\right)^2} = \epsilon/\sqrt{2} < \epsilon. \tag{14.18}$$

Corollary 12.6 guarantees the existence of symmetric phase factors Φ_C, Φ_S that represent the polynomials $\beta^{-1}C(x)$ and $\beta^{-1}S(x)$, respectively. Using these phase factors, we obtain block encodings $U_C \in \mathrm{BE}_{\beta,m+1}(\cos(Ht), \epsilon/2)$ and $U_S \in \mathrm{BE}_{\beta,m+1}(\sin(Ht), \epsilon/2)$. Finally, we use one more ancilla qubit and LCU to combine U_C and U_S into the desired block encoding U_Φ. In the corresponding select oracle (see Fig. 13.3), $C(H)$ does not require any controlled-U_H, whereas $S(H)$ uses a controlled-U_H circuit once. □

Example 14.4. Fig. 14.1 shows the QSP representation error for approximating $\cos(tx)$ using $P_{\mathrm{Re}}(x) = C_d(x)$ with $t = 4\pi, \beta = 1.001, d = 24$. The approximation improves significantly with a larger degree $d = 50$ (see Fig. 14.2). The phase factors are obtained via QSPPACK. The results for the sine function are similar. ◇

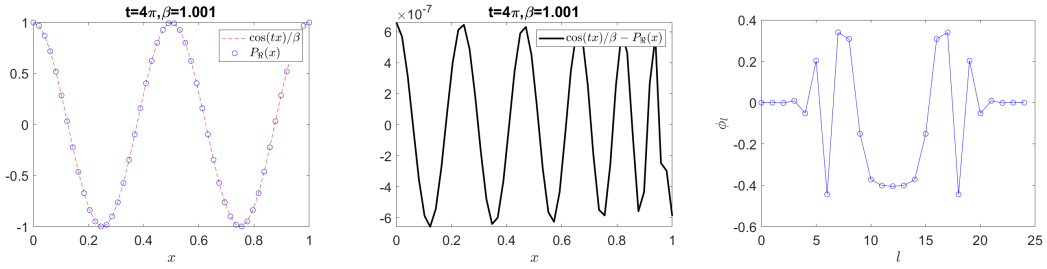


FIGURE 14.1. QSP representation of $\cos(tx)/\beta$ with $t = 4\pi, \beta = 1.001, d = 24$. The phase factors plotted remove a factor of $\pi/4$ on both ends (see **??**).

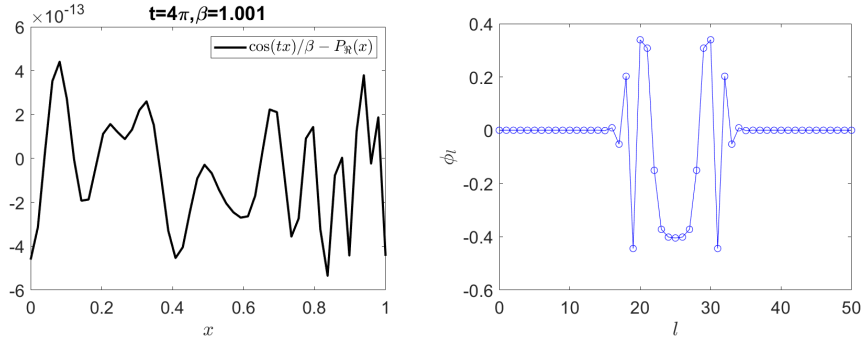


FIGURE 14.2. Error of the QSP representation of $\cos(tx)/\beta$ with $t = 4\pi, \beta = 1.001, d = 50$. The phase factors plotted remove a factor of $\pi/4$ on both ends (see **??**).

Example 14.5 (Oblivious amplitude amplification for Hamiltonian simulation)**.** In the Hamiltonian simulation problem, we use LCU to obtain $U_\Phi \in \mathrm{BE}_{2\beta,m+2}(e^{-iHt},\epsilon)$, where $\beta = 1+\epsilon$. Since

$$\left\|e^{-iHt} - 2\left\langle 0^{m+2}|U_\Phi|0^{m+2}\right\rangle\right\| \le \left\|e^{-iHt} - 2\beta\left\langle 0^{m+2}|U_\Phi|0^{m+2}\right\rangle\right\| + 2\epsilon\left\|U_\Phi\right\| \le 3\epsilon, \tag{14.19}$$

we also have $U_\Phi \in \mathrm{BE}_{2,m+2}(e^{-iHt},3\epsilon)$. According to the perturbation analysis of oblivious amplitude amplification in Example 13.12, we can use 2 uses of U_Φ and 1 use of $U_\Phi^\dagger$ to construct a block encoding $\mathfrak{U} \in \mathrm{BE}_{1,m+2}(e^{-iHt},9\epsilon)$. It uses 3 applications of controlled U_H.

If we would like to use the oblivious amplitude amplified Hamiltonian simulation as a subroutine and to simulate for an even longer period $T = Lt$ for some integer L, we can simply use $\mathfrak{U}^L$ and the error is bounded by $9\epsilon L$ using the linear error growth property. ⋄

14.2. Truncated Taylor series method

We now introduce an alternative approach for simulating time-independent Hamiltonians: the **truncated Taylor series method**, which predates the QSP-based methods. While it does not achieve optimal query complexity, it offers a very different perspective on Hamiltonian simulation. Moreover, it serves as a natural precursor to more advanced algorithms for time-dependent Hamiltonians, such as the truncated Dyson series method.

Assume $\|H\| \le 1$ for simplicity. For a short time Δt, the truncated Taylor method approximates the short-time evolution operator by truncating the Taylor series after order $K-1$:

$$e^{-iH\Delta t} \approx \sum_{k=0}^{K-1} \frac{(-iH\Delta t)^k}{k!}. \tag{14.20}$$

The choice of K will be specified later.

We first construct a sequence of unitary operations U_k that correspond to the terms $(-iH)^k$ using products of block encodings. Then we implement a select oracle

$$U_{\mathrm{SELECT}} = \sum_{k=0}^{K-1} |k\rangle\langle k| \otimes U_k. \tag{14.21}$$

We also assume access to a prepare oracle (and assume $\log_2 K$ is an integer for simplicity)

$$V_{\mathrm{PREP}}\left|0^{\log_2 K}\right\rangle = \frac{1}{\|c\|_1}\sum_{k=0}^{K-1}\sqrt{c_k}\,|k\rangle\,, \quad c_k = \frac{(\Delta t)^k}{k!}. \tag{14.22}$$

We choose $\Delta t = \ln 2$ so that the full Taylor series satisfies $\sum_{k=0}^{\infty}(\Delta t)^k/k! = e^{\Delta t} = 2$. For finite truncation order K, one has $\|c\|_1 \le 2$ and $|\|c\|_1 - 2|$ is controlled by the Taylor remainder. Then LCU implements a block encoding of $e^{-iH\Delta t}$ with normalization factor $\|c\|_1 \le 2$. After oblivious amplitude amplification, this yields a block encoding

$$W \in \mathrm{BE}_{1,a}(e^{-iH\Delta t},\epsilon'), \tag{14.23}$$

where $a = \mathcal{O}(\log_2 K + m)$, if U_{SELECT} is implemented using the compression gadget in Example 11.10.

For simulations over longer durations $t = L\Delta t$, we concatenate the segments:

$$e^{-iHt} \approx \left(\sum_{k=0}^{K-1}\frac{(-iH\Delta t)^k}{k!}\right)^L. \tag{14.24}$$

Each term in the Taylor series can be implemented using a block encoding, and their linear combination forms a block encoding of the approximated evolution operator. Using oblivious amplitude amplification and the fact that the target $e^{-iH\Delta t}$ is unitary, we have

$$\left\|(\langle 0^a| W |0^a\rangle)^L - e^{-iHL\Delta t}\right\| \leq L\epsilon'. \tag{14.25}$$

To approximate e^{-iHt} to precision ϵ, it is sufficient to choose $\epsilon' = \epsilon/L = \epsilon\Delta t/t$. The success probability satisfies $p \geq (1 - L\epsilon')^2$.

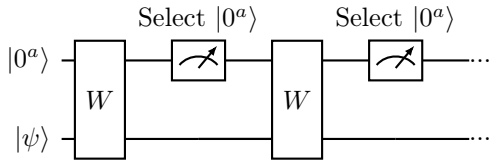


FIGURE 14.3. Circuit for simulating time-independent Hamiltonian evolution via the truncated Taylor series method.

Finally, we bound the remainder of the Taylor expansion.

$$\left\|\sum_{k=K}^{\infty} \frac{(-iH\Delta t)^k}{k!}\right\| \leq \frac{(\|H\|\,\Delta t)^K}{K!} e^{\|H\|\Delta t} \leq \left(\frac{e\,\|H\|\,\Delta t}{K}\right)^K e^{\|H\|\Delta t} \leq \epsilon'. \tag{14.26}$$

Here we use the relation

$$K^K \leq e^K K!, \tag{14.27}$$

and we may choose K using the refined estimate Eq. (14.6). Concretely, taking $a = e\,\|H\|\,\Delta t$ and replacing ϵ' by $\epsilon' e^{-\|H\|\Delta t}$ gives an inequality of the form $(a/K)^K \leq \epsilon' e^{-\|H\|\Delta t}$; since $a = \mathcal{O}(1)$ for $\Delta t = \ln 2$ and $\|H\| \leq 1$, Eq. (14.6) yields

$$K = \mathcal{O}\left(\frac{\log(1/\epsilon')}{\log\log(1/\epsilon')}\right) = \mathcal{O}\left(\frac{\log(t/\epsilon)}{\log\log(t/\epsilon)}\right). \tag{14.28}$$

The total query complexity to U_H is $\mathcal{O}\left(\frac{t\log(t/\epsilon)}{\log\log(t/\epsilon)}\right)$, and the number of ancilla qubits is $\mathcal{O}(m + \log\log(t/\epsilon))$.

14.3. Time-ordered operator exponentials

Let us first consider a simple time-dependent Hamiltonian $H(t) = (T-t)I + tX$, for which the Schrödinger equation reads

$$\partial_t |\psi(t)\rangle = -i((T-t)I + tX)|\psi(t)\rangle\,, \quad |\psi(0)\rangle = |\psi_0\rangle\,. \tag{14.29}$$

For some initial state $|\psi_0\rangle$, we can rewrite the problem as an operator differential equation for the time evolution operator $U(t)$ such that $U(t)|\psi_0\rangle = |\psi(t)\rangle$ for any t. The solution to the differential equation is

$$U(t) = e^{-i\int_0^t ((T-s)I + sX)\,\mathrm{d}s} = e^{-i(I(Tt - t^2/2) + t^2X/2)}. \tag{14.30}$$

This can be verified by differentiating the right-hand side, using that $H(s)$ commutes with $H(s')$ for all s, s' in this example.

On the other hand, if we change the Hamiltonian to $H(t) = (T-t)Z + tX$, then

$$U(t) \neq e^{-i(Z(Tt-t^2/2)+t^2X/2)}. \tag{14.31}$$

The exponential of the integrated Hamiltonian does not solve the differential equation in this case because $H(t)$ fails to commute with itself at different times. Specifically, for $t \neq t'$ one has $\|[H(t), H(t')]\| \neq 0$ (assuming $T \neq 0$). This prevents us from differentiating $e^{-i\int_0^t H(s)\,\mathrm{d}s}$ by the usual chain rule.

This example illustrates that time dependence introduces genuinely new features. In particular, the time dependence of $H(t)$ need not be analytic, and high-order derivatives may not exist. Such issues can affect the validity of high-order Taylor-type arguments, and have led to misunderstandings in numerical analysis for non-analytic time-dependent problems [CC02]. We therefore begin by formalizing the solution to the operator differential equations that arise in time-dependent simulation and introducing the **Dyson series expansion**, which underlies the truncated Dyson series method.

THEOREM 14.6 (Time-ordered operator exponentials and Dyson series). *Let $A : [0,t] \to \mathbb{C}^{d\times d}$ for some positive integer d be a time-dependent linear operator that is continuous on $[0,t] \subset \mathbb{R}$. The operator differential equation $\partial_t U(t) = A(t)U(t)$ with initial condition $U(0) = I$ has a unique solution, called an ordered operator exponential, and for any $t > 0$ can be expressed as the series expansion*

$$U(t) = \mathcal{T}e^{\int_0^t A(s)\,\mathrm{d}s} = I + \sum_{n=1}^{\infty} \int_0^t \int_0^{s_n} \int_0^{s_{n-1}} \cdots \int_0^{s_2} A(s_n)\cdots A(s_1)\,\mathrm{d}s_1 \cdots \mathrm{d}s_n, \tag{14.32}$$

which is known as the Dyson series expansion.

PROOF. From the fundamental theorem of calculus,

$$\int_0^t \partial_s U(s)\,\mathrm{d}s = U(t) - U(0) = U(t) - I. \tag{14.33}$$

Substituting $\partial_s U(s) = A(s)U(s)$ yields the integral equation

$$U(t) = I + \int_0^t A(s)U(s)\,\mathrm{d}s. \tag{14.34}$$

Iterating Eq. (14.34) p times gives

$$U(t) = I + \sum_{n=1}^{p} \int_0^t \int_0^{s_n} \int_0^{s_{n-1}} \cdots \int_0^{s_2} A(s_n)\cdots A(s_1)\,\mathrm{d}s_1 \cdots \mathrm{d}s_n + R_{p+1}(t), \tag{14.35}$$

where

$$R_{p+1}(t) = \int_0^t \int_0^{s_{p+1}} \int_0^{s_p} \cdots \int_0^{s_2} A(s_{p+1})\cdots A(s_1)\,U(s_1)\,\mathrm{d}s_1 \cdots \mathrm{d}s_{p+1}. \tag{14.36}$$

It remains to show that $\lim_{p\to\infty} \|R_{p+1}(t)\| = 0$.

Let $M := \sup_{s\in[0,t]} \|A(s)\| < \infty$, which exists because A is continuous on a compact interval. Moreover, one has the bound

$$\|U(s)\| \leq \exp\Big(\int_0^s \|A(\tau)\|\;\mathrm{d}\tau\Big) \leq e^{Ms} \leq e^{Mt} \qquad (0 \leq s \leq t). \tag{14.37}$$

Using submultiplicativity and the volume of the simplex $\{0 \le s_1 \le \cdots \le s_{p+1} \le t\}$, we obtain

$$\|R_{p+1}(t)\| \le e^{Mt}\, M^{p+1} \int_0^t \int_0^{s_{p+1}} \cdots \int_0^{s_2} \mathrm{d}s_1 \cdots \mathrm{d}s_{p+1} = e^{Mt}\, \frac{(Mt)^{p+1}}{(p+1)!}. \tag{14.38}$$

Since $(p+1)!$ dominates $(Mt)^{p+1}$ as $p \to \infty$, this implies $\lim_{p\to\infty} \|R_{p+1}(t)\| = 0$, and hence the Dyson series equals $U(t)$. □

The Dyson series can be viewed as the natural analogue of the Taylor series when $A(t)$ is constant. Specifically, if $A(t) = A$ for all t, then

$$U(t) = I + \int_0^t A \, \mathrm{d}s_1 + \int_0^t \int_0^{s_2} A^2 \, \mathrm{d}s_1 \, \mathrm{d}s_2 + \cdots = I + At + \frac{A^2 t^2}{2!} + \cdots . \tag{14.39}$$

An alternative way to characterize the evolution operator is through **time-ordering**. The idea is to use a time-ordering symbol that reorders products of operators according to their time arguments. As an example, the time-ordered product of two operators $\mathcal{T}[A(s_1)A(s_2)]$ is given by

$$\mathcal{T}[A(s_1)A(s_2)] = \begin{cases} A(s_1)A(s_2), & s_1 \ge s_2; \\ A(s_2)A(s_1), & s_1 < s_2. \end{cases} \tag{14.40}$$

Note that the two possible outcomes are in general different, since the matrices A at different times might not commute. Using the identity

$$\int_0^t \cdots \int_0^t \mathcal{T}[A(s_1)\cdots A(s_n)]\, \mathrm{d}s_1 \cdots \mathrm{d}s_n = n! \int_0^t \int_0^{s_n} \int_0^{s_{n-1}} \cdots \int_0^{s_2} A(s_n)\cdots A(s_1)\, \mathrm{d}s_1 \cdots \mathrm{d}s_n, \tag{14.41}$$

we may define the time-ordered matrix exponential as

$$\mathcal{T}\Big[e^{\int_0^t A(s)\,\mathrm{d}s}\Big] = I + \sum_{n=1}^{\infty} \frac{1}{n!} \int_0^t \cdots \int_0^t \mathcal{T}[A(s_1)\cdots A(s_n)]\, \mathrm{d}s_1 \cdots \mathrm{d}s_n. \tag{14.42}$$

Therefore the solution of $\partial_t U(t) = A(t)U(t)$ with $U(0) = I$ can be written as

$$U(t) = \mathcal{T}\Big[e^{\int_0^t A(s)\,\mathrm{d}s}\Big]. \tag{14.43}$$

The time-ordering symbol is best viewed as a prescription rather than as an operator acting on the underlying Hilbert space: for each tuple $(s_1, \ldots, s_n)$, it replaces the product $A(s_1)\cdots A(s_n)$ by the product with the factors sorted in nonincreasing time order. For this reason, it is often preferable to work directly with the Dyson series definition of the ordered operator exponential. However, the time-ordered exponential notation will be useful in the truncated Dyson series algorithm.

14.4. Block encoding of time-dependent operators

A time-dependent Hamiltonian maps a tuple of the time and a vector in the Hilbert space of the simulation to another vector in the Hilbert space which is of the form $H : \mathbb{R} \times \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$. This means that a block-encoding of a time-dependent Hamiltonian must do more than just block encoding the space that the Hamiltonian acts on: it must also block encode the time.

Let us divide the interval $[0, T]$ into 2^{n_t} equal steps of size $\Delta t = \frac{T}{2^{n_t}}$. Then for any $t \in [0, T)$, we can approximate the time-dependent Hamiltonian by the piecewise-constant function

$$\widetilde{H}(t) = H(t_\ell), \quad \ell = \left\lfloor \frac{t}{\Delta t} \right\rfloor, \qquad t_\ell = \ell\, \Delta t. \tag{14.44}$$

By Taylor's theorem, the local error in the Hamiltonian is

$$\left\| H(t) - \widetilde{H}(t) \right\| \leq \max_{s\in[0,T]} \|H'(s)\| \ \Delta t, \tag{14.45}$$

and the error in the time-ordered evolution satisfies

$$\left\| \mathcal{T} e^{-i\int_0^T H(s)\ \mathrm{d}s} - \mathcal{T} e^{-i\int_0^T \widetilde{H}(s)\ \mathrm{d}s} \right\| \leq \max_{s\in[0,T]} \|H'(s)\| \ T\,\Delta t. \tag{14.46}$$

Thus the number of temporal points that we use in the discretization needs to increase as the derivative of the Hamiltonian increases. Specifically if we want to ensure that the error in the evaluation of the time evolution operator due to discretization is at most ϵ_T, then it suffices to choose

$$n_t = \mathcal{O}\left(\log\left(\frac{\max_s \|H'(s)\| \, T^2}{\epsilon_T} \right) \right). \tag{14.47}$$

Therefore, even for rapidly varying Hamiltonians, the number of qubits required for the time register scales only logarithmically in $\max_s \|H'(s)\|$ and $1/\epsilon_T$.

We can now block-encode the discretized Hamiltonian by using a clock register for the time and a standard block encoding on the system register.

Definition 14.7 (HAM-T oracle)**.** *Let $H(t) \in \mathbb{C}^{2^n \times 2^n}$ be a time-dependent Hamiltonian. Define a unitary oracle* `HAM-T` *acting on $n_t + m + n$ qubits, where m ancillas are used for the system block encoding. For a normalization constant $\alpha_T > 0$, define* `HAM-T` *by*

$$(I^{\otimes n_t} \otimes \langle 0^m| \otimes I^{\otimes n})\texttt{HAM-T}(I^{\otimes n_t} \otimes |0^m\rangle \otimes I^{\otimes n}) = \sum_{\ell=0}^{2^{n_t}-1} |\ell\rangle\langle\ell| \otimes \frac{H(\ell T/2^{n_t})}{\alpha_T}. \tag{14.48}$$

This oracle provides a block-encoding of the block-diagonal operator $\sum_\ell |\ell\rangle\langle\ell| \otimes H(\ell T/2^{n_t})$.

To construct such an oracle, we still need to implement each $H(t_\ell)$ via a unitary block encoding. We do so using a linear-combination-of-unitaries (LCU) decomposition together with a clock register. The computational basis on the clock is a convenient choice, though other bases (for example, Fourier modes) can also be used.

Definition 14.8 (Standard form for time-dependent LCU)**.** *Let $H(t) \in \mathbb{C}^{2^n \times 2^n}$ be expressed as*

$$H(t) = \sum_{j=0}^{M-1} c_j(t) U_j(t), \tag{14.49}$$

where $c_j(t) \geq 0$ and each $U_j(t)$ is unitary on $\mathbb{C}^{2^n}$. Define SELECT *so that its action on a time index ℓ and Hamiltonian index j satisfies*

$$(\langle \ell| \langle j| \otimes I^{\otimes n})\,\mathrm{SELECT}(|\ell\rangle |j\rangle \otimes I^{\otimes n}) = U_j(\ell T/2^{n_t}). \tag{14.50}$$

Similarly, the time-dependent coefficients are specified by a state-preparation oracle PREP-T*, which is defined for some $\alpha \geq \max_\ell \sum_{j=0}^{M-1} c_j(\ell T/2^{n_t})$ by*

$$(\text{PREP-T})\,|\ell\rangle\,|0^{\lceil \log M\rceil}\rangle = |\ell\rangle \sum_{j=0}^{M-1} \frac{\sqrt{c_j(\ell T/2^{n_t})}}{\sqrt{\alpha}} |j\rangle\,. \tag{14.51}$$

By Lemma 9.5, the resulting block encoding is

$$\texttt{HAM-T} = ((\text{PREP-T})^\dagger \otimes I)\ \text{SELECT}\ ((\text{PREP-T}) \otimes I). \tag{14.52}$$

Note above that we again make the seemingly restrictive assumption that $c_j(t) \geq 0$. This assumption is made only to simplify the discussion of the block-encoding as the sign of each coefficient can always be absorbed into the definition of the unitary.

Example 14.9. Let us clarify this construction by considering an elementary time-dependent Hamiltonian,

$$H(t) = X + \sin(\omega t)Z. \tag{14.53}$$

This violates the assumption $c_j(t) \geq 0$ since $\sin(\omega t)$ changes sign. We can instead write

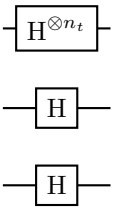
$$H(t) = X + \frac{1}{2}(-ie^{i\omega t})Z + \frac{1}{2}(ie^{-i\omega t})Z. \tag{14.54}$$

The Hamiltonian is now in standard form, with coefficients $c_j(t) = [1, 1/2, 1/2]$ and unitaries $U_j(t) = [X, -ie^{i\omega t}Z, ie^{-i\omega t}Z]$. In this case, we can further rewrite it as

$$H(t) = \frac{X}{2} + \frac{X}{2} + \frac{1}{2}(-ie^{i\omega t})Z + \frac{1}{2}(ie^{-i\omega t})Z. \tag{14.55}$$

At first sight, splitting a single X term into two identical terms seems wasteful. However, in LCU constructions the cost often depends more strongly on the number of distinct coefficient values than on the raw number of terms. Here we can implement PREP-T by preparing a uniform superposition over the index registers.

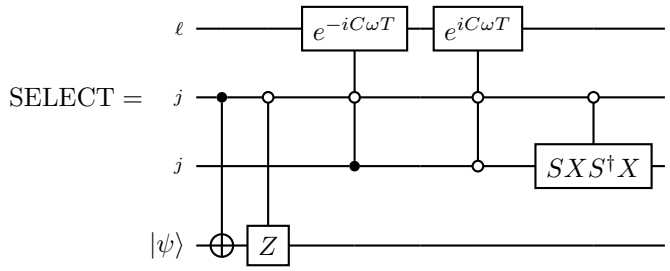
PREP-T =



Next for notational simplicity we introduce the following operator which serves to convert bit strings corresponding to the time to a fraction of the total evolution time. That is to say, we wish to construct an operator such that $|\ell\rangle \mapsto \frac{\ell}{2^{n_t}}$ for any $\ell \in \{0,1\}^{n_t}$.

$$C := \sum_{\ell=0}^{2^{n_t}-1} \frac{\ell}{2^{n_t}} |\ell\rangle\langle\ell|. \tag{14.56}$$

C is a Hermitian clock operator (not a unitary). Identifying a basis string $\ell \in \{0,1\}^{n_t}$ with the integer $\ell \in \{0, \ldots, 2^{n_t} - 1\}$, the unitary $e^{\pm i\omega TC}$ applies the phase $e^{\pm i\omega T\ell/2^{n_t}}$ to $|\ell\rangle$, which realizes the factors $e^{\pm i\omega t_\ell}$ by phase kickback. We now implement SELECT by encoding the control logic for the unitaries as follows: the first qubit selects whether we apply an X or Z term, and the second selects the sign of the frequency. The resulting SELECT circuit is



The first two gates in the circuit decide whether an X or Z Hamiltonian term is selected, whereas the last terms apply the required phases to the unitaries in the event that one of the two Z terms in the standard form decomposition of $H(t)$ is selected via the phase kickback effect. We leave it as an exercise to the reader to validate that the operation $e^{-iC\omega T}$ can be efficiently implemented using a sequence of n_t R_z rotations on the time register (and, when needed inside the select oracle, their controlled versions). The final term applies the respective $+i, -i$ phase to the Hamiltonian terms as seen from the fact that

$$SXS^\dagger X = \begin{bmatrix} -i & 0 \\ 0 & i \end{bmatrix}, \tag{14.57}$$

which provides precisely the phases required by the standard form LCU expression.

◇

14.5. Truncated Dyson series for time-dependent Hamiltonian simulation

With the block encoding of a time-dependent Hamiltonian in hand, we turn to simulating the time-ordered evolution $\mathcal{T}e^{-i\int_0^t H(s)\,\mathrm{d}s}$. The approach was pioneered by [LW19, KSB19] and is based on implementing the **truncated Dyson series** of Theorem 14.6 as a linear combination of unitaries. This method generalizes the intuition behind the truncated Taylor series method in Section 14.2, but it is more involved because the terms must be ordered according to the times at which the Hamiltonian is queried. Here we present a simplified discussion that is not optimized for gate counts or ancilla qubits, but has essentially the same query complexity. In settings where Hamiltonian-oracle queries dominate the cost, this query complexity is a useful proxy for the overall complexity.

As in the truncated Taylor series algorithm, the truncated Dyson series algorithm simulates the evolution by dividing time into short segments and then concatenating the resulting circuits. This is needed because the block-encoding normalization factor scales exponentially with the segment duration, as in the truncated Taylor series method. The performance guarantee for a single short segment is stated next.

Lemma 14.10 (Truncated Dyson Hamiltonian simulation for short segments)**.** *Let $H : [0, T] \to \mathbb{C}^{2^n \times 2^n}$ be a differentiable time-dependent Hamiltonian that has a standard-form LCU decomposition given by the oracles* PREP-T *and* SELECT*. There exists a quantum algorithm that for any $\epsilon > 0$ and $T \in [0, \ln(2)/\alpha_T]$ implements a unitary V such that for a constant $\beta \leq 2 + \mathcal{O}(\epsilon)$,*

$$\left\| \beta(\langle 0| \otimes I)V(|0\rangle \otimes I) - \mathcal{T}e^{-i\int_0^T H(s)\,\mathrm{d}s} \right\| \leq \epsilon, \tag{14.58}$$

using a number of qubits encoding the time, n_t, that satisfies (14.47) *and a number of queries to* `HAM-T` *that is in $\mathcal{O}\big(\log(1/\epsilon)/\log\log(1/\epsilon)\big)$.*

PROOF. The first step of the algorithm involves building a truncated Dyson series expansion of the time-ordered operator exponential up to order K. We focus on queries to `HAM-T`, because a single query to this oracle can be simulated using $\mathcal{O}(1)$ queries to PREP-T and SELECT as in Definition 14.8.

We next show that the simulation error can be made at most ϵ. There are two sources of error: truncating the Dyson series at order K, and discretizing time using n_t bits (as discussed in (14.44)). Using the piecewise-constant approximation $\widetilde{H}$, we write

$$\begin{aligned}\left\|\beta(\langle 0|\otimes I)V(|0\rangle\otimes I)-\mathcal{T}e^{-i\int_0^T H(s)\,\mathrm{d}s}\right\| \le{}& \left\|\mathcal{T}e^{-i\int_0^T H(s)\,\mathrm{d}s}-\mathcal{T}e^{-i\int_0^T \widetilde{H}(s)\,\mathrm{d}s}\right\| \\ &+\left\|\beta(\langle 0|\otimes I)V(|0\rangle\otimes I)-\mathcal{T}e^{-i\int_0^T \widetilde{H}(s)\,\mathrm{d}s}\right\|.\end{aligned} \tag{14.59}$$

We know from (14.46) that by choosing n_t to satisfy (14.47), we can ensure the discretization error is at most $\epsilon_T = \epsilon/2$:

$$\left\|\mathcal{T}e^{-i\int_0^T H(s)\,\mathrm{d}s}-\mathcal{T}e^{-i\int_0^T \widetilde{H}(s)\,\mathrm{d}s}\right\| \le \epsilon/2. \tag{14.60}$$

Thus it suffices to ensure that the error in the truncated Dyson series for the rounded Hamiltonian can be made at most $\epsilon/2$.

The error in the Dyson series approximation at order K has already been computed in (**??**). Since V is a block-encoding of the truncated Dyson series of the discretized-time Hamiltonian, and the truncation bound depends only on an a priori bound on $\left\|\widetilde{H}(s)\right\|$, we obtain

$$\left\|\beta(\langle 0|\otimes I)V(|0\rangle\otimes I)-\mathcal{T}e^{-i\int_0^T \widetilde{H}(s)\,\mathrm{d}s}\right\| \le \frac{(\sup_s\left\|\widetilde{H}(s)\right\| T)^{K+1}}{(K+1)!} \le \frac{(\alpha_T T)^{K+1}}{(K+1)!}. \tag{14.61}$$

Then, using the same argument as in the analysis of the truncated Taylor series simulation algorithm and the refined estimate Eq. (14.6), we obtain

$$K = \mathcal{O}\left(\frac{\log(\sup_s\|H(s)\|\,T/\epsilon)}{\log\log(\sup_s\|H(s)\|\,T/\epsilon)}\right) = \mathcal{O}\left(\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right), \tag{14.62}$$

where the last result follows from the assumption in the lemma statement that $T \le \ln(2)/\alpha_T$. This shows that for K chosen in this fashion the truncation error is at most $\epsilon/2$, and hence the overall error is at most ϵ after accounting for the discretization error.

We now show that we can construct the Dyson series expansion to order K using at most K queries to `HAM-T`. Using the definition of the Dyson series as given by (14.42),

$$\mathcal{T}e^{-i\int_0^T \widetilde{H}(s)\,\mathrm{d}s} \approx I + \sum_{k=1}^{K}\frac{(-i)^k}{k!}\int_0^T\int_0^T\int_0^T\cdots\int_0^T \mathcal{T}\big[\widetilde{H}(s_k)\cdots\widetilde{H}(s_1)\big]\,\mathrm{d}s_1\cdots\mathrm{d}s_k. \tag{14.63}$$

Then, because the Hamiltonian is piecewise constant, we can express this integral as

$$\int_0^T\int_0^T\int_0^T\cdots\int_0^T \mathcal{T}\big[\widetilde{H}(s_k)\cdots\widetilde{H}(s_1)\big]\,\mathrm{d}s_1\cdots\mathrm{d}s_k = \sum_{s_1,\ldots,s_k\in\mathcal{S}} \mathcal{T}\big[\widetilde{H}(s_k)\cdots\widetilde{H}(s_1)\big](\Delta t)^k, \tag{14.64}$$

where $\mathcal{S} = \{\ell\Delta t \mid \ell\in[2^{n_t}]\}$. This can be written as a linear combination of unitaries; however, the domain of integration grows exponentially with k. Fortunately, the cost of LCU depends on the one-norm of the coefficients rather than the number of terms, as in Lemma 9.5. Thus the cost of implementing this sum need not be exponential in k. The main object that we need to implement

is an operator of the form $\mathcal{T}\widetilde{H}(s_k)\cdots\widetilde{H}(s_1)$ which sorts the different Hamiltonian terms based on the time that each is evaluated at.

To implement time ordering, we need each summand in the expansion to have its own clock. Specifically, we wish to apply a transformation of the form

$$|\ell_1\rangle|\ell_2\rangle\cdots|\ell_k\rangle\otimes|\psi\rangle \to |\ell_1\rangle|\ell_2\rangle\cdots|\ell_k\rangle\otimes\mathcal{T}\widetilde{H}(s_k)\cdots\widetilde{H}(s_2)\widetilde{H}(s_1)|\psi\rangle \tag{14.65}$$

where $s_p = \ell_p\Delta t$. This transformation would allow us to build each of the terms in the truncated Dyson series (for the rounded Hamiltonian $\widetilde{H}$). However, the details of a reversible sorting process that can achieve this are complicated. An optimized version of the sorting process is described in [LW19] that includes all ancillae and gate operations needed in addition to the query operations. □

With the ability to simulate a short time segment for which $\alpha_T T = \ln(2)+\mathcal{O}(\epsilon)$, we can simulate for arbitrary durations by concatenating segments and boosting success probabilities via amplitude amplification.

THEOREM 14.11 (Truncated Dyson Simulations for Long Segments). *Let $H:\mathbb{R}\to\mathbb{C}^{2^n\times 2^n}$ be a differentiable time-dependent Hamiltonian that has a standard-form LCU decomposition given by the oracles* PREP-T *and* SELECT. *There exists a quantum algorithm that for any $\epsilon>0$ and $T>0$ implements a unitary U such that*

$$\left\|(\langle 0|\otimes I)U(|0\rangle\otimes I) - \mathcal{T}e^{-i\int_0^T H(s)\,\mathrm{d}s}\right\| \le \epsilon, \tag{14.66}$$

using $n_t = \mathcal{O}\left(\log\left(\frac{\max_{s\in[0,T]}\|H'(s)\|T^2}{\epsilon}\right)\right)$ qubits to encode the time, and $\widetilde{\mathcal{O}}\left(\alpha_T T\log(1/\epsilon)\right)$ queries to `HAM-T`.

PROOF SKETCH. The core idea of the strategy is to divide the total simulation time T into $r-1$ equal segments of duration $\ln(2)/\alpha_T$ and a final segment of length $T-(r-1)\ln(2)/\alpha_T$. Each of the first $r-1$ segments is chosen so that the success probability of the corresponding simulation is close to $1/4$, allowing it to be boosted to nearly 1 using a single round of oblivious amplitude amplification (see Section 11.4). The final, possibly shorter segment does not have this optimal success probability, but its amplitude can still be amplified using fixed-point amplitude amplification (see Section 13.4). The overall simulation circuit is obtained by concatenating the circuits for each segment. □

There is one subtlety in this argument. In practice, the value of α_T may depend on the chosen time interval. That is, if the Hamiltonian has a small coefficient norm over a particular segment, then this simulation method suggests using a longer timestep. However, increasing the timestep in turn changes the value of α_T for that interval. This introduces an opportunity for optimization: by choosing segment durations optimally, the overall query complexity can scale as $\widetilde{\mathcal{O}}\left(\int_0^T \alpha_T(t)\,\mathrm{d}t\cdot\log(1/\epsilon)\right)$ queries to `HAM-T`. This can be advantageous when $\alpha_T(t)$ is large only locally in time but the integral remains small.

Comparing the result here to Section 14.1, we find that the complexity does not reach the optimal scaling achievable via qubitization. This is expected, since truncated Dyson (like truncated Taylor) methods operate fundamentally differently from quantum signal processing and qubitization. Qubitization relies on constructing a walk operator whose spectrum directly corresponds to that of the Hamiltonian. While time-dependent Hamiltonians possess well-defined instantaneous spectra, they generally do not admit global eigenvalues or eigenvectors. As a result, it is not

possible to define a single fixed walk operator that encodes the entire time evolution and can be efficiently transformed via quantum singular value transformation. To date, no known method fully bridges the gap in query complexity between time-independent and time-dependent Hamiltonian simulation.

14.6. Interaction picture simulation

As the truncated Dyson series method has slightly worse asymptotic scaling in t and ϵ compared to qubitization, it may seem surprising that it can still be advantageous for simulating time-independent Hamiltonians in certain regimes. The key idea is the interaction picture, which rewrites the dynamics so that the remaining time-dependent Hamiltonian term has a significantly smaller operator norm. In such cases, this reduction can offset the asymptotic overhead of truncated-Dyson methods relative to the optimal scaling achieved by qubitization.

The idea behind **interaction picture simulation** is simple. Assume that we have a time-independent Hamiltonian of the form $H = \alpha A + \beta B$, where A and B are Hermitian matrices with $\|A\| \leq 1$ and $\|B\| \leq 1$, and where $\beta \gg \alpha \geq 0$. In this regime, the dynamics is dominated by B. We can factor out this dominant part by switching to a time-dependent frame (the interaction picture with respect to B). For a state $|\psi(t)\rangle$ that obeys the Schrödinger equation under H, define

$$|\psi(t)\rangle_I := e^{i\beta Bt} |\psi(t)\rangle . \tag{14.67}$$

We now compute the corresponding evolution equation for $|\psi(t)\rangle_I$.

Proposition 14.12. *Let $H = \alpha A + \beta B$ be a Hermitian operator acting on a finite-dimensional Hilbert space with Hermitian A, B. If $|\psi(t)\rangle$ satisfies $\partial_t |\psi(t)\rangle = -iH |\psi(t)\rangle$ and $|\psi(t)\rangle_I := e^{i\beta Bt} |\psi(t)\rangle$, then $|\psi(t)\rangle_I$ satisfies*

$$\partial_t |\psi(t)\rangle_I = -iH_I(t) |\psi(t)\rangle_I , \tag{14.68}$$

*where $H_I(t)$ is called the **interaction Hamiltonian** with respect to B and is defined by*

$$H_I(t) := \alpha e^{i\beta Bt} A e^{-i\beta Bt}. \tag{14.69}$$

Proof. This follows from the product rule and the Schrödinger equation for $|\psi(t)\rangle$.

$$\begin{aligned} \partial_t |\psi(t)\rangle_I &= i\beta B e^{i\beta Bt} |\psi(t)\rangle + e^{i\beta Bt} \partial_t |\psi(t)\rangle \\ &= i\beta B e^{i\beta Bt} |\psi(t)\rangle - i e^{i\beta Bt} (\alpha A + \beta B) |\psi(t)\rangle \\ &= -i e^{i\beta Bt} (\alpha A) |\psi(t)\rangle = -i e^{i\beta Bt} (\alpha A) e^{-i\beta Bt} |\psi(t)\rangle_I \\ &= -iH_I(t) |\psi(t)\rangle_I . \end{aligned} \tag{14.70}$$

□

Since unitary conjugation preserves the operator norm, we have

$$\|H_I(t)\| = \alpha \|A\| \leq \alpha. \tag{14.71}$$

Thus the interaction picture isolates a time-dependent Hamiltonian whose operator norm is governed by α rather than β. The tradeoff is that $H_I(t)$ can vary rapidly in time when β is large. Indeed, differentiating gives

$$H_I'(t) = i\alpha\beta\, e^{i\beta Bt} [B, A] e^{-i\beta Bt}, \tag{14.72}$$

so the relevant variation scale depends on β (and on the commutator $[B, A]$). In truncated-Dyson approaches, this rapid time variation affects the time-discretization overhead (for example, the number of time-register qubits), which scales only logarithmically in such variation bounds.

This advantage, however, comes with an important caveat: one must be able to efficiently implement the evolution $e^{i\beta Bt}$ on a quantum computer. For example, if B is one-sparse (such as a Pauli string), then this evolution can be implemented efficiently. More generally, the benefit applies whenever B can be fast forwarded, meaning that its time evolution can be simulated in sublinear time in βt. While fast forwardability can arise for specific terms in a Hamiltonian decomposition, it does not hold generically. In fact, as discussed in the next section, there are Hamiltonians that provably cannot be simulated in fewer than $\mathcal{O}(\beta t)$ queries.

14.7. No fast forwarding theorem

Notes and further reading

The use of linear combination of unitaries (LCU) for Hamiltonian simulation was developed in [CW12], and the truncated Taylor series method with nearly optimal dependence was developed in [BCK15]. The truncated Dyson series method for time-dependent simulation was developed in [LW19, KSB19]. The truncated Dyson series method is closely related to the truncated Magnus expansion method, which can be particularly suitable for highly oscillatory time-dependent problems; see, for example, [AFL22, CZA24**?**] and the references therein.

Ref. [BACS07] proved the first no-fast-forwarding theorem for time-independent Hamiltonian simulation. For more general differential equations related to Chapter 20, see [ALWZ25].

CHAPTER 15

Operator splitting based Hamiltonian simulation

In this chapter, we consider a specific class of Hamiltonian simulation algorithms where the Hamiltonian is given as a sum of simpler components: $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$, with each term H_γ generating an evolution $U_\gamma(t) = e^{-itH_\gamma}$ that is assumed to be easy to implement. The central question is: how can we combine these individually implementable unitaries to approximate the full evolution $U(t) = e^{-iHt}$? This leads to the idea of approximating the exponential of a sum by a product of exponentials, a strategy known as **operator splitting**. This approximation then seeks to find an approximation of the form

$$U(t) \approx \prod_{j=1}^{N_{\exp}} e^{-it_j H_{\gamma_j}}, \tag{15.1}$$

for some sequence of times t_j and Hamiltonian terms γ_j. Here the above holds approximately, but not precisely, because matrix multiplication is not commutative.

The above product formula approximation yields a divide and conquer scheme wherein the Hamiltonian is subdivided into smaller simulations wherein the evolution can be either brute forced or simulated using the methods in Chapter 14. This family of techniques is also commonly referred to as **product formula**, **Trotter decomposition**, Trotter splitting, or simply, splitting methods. Terms such as the Trotter-Suzuki expansion are sometimes used interchangeably with operator splitting, although they more precisely refer to specific recursive constructions derived via composition techniques. In contrast, operator splitting encompasses a broader class of methods that approximate time evolution using simpler component dynamics. Operator splitting also played a foundational role in the early development of quantum computation theory. In particular, it was used to argue that a universal quantum simulator could be realized (for local Hamiltonians) by applying a Trotter decomposition to the time evolution operator [Llo96].

It is also important to note that approximating $U(t)$ is not merely a numerical task in scientific computing. In quantum algorithms, the ability to implement $U(t)$ from easily accessible building blocks $U_\gamma(t)$ also enables indirect access to the Hamiltonian H itself. While the formal inverse operation $H = (\log U(t))/(-it)$ is generally not computed explicitly, eigenvalue transformations and other matrix processing tasks can often proceed using $U(t)$ directly, without recovering H in closed form. An approximate implementation of $U(t)$ also serves as a powerful input model for representing matrices. This is known as the **Hamiltonian simulation based input model**, also claled the **Hamiltonian evolution based input model**. Among the various approaches available, Trotter-style product formulas remain the most widely used and practically implementable method for realizing this model on quantum hardware.

We begin with the simplest case where the Hamiltonians commute exactly, and then develop first- and second-order product formulas. We proceed to higher-order generalizations, error bounds

involving commutators, and more refined norm-based estimates. We also discuss recent extensions, including randomized time-sampling strategies and adaptations to time-dependent Hamiltonians.

15.1. Warmup: the commuting case

First let us begin with the case where all terms in the Hamiltonian that we wish to simulate commute. Commuting matrices act very much like ordinary numbers. When all terms H_γ commute with each other, we simply have

$$U(t) = \prod_{\gamma=1}^{\Gamma} e^{-iH_\gamma t} = \prod_{\gamma} U_\gamma(t). \tag{15.2}$$

This arises from an elementary calculation involving the definition of the operator exponential, which we include below for completeness.

Lemma 15.1. *Let $\{A_\gamma : \gamma = 1, \dots, \Gamma\}$ be normal matrices acting on a finite Hilbert space $\mathcal{H}$ such that $[A_\gamma, A_{\gamma'}] = 0$ for all $\gamma, \gamma' \in \{1, \dots, \Gamma\}$. Then*

$$e^{\sum_\gamma A_\gamma} = \prod_{\gamma} e^{A_\gamma}. \tag{15.3}$$

Proof. If $\Gamma = 1$ then there is only one term in the product formula. Let us then continue to the case where $\Gamma = 2$. Assume that $[A, B] = 0$ for operators $A, B \in L(\mathcal{H})$ then we wish to show that $e^{A+B} = e^A e^B$. We can demonstrate this through Taylor's theorem (which holds because $\mathcal{H}$ is finite dimensional)

$$e^{A+B} = \sum_{j=0}^{\infty} (A+B)^j / j!. \tag{15.4}$$

Note that $(A+B)^j$ is simply the sum of all unordered sequences of A, B of length j. If $[A, B] = 0$ then $AB = BA$. Thus for example $(A+B)^2 = A^2 + AB + BA + B^2 = A^2 + 2AB + B^2$. Repeating this same pattern for larger j yields

$$(A+B)^j = \sum_{p=0}^{j} \binom{j}{p} A^{j-p} B^p. \tag{15.5}$$

Thus in the commuting case

$$e^{A+B} = \sum_{j=0}^{\infty} \sum_{p=0}^{j} \binom{j}{p} A^{j-p} B^p / j! = \sum_{j=0}^{\infty} \sum_{p=0}^{j} \frac{A^{j-p} B^p}{(j-p)! p!} = e^A e^B. \tag{15.6}$$

Thus we have demonstrated that the result holds for $\gamma = 2$.

Now let $p \geq 2$ and assume that $\prod_{\gamma=1}^{p} e^{A_\gamma} = e^{\sum_{\gamma=1}^{p} A_\gamma}$. We then have that

$$\prod_{\gamma=1}^{p} e^{A_\gamma} e^{A_{p+1}} = e^{\sum_\gamma A_\gamma} e^{A_{p+1}}. \tag{15.7}$$

Then from the above case we can combine these two exponentials because this reduces to the case $\Gamma = 2$. Thus

$$e^{\sum_\gamma A_\gamma} e^{A_{p+1}} = e^{\sum_{\gamma=1}^{p+1} A_\gamma} \tag{15.8}$$

Therefore by induction the claim holds for all $p \geq 2$ and thus the result follows. □

Example 15.2 (Sum of Pauli-Z gates)**.** Consider a sum of n Pauli-Z gates $H = \sum_{i=1}^{n} Z_i$. For instance, when $n = 2$,

$$H = Z \otimes I + I \otimes Z = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \end{pmatrix}. \tag{15.9}$$

This is not a unitary matrix (even after normalization by a factor of 2). However, since all Z_i's commute, the Hamiltonian simulation problem

$$U(t) = e^{-itH} = \prod_{i=1}^{n} e^{-itZ_i} \tag{15.10}$$

can be solved exactly with n single qubit gates for any t. ◇

Now that we have shown that a sum of elementary Pauli-Z Hamiltonian can be simulated on a quantum computer a natural next step is to generalize this to a sum of commuting k-local Hamiltonians, which are Hamiltonians that are a sum of Pauli operations that each act non-trivially on at most k qubits. We will further also provide the explicit number of gates needed to perform this product.

THEOREM 15.3. *Let $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$ be a Hamiltonian in $L(\mathbb{C}^{2^n})$ where each $H_\gamma = \alpha_\gamma P_{\gamma_1} \cdots P_{\gamma_n}$ and each $\gamma_j \in \{0, 1, 2, 3\}$ corresponds to a Pauli operator from the set $\{I, X, Y, Z\}$. Further assume that H is k-local, i.e., for each γ at least $n - k$ of the indices satisfy $\gamma_j = 0$, and that $[H_\gamma, H_{\gamma'}] = 0$ for all γ, γ'. We then have that*

1. $e^{-iHt} = \prod_\gamma e^{-iH_\gamma t}$
2. $\prod_\gamma e^{-iH_\gamma t}$ *can be exactly implemented on a quantum computer using at most* Γ R_z *gates and at most* $2(k-1)\Gamma$ CNOT *gates and* $2k\Gamma$ *single qubit Clifford gates.*

PROOF. The first claim in the Theorem is a direct consequence of Lemma 15.1. We will now examine the second claim. Our strategy for implementing these terms is simple. We will first diagonalize each Pauli operator to turn it into a product of Pauli-Z gates. Then we will construct a simulation circuit for each of the diagonal operations by computing the eigenvalues of the diagonal operations.

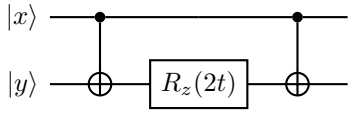
To see how this is achieved let us begin with a simple case: $e^{-iZ\otimes Zt}$. From the eigenvalue decomposition we see that the eigenvalues of the exponential are simply the exponential of the eigenvalues of the exponent. These are particularly easy to find here because Z is diagonal:

$$Z \otimes Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow e^{-iZ\otimes Zt} = \begin{bmatrix} e^{-it} & 0 & 0 & 0 \\ 0 & e^{it} & 0 & 0 \\ 0 & 0 & e^{it} & 0 \\ 0 & 0 & 0 & e^{-it} \end{bmatrix} \tag{15.11}$$

From this we observe that any input $|x\rangle |y\rangle$ obtains a phase of e^{-it} if $x = y = 0$ or $x = y = 1$, and a phase of e^{it} otherwise. Note that this pattern can be more succinctly represented by seeing that, irrespective of x, y, $e^{-i(-1)^{x+y}t}$ is achieved. That is to say

$$|x\rangle |y\rangle \mapsto e^{-i(-1)^{x+y}t} |x\rangle |y\rangle . \tag{15.12}$$

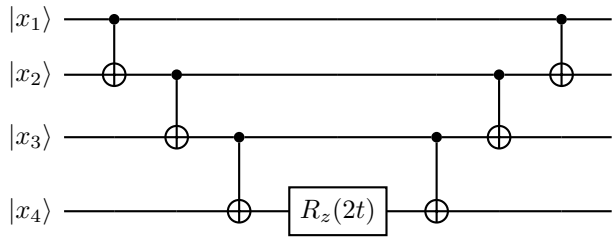
Now recall that $Xe^{-iZt}X = e^{-i(XZX)t} = e^{iZt}$. This implies that we can flip the direction of a Z-rotation by conjugating with an X gate. In effect, when $x = 1$ the sign of the phase rotation for $e^{-iZ\otimes Zt}$ is flipped. Thus we can use the following circuit for $e^{-iZ\otimes Zt}$.

(15.13)
$$e^{-iZ\otimes Zt} =$$


Next let us consider a Hamiltonian that has k Pauli-Z operations in it. We can go through the exact same reasoning to see that

(15.14)
$$e^{-i(\prod_{j=1}^{k} Z_j)t}\,|x_1\rangle\cdots|x_k\rangle = e^{-i(-1)^{\sum_j x_j}t}\,|x_1\rangle\cdots|x_k\rangle\,.$$

Again we see that the phase is flipped based on the parity of $\sum_j x_j$, which can be obtained by applying a chain of CNOT operations and then applying a single R_z rotation on one qubit, as in Eq. (15.13). The circuit for this for $k = 4$ is given below

(15.15)
$$e^{-iZ^{\otimes 4}t} =$$


This circuit is often simplified to have all of the targets for the CNOT operations either on the bottom-most qubit [WBAG11].

We see then that in general any Pauli operator consisting of k Pauli-Z operations can be implemented exactly using $2(k-1)$ CNOT operations and a single R_z gate.

Now let us consider the case of $e^{-iX\otimes Yt}$ for example. We can again use the property that for any unitary U and normal matrix A, $Ue^{A}U^\dagger = e^{UAU^\dagger}$. We then wish to choose U to diagonalize $X \otimes Y$. We can use the fact that $X = \mathrm{H}\,Z\,\mathrm{H}$ and $Y = S\,\mathrm{H}\,Z\,\mathrm{H}\,S^\dagger$ to see that

(15.16)
$$e^{-iX\otimes Yt} = (\mathrm{H}\otimes S\,\mathrm{H})e^{-iZ\otimes Zt}(\mathrm{H}\otimes \mathrm{H}\,S^\dagger)$$

This logic can be extended to the case where there are at most k non-Pauli-Z operators in the exponent. We see that in each case we will have to apply one single-qubit Clifford operation per qubit to diagonalize and then another single-qubit Clifford gate to return to the original basis. Thus the cost for simulating a generic $e^{-i\alpha_\gamma P_\gamma t}$ term in the decomposition is $2k$ single-qubit Clifford gates plus the cost of simulating $Z^{\otimes k}$ (which is $2(k-1)$ CNOT operations and a single R_z gate).

By the first claim, the overall simulation of the unitary consists of Γ exponentials each of which is the exponential of a Pauli operator with at most k terms. Thus the cost is at most Γ times the aforementioned costs, which yield the result stated in the theorem. □

Example 15.4 (2D toric code)**.** Let $N = 2L^2$ spins lie on the edges of the toroidal lattice, with Hilbert space $\mathcal{H} \cong \mathbb{C}^{2^N}$. The Hamiltonian for Kitaev's 2D toric code [Kit03] is

(15.17)
$$H^{\text{toric}} = -\sum_s \mathsf{X}_s - \sum_p \mathsf{Z}_p\,,$$

where s and p range over stars and plaquettes, respectively. A star consists of the four edges incident to a vertex, and a plaquette consists of the four edges surrounding a face, as shown in Fig. 15.1. The corresponding operators are

$$\mathsf{X}_s = \prod_{i \in s} \sigma_i^x, \quad \mathsf{Z}_p = \prod_{i \in p} \sigma_i^z, \tag{15.18}$$

which are all 4-local Pauli operators. Moreover,

$$[\mathsf{X}_s, \mathsf{Z}_p] = [\mathsf{X}_s, \mathsf{X}_{s'}] = [\mathsf{Z}_p, \mathsf{Z}_{p'}] = 0, \tag{15.19}$$

for all stars s, s' and plaquettes p, p'. Hence H^{toric} is a sum of commuting local Hamiltonian terms. Because of the periodic boundary condition, we also have

$$\prod_s \mathsf{X}_s = 1, \quad \prod_p \mathsf{Z}_p = 1, \tag{15.20}$$

so these commuting terms are not all independent. Even so, by Theorem 15.3, for every t we have

$$e^{-itH^{\text{toric}}} = \prod_s e^{it\mathsf{X}_s} \prod_p e^{it\mathsf{Z}_p}, \tag{15.21}$$

and this product formula is exact. Since there are L^2 star terms and L^2 plaquette terms, each supported on 4 qubits, the simulation can be implemented using at most $2L^2$ R_z gates, $12L^2$ CNOT gates, and $16L^2$ single-qubit Clifford gates.

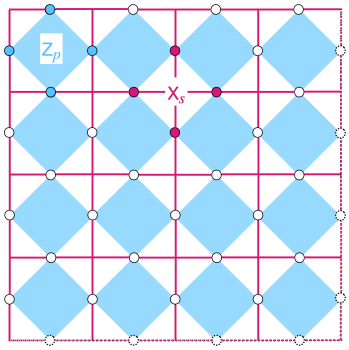


FIGURE 15.1. Illustration of 2D toric code. Each blue plaquette supports the 4-local operator $\mathsf{Z}_p = \prod_{i \in p} \sigma_i^z$, and each red star supports the 4-local operator $\mathsf{X}_s = \prod_{i \in s} \sigma_i^x$. The dashed line on the right and bottom is identified with the solid line on the left and top, respectively, due to the periodic boundary condition.

◇

15.2. First order and second order operator splitting

In quantum physics, a common task is to analyze time evolution under a Hamiltonian that is a sum of non-commuting components. The basic idea of operator splitting methods is that by rapidly alternating the application of the individual terms, the resulting dynamics can approximate the evolution generated by the full sum. This "term flickering" effect is analogous to how a rapidly flickering screen seems to be static but actually is changing incredibly quickly compared to the timescale of a video. Here this reduces the non-commuting case to controlling the errors introduced

by switching, which are governed by commutators between the terms. Quantitative bounds on these errors determine how small the switching time step must be in order to reach a target accuracy.

To analyze the complexity of operator splitting methods, we need to understand how these non-commutativity errors accumulate. The Duhamel principle (Proposition 3.22) provides a convenient way to express the effect of an inhomogeneous perturbation.

Proposition 15.5 (First-order Trotter formula, $\Gamma = 2$)**.** *Let $H = H_1 + H_2$ where $H_1, H_2 \in \mathbb{C}^{N\times N}$ are Hermitian matrices. Let $|||H||| := \|H_1\| + \|H_2\|$. The first-order Trotter approximation is defined as $\widetilde{U}(t) := e^{-itH_1}e^{-itH_2}$. For any $t \geq 0$, the local simulation error is bounded by*

$$\left\| e^{-itH} - \widetilde{U}(t) \right\| \leq t^2 |||H|||^2. \tag{15.22}$$

Consequently, to simulate e^{-itH} for a total time t with precision ϵ, the interval $[0, t]$ can be divided into L steps of size $\Delta t = t/L$, where

$$L = \left\lceil \frac{t^2 |||H|||^2}{\epsilon} \right\rceil. \tag{15.23}$$

Proof. We analyze the error using the Taylor expansion with integral remainder. For a twice differentiable matrix-valued function $f(t)$, we have

$$f(t) = f(0) + f'(0)t + \int_0^t f''(s)(t-s)\,\mathrm{d}s. \tag{15.24}$$

Let us apply this expansion to each matrix entry of the exact and approximate propagators. First, consider the exact propagator $U(t) = e^{-itH}$. Its expansion is

$$U(t) = I - itH + \int_0^t (-iH)^2 e^{-isH}(t-s)\,\mathrm{d}s. \tag{15.25}$$

Next, we examine the approximate propagator $\widetilde{U}(t) = e^{-itH_1}e^{-itH_2}$. Differentiating and using that each H_j commutes with e^{-itH_j}, we find

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\widetilde{U}(t) &= \left(\frac{\mathrm{d}}{\mathrm{d}t}e^{-itH_1}\right)e^{-itH_2} + e^{-itH_1}\left(\frac{\mathrm{d}}{\mathrm{d}t}e^{-itH_2}\right) \\ &= (-iH_1)e^{-itH_1}e^{-itH_2} + e^{-itH_1}(-iH_2)e^{-itH_2} \\ &= e^{-itH_1}(-iH_1 - iH_2)e^{-itH_2} \\ &= e^{-itH_1}(-iH)e^{-itH_2}. \end{aligned} \tag{15.26}$$

Evaluating this at $t = 0$ yields $\widetilde{U}'(0) = -iH = U'(0)$, so the zeroth and first-order terms match the exact evolution. Differentiating a second time, we obtain

$$\begin{aligned} \frac{\mathrm{d}^2}{\mathrm{d}t^2}\widetilde{U}(t) &= e^{-itH_1}(-iH_1)(-iH)e^{-itH_2} + e^{-itH_1}(-iH)(-iH_2)e^{-itH_2} \\ &= -e^{-itH_1}(H_1^2 + H_1H_2 + H_1H_2 + H_2^2)e^{-itH_2} \\ &= -e^{-itH_1}(H_1^2 + 2H_1H_2 + H_2^2)e^{-itH_2}. \end{aligned} \tag{15.27}$$

The Taylor expansion for the approximation is thus

$$\widetilde{U}(t) = I - itH - \int_0^t e^{-isH_1}(H_1^2 + 2H_1H_2 + H_2^2)e^{-isH_2}(t-s)\,\mathrm{d}s. \tag{15.28}$$

Subtracting the two expansions, the difference is given by the remainder terms:

$$U(t) - \widetilde{U}(t) = \int_0^t \left[(-iH)^2 e^{-isH} + e^{-isH_1}(H_1^2 + 2H_1H_2 + H_2^2)e^{-isH_2}\right](t-s)\,\mathrm{d}s. \tag{15.29}$$

We bound the spectral norm of the integrands. Since e^{-isH}, e^{-isH_1}, and e^{-isH_2} are unitary, each has operator norm 1. For the exact term, $\left\|(-iH)^2 e^{-isH}\right\| \leq \|H\|^2 \leq |||H|||^2$. For the approximate term,

$$\left\|e^{-isH_1}(H_1^2 + 2H_1H_2 + H_2^2)e^{-isH_2}\right\| \leq \|H_1\|^2 + 2\|H_1\|\|H_2\| + \|H_2\|^2 = |||H|||^2. \tag{15.30}$$

Applying the triangle inequality to the integral yields

$$\left\|U(t) - \widetilde{U}(t)\right\| \leq \int_0^t \left(|||H|||^2 + |||H|||^2\right)(t-s)\,\mathrm{d}s = 2|||H|||^2\frac{t^2}{2} = t^2|||H|||^2, \tag{15.31}$$

which proves Eq. (15.22).

Finally, for the global simulation over time t, we apply the product formula L times with time step $\Delta t = t/L$. Using the triangle inequality over L steps, the total error is bounded by

$$\left\|U(t) - \widetilde{U}(\Delta t)^L\right\| \leq L\left\|U(\Delta t) - \widetilde{U}(\Delta t)\right\| \leq L(\Delta t)^2|||H|||^2 = \frac{t^2}{L}|||H|||^2. \tag{15.32}$$

Setting this bound equal to ϵ and solving for L gives the stated complexity. □

Example 15.6. The Hamiltonian for the TFIM with n sites is $H = H_1 + H_2$, where

$$H_1 = -\sum_{i=1}^{n-1} JZ_iZ_{i+1}, \quad H_2 = -\sum_{i=1}^{n} gX_i. \tag{15.33}$$

Since the operators within H_1 commute, and similarly the operators within H_2 commute, the corresponding exponential propagators can be exactly decomposed into a product of local unitaries:

$$e^{-itH_1} = e^{itJ\sum_{i=1}^{n-1} Z_iZ_{i+1}} = \prod_{i=1}^{n-1} e^{itJZ_iZ_{i+1}}, \tag{15.34}$$

$$e^{-itH_2} = e^{itg\sum_{i=1}^{n} X_i} = \prod_{i=1}^{n} e^{itgX_i}. \tag{15.35}$$

Each term $e^{itJZ_iZ_{i+1}}$ (a two-qubit rotation) and e^{itgX_i} (a single-qubit rotation) can be implemented independently, with no additional approximation error arising from the decomposition within H_1 or within H_2.

To estimate the complexity of the first-order Trotter simulation $S_1(t) = e^{-itH_1}e^{-itH_2}$, we bound $|||H|||$ by the sum of the norms of the individual terms:

$$|||H||| = \|H_1\| + \|H_2\| \leq \sum_{i=1}^{n-1}\|-JZ_iZ_{i+1}\| + \sum_{i=1}^{n}\|-gX_i\|. \tag{15.36}$$

Since $\|Z_iZ_{i+1}\| = 1$ and $\|X_i\| = 1$, assuming $J, g > 0$, we have

$$|||H||| \leq J(n-1) + gn = \mathcal{O}(n). \tag{15.37}$$

Therefore, according to the error estimate $\mathcal{O}(t^2|||H|||^2/L)$ derived in Proposition 15.5, the required number of Trotter steps L to achieve precision ϵ is $L = \mathcal{O}(|||H|||^2t^2/\epsilon)$. The total query complexity is $\mathcal{O}(L)$, which scales as $\mathcal{O}\left(\frac{n^2t^2}{\epsilon}\right)$. ◇

Example 15.7. The 1D Heisenberg model Hamiltonian is given by $H = J\sum_{i=1}^{n-1}(\mathbf{S}_i \cdot \mathbf{S}_{i+1})$, where $\mathbf{S}_i = (X_i, Y_i, Z_i)/2$, so that

$$H = \frac{J}{4}\sum_{i=1}^{n-1}\left(X_iX_{i+1} + Y_iY_{i+1} + Z_iZ_{i+1}\right). \tag{15.38}$$

We consider the common splitting $H = H_x + H_y + H_z$, where

$$H_x = \frac{J}{4}\sum_{i=1}^{n-1} X_iX_{i+1}, \quad H_y = \frac{J}{4}\sum_{i=1}^{n-1} Y_iY_{i+1}, \quad H_z = \frac{J}{4}\sum_{i=1}^{n-1} Z_iZ_{i+1}. \tag{15.39}$$

Here, the total number of terms is $\Gamma = 3$. The first-order Trotter product is $S_1(t) = e^{-itH_x}e^{-itH_y}e^{-itH_z}$. The local terms within H_x, H_y, and H_z commute, allowing for exact decomposition of the exponential propagators:

$$e^{-itH_x} = \prod_{i=1}^{n-1} e^{-it\frac{J}{4}X_iX_{i+1}}, \tag{15.40}$$

with similar expressions for e^{-itH_y} and e^{-itH_z}.

To estimate the query complexity, we compute the sum of the operator norms $|||H||| = \|H_x\| + \|H_y\| + \|H_z\|$. Since $\|X_iX_{i+1}\| = \|Y_iY_{i+1}\| = \|Z_iZ_{i+1}\| = 1$, and assuming $J > 0$:

$$|||H||| \leq \sum_{k\in\{x,y,z\}}\sum_{i=1}^{n-1}\left\|\frac{J}{4}\sigma_i^k\sigma_{i+1}^k\right\| = \frac{3J}{4}(n-1) = \mathcal{O}(n). \tag{15.41}$$

Consequently, the total query complexity for the 1D Heisenberg model scales as $\mathcal{O}\left(\frac{|||H|||^2t^2}{\epsilon}\right) = \mathcal{O}\left(\frac{n^2t^2}{\epsilon}\right)$.

Similarly, for the 2D Heisenberg model on an $n \times n$ square lattice ($N = n^2$ total qubits), the number of nearest-neighbor bonds (edges) is approximately $2N = 2n^2$. The operator norm is bounded by the total number of terms times the maximal norm of a single term. The norm sum scales linearly with the number of sites N:

$$|||H||| = \|H_x\| + \|H_y\| + \|H_z\| \leq \frac{3J}{4}(\text{Number of bonds}) \approx \frac{3J}{4}(2n^2) = \mathcal{O}(n^2). \tag{15.42}$$

The increased number of interactions leads to a larger norm sum. The query complexity then scales as $\mathcal{O}\left(\frac{|||H|||^2t^2}{\epsilon}\right) = \mathcal{O}\left(\frac{(n^2)^2t^2}{\epsilon}\right) = \mathcal{O}\left(\frac{n^4t^2}{\epsilon}\right)$. ◇

The cost of the first-order Trotter expansion is often undesirable as a function of the simulation parameters t, n, and ϵ. This scaling can be improved using the **second-order Trotter method**, also known as **symmetric Trotter splitting** or **Strang splitting**.

Proposition 15.8 (Second-order Trotter formula, $\Gamma = 2$)**.** *Let $H = H_1 + H_2$ where $H_1, H_2 \in \mathbb{C}^{N\times N}$ are Hermitian matrices. The second-order (symmetric) Trotter expansion is defined as*

$$\widetilde{U}_2(t) := e^{-i\frac{t}{2}H_2}e^{-itH_1}e^{-i\frac{t}{2}H_2}. \tag{15.43}$$

For any $t \geq 0$, the local simulation error satisfies the bound

$$\left\|e^{-itH} - \widetilde{U}_2(t)\right\| \leq \frac{t^3}{3}|||H|||^3. \tag{15.44}$$

Furthermore, for a simulation over a total time t divided into L segments of size $\Delta t = t/L$, the global error scales quadratically with the time step:

$$\left\| e^{-itH} - \left(\widetilde{U}_2(\Delta t)\right)^L \right\| = \mathcal{O}(t\Delta t^2 |||H|||^3). \tag{15.45}$$

Consequently, to achieve a precision ϵ, the required number of segments is $L = \mathcal{O}((|||H|||t)^{3/2}\epsilon^{-1/2})$, yielding a query complexity of $\mathcal{O}(L)$.

Proof. We first address the query complexity. While a single step Eq. (15.43) uses three exponentials, concatenating L steps allows adjacent half-steps to merge:

$$\left(e^{-i\frac{\Delta t}{2}H_2} e^{-i\Delta t H_1} e^{-i\frac{\Delta t}{2}H_2}\right)^L = e^{-i\frac{\Delta t}{2}H_2} \left(e^{-i\Delta t H_1} e^{-i\Delta t H_2}\right)^{L-1} e^{-i\Delta t H_1} e^{-i\frac{\Delta t}{2}H_2}. \tag{15.46}$$

The total number of exponentials is $2(L-1)+3 = 2L+1$, which is asymptotically equivalent to the first-order method.

To analyze the error, we employ the Taylor expansion with an integral remainder. The exact propagator expands as

$$U(t) = I - itH + \frac{(-itH)^2}{2} + \int_0^t (-iH)^3 e^{-isH} \frac{(t-s)^2}{2!}\, \mathrm{d}s. \tag{15.47}$$

We now compute the derivatives of $\widetilde{U}_2(t)$ at $t = 0$. For compactness, we use the multinomial coefficient notation:

$$\binom{m}{q_1, \ldots, q_k} := \frac{m!}{q_1! \cdots q_k!}. \tag{15.48}$$

The first derivative is

$$\left.\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{U}_2(t)\right|_{t=0} = -\frac{i}{2}H_2 - iH_1 - \frac{i}{2}H_2 = -iH, \tag{15.49}$$

which matches the first-order term of the exact propagator. For the second derivative, using the product rule on the three factors leads to a sum over indices $q_1 + q_2 + q_3 = 2$:

$$\left.\frac{\mathrm{d}^2}{\mathrm{d}t^2}\widetilde{U}_2(t)\right|_{t=0} = \sum_{q_1+q_2+q_3=2} \binom{2}{q_1, q_2, q_3} \left(-i\frac{H_2}{2}\right)^{q_1} (-iH_1)^{q_2} \left(-i\frac{H_2}{2}\right)^{q_3}. \tag{15.50}$$

Expanding the sum yields

$$\begin{aligned} \frac{\mathrm{d}^2}{\mathrm{d}t^2}\widetilde{U}_2(0) &= (-i)^2 \left[\left(\frac{H_2}{2}\right)^2 + H_1^2 + \left(\frac{H_2}{2}\right)^2 + 2\left(\frac{H_2}{2}\right)H_1 + 2H_1\left(\frac{H_2}{2}\right) + 2\left(\frac{H_2}{2}\right)\left(\frac{H_2}{2}\right)\right] \\ &= (-i)^2 \left[H_1^2 + H_1H_2 + H_2H_1 + H_2^2\right] = (-iH)^2. \end{aligned} \tag{15.51}$$

Thus, the second-order terms also match. The error is therefore controlled by the third derivative. The norm of the third derivative of the approximation is bounded by

$$\begin{aligned} \left\| \frac{\mathrm{d}^3}{\mathrm{d}t^3}\widetilde{U}_2(t) \right\| &\le \sum_{q_1+q_2+q_3=3} \binom{3}{q_1, q_2, q_3} \left(\frac{1}{2}\|H_2\|\right)^{q_1} \|H_1\|^{q_2} \left(\frac{1}{2}\|H_2\|\right)^{q_3} \\ &= \left(\|H_1\| + \frac{1}{2}\|H_2\| + \frac{1}{2}\|H_2\|\right)^3 = |||H|||^3. \end{aligned} \tag{15.52}$$

The remainder term for the difference $U(t) - \widetilde{U}_2(t)$ is bounded by the sum of the exact and approximate remainders:

$$\left\| U(t) - \widetilde{U}_2(t) \right\| \leq \int_0^t \left(\|H\|^3 + |||H|||^3 \right) \frac{(t-s)^2}{2!} \, \mathrm{d}s \leq 2|||H|||^3 \frac{t^3}{6} = \frac{t^3}{3} |||H|||^3. \tag{15.53}$$

Finally, using the triangle inequality over L steps, the global error is bounded by

$$L \times \mathcal{O}(\Delta t^3 |||H|||^3) = \mathcal{O}(t \Delta t^2 |||H|||^3). \tag{15.54}$$

Solving for L given a target error ϵ completes the proof. □

Example 15.9. Consider the Transverse Field Ising Model (TFIM) discussed in Example 15.6, where $|||H||| = \mathcal{O}(n)$. We can now quantitatively compare the cost of the first-order and second-order Trotter schemes.

For the first-order scheme, the query complexity was found to be

$$\mathcal{O}\left(\frac{n^2 t^2}{\epsilon} \right). \tag{15.55}$$

For the second-order scheme, substituting $|||H||| \sim n$ into the complexity bound derived in Proposition 15.8, we obtain

$$\mathcal{O}\left(\frac{n^{3/2} t^{3/2}}{\epsilon^{1/2}} \right). \tag{15.56}$$

The second-order method offers a substantial asymptotic improvement, and symmetric splitting is often a standard baseline for product-formula simulation. ◇

15.3. Higher order operator splitting formula

How do we generate high-order Trotter methods? One systematic approach is via **composition methods**. For $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$, one choice of a first-order Trotter method is

$$\widetilde{U}_1(t) := e^{-itH_\Gamma} \cdots e^{-itH_1}. \tag{15.57}$$

The corresponding **adjoint method** is

$$\widetilde{U}_1^\star(t) := (\widetilde{U}_1(-t))^\dagger = e^{-itH_1} \cdots e^{-itH_\Gamma}. \tag{15.58}$$

The second-order Trotter formula can be written as the composition of the first-order method and its adjoint:

$$\widetilde{U}_2(t) := \widetilde{U}_1^\star(t/2) \widetilde{U}_1(t/2). \tag{15.59}$$

If a method is equal to its adjoint, it is called symmetric. A symmetric method must have even order of accuracy [HLW06, Theorem 3.2]. Thus $\widetilde{U}_2(t)$ must be (at least) a second-order method, agreeing with the previous analysis.

The same viewpoint applies to general product formulas. Let $U(t) := e^{-itH}$. We say that a product formula $\widetilde{U}_p(t)$ has order p if the short-time error admits an expansion

$$\widetilde{U}_p(t) = U(t) + Ct^{p+1} + \mathcal{O}(t^{p+2}), \quad t \to 0, \tag{15.60}$$

for some bounded operator C. Given real coefficients $s_1, \ldots, s_m$, we define the corresponding composition product formula by

$$\widetilde{U}_p^{\mathrm{comp}}(t) := \widetilde{U}_p(s_m t) \cdots \widetilde{U}_p(s_1 t). \tag{15.61}$$

The order of accuracy for composition methods can be analyzed by Taylor expansion (see, e.g., [HLW06, Theorem 4.1]).

THEOREM 15.10. *Assume $\widetilde{U}_p(t)$ has order p in the sense of Eq.* (15.60). *If*

$$\sum_{j=1}^{m} s_j = 1, \qquad \sum_{j=1}^{m} s_j^{p+1} = 0, \tag{15.62}$$

then the composition method $\widetilde{U}_p^{\mathrm{comp}}(t)$ in Eq. (15.61) *has order at least $p+1$.*

PROOF. Write $U_j := U(s_j t)$ and $\widetilde{U}_j := \widetilde{U}_p(s_j t)$ for brevity. By Eq. (15.60),

$$\widetilde{U}_j = U_j + C(s_j t)^{p+1} + \mathcal{O}(t^{p+2}). \tag{15.63}$$

Using the group property $U(s_m t)\cdots U(s_1 t) = U\big((\sum_{j=1}^m s_j)t\big)$ together with $U_j = I + \mathcal{O}(t)$, we obtain

$$\widetilde{U}_p^{\mathrm{comp}}(t) = U(t) + Ct^{p+1}\sum_{j=1}^{m} s_j^{p+1} + \mathcal{O}(t^{p+2}). \tag{15.64}$$

Under Eq. (15.62), the t^{p+1} term vanishes, and therefore $\widetilde{U}_p^{\mathrm{comp}}(t) = U(t) + \mathcal{O}(t^{p+2})$, i.e., the composition has order at least $p+1$. □

Example 15.11 (Yoshida–Suzuki triple jump method)**.** In Eq. (15.62), the smallest m for which the system admits a solution is $m = 3$. We then have some freedom in solving the two equations. If we impose symmetry $s_1 = s_3$, then we obtain

$$s_1 = s_3 = \frac{1}{2 - 2^{1/(p+1)}}, \quad s_2 = -\frac{2^{1/(p+1)}}{2 - 2^{1/(p+1)}}. \tag{15.65}$$

This procedure can be repeated. Starting from a symmetric method of order 2, applying Eq. (15.65) with $p = 2$ yields a method of order at least 3; by symmetry of the coefficients, the resulting method is in fact of order 4. Repeating Eq. (15.65) with $p = 4$ yields a symmetric 9-stage composition method of order 6, then with $p = 6$ a 27-stage symmetric composition method of order 8, and so on. For an order-p method, the total number of steps is $3^{p/2-1}$. ◇

Example 15.12 (Trotter–Suzuki method)**.** Consider the case $m = 5$ with $s_1 = s_2 = s_4 = s_5$. The conditions in Eq. (15.62) become

$$4s_1 + s_3 = 1, \qquad 4s_1^{p+1} + s_3^{p+1} = 0. \tag{15.66}$$

The second equation implies $s_3 = -4^{1/(p+1)} s_1$. This gives the formula [Suz91]

$$s_1 = s_2 = s_4 = s_5 = \frac{1}{4 - 4^{1/(p+1)}}, \qquad s_3 = -\frac{4^{1/(p+1)}}{4 - 4^{1/(p+1)}}. \tag{15.67}$$

If the basic method $\widetilde{U}_p(t)$ is symmetric and the coefficients are chosen as above, then $\widetilde{U}_p^{\mathrm{comp}}(t)$ is also symmetric, and therefore its order must be even. In particular, if p is even, then "order at least $p+1$" implies order at least $p+2$.

Applying this construction to the second-order Trotter method $\widetilde{U}_2(t)$ yields the **Trotter–Suzuki formula**. It recursively defines

$$\widetilde{U}_{2k}(t) := \widetilde{U}_{2k-2}^2(s_k t)\,\widetilde{U}_{2k-2}((1-4s_k)t)\,\widetilde{U}_{2k-2}^2(s_k t), \qquad s_k := \frac{1}{4 - 4^{1/(2k-1)}}, \quad k \geq 2. \tag{15.68}$$

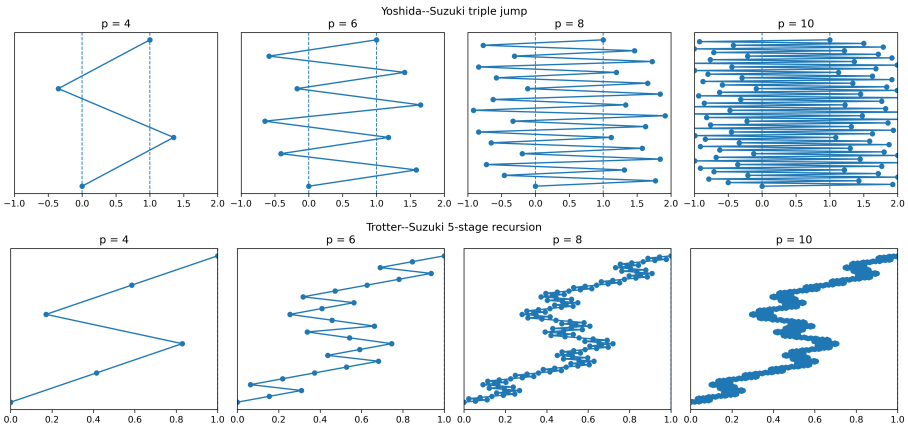


FIGURE 15.2. Comparison of steps of composition methods: the upper panel shows the Yoshida–Suzuki triple jump method, and the lower panel shows the Trotter–Suzuki 5-stage method.

Note that $\widetilde{U}_{2k}$ uses five applications of $\widetilde{U}_{2k-2}$, and hence can be written using at most 5^{k-1} applications of $\widetilde{U}_2(\cdot)$. The approximation order is $2k$, i.e.,

$$\left\| \widetilde{U}_{2k}(t) - e^{-iHt} \right\| = \mathcal{O}(t^{2k+1}), \quad t \to 0. \tag{15.69}$$

Expanding the recursion for $\widetilde{U}_{2k}(t)$ yields a composition method written directly in terms of $\widetilde{U}_2(\cdot)$. A comparison of the resulting step-size patterns for the Yoshida–Suzuki triple jump method and the Trotter–Suzuki method is shown in Fig. 15.2. The Yoshida–Suzuki coefficients oscillate between relatively large positive and negative values, which is often undesirable. By contrast, the 5-stage Trotter–Suzuki construction exhibits a pattern called "Suzuki fractals", with much smaller coefficient magnitudes. ◇

After expanding all compositions, an operator-splitting based simulation can be written as

$$\widetilde{U}(t) := \prod_{k=1}^{K} \prod_{\gamma=1}^{\Gamma} e^{-ita_{k,\gamma} H_{\pi_k(\gamma)}}, \tag{15.70}$$

where $a_{k,\gamma} \in \mathbb{R}$ and, for each k, π_k is a permutation of $\{1, \ldots, \Gamma\}$. In the Trotter–Suzuki formula above, $\widetilde{U}_{2k}(t)$ can be written in the form of Eq. (15.70) with $\Gamma = 2$ and $K = 2 \times 5^{k-1}$. We define the sum of operator norms:

$$|||H||| := \sum_{\gamma=1}^{\Gamma} \|H_\gamma\| . \tag{15.71}$$

The following result is similar to [CST+21, Lemma 1]. It provides an explicit constant without resorting to asymptotic notation.

THEOREM 15.13 (Sum of operator norm error scaling). *Let $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$ be a Hermitian operator acting on a finite-dimensional Hilbert space, and let $t \in \mathbb{R}$. Let $\widetilde{U}(t) = \prod_{k=1}^{K} \prod_{\gamma=1}^{\Gamma} e^{-ita_{k,\gamma} H_{\pi_k(\gamma)}}$ be a p-th order Trotter expansion with $|a_{k,\gamma}| \leq 1$. Then for any $t \geq 0$,*

(1) *The short-time error in the p-th order Trotter formula is*

$$\|\widetilde{U}(t) - e^{-itH}\| \leq \frac{t^{p+1}(K^{p+1}+1)}{(p+1)!}|||H|||^{p+1}. \tag{15.72}$$

(2) *We have that* $\|\widetilde{U}(t/L)^L - e^{-itH}\| \leq \epsilon$ *for the following choice of* L*:*

$$L = \left\lceil (tK|||H|||)^{1+\frac{1}{p}} \epsilon^{-\frac{1}{p}} \right\rceil. \tag{15.73}$$

The total number of operator exponentials of the form $e^{-ita_{k,\gamma}H_{\pi_k(\gamma)}}$ *required by the simulation is*

$$\Gamma KL = \Gamma \left\lceil K^{2+\frac{1}{p}} (t|||H|||)^{1+\frac{1}{p}} \epsilon^{-\frac{1}{p}} \right\rceil. \tag{15.74}$$

(3) *If we focus on the scaling with respect to* $t, |||H|||, \epsilon$*, then as* $p \to \infty$ *the gate complexity can be expressed as*

$$\mathcal{O}((t|||H|||)^{1+o(1)} \epsilon^{-o(1)}). \tag{15.75}$$

Proof. Define $F(t) := \widetilde{U}(t) - e^{-itH}$. Since $\widetilde{U}(t)$ is a p-th order formula, we have $F^{(j)}(0) = 0$ for all $0 \leq j \leq p$. Then by Taylor's theorem,

$$\widetilde{U}(t) - e^{-itH} = \frac{t^{p+1}}{p!} \int_0^1 \mathrm{d}u\, (1-u)^p \left(\widetilde{U}^{(p+1)}(ut) - (-iH)^{p+1} e^{-iutH} \right), \tag{15.76}$$

where

$$\widetilde{U}^{(p+1)}(ut) = \sum_{q_{1,1}+\cdots+q_{K,\Gamma}=p+1} \binom{p+1}{q_{1,1},\ldots,q_{K,\Gamma}} \prod_{k=1}^{K} \prod_{\gamma=1}^{\Gamma} (-ia_{k,\gamma} H_{\pi_k(\gamma)})^{q_{k,\gamma}} e^{-iuta_{k,\gamma}H_{\pi_k(\gamma)}}. \tag{15.77}$$

Using $|a_{k,\gamma}| \leq 1$ for any k, γ, we bound

$$\begin{aligned} \|\widetilde{U}^{(p+1)}(ut)\| &\leq \sum_{q_{1,1}+\cdots+q_{K,\Gamma}=p+1} \binom{p+1}{q_{1,1},\ldots,q_{K,\Gamma}} \prod_{k=1}^{K} \prod_{\gamma=1}^{\Gamma} \|H_{\pi_k(\gamma)}\|^{q_{k,\gamma}} \\ &= \left(\sum_{k=1}^{K} \sum_{\gamma=1}^{\Gamma} \|H_{\pi_k(\gamma)}\| \right)^{p+1} = (K|||H|||)^{p+1}. \end{aligned} \tag{15.78}$$

Therefore

$$\|\widetilde{U}(t) - e^{-itH}\| \leq \frac{t^{p+1}}{(p+1)!} \left[(K|||H|||)^{p+1} + |||H|||^{p+1} \right]. \tag{15.79}$$

This proves Eq. (15.72).

We now use Stirling's approximation,

$$\sqrt{2\pi} n^n \sqrt{n} \leq e^n n! \leq e n^n \sqrt{n}. \tag{15.80}$$

which in particular implies

$$n^n \leq e^n n! \leq e^n n^n. \tag{15.81}$$

For long-time simulation, we choose

$$L \frac{(t/L)^{p+1}(K^{p+1}+1)}{(p+1)!} |||H|||^{p+1} = \epsilon, \tag{15.82}$$

which gives

$$
\begin{aligned}
L =& \frac{(K^{p+1}+1)^{\frac{1}{p}}\,(t|||H|||)^{1+\frac{1}{p}}}{\epsilon^{\frac{1}{p}}\,((p+1)!)^{\frac{1}{p}}} \leq \left(\frac{K^{p+1}+1}{\sqrt{2\pi(p+1)}}\right)^{\frac{1}{p}} \left(\frac{e}{p+1}\right)^{1+\frac{1}{p}} (t|||H|||)^{1+\frac{1}{p}}\,\epsilon^{-\frac{1}{p}} \\
&\leq (tK|||H|||)^{1+\frac{1}{p}}\,\epsilon^{-\frac{1}{p}}.
\end{aligned} \tag{15.83}
$$

In the first inequality we used Stirling's approximation, and in the second inequality we used the fact that

$$
\left(\frac{2}{\sqrt{2\pi(p+1)}}\right)^{\frac{1}{p}} \left(\frac{e}{p+1}\right)^{1+\frac{1}{p}} \leq 1, \quad p \geq 1, \tag{15.84}
$$

which can be verified through direct computation. Finally, since $\widetilde{U}(t/L)$ and $e^{-itH/L}$ are unitary, we have

$$
\|\widetilde{U}(t/L)^L - e^{-itH}\| \leq L\|\widetilde{U}(t/L) - e^{-itH/L}\|, \tag{15.85}
$$

which justifies the choice of L above. This proves Eq. (15.73) and Eq. (15.74). □

When $p \to \infty$, the scaling $\mathcal{O}((t|||H|||)^{1+o(1)}\,\epsilon^{-o(1)})$ is near optimal with respect to $t, |||H|||, \epsilon$. Plugging in $p = 1, 2$, we also recover the previous results on first and second order Trotter methods.

Example 15.14 (Optimizing the cost of high-order Trotter formula)**.** Should p be chosen to be as large as possible? To answer this question, we need to take into account the preconstant $K^{2+\frac{1}{p}}$ in the gate complexity. For the Trotter–Suzuki formula of even order p, we have $K = 2 \times 5^{p/2-1}$. Therefore $K^{2+\frac{1}{p}} \leq 5^p$. An upper bound on the total number of gates is then

$$
\Gamma K^{2+\frac{1}{p}}\,(t|||H|||)^{1+\frac{1}{p}}\,\epsilon^{-\frac{1}{p}} \leq \Gamma t|||H|||5^p(t|||H|||\epsilon^{-1})^{\frac{1}{p}} \tag{15.86}
$$

For a fixed problem, we can assume that $t, |||H|||$ are fixed. Then the optimal value of p should be determined by the solution of the minimization problem

$$
\min_{p\geq 1} \quad 5^p(t|||H|||\epsilon^{-1})^{\frac{1}{p}}. \tag{15.87}
$$

Treating p as a continuous variable, the minimizer is

$$
p = \max\left\{1, \sqrt{\frac{\log(t|||H|||\epsilon^{-1})}{\log 5}}\right\}. \tag{15.88}
$$

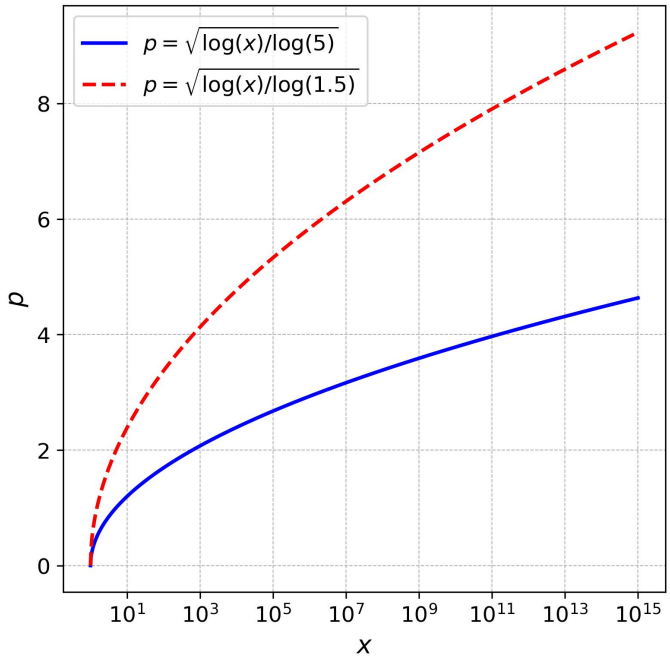


FIGURE 15.3. Optimal order p versus $x = t|||H|||\epsilon^{-1}$, assuming $K^{2+\frac{1}{p}}$ is upper bounded by 5^p and 1.5^p, respectively.

One should note that the function $\sqrt{\log(x)}$ grows very slowly as a function of x. Given $x = t|||H|||\epsilon^{-1}$, the optimal Trotter order p can be determined as shown in Fig. 15.3. The graph illustrates that the second-order Trotter method is optimal when $x \leq 10^3$, while the fourth-order method remains optimal up to $x \leq 10^{11}$. In practical applications, high-order methods can be more competitive than suggested by the above analysis, especially when more efficient methods than the Trotter–Suzuki decomposition are applied. Fig. 15.3 shows that if the factor $K^{2+\frac{1}{p}}$ can be replaced by a more modest growth 1.5^p, then the 6-th order and 8-th order methods become competitive when $t|||H|||\epsilon^{-1}$ reaches 10^7 and 10^{11}, respectively. Therefore very high-order Trotter methods provide theoretical insights, but they are often less practical in real-world situations.

◇

15.4. Commutator error bound and vector norm error bound

The error analysis in Proposition 15.5 relied on the sum of operator norms $|||H|||$. This yields a worst-case bound $\mathcal{O}(t^2|||H|||^2)$, but it does not reflect an important structural feature: if H_1 and H_2 commute (i.e., $[H_1, H_2] = 0$), then the first-order Trotter splitting is exact for all t.

To see the origin of this phenomenon, compare the Taylor expansions of the approximate and exact propagators. A direct calculation shows that the first nonzero term in the difference occurs at order t^2:

$$e^{-itH_1}e^{-itH_2} - e^{-it(H_1+H_2)} = -\frac{t^2}{2}[H_1, H_2] + \mathcal{O}(t^3). \tag{15.89}$$

This indicates that the local error is controlled by the commutator, a feature known as **commutator error scaling**. To prove a rigorous bound proportional to $\|[H_1, H_2]\|$, we first state a lemma regarding the time evolution of conjugated operators.

Lemma 15.15. *Let $A, B \in \mathbb{C}^{N\times N}$ be Hermitian matrices. Then*

$$e^{-itA} B e^{itA} = B - i \int_0^t e^{-isA}[A, B] e^{isA} \, \mathrm{d}s. \tag{15.90}$$

Consequently, the difference between a conjugated operator and the operator itself is bounded by

$$\left\| e^{-itA} B e^{itA} - B \right\| \leq t \, \|[A, B]\| . \tag{15.91}$$

Proof. Define the matrix-valued function $f(t) := e^{-itA} B e^{itA}$. Differentiating with respect to t, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} f(t) = -iA e^{-itA} B e^{itA} + i e^{-itA} B A e^{itA} = -i e^{-itA}[A, B] e^{itA}. \tag{15.92}$$

Applying the fundamental theorem of calculus, $f(t) = f(0) + \int_0^t f'(s) \, \mathrm{d}s$, yields Eq. (15.90). Taking the spectral norm of both sides and using the unitarity of e^{-isA} gives the stated bound. □

We can now improve the error analysis for the first-order Trotter formula.

Proposition 15.16 (Commutator scaling of first-order Trotter error)**.** *Let $H = H_1 + H_2$ with H_1, H_2 Hermitian. For the first-order Trotter approximation $\widetilde{U}(t) = e^{-itH_1} e^{-itH_2}$, the error is bounded by*

$$\left\| \widetilde{U}(t) - e^{-itH} \right\| \leq \frac{t^2}{2} \, \|[H_1, H_2]\| . \tag{15.93}$$

Proof. We begin by determining the differential equation satisfied by the approximate propagator $\widetilde{U}(t)$. Differentiating with respect to time gives

$$\begin{aligned} i\partial_t \widetilde{U}(t) &= \left(i\partial_t e^{-itH_1}\right) e^{-itH_2} + e^{-itH_1} \left(i\partial_t e^{-itH_2}\right) \\ &= H_1 e^{-itH_1} e^{-itH_2} + e^{-itH_1} H_2 e^{-itH_2}. \end{aligned} \tag{15.94}$$

We compare this to the ideal evolution $i\partial_t U(t) = (H_1 + H_2) U(t)$. Rearranging to isolate $H = H_1 + H_2$ yields

$$\begin{aligned} i\partial_t \widetilde{U}(t) &= (H_1 + H_2)\widetilde{U}(t) + e^{-itH_1} H_2 e^{-itH_2} - H_2 e^{-itH_1} e^{-itH_2} \\ &= H\widetilde{U}(t) + [e^{-itH_1}, H_2] e^{-itH_2}. \end{aligned} \tag{15.95}$$

Thus $\widetilde{U}(t)$ satisfies an inhomogeneous Schrödinger equation with inhomogeneity $B(t) := [e^{-itH_1}, H_2] e^{-itH_2}$. Applying Duhamel's principle (Proposition 3.22), we obtain

$$\left\| \widetilde{U}(t) - U(t) \right\| \leq \int_0^t \|B(s)\| \, \mathrm{d}s = \int_0^t \left\| [e^{-isH_1}, H_2] \right\| \mathrm{d}s. \tag{15.96}$$

To bound the integrand, we observe that the norm is invariant under unitary multiplication. Multiplying the term inside the norm by e^{isH_1} on the left gives

$$\left\| [e^{-isH_1}, H_2] \right\| = \left\| e^{isH_1} H_2 e^{-isH_1} - H_2 \right\| . \tag{15.97}$$

Using Lemma 15.15 with $A = -H_1$ and $B = H_2$, we have

$$\left\| e^{isH_1} H_2 e^{-isH_1} - H_2 \right\| \leq s \, \|[H_1, H_2]\| . \tag{15.98}$$

Substituting this back into the integral yields the final result:

$$\left\|\widetilde{U}(t) - U(t)\right\| \leq \int_0^t s \left\|[H_1, H_2]\right\| \mathrm{d}s = \frac{t^2}{2} \left\|[H_1, H_2]\right\|. \tag{15.99}$$

□

The commutator error bound implies the operator norm bound. Indeed,

$$\left\|[H_1, H_2]\right\| \leq 2 \left\|H_1\right\| \left\|H_2\right\| \leq (\left\|H_1\right\| + \left\|H_2\right\|)^2 = \left\|\!\left|H\right|\!\right\|^2, \tag{15.100}$$

which is sharper than the bound in proved in Proposition 15.5 by a factor of 2. Furthermore, commutator type error bounds can be substantially sharper than the operator norm bounds when the terms are close to commuting.

Example 15.17 (Backward error analysis and modified Hamiltonian)**.** Standard forward error analysis bounds the distance between the exact and approximate propagators. Alternatively, **backward error analysis** asks (recall Section 7.1): For what Hamiltonian $H_{\mathrm{mod}}(t)$ is the approximate propagator $\widetilde{U}(t)$ the **exact** solution?

Consider the first-order Trotter splitting $\widetilde{U}(t) = e^{-itH_1}e^{-itH_2}$. We determine the differential equation satisfied by $\widetilde{U}(t)$ by differentiating with respect to time:

$$\begin{aligned} i\partial_t \widetilde{U}(t) &= \left(i\partial_t e^{-itH_1}\right) e^{-itH_2} + e^{-itH_1}\left(i\partial_t e^{-itH_2}\right) \\ &= H_1 e^{-itH_1}e^{-itH_2} + e^{-itH_1}H_2 e^{-itH_2} \\ &= \left(H_1 + e^{-itH_1}H_2 e^{itH_1}\right)\widetilde{U}(t). \end{aligned} \tag{15.101}$$

Thus, $\widetilde{U}(t)$ exactly solves the Schrödinger equation $i\partial_t\psi(t) = H_{\mathrm{mod}}(t)\psi(t)$ governed by the time-dependent **modified Hamiltonian**:

$$H_{\mathrm{mod}}(t) = H_1 + e^{-itH_1}H_2 e^{itH_1} = H + \delta H(t), \tag{15.102}$$

where the deviation from the target Hamiltonian is $\delta H(t) = e^{-itH_1}H_2 e^{itH_1} - H_2$.

Using the conjugation expansion derived in Lemma 15.15, we can expand the backward error term $\delta H(t)$ for small t:

$$\delta H(t) = -it[H_1, H_2] + \mathcal{O}(t^2). \tag{15.103}$$

This result offers a physical intuition for the Trotter error: the simulation implements a Hamiltonian that drifts linearly away from H with time. The direction of the drift is determined by the commutator $[H_1, H_2]$.

We can relate this back to the forward error using the Magnus expansion. The propagator generated by $H_{\mathrm{mod}}(t)$ can be approximated by the exponential of the average Hamiltonian:

$$\widetilde{U}(t) \approx \exp\left(-i\int_0^t H_{\mathrm{mod}}(s)\,\mathrm{d}s\right) = \exp\left(-itH - i\int_0^t \delta H(s)\,\mathrm{d}s\right). \tag{15.104}$$

Integrating the leading order term of the backward error yields

$$\int_0^t \delta H(s)\,\mathrm{d}s \approx \int_0^t -is[H_1, H_2]\,\mathrm{d}s = -\frac{it^2}{2}[H_1, H_2]. \tag{15.105}$$

This recovers the quadratic commutator scaling $\mathcal{O}(t^2 \left\|[H_1, H_2]\right\|)$ found in the forward error analysis (Proposition 15.16). ◇

Exercise 15.1. Consider the Hamiltonian simulation problem for $H = H_1 + H_2 + H_3$. Show that the first order Trotter formula

$$\widetilde{U}(t) = e^{-itH_1}e^{-itH_2}e^{-itH_3}$$

has a commutator type error bound

$$\left\|\widetilde{U}(t) - U(t)\right\| \leq \frac{t^2}{2}(\|[H_1, H_2]\| + \|[H_1, H_3]\| + \|[H_2, H_3]\|) = \frac{t^2}{4}\sum_{\gamma_1,\gamma_2=1}^{3}\|[H_{\gamma_1}, H_{\gamma_2}]\|. \tag{15.106}$$

Example 15.18. For the 1D TFIM model with nearest neighbor interactions, the commutator $[Z_iZ_j, X_k]$ is non-zero only when $k = i$ or $k = j$. So the norm of the commutator $\|[H_1, H_2]\|$ scales linearly with the system size, i.e., $\|[H_1, H_2]\| = \mathcal{O}(n)$. Therefore, to achieve a precision of ϵ for constant simulation time t, using the first order Trotter method, the scaling of the total number of time steps L relative to the system size is $\mathcal{O}(n^2/\epsilon)$ if we use the estimate based on the operator norm. However, the scaling is only $\mathcal{O}(n/\epsilon)$ if we use the estimate based on the commutator norm. ⋄

Example 15.19. Consider the 1D Hubbard model, the Hamiltonian can be written in the form $H = H_1 + H_2$, where H_1 represents the kinetic energy of the electrons and H_2 describes the on-site interaction. The norms of H_1 and H_2 are proportional to the system size, thus $\|H_1\|, \|H_2\| = \mathcal{O}(n)$. However, since the commutator $[H_1, H_2]$ is non-zero only for terms where the kinetic and interaction energies correspond to the same site, we have $\|[H_1, H_2]\| = \mathcal{O}(n)$. Therefore, to achieve a precision of ϵ for constant simulation time t, using the first order Trotter method, the scaling of the total number of time steps L with respect to the system size is $\mathcal{O}(n^2/\epsilon)$ if we use the estimate based on the operator norm, but it is only $\mathcal{O}(n/\epsilon)$ if we use the estimate based on the commutator norm. ⋄

The commutator viewpoint extends to higher-order product formulas. We record the following result, parallel to Theorem 15.13, without proof; see [CST+21, Theorem 6].

THEOREM 15.20 (Commutator error scaling of Trotter expansion). *Let $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$ with H_γ Hermitian and $t \in \mathbb{R}$. Let $\widetilde{U}(t) = \prod_{k=1}^{K}\prod_{\gamma=1}^{\Gamma} e^{-ita_{k,\gamma}H_{\pi_k(\gamma)}}$ be a p-th order Trotter expansion, and assume $|a_{k,\gamma}| \leq 1$ for all k, γ. Define*

$$|||H|||_{p+1,\mathrm{comm}} = \sum_{\gamma_1,\gamma_2,\ldots,\gamma_{p+1}=1}^{\Gamma} \left\|\left[H_{\gamma_{p+1}}, \left[H_{\gamma_p}, \cdots [H_{\gamma_2}, H_{\gamma_1}]\cdots\right]\right]\right\|. \tag{15.107}$$

In particular, $|||H|||_{p+1,\mathrm{comm}} = \mathcal{O}\big(2^p\Gamma^{p+1}(\max_j \|H_j\|)^{p+1}\big)$. Then for short $t \geq 0$,

$$\|\widetilde{U}(t) - e^{-itH}\| = \mathcal{O}\left(t^{p+1}K^{p+1}|||H|||_{p+1,\mathrm{comm}}\right). \tag{15.108}$$

For long time simulation, to reach precision ϵ, we can choose the number of segments

$$L = \mathcal{O}\left((tK)^{1+\frac{1}{p}}\,|||H|||_{p+1,\mathrm{comm}}^{\frac{1}{p}}\epsilon^{-\frac{1}{p}}\right), \tag{15.109}$$

and the number of gates is

$$\Gamma KL = \mathcal{O}\left(\Gamma K^{2+\frac{1}{p}}t^{1+\frac{1}{p}}|||H|||_{p+1,\mathrm{comm}}^{\frac{1}{p}}\epsilon^{-\frac{1}{p}}\right). \tag{15.110}$$

The commutator bounds above are operator-norm statements. Since the operator norm is defined by $\|A\| = \sup_{\|\psi\|=1}\|A\psi\|$, it captures a worst-case error over all states. In many applications one is instead interested in a fixed initial state $\psi(0)$; then the state-dependent error

$\left\|(U(t)-\widetilde{U}(t))\psi(0)\right\|$ can be much smaller than $\left\|U(t)-\widetilde{U}(t)\right\|$. This phenomenon is known as **vector norm scaling**.

Proposition 15.21 (Vector norm scaling for first-order Trotter)**.** *Let $H = H_1 + H_2$ with H_1, H_2 Hermitian, and let $U(t) = e^{-itH}$ and $\widetilde{U}(t) = e^{-itH_1}e^{-itH_2}$. Then for any $\psi(0) \in \mathbb{C}^N$ and any $t \geq 0$,*

$$\begin{aligned}\left\|\widetilde{U}(t)\psi(0) - U(t)\psi(0)\right\| &\leq \int_0^t \int_0^s \left\|[H_1, H_2]e^{-iuH_1}e^{-isH_2}\psi(0)\right\| \, \mathrm{d}u \, \mathrm{d}s \\ &\leq \frac{t^2}{2} \sup_{0\leq u\leq s\leq t} \left\|[H_1, H_2]e^{-iuH_1}e^{-isH_2}\psi(0)\right\|.\end{aligned} \tag{15.111}$$

In particular, since $\left\|[H_1, H_2]e^{-iuH_1}e^{-isH_2}\psi(0)\right\| \leq \|[H_1, H_2]\| \, \|\psi(0)\|$, this implies the operator-norm commutator bound $\left\|\widetilde{U}(t) - U(t)\right\| \leq \frac{t^2}{2}\|[H_1, H_2]\|$.

Proof. As in Proposition 15.16, applying Duhamel's principle to the inhomogeneous equation satisfied by $\widetilde{U}(t)\psi(0)$ yields

$$\left\|\widetilde{U}(t)\psi(0) - U(t)\psi(0)\right\| \leq \int_0^t \left\|[e^{-isH_1}, H_2]e^{-isH_2}\psi(0)\right\| \, \mathrm{d}s. \tag{15.112}$$

Using unitary invariance of the norm, we multiply by e^{isH_1} on the left and obtain

$$\left\|[e^{-isH_1}, H_2]e^{-isH_2}\psi(0)\right\| = \left\|e^{isH_1}[e^{-isH_1}, H_2]e^{-isH_2}\psi(0)\right\|. \tag{15.113}$$

Since $e^{isH_1}[e^{-isH_1}, H_2] = H_2 - e^{isH_1}H_2e^{-isH_1} = -(e^{isH_1}H_2e^{-isH_1} - H_2)$, this implies

$$\left\|[e^{-isH_1}, H_2]e^{-isH_2}\psi(0)\right\| = \left\|(e^{isH_1}H_2e^{-isH_1} - H_2)e^{-isH_2}\psi(0)\right\|. \tag{15.114}$$

Finally, applying Eq. (15.90) with $A = -H_1$ and $B = H_2$ gives

$$e^{isH_1}H_2e^{-isH_1} - H_2 = i\int_0^s e^{iuH_1}[H_1, H_2]e^{-iuH_1} \, \mathrm{d}u. \tag{15.115}$$

Therefore,

$$\left\|[e^{-isH_1}, H_2]e^{-isH_2}\psi(0)\right\| \leq \int_0^s \left\|[H_1, H_2]e^{-iuH_1}e^{-isH_2}\psi(0)\right\| \, \mathrm{d}u, \tag{15.116}$$

and substituting into the previous bound yields the stated double-integral estimate. The supremum bound follows from $\int_0^t \int_0^s \mathrm{d}u \, \mathrm{d}s = t^2/2$. □

Vector norm scaling is particularly useful when the Hamiltonian involves unbounded operators, such as differential operators in the Schrödinger equation. In such cases, the operator norm may diverge or scale poorly with discretization parameters, while the vector norm error can remain well behaved for sufficiently smooth states.

Example 15.22 (Application to the first-quantized Schrödinger equation)**.** Consider a particle in a one-dimensional potential $V(x)$ on the domain $\Omega = [0, 1]$ with periodic boundary conditions. The Hamiltonian is $H = H_1 + H_2$, where $H_1 = -\partial_x^2$ (kinetic energy) and $H_2 = V(x)$ (potential energy). We discretize the domain using a uniform grid of N points.

Using the operator norm estimates from previous sections, we have $\|H_1\| = \mathcal{O}(N^2)$ and $\|H_2\| = \mathcal{O}(1)$. The standard Trotter error bound would imply a complexity scaling of:

$$L_{\mathrm{op}} = \mathcal{O}\left(\frac{t^2\|\|H\|\|^2}{\epsilon}\right) = \mathcal{O}\left(\frac{t^2N^4}{\epsilon}\right), \tag{15.117}$$

which is prohibitively expensive for fine discretizations.

However, utilizing Proposition 15.21, we examine the commutator term acting on wavefunctions of the form $e^{-iuH_1}e^{-isH_2}\psi(0)$. For any smooth function $\psi(x)$,

$$[H_1, H_2]\psi = [-\partial_x^2, V]\psi = -\partial_x^2(V\psi) + V\partial_x^2\psi = -(V''\psi + 2V'\psi'). \tag{15.118}$$

In the non-asymptotic bound in Proposition 15.21, the commutator acts on split-evolved states

$$\phi_{u,s} := e^{-iuH_1}e^{-isH_2}\psi(0), \qquad 0 \le u \le s \le t. \tag{15.119}$$

Using the explicit commutator expression, we obtain

$$\|[H_1, H_2]\phi_{u,s}\| \le \|V''\|_\infty \|\phi_{u,s}\| + 2\|V'\|_\infty \left\|\phi'_{u,s}\right\|. \tag{15.120}$$

Moreover, under periodic boundary conditions, e^{-iuH_1} is a Fourier multiplier and commutes with ∂_x, so $\|\phi_{u,s}\| = \left\|e^{-isV}\psi(0)\right\| = \|\psi(0)\|$ and $\left\|\phi'_{u,s}\right\| = \left\|(e^{-isV}\psi(0))'\right\|$. Since

$$(e^{-isV}\psi(0))' = e^{-isV}(\psi'(0) - isV'\psi(0)), \tag{15.121}$$

we have the bound

$$\left\|\phi'_{u,s}\right\| \le \|\psi'(0)\| + s\|V'\|_\infty \|\psi(0)\|. \tag{15.122}$$

The bound above is written in terms of $\psi'(0)$. To obtain a statement that depends on the regularity of the **true** solution at all times, let $\psi(s) := e^{-isH}\psi(0)$ denote the exact Schrödinger evolution. Assume that $\sup_{0\le s\le t}\|\psi'(s)\|$ is finite and does not scale with the discretization size N (for instance, this holds if the initial state has bounded H^1-regularity and the potential is sufficiently regular).

To extract the global complexity scaling, apply first-order splitting with L steps of size $\Delta t = t/L$. Let $U_{\Delta t} := e^{-i\Delta t H}$ and $\widetilde{U}_{\Delta t} := e^{-i\Delta t H_1}e^{-i\Delta t H_2}$. For any initial state $\psi(0)$, a telescoping expansion gives

$$\widetilde{U}_{\Delta t}^L\psi(0) - U_{\Delta t}^L\psi(0) = \sum_{j=0}^{L-1} \widetilde{U}_{\Delta t}^{L-1-j}\big(\widetilde{U}_{\Delta t} - U_{\Delta t}\big)U_{\Delta t}^j\psi(0). \tag{15.123}$$

Since both $\widetilde{U}_{\Delta t}$ and $U_{\Delta t}$ are unitary, taking norms yields

$$\left\|\widetilde{U}_{\Delta t}^L\psi(0) - U_{\Delta t}^L\psi(0)\right\| \le \sum_{j=0}^{L-1}\left\|\big(\widetilde{U}_{\Delta t} - U_{\Delta t}\big)\psi(t_j)\right\|, \qquad \psi(t_j) := U_{\Delta t}^j\psi(0). \tag{15.124}$$

Thus it suffices to bound the one-step (local) error applied to the exact state at the beginning of each step.

We now apply Proposition 15.21 to a single step with time horizon Δt and initial state $\psi(t_j)$. In the commutator term we encounter

$$e^{-iuH_1}e^{-isH_2}\psi(t_j), \qquad 0 \le u \le s \le \Delta t. \tag{15.125}$$

Using periodic boundary conditions as above, e^{-iuH_1} commutes with ∂_x, and we obtain

$$\left\|\partial_x\big(e^{-iuH_1}e^{-isH_2}\psi(t_j)\big)\right\| = \left\|\partial_x\big(e^{-isV}\psi(t_j)\big)\right\| \le \|\psi'(t_j)\| + s\|V'\|_\infty\|\psi(t_j)\|. \tag{15.126}$$

Since $\|\psi(t_j)\| = \|\psi(0)\|$ and $s \le \Delta t$, for $\Delta t \le 1$ we can bound the right-hand side by

$$\|\psi'(t_j)\| + \|V'\|_\infty\|\psi(0)\| \le \sup_{0\le s\le t}\|\psi'(s)\| + \|V'\|_\infty\|\psi(0)\|. \tag{15.127}$$

Therefore the supremum term in Proposition 15.21 (with t replaced by Δt) is bounded by a constant depending on $\sup_{0\le s\le t}\|\psi'(s)\|$, $\|V'\|_\infty$, and $\|V''\|_\infty$, but not on N. This yields a local error per

step of order $\mathcal{O}(\Delta t^2)$ and hence a global error of order $\mathcal{O}(L\Delta t^2) = \mathcal{O}(t\Delta t) = \mathcal{O}(t^2/L)$. Choosing $L = \mathcal{O}(t^2/\epsilon)$ yields

$$L_{\text{vec}} = \mathcal{O}\left(\frac{t^2}{\epsilon}\right). \tag{15.128}$$

This result demonstrates that the number of time steps required is independent of the discretization parameter N. This explains the practical efficiency of splitting methods for quantum dynamics. ⋄

15.5. Operator splitting with randomized Hamiltonian evolution time

The Hamiltonian simulation problem can be equivalently formulated within the density matrix formalism using the Liouville–von Neumann equation

$$\partial_t \rho(t) = -i[H, \rho(t)] \coloneqq \mathcal{L}(\rho(t)), \tag{15.129}$$

where the linear map $\mathcal{L} : \mathbb{C}^{N\times N} \to \mathbb{C}^{N\times N}$ is known as the **Liouvillian**. The solution to the von Neumann equation admits a closed-form expression given by the exponential of the Liouvillian:

$$\rho(t) = e^{t\mathcal{L}}(\rho(0)) = \sum_{k=0}^{\infty} \frac{t^k \mathcal{L}^k}{k!}(\rho(0)). \tag{15.130}$$

Proposition 15.23 (Bounds on the Liouvillian)**.** *Let $H \in \mathbb{C}^{N\times N}$ be a Hermitian matrix and let $\mathcal{L}(\cdot) = -i[H, \cdot]$ be the associated Liouvillian. Then the induced trace norm and the diamond norm satisfy*

$$\|\mathcal{L}\|_{1\to 1} \le 2\|H\| \quad \textit{and} \quad \|\mathcal{L}\|_{\diamond} \le 2\|H\|. \tag{15.131}$$

Proof. First, we consider the induced trace norm defined as $\|\mathcal{L}\|_{1\to 1} := \sup_{\|X\|_1=1} \|\mathcal{L}(X)\|_1$. Let $X \in \mathbb{C}^{N\times N}$ be an arbitrary matrix. By the triangle inequality and the definition of the commutator, we have

$$\|\mathcal{L}(X)\|_1 = \|-i(HX - XH)\|_1 \le \|HX\|_1 + \|XH\|_1. \tag{15.132}$$

We apply Hölder's inequality for Schatten norms, specifically $\|AB\|_1 \le \|A\|\,\|B\|_1$ (where $\|\cdot\|$ is the spectral norm and $\|\cdot\|_1$ is the trace norm). This yields

$$\|HX\|_1 \le \|H\|\,\|X\|_1 \quad \text{and} \quad \|XH\|_1 \le \|X\|_1\,\|H\|. \tag{15.133}$$

Summing these gives

$$\|\mathcal{L}(X)\|_1 \le 2\|H\|\,\|X\|_1. \tag{15.134}$$

Taking the supremum over all X with unit trace norm proves $\|\mathcal{L}\|_{1\to 1} \le 2\|H\|$.

Recall that the diamond norm is defined as $\|\mathcal{L}\|_{\diamond} := \sup_{k\ge 1} \|\mathcal{L} \otimes \mathcal{I}_k\|_{1\to 1}$, where $\mathcal{I}_k$ is the identity map on an ancillary system of dimension k. For any dimension k and any extended matrix $X_{\text{ext}} \in \mathbb{C}^{Nk\times Nk}$, the action of the extended map is

$$(\mathcal{L} \otimes \mathcal{I}_k)(X_{\text{ext}}) = -i(H \otimes I_k)X_{\text{ext}} + iX_{\text{ext}}(H \otimes I_k) = -i[H \otimes I_k, X_{\text{ext}}]. \tag{15.135}$$

This has the exact same commutator form as the original map, but with the Hamiltonian $\widetilde{H} = H \otimes I_k$. We can therefore apply the result derived in the first part of the proof directly to $\widetilde{H}$:

$$\|\mathcal{L} \otimes \mathcal{I}_k\|_{1\to 1} \le 2\|H \otimes I_k\|. \tag{15.136}$$

Since the spectral norm is multiplicative under tensor products with the identity, we have $\|H \otimes I_k\| = \|H\| \|I_k\| = \|H\|$. Thus, for any k,

$$\|\mathcal{L} \otimes \mathcal{I}_k\|_{1\to 1} \leq 2 \|H\| . \tag{15.137}$$

Taking the supremum over k confirms that $\|\mathcal{L}\|_\diamond \leq 2 \|H\|$. □

The **quantum stochastic drift protocol** (qDRIFT) [Cam19] provides a distinct approach to Hamiltonian simulation, particularly effective when the Hamiltonian is composed of a large number of terms. We consider a Hamiltonian H expressed as a linear combination $H = \sum_{\gamma=1}^{\Gamma} c_\gamma H_\gamma$. We assume the standard setting where H is Hermitian, the coefficients c_γ are positive real numbers, and each component H_γ is Hermitian. Without loss of generality, we assume that each component is normalized, $\|H_\gamma\| = 1$, absorbing the normalization factors into the coefficients c_γ.

The core idea of qDRIFT is to apply the individual evolution operators $U_\gamma(t) = e^{-itH_\gamma}$ stochastically. The probability of selecting the γ-th term is determined by the relative magnitude of its coefficient. We define the L_1 norm of the coefficients as

$$\|c\|_1 = \sum_{\gamma=1}^{\Gamma} c_\gamma, \tag{15.138}$$

and the corresponding probability distribution as

$$p_\gamma = c_\gamma / \|c\|_1 . \tag{15.139}$$

This stochastic application of unitaries defines a quantum channel $\mathcal{Q}_t$:

$$\mathcal{Q}_t(\rho) = \sum_{\gamma=1}^{\Gamma} p_\gamma e^{-iH_\gamma t} \rho e^{iH_\gamma t}. \tag{15.140}$$

The key insight is that the channel $\mathcal{Q}_t$ closely approximates the ideal evolution $e^{\mathcal{L}t'}$, where $\mathcal{L}(\rho) = -i[H, \rho]$ and the effective time step is $t' = t/\|c\|_1$. To simulate the evolution for a total time T, the channel is applied repeatedly N_t times, such that $T = N_t t'$. The procedure is formalized in Algorithm 15.1.

The efficiency of the qDRIFT protocol depends on the number of steps N_t required to achieve a desired precision ϵ, measured using the diamond norm.

Proposition 15.24 (Convergence of qDRIFT)**.** *Let $H = \sum_{\gamma=1}^{\Gamma} c_\gamma H_\gamma$ be a Hamiltonian such that $\|H_\gamma\| = 1$ and $c_\gamma > 0$ for all γ. Let $\|c\|_1 = \sum_\gamma c_\gamma$. To simulate the evolution $e^{\mathcal{L}T}$ up to error ϵ in the diamond norm, it suffices to apply the qDRIFT channel $\mathcal{Q}_t$ defined in Eq.* (15.140) *for N_t steps, where*

$$N_t = \mathcal{O}\left(\frac{T^2 \|c\|_1^2}{\epsilon}\right). \tag{15.141}$$

Proof. We analyze the error introduced in a single step. Let t be the time parameter for the channel $\mathcal{Q}_t$. We aim to bound the diamond norm distance between $\mathcal{Q}_t$ and the ideal evolution $e^{\mathcal{L}t'}$, where $t' = t/\|c\|_1$.

We begin by defining a linearized approximation of the evolution. Let $W_\gamma = I - iH_\gamma t$. Since $\|H_\gamma\| = 1$, we have $\|W_\gamma\| \leq 1 + t \leq e^t$. The approximation error between the unitary $U_\gamma = e^{-iH_\gamma t}$

Algorithm 15.1 qDRIFT Protocol

Require: Hamiltonian $H = \sum_{\gamma=1}^{\Gamma} c_\gamma H_\gamma$ (with $\|H_\gamma\| = 1, c_\gamma > 0$).
Access to unitaries $e^{-iH_\gamma t}$.
Initial state $|\psi\rangle$. Total evolution time $T > 0$. Number of steps N_t.
Ensure: Output state $|\Psi_{N_t}\rangle$, which is a sample from the distribution defined by the channel $\mathcal{Q}_t^{N_t}(|\psi\rangle\langle\psi|)$, approximating $e^{-iHT}|\psi\rangle$.
1: Calculate $\|c\|_1 \leftarrow \sum_{\gamma=1}^{\Gamma} c_\gamma$.
2: Define the probability distribution $p_\gamma \leftarrow c_\gamma / \|c\|_1$ for $\gamma = 1, \ldots, \Gamma$.
3: Set the time step parameter $t \leftarrow \frac{T\|c\|_1}{N_t}$.
4: Initialize $|\Psi_0\rangle \leftarrow |\psi\rangle$.
5: **for** $k = 1$ to N_t **do**
6: Randomly sample an index $\gamma_k \in \{1, \ldots, \Gamma\}$ according to the distribution $\{p_\gamma\}$.
7: Apply the unitary evolution: $|\Psi_k\rangle \leftarrow e^{-iH_{\gamma_k} t}|\Psi_{k-1}\rangle$.
8: **end for**
9: **return** $|\Psi_{N_t}\rangle$.

and W_γ is bounded by the Taylor series remainder:

$$\|U_\gamma - W_\gamma\| \leq \sum_{k=2}^{\infty} \frac{t^k}{k!} = e^t - (1+t) \leq \frac{t^2}{2} e^t. \tag{15.142}$$

Let $\mathcal{U}_\gamma(\rho) = U_\gamma \rho U_\gamma^\dagger$ and $\mathcal{W}_\gamma(\rho) = W_\gamma \rho W_\gamma^\dagger$. We bound the diamond norm distance between these two maps using Lemma 3.59.

$$\|\mathcal{U}_\gamma - \mathcal{W}_\gamma\|_\diamond \leq \|U_\gamma\| \left\|U_\gamma^\dagger - W_\gamma^\dagger\right\| + \|U_\gamma - W_\gamma\| \left\|W_\gamma^\dagger\right\|. \tag{15.143}$$

Since U_γ is unitary, $\|U_\gamma\| = 1$. We also have $\left\|U_\gamma^\dagger - W_\gamma^\dagger\right\| = \|U_\gamma - W_\gamma\|$ and $\left\|W_\gamma^\dagger\right\| = \|W_\gamma\|$ (as H_γ is Hermitian). Using the bounds derived above:

$$\begin{aligned} \|\mathcal{U}_\gamma - \mathcal{W}_\gamma\|_\diamond &\leq \|U_\gamma - W_\gamma\| \left(1 + \|W_\gamma\|\right) \\ &\leq \left(\frac{t^2}{2} e^t\right)(1 + e^t) = \frac{t^2}{2}(e^t + e^{2t}). \end{aligned} \tag{15.144}$$

For $t \geq 0$, $e^t \leq e^{2t}$, so $\|\mathcal{U}_\gamma - \mathcal{W}_\gamma\|_\diamond \leq t^2 e^{2t}$.

Let $\mathcal{W}_t = \sum_\gamma p_\gamma \mathcal{W}_\gamma$ be the averaged linearized map. The qDRIFT channel is $\mathcal{Q}_t = \sum_\gamma p_\gamma \mathcal{U}_\gamma$. By the convexity of the diamond norm:

$$\|\mathcal{Q}_t - \mathcal{W}_t\|_\diamond \leq \sum_\gamma p_\gamma \|\mathcal{U}_\gamma - \mathcal{W}_\gamma\|_\diamond \leq t^2 e^{2t}. \tag{15.145}$$

Next, we examine the structure of $\mathcal{W}_t$:

$$\begin{aligned} \mathcal{W}_t(\rho) &= \sum_\gamma p_\gamma (I - iH_\gamma t)\rho(I + iH_\gamma t) \\ &= \rho - it \sum_\gamma \frac{c_\gamma}{\|c\|_1} [H_\gamma, \rho] + t^2 \sum_\gamma p_\gamma H_\gamma \rho H_\gamma \\ &= (\mathcal{I} + t'\mathcal{L})(\rho) + t^2 \mathcal{D}(\rho), \end{aligned} \tag{15.146}$$

where $\mathcal{D}(\rho) = \sum_\gamma p_\gamma H_\gamma \rho H_\gamma$. Let $\mathcal{I}_1 = \mathcal{I} + t'\mathcal{L}$ be the first-order approximation of the ideal evolution. The difference between $\mathcal{W}_t$ and $\mathcal{I}_1$ is $t^2\mathcal{D}$. We bound its diamond norm. Since H_γ are Hermitian and $p_\gamma \geq 0$, the map $\mathcal{D}$ is completely positive. By **??**, its diamond norm is equal to the operator norm of its adjoint acting on the identity, $\|\Phi\|_\diamond = \left\|\Phi^\dagger(I)\right\|$. Since H_γ are Hermitian, $\mathcal{D}$ is self-adjoint. Thus,

$$\|\mathcal{D}\|_\diamond = \|\mathcal{D}(I)\| = \left\|\sum_\gamma p_\gamma H_\gamma^2\right\|. \tag{15.147}$$

We bound this operator norm using the triangle inequality:

$$\|\mathcal{D}\|_\diamond \leq \sum_\gamma p_\gamma \left\|H_\gamma^2\right\| = \sum_\gamma p_\gamma \|H_\gamma\|^2 = 1. \tag{15.148}$$

Thus, $\|\mathcal{W}_t - \mathcal{I}_1\|_\diamond \leq t^2$. Combining the bounds via the triangle inequality yields

$$\|\mathcal{Q}_t - \mathcal{I}_1\|_\diamond \leq \|\mathcal{Q}_t - \mathcal{W}_t\|_\diamond + \|\mathcal{W}_t - \mathcal{I}_1\|_\diamond \leq t^2 e^{2t} + t^2 \leq 2t^2 e^{2t}. \tag{15.149}$$

Finally, we bound the error in the Taylor expansion of the ideal evolution $e^{\mathcal{L}t'}$.

$$\left\|e^{\mathcal{L}t'} - \mathcal{I}_1\right\|_\diamond \leq \sum_{k=2}^{\infty} \frac{(t')^k \|\mathcal{L}\|_\diamond^k}{k!}. \tag{15.150}$$

We use the bound $\|\mathcal{L}\|_\diamond \leq 2\|H\|$. Furthermore, $\|H\| \leq \sum_\gamma c_\gamma \|H_\gamma\| = \|c\|_1$. Thus, the expansion parameter is bounded by $t' \|\mathcal{L}\|_\diamond \leq (t/\|c\|_1)(2\|c\|_1) = 2t$. Therefore,

$$\left\|e^{\mathcal{L}t'} - \mathcal{I}_1\right\|_\diamond \leq \sum_{k=2}^{\infty} \frac{(2t)^k}{k!} = e^{2t} - (1+2t) \leq 2t^2 e^{2t}. \tag{15.151}$$

The total error per step is the sum of these contributions:

$$\left\|e^{\mathcal{L}t'} - \mathcal{Q}_t\right\|_\diamond \leq \left\|e^{\mathcal{L}t'} - \mathcal{I}_1\right\|_\diamond + \|\mathcal{I}_1 - \mathcal{Q}_t\|_\diamond \leq 4t^2 e^{2t}. \tag{15.152}$$

Since both $e^{\mathcal{L}t'}$ and $\mathcal{Q}_t$ are quantum channels (CPTP maps), their diamond norms are 1. By the stability of the diamond norm under composition, errors accumulate at most linearly over N_t steps. The total error for the evolution up to time $T = N_t t'$ is

$$\left\|e^{\mathcal{L}T} - \mathcal{Q}_t^{N_t}\right\|_\diamond \leq N_t \left\|e^{\mathcal{L}t'} - \mathcal{Q}_t\right\|_\diamond \leq 4N_t t^2 e^{2t}. \tag{15.153}$$

Substituting $N_t = T\|c\|_1 / t$, the total error is $4T\|c\|_1 te^{2t}$. To ensure this is bounded by ϵ, we require $4T\|c\|_1 te^{2t} \leq \epsilon$. By choosing $t = \Theta(\epsilon/(T\|c\|_1))$, the factor e^{2t} approaches 1 as $\epsilon \to 0$ (assuming $T\|c\|_1$ is fixed), or is otherwise bounded by a constant if t is small. Consequently, the required number of steps is $N_t = \mathcal{O}\left(\frac{T^2\|c\|_1^2}{\epsilon}\right)$. □

We can generalize this protocol to Hamiltonians $H = \sum_\gamma H_\gamma$ where the individual terms are not necessarily normalized. We define the 1-norm of the Hamiltonian as

$$|||H||| := \sum_\gamma \|H_\gamma\|. \tag{15.154}$$

We can rewrite $H = \sum_\gamma \|H_\gamma\| (H_\gamma / \|H_\gamma\|)$. Applying qDRIFT to this decomposition, we identify $c_\gamma = \|H_\gamma\|$ and $\|c\|_1 = |||H|||$. The probability distribution becomes $p_\gamma = \|H_\gamma\| / |||H|||$, and the

evolution applied in each step corresponds to the normalized Hamiltonian terms. The resulting complexity scaling is

$$N_t = \mathcal{O}\left(\frac{T^2 |||H|||^2}{\epsilon}\right), \tag{15.155}$$

which matches the error scaling of the first-order Trotter method with respect to T and ϵ.

The primary advantage of the qDRIFT method is that its cost depends on the aggregate norm $|||H|||$ and is independent of the number of terms Γ. This is particularly beneficial when the Hamiltonian comprises a large number of terms, provided their cumulative norm remains manageable.

However, qDRIFT presents some disadvantages compared to deterministic methods like Trotterization. Its inherent randomized nature can complicate its use as a subroutine in algorithms requiring a coherent implementation of the unitary evolution e^{-iHT}. Furthermore, the error bound in qDRIFT depends directly on $|||H|||^2$, rather than on the norms of the commutators between the terms. This often results in a larger prefactor in the error compared to higher-order Trotter formulas, especially when the Hamiltonian terms nearly commute.

Moreover, the nature of the output state differs significantly. If the target precision ϵ is defined in terms of the diamond norm (or trace distance), both the first-order Trotter method and qDRIFT exhibit a cost scaling of $\mathcal{O}(\epsilon^{-1})$. However, the Trotter method implements a unitary evolution, resulting in a pure state if the input is pure. In contrast, qDRIFT implements a quantum channel, generally producing a mixed state (represented as an ensemble of the pure states generated by the algorithm). If we measure accuracy in terms of infidelity (recall **??**), the infidelity of the deterministic first-order Trotter method scales as $\mathcal{O}(\epsilon^2)$. This is often quadratically better than the infidelity achievable by a randomized method such as qDRIFT, where the infidelity typically scales linearly with the trace distance error, $\mathcal{O}(\epsilon)$.

15.6. Sparse Hamiltonian simulation with product formulas

15.7. Operator splitting for time-dependent problems

15.8. Lieb-Robinson Bounds for Local Hamiltonians

15.9. Phase Estimation and Operator Splitting

Notes and further reading

The high-order operator splitting methods discussed in this chapter are constructed by composing lower-order methods. Such composition techniques were extensively developed in the 1990s [Suz90, Yos90, McL95], and we refer readers to the classical textbook [HLW06] for a comprehensive treatment of composition methods in the context of geometric numerical integration. The time-independent Schrödinger equation offers a concrete setting for analyzing the error behavior of operator splitting methods, both in low- and high-order regimes [JL00, Tha08, DT10]. For instance, [JL00] first observed that in some quantum settings, the actual error scales only with the size of commutators (with the vector norm scaling), leading to error bounds that are significantly smaller than one might expect from standard order analysis.

In the context of quantum algorithms, these ideas have been extended and refined to address bounded operators such as spin systems, as well as unbounded operators like Schrödinger operators in first quantization. A detailed and increasingly nuanced understanding of error behavior in Hamiltonian simulation has since emerged as a highly active area of research. Theoretical developments have led to continued improvements in both asymptotic and practical error

bounds [WBHS10, BCG14, BCK15, CMN+18, COS19, Low19, CS19, CST+21, CHKT21, ŞS20, AFL21, SBW+21, AFL22, RWW24, HHB+25].

Beyond the Trotter-Suzuki family of composition methods, a wide variety of other approaches have been developed for constructing high-order operator splitting schemes for Hamiltonian simulation. These include Magnus expansion techniques [BCOR09, CZA24, FLS25], multiproduct formulas [LKW19, AAT24], and other generalizations, each with their own trade-offs in terms of cost, accuracy, and suitability for quantum implementation.

CHAPTER 16

Quantum phase estimation

Quantum phase estimation is one of the most versatile and powerful tools in the arsenal of the quantum algorithm designer. The basic idea behind phase estimation is to perform an experiment that interferes two different branches of a quantum superposition on which an unknown phase is applied and another where it is not. We then aim to infer from the interference pattern that emerges the value of that phase.

While the quantum application of this idea is relatively new, the roots of the phase estimation algorithm run deep. The idea was essentially pioneered by Jules Jamin in 1856, where the ideas were subsequently developed into a field that we now know as interferometry which allows precise estimation of phase differences to be estimated via one (of many possible) wave interference experiment. These ideas have profoundly changed radio astronomy (where interference between radio waves are used to perform high-sensitivity experiments). For readers well versed in interferometry, the quantum phase estimation is mathematically equivalent to a (multi-pass) Mach-Zehnder interferometer from optics with the role of beam splitters replaced with a Hadamard gate and the phase delay on one arm replaced with a qubit-controlled unitary operation. For those unfamiliar with interferometry, we define the problem of quantum phase estimation below.

The setup of the phase estimation problem is as follows. Let U be a unitary, and $|\psi\rangle$ is an eigenvector, i.e.,

$$U|\psi\rangle = e^{i2\pi\varphi}|\psi\rangle, \quad \varphi \in [0,1). \tag{16.1}$$

The goal is to find the phase φ up to certain precision. We will use the 1-periodic distance to measure the distance between phases

$$|\theta|_1 \equiv |\theta|_{\bmod 1} := \min\{(\theta \bmod 1), 1-(\theta \bmod 1)\} \in [0,1/2] \tag{16.2}$$

Then for $\varphi, \varphi' \in [0,1)$,

$$\left|e^{2\pi i\varphi} - e^{2\pi i\varphi'}\right| = 2\left|\sin(\pi(\varphi-\varphi'))\right| = 2\sin(\pi|\varphi-\varphi'|_1) \geq 4|\varphi-\varphi'|_1. \tag{16.3}$$

Here we have used the fact

$$|\sin(\pi\theta)| \geq 2|\theta|, \quad \theta \in [-1/2, 1/2]. \tag{16.4}$$

If we only have an approximate implementation of U denoted by $\widetilde{U}$ satisfying

$$\left\|U - \widetilde{U}\right\| \leq \epsilon, \tag{16.5}$$

then according to the perturbation theory of unitary matrices in Theorem 7.8, the perturbation of the eigenvalue on the unit circle satisfies $\left|e^{2\pi i\varphi} - e^{2\pi i\varphi'}\right| \leq \epsilon$, and the perturbation of the corresponding phase angle is bounded by $\epsilon/4$.

Phase estimation is a quantum primitive with numerous applications: prime factorization (Shor's algorithm), linear system (HHL), eigenvalue problem, amplitude estimation, quantum counting, quantum walk, etc.

Using a classical computer, we can estimate φ using $U|\psi\rangle \oslash |\psi\rangle$, where $\oslash$ stands for the element-wise division operation. Specifically, if $|\psi\rangle$ is indeed an eigenvector and $\langle j|\psi\rangle \neq 0$ for some j in the computational basis, then we can extract the phase from

$$\langle j|U|\psi\rangle \,/\, \langle j|\psi\rangle = e^{i2\pi\varphi}. \tag{16.6}$$

Unfortunately, such a element-wise division operation cannot be efficiently implemented on quantum computers, and alternative methods are needed.

Quantum phase estimation has numerous variants, and still receives intensive research attention till today. This chapter only introduces some of the simplest variants.

We distinguish between two primary tasks in quantum phase estimation: (1) Eigenvalue estimation, and (2) Eigenstate preparation. The QPE algorithm in Section 16.2 accomplishes both tasks simultaneously. There is a variant of the QPE algorithm that refines the state iteratively and uses only one ancilla qubit, which will be discussed in Section 16.7. In addition, there exist spectral estimation algorithms that use a single ancilla qubit but focus solely on task (1), and we will provide an example of such an algorithm in **??**.

16.1. Quantum Fourier transform

The Fourier transform is ubiquitous in scientific computing, and the fast Fourier transform (FFT) is a backbone for many fast classical algorithms. The quantum Fourier transform (QFT) plays a similar role in quantum algorithms, appearing in phase estimation, Shor's algorithm, and related constructions such as the fast Fermionic Fourier transform (FFFT) [BWM+18]. Beyond serving as a subroutine, it provides a concrete mechanism for converting information stored in phase into information stored in the computational basis, and it leads naturally to generalized Pauli matrices on multi-qubit systems. For these reasons, a clear understanding of the quantum Fourier transform is useful throughout quantum algorithms.

For any j in the computational basis of an n-qubit system with $N = 2^n$, the (discrete) forward Fourier transform is defined as follows:

$$U_{\mathrm{FT}}|j\rangle = \frac{1}{\sqrt{N}} \sum_{k\in[N]} e^{i2\pi \frac{kj}{N}} |k\rangle\,. \tag{16.7}$$

The transformation is also often abbreviated in terms of the root of unity $\omega = e^{i2\pi/N}$ as

$$U_{\mathrm{FT}}|j\rangle = \frac{1}{\sqrt{N}} \sum_{k\in[N]} \omega^{kj} |k\rangle\,. \tag{16.8}$$

In particular

$$U_{\mathrm{FT}}|0^n\rangle = \frac{1}{\sqrt{N}} \sum_{k\in[N]} |k\rangle = \mathrm{H}^{\otimes n}|0^n\rangle\,. \tag{16.9}$$

The (discrete) inverse Fourier transform is

$$U_{\mathrm{FT}}^{\dagger}|j\rangle = \frac{1}{\sqrt{N}} \sum_{k\in[N]} e^{-i2\pi \frac{kj}{N}} |k\rangle\,. \tag{16.10}$$

Using the binary representation of integers

$$k = (k_{n-1} \cdots k_0.), \quad j = (j_{n-1} \cdots j_0.) \tag{16.11}$$

we have

$$\begin{aligned}\frac{kj}{N} =& k_0 \frac{j}{2^n} + k_1 \frac{j}{2^{n-1}} + \cdots + k_{n-1} \frac{j}{2} \\ =& k_0(.j_{n-1} \cdots j_0) + k_1(j_{n-1}.j_{n-2} \cdots j_0) + \cdots + k_{n-1}(j_{n-1} \cdots j_1.j_0).\end{aligned}$$

Since $e^{i2\pi x}$ depends only on the fractional part of x, we may discard the integer parts in the binary expansions above. Therefore the exponential can be written as

$$e^{i2\pi \frac{kj}{N}} = e^{i2\pi k_0(.j_{n-1} \cdots j_0)} e^{i2\pi k_1(.j_{n-2} \cdots j_0)} \cdots e^{i2\pi k_{n-1}(.j_0)}. \tag{16.12}$$

A standard calculation gives the following factorized form:

(16.13)

$$\begin{aligned}U_{\mathrm{FT}} |j_{n-1} \cdots j_0\rangle =& \frac{1}{\sqrt{2^n}} \sum_{k_{n-1},\ldots,k_0} e^{i2\pi k_0(.j_{n-1} \cdots j_0)} e^{i2\pi k_1(.j_{n-2} \cdots j_0)} \cdots e^{i2\pi k_{n-1}(.j_0)} |k_{n-1} \cdots k_0\rangle \\ =& \frac{1}{\sqrt{2^n}} \left(\sum_{k_{n-1}} e^{i2\pi k_{n-1}(.j_0)} |k_{n-1}\rangle \right) \otimes \left(\sum_{k_{n-2}} e^{i2\pi k_{n-2}(.j_1 j_0)} |k_{n-2}\rangle \right) \\ & \otimes \cdots \otimes \left(\sum_{k_0} e^{i2\pi k_0(.j_{n-1} \cdots j_0)} |k_0\rangle \right) \\ =& \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{i2\pi(.j_0)} |1\rangle \right) \otimes \left(|0\rangle + e^{i2\pi(.j_1 j_0)} |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + e^{i2\pi(.j_{n-1} \cdots j_0)} |1\rangle \right).\end{aligned}$$

Eq. (16.13) involves a series of controlled rotations of the form

$$|0\rangle \to \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi(.j_{n-1} \cdots j_0)} |1\rangle \right). \tag{16.14}$$

Before describing the full QFT circuit, let us first work out a circuit for implementing this controlled rotation. We use the relation

$$e^{i2\pi(.j_{n-1} \cdots j_0)} = e^{i2\pi(.j_{n-1})} e^{i2\pi(.0 j_{n-2})} \cdots e^{i2\pi(.0 \cdots 0 j_0)}. \tag{16.15}$$

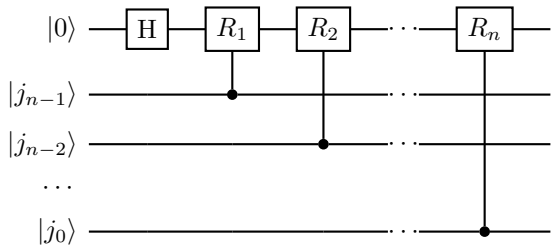
Example 16.1 (Implementation of controlled rotation)**.** Consider the implementation of

$$|0\rangle |j\rangle \to \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi(.j_{n-1} \cdots j_0)} |1\rangle \right) |j\rangle, \tag{16.16}$$

Let us define

$$R_j = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2^{j-1}} \end{pmatrix}. \tag{16.17}$$

In particular, $R_1 = Z$. The quantum circuit is



◇

The implementation of QFT follows the same principle, but **does not** require a separate signal qubit to store the phase information. Let us see a few examples.

When $n = 1$, we need to implement

$$|j_0\rangle \to \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi(.j_0)}|1\rangle\right). \tag{16.18}$$

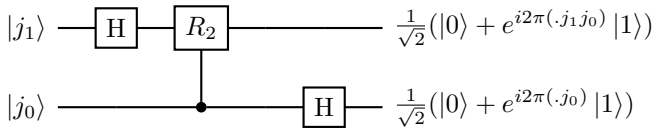
This is the Hadamard gate:

$$|j_0\rangle \to \mathrm{H}|j_0\rangle. \tag{16.19}$$

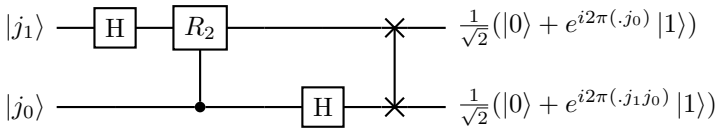
When $n = 2$, we need to implement

$$|j\rangle \to \frac{1}{\sqrt{2^2}}\left(|0\rangle + e^{i2\pi(.j_0)}|1\rangle\right) \otimes \left(|0\rangle + e^{i2\pi(.j_1 j_0)}|1\rangle\right). \tag{16.20}$$

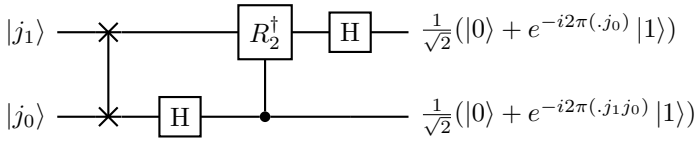
This can be implemented using the following circuit:



Comparing the result with that in Eq. (16.20), we find that the ordering of the qubits is reversed. To recover the desired result in QFT, we can apply a SWAP gate to the outcome, i.e.,



In order to implement the inverse Fourier transform, we only need to apply the Hermitian conjugate as



Similarly one can construct the circuit for U_{FT} and its inverse for $n = 3$.

In general, the QFT circuit is given by Fig. 16.1. Compare the circuit in Fig. 16.1 with Eq. (16.13), we find again that the ordering is reversed in the output. To restore the correct order as defined in QFT, we can use $\mathcal{O}(n/2)$ swap operations. The total gate complexity of QFT is $\mathcal{O}(n^2)$. We summarize this in the following theorem.

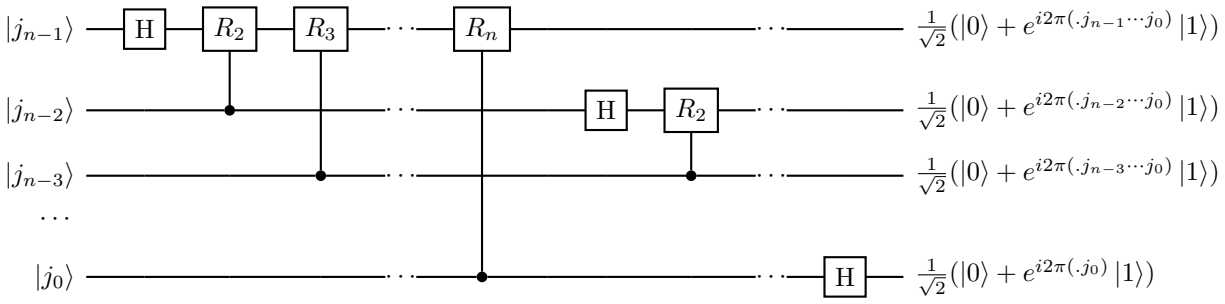


FIGURE 16.1. Circuit diagram for quantum Fourier transform (before applying swap operations).

THEOREM 16.2. *Let $U_{\mathrm{FT}} \in L(\mathbb{C}^{2^n})$ be the n-qubit quantum Fourier transform. There exists a quantum circuit of length $\mathcal{O}(n^2)$ containing gates drawn from the library $\mathrm{H}, \mathrm{CNOT}, R_z(\theta)$ that can precisely implement U_{FT}.*

Exercise 16.1. For a 3 qubit system, explicitly construct the circuit for U_{FT} and its inverse.

Exercise 16.2. For a n-qubit system, write down the quantum circuit for the swap operation used in QFT.

16.1.1. Generalized Pauli matrices. As we saw above, the quantum Fourier transform generalizes the Hadamard transform to dimensions greater than 2. The Hadamard gate is an element of the Clifford group, and hence it maps the single-qubit Pauli matrices to themselves under conjugation. For $n \geq 2$, the quantum Fourier transform is not a Clifford unitary, so it does not stabilize the n-qubit Pauli group. Nevertheless, there is a natural family of Pauli-like matrices that is stabilized by conjugation with the quantum Fourier transform. These are Sylvester's **generalized Pauli matrices** (also known as **Weyl-Heisenberg matrices**), which will also be useful later in the discussion of phase estimation and arithmetic.

Sylvester's generalized Pauli matrices are defined as follows.

Definition 16.3 (Sylvester's generalized Pauli matrices)**.** *Let $\mathcal{Z}_n \in L(\mathbb{C}^{2^n})$ be the n-qubit generalized Z matrix, defined by*

$$\mathcal{Z}_n := \sum_{j=0}^{2^n-1} e^{i2\pi j/2^n} |j\rangle\langle j|. \tag{16.21}$$

and let $\mathcal{X}_n \in L(\mathbb{C}^{2^n})$ be the n-qubit generalized X matrix, defined by

$$\mathcal{X}_n := U_{\mathrm{FT}}^\dagger \mathcal{Z}_n U_{\mathrm{FT}}. \tag{16.22}$$

Up to global phases, the 2^{2n} elements of the n-qubit generalized Pauli group are

$$P_{k,j} = \mathcal{X}_n^k \mathcal{Z}_n^j = \sum_{\ell=0}^{2^n-1} \omega^{j\ell} |\ell + k\rangle\langle\ell|, \tag{16.23}$$

where the addition $\ell + k$ is taken modulo 2^n.

This shows that $\mathcal{X}_n$ plays the role of a bit-flip (Pauli-X) matrix in this setting: it is the Fourier transform of the diagonal generalized Pauli-Z matrix. Although the generalized Pauli group has 2^{2n} elements, the matrices have enough structure that we can often manipulate them without expanding them entrywise.

Example 16.4. As a simple example of finding an explicit expression for the generalized Pauli matrices, let us verify the property that

$$\mathcal{X}_n = \sum_{\ell=0}^{2^n-1} |\ell+1\rangle\langle\ell|, \tag{16.24}$$

where $\ell+1$ is understood modulo 2^n. Using the definition of U_{FT} and $\mathcal{Z}_n$ we compute

$$\begin{aligned} U_{\mathrm{FT}}^\dagger \mathcal{Z}_n U_{\mathrm{FT}} &= \frac{1}{2^n} \sum_{j,m=0}^{2^n-1} \left(\sum_{\ell=0}^{2^n-1} e^{i2\pi\ell(m-j+1)/2^n} \right) |j\rangle\langle m| \\ &= \sum_{m=0}^{2^n-1} |m+1\rangle\langle m|, \end{aligned} \tag{16.25}$$

where we used $\sum_{\ell=0}^{2^n-1} e^{i2\pi\ell t/2^n} = 2^n$ if $t \equiv 0 \pmod{2^n}$ and 0 otherwise. ◇

This example illustrates a basic property of the Fourier transform: it turns phase rotations into shifts in the computational basis. This observation leads to an in-place adder construction known as the Draper adder.

Corollary 16.5 (Draper adder)**.** *There exists a quantum algorithm that implements a unitary n-qubit modular adder that maps, for any n-bit integers p, q, $|p\rangle|q\rangle \mapsto |p\rangle|q+p \mod 2^n\rangle$. This algorithm uses no ancillary qubits and requires $\mathcal{O}(n^2)$ two-qubit operations.*

Proof. Write $a = \sum_{r=0}^{n-1} a_r 2^r$ with $a_r \in \{0,1\}$. Then

$$\begin{aligned} |a\rangle|b\rangle &\to |a\rangle\ \mathcal{X}_n^a |b\rangle \\ &= |a\rangle\ \mathcal{X}_n^{a_0 2^0} \cdots \mathcal{X}_n^{a_{n-1} 2^{n-1}} |b\rangle \\ &= |a\rangle\ U_{\mathrm{FT}}^\dagger \mathcal{Z}_n^{a_0 2^0} \cdots \mathcal{Z}_n^{a_{n-1} 2^{n-1}} U_{\mathrm{FT}} |b\rangle\,. \end{aligned} \tag{16.26}$$

Each $\mathcal{Z}_n$ can be implemented using a sequence of n single-qubit phase rotations. Specifically, using the same binary expansion used above in the Fourier transform,

$$\mathcal{Z}_n = \bigotimes_{m=0}^{n-1} e^{i2\pi 2^{m-n}|1\rangle\langle 1|}. \tag{16.27}$$

In turn for any integer power $p \geq 0$ we then have that

$$\mathcal{Z}_n^p = \bigotimes_{m=0}^{n-1} e^{i2\pi p 2^{m-n}|1\rangle\langle 1|}, \tag{16.28}$$

which requires only n single-qubit rotations. In the adder construction, these rotations are controlled by the bits of a, and hence are implemented as controlled phase rotations between the registers. There are $\mathcal{O}(n^2)$ such controlled rotations, each realizable using $\mathcal{O}(1)$ CNOT gates and single-qubit R_z gates, so the overall two-qubit gate complexity is $\mathcal{O}(n^2)$.

As noted in Theorem 16.2, the n-qubit quantum Fourier transform can be implemented using $\mathcal{O}(n^2)$ two-qubit gates. Thus the overall two-qubit gate complexity is $\mathcal{O}(n^2) + \mathcal{O}(n^2) = \mathcal{O}(n^2)$. Finally, since neither the quantum Fourier transform nor the controlled rotations require ancillary qubits, the entire adder algorithm can be implemented in place. □

We see from the above corollary that, in this setting, the identity $Z = \mathrm{H}\,X\,\mathrm{H}$ generalizes to a construction of adders that do not require additional qubits, unlike traditional reversible circuits such as the carry ripple adder. This example will set the stage in the following section, where we use similar reasoning to show that the quantum Fourier transform can be used to learn the eigenvalues of a unitary, even when those eigenvalues are not of the form $e^{i2\pi j/2^n}$ for $j \in \mathbb{Z}_{2^n}$.

The generalized Pauli matrices further satisfy similar commutation relations to ordinary Pauli matrices that reduce to the familiar anti-commutation relation when $n = 1$:

$$\mathcal{X}_n \mathcal{Z}_n = e^{-i2\pi/2^n} \mathcal{Z}_n \mathcal{X}_n. \tag{16.29}$$

Further, these generalized Pauli matrices form a complete orthonormal basis with respect to the Hilbert-Schmidt inner product: $\langle P_{j,k}, P_{\ell,m}\rangle = 2^{-n} \operatorname{Tr}[P_{j,k}^\dagger P_{\ell,m}] = \delta_{(j,k),(\ell,m)}$. Consequently, any matrix M can be decomposed as $M = \sum_{j,k} P_{j,k} \langle P_{j,k}, M\rangle$.

16.2. Quantum phase estimation

In this section, we introduce the (standard) quantum phase estimation (QPE), which uses a quantum circuit based on the quantum Fourier transform (QFT), and requires d ancilla qubits to store the phase information.

From nowor now, we assume that $\varphi = 0.j_{d-1}\cdots j_0$ has an exact d-bit representation. From the availability of U^j we can define a controlled unitary operation

$$\mathcal{U} = \sum_{j\in[2^d]} |j\rangle\langle j| \otimes U^j. \tag{16.30}$$

When $d = 1$, $\mathcal{U}$ is simply the controlled U operation. For a general d, it seems that we need to implement all 2^d different U^j. However, this is not necessary. Writing $j = \sum_{i=0}^{d-1} j_i 2^i$ with $j_i \in \{0, 1\}$, we have $U^j = U^{\sum_{i=0}^{d-1} j_i 2^i} = \prod_{i=0}^{d-1} U^{j_i 2^i}$. Therefore, similarly to the operations in QFT,

$$\begin{aligned}
\mathcal{U} &= \sum_{j\in[2^d]} |j\rangle\langle j| \otimes U^j \\
&= \sum_{j_{d-1},\ldots,j_0} (|j_{d-1}\rangle\langle j_{d-1}|) \otimes \cdots \otimes (|j_0\rangle\langle j_0|) \otimes \prod_{i=0}^{d-1} U^{j_i 2^i} \\
&= \prod_{i=0}^{d-1}{}' \left(\sum_{j_i} |j_i\rangle\langle j_i| \otimes U^{j_i 2^i} \right) \\
&= \prod_{i=0}^{d-1}{}' \left(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U^{2^i} \right).
\end{aligned} \tag{16.31}$$

Here the primed product $\prod'$ is a slightly awkward notation: the factors act on different control qubits in the first register, while they are multiplied (i.e., applied sequentially) on the second register. It is in fact much clearer to observe the structure in the quantum circuit in Fig. 16.2.

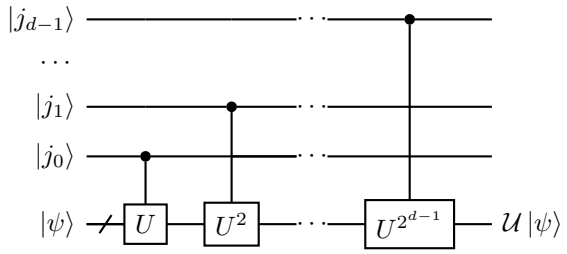


FIGURE 16.2. Circuit for controlled matrix power of U.

Now let the initial state in the ancilla qubits be $|0^d\rangle$. Using $\mathrm{H}^{\otimes d}$ (equivalently, the QFT acting on $|0^d\rangle$) and $\mathcal{U}$, we transform the initial state according to

$$\begin{aligned} |0^d\rangle\,|\psi_0\rangle &\xrightarrow{\mathrm{H}^{\otimes d}\otimes I} \frac{1}{\sqrt{2^d}}\sum_{j\in[2^d]}|j\rangle\,|\psi_0\rangle \\ &\xrightarrow{\mathcal{U}} \frac{1}{\sqrt{2^d}}\sum_{j\in[2^d]}|j\rangle\,U^j\,|\psi_0\rangle = \frac{1}{\sqrt{2^d}}\sum_{j\in[2^d]}|j\rangle\,e^{i2\pi\varphi j}\,|\psi_0\rangle \\ &\xrightarrow{U_{\mathrm{FT}}^\dagger\otimes I} \sum_{k'\in[2^d]}\left(\frac{1}{2^d}\sum_{j\in[2^d]}e^{i2\pi j\left(\varphi-\frac{k'}{2^d}\right)}\right)|k'\rangle\,|\psi_0\rangle\,. \end{aligned} \tag{16.32}$$

Since $\varphi = \frac{k}{2^d}$ for some $k \in [2^d]$, measuring the first register yields the outcome k with certainty, and hence recovers the phase information. Therefore the quantum circuit for QFT-based QPE is given by Fig. 16.3. Here we have used Eq. (16.9). We should note that $U_{\mathrm{FT}}^\dagger$ includes the swapping operations.

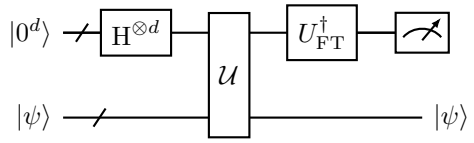


FIGURE 16.3. Quantum circuit for quantum phase estimation using quantum Fourier transform.

Example 16.6 (Hadamard test viewed as QPE)**.** When $d = 1$, note that $U_{\mathrm{FT}}^\dagger = \mathrm{H}$, so the QFT based QPE in Fig. 16.3 is exactly the Hadamard test in Fig. 2.1. Note that φ does not need to be exactly represented by a one bit number! ◇

16.3. Analysis of Fourier-based quantum phase estimation

Recall that we assume U has the eigendecomposition:

$$U\,|\psi_j\rangle = e^{i2\pi\varphi_j}\,|\psi_j\rangle\,. \tag{16.33}$$

Without loss of generality we label the eigenphases so that $0 \le \varphi_0 \le \varphi_1 \le \cdots \le \varphi_{N-1} < 1$. In order to use QPE to find the value of φ_0, so far we have assumed that

(1) The initial state $|\phi\rangle = |\psi_0\rangle$ is an exact eigenstate.
(2) The eigenvalue φ_0 has a d-bit binary representation.

In practical calculations, neither condition can be satisfied **exactly**, and thus we must analyze their effects on the accuracy of QPE.

We first focus on the case that only the condition (2) is violated, i.e., φ_0 cannot be exactly represented by a d-bit number. To estimate φ_0 to d bits of precision with high probability, we apply the QPE circuit using $t > d$ ancilla qubits and interpret the measurement outcome k' as the estimator $\widetilde{\varphi}_{k'} = k'/T$. The exact relation between the t and the desired accuracy d will be determined later. Let $T = 2^t$. Similar to Eq. (16.32), we obtain the state

$$\begin{aligned} |0^t\rangle |\psi_0\rangle &\to \sum_{k' \in [T]} \left(\frac{1}{T} \sum_{j \in [T]} e^{i2\pi j \left(\varphi_0 - \frac{k'}{T}\right)} \right) |k'\rangle |\psi_0\rangle \\ &= \sum_{k'} \gamma_{0,k'} |k'\rangle |\psi_0\rangle . \end{aligned} \tag{16.34}$$

Here

$$\gamma_{0,k'} = \frac{1}{T} \sum_{j \in [T]} e^{i2\pi j \left(\varphi_0 - \frac{k'}{T}\right)} = \frac{1}{T} \frac{1 - e^{i2\pi T(\varphi_0 - \widetilde{\varphi}_{k'})}}{1 - e^{i2\pi(\varphi_0 - \widetilde{\varphi}_{k'})}}, \quad \widetilde{\varphi}_{k'} = \frac{k'}{T}. \tag{16.35}$$

Therefore if φ_0 has an exact t-bit representation, i.e., $\varphi_0 = \widetilde{\varphi}_{k'_0}$ for some k'_0, then $\gamma_{0,k'} = \delta_{k',k'_0}$. We recover the previous result that one run of the QPE circuit gives the value φ_0 deterministically.

Now assume that $\varphi_0 \neq \widetilde{\varphi}_{k'}$ for any k'. The probability of measuring the first register and obtaining k' is

$$\mathbb{P}(k') = |\gamma_{0,k'}|^2 = \frac{1}{T^2} \frac{\sin^2(\pi T(\varphi_0 - \widetilde{\varphi}_{k'}))}{\sin^2(\pi(\varphi_0 - \widetilde{\varphi}_{k'}))} =: F_T(\varphi_0 - \widetilde{\varphi}_{k'}). \tag{16.36}$$

The function

$$F_T(x) = \frac{1}{T^2} \frac{\sin^2(\pi T x)}{\sin^2(\pi x)} . \tag{16.37}$$

is called the **Fejér kernel**.

Note that $e^{i2\pi x}$ is a periodic function with period 1, we can only determine the value of x mod 1. Therefore we use the periodic distance Eq. (16.2). In terms of the phase, we would like to find k'_0 such that

$$\left|\varphi_0 - \widetilde{\varphi}_{k'_0}\right|_1 < \epsilon. \tag{16.38}$$

Here $\epsilon = 2^{-d} = 2^{t-d}/T$ is the precision parameter. In particular, for any k' we have

$$|\varphi_0 - \widetilde{\varphi}_{k'}|_1 \le \frac{1}{2}. \tag{16.39}$$

Using the relation that $|\sin(\pi\theta)| \ge 2|\theta|$ for any $\theta \in [-1/2, 1/2]$, we obtain

$$F_T(\varphi_0 - \widetilde{\varphi}_{k'}) = |\gamma_{0,k'}|^2 \le \frac{1}{4T^2 |\varphi_0 - \widetilde{\varphi}_{k'}|_1^2}. \tag{16.40}$$

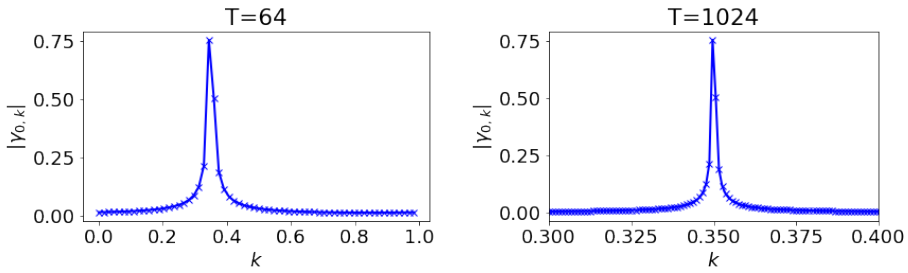


FIGURE 16.4. For $\varphi_0 = 0.35$, the shape of $|\gamma_{0,k}|$ with $T = 64$ and $T = 1024$. These results show a distinct concentration of the probability of success as T increases, which (after noting the smaller range in the $T-1024$ plot) corresponds to a reduction of uncertainty or roughly a factor of 16 as expected by the discussion in Section 16.3.

Let k_0' be the measurement outcome, which can be viewed as a random variable. The probability of obtaining some $\widetilde{\varphi}_{k_0'}$ that is at least ϵ distance away from φ_0 is

$$\begin{aligned}\mathbb{P}(\left|\varphi_0 - \widetilde{\varphi}_{k_0'}\right|_1 \geq \epsilon) &= \sum_{|\varphi_0 - \widetilde{\varphi}_{k'}|_1 \geq \epsilon} F_T(\varphi_0 - \widetilde{\varphi}_{k'}) \\ &\leq \sum_{|\varphi_0 - \widetilde{\varphi}_{k'}|_1 \geq \epsilon} \frac{1}{4T^2 \left|\varphi_0 - \widetilde{\varphi}_{k'}\right|_1^2} \\ &\leq \frac{2}{4T} \int_{\epsilon}^{\infty} \frac{1}{x^2}\,\mathrm{d}x + \frac{2}{4T^2\epsilon^2} = \frac{1}{2T\epsilon} + \frac{1}{2(T\epsilon)^2}.\end{aligned} \tag{16.41}$$

Set $t - d = \lceil \log_2 \delta^{-1} \rceil$, then $T\epsilon = 2^{t-d} \geq \delta^{-1}$. Hence for any $0 < \delta < 1$, the failure probability satisfies

$$\mathbb{P}(\left|\varphi_0 - \widetilde{\varphi}_{k_0'}\right|_1 \geq \epsilon) \leq \frac{\delta + \delta^2}{2} \leq \delta. \tag{16.42}$$

In other words, in order to obtain the phase φ_0 to accuracy $\epsilon = 2^{-d}$ with a success probability at least $1 - \delta$, we need $d + \lceil \log_2 \delta^{-1} \rceil$ ancilla qubits to store the value of the phase. On top of that, the simulation time needs to be $T = (\epsilon\delta)^{-1}$.

Remark 16.7 (Quantum median method)**.** One problem with QPE is that in order to obtain a success probability $1-\delta$, we must use $\log_2 \delta^{-1}$ ancilla qubits, and the maximal simulation time also needs to be increased by a factor δ^{-1}. The increase of the maximal simulation time is particularly undesirable since it increases the circuit depth and hence the required coherence time of the quantum device. When $|\psi\rangle$ is an exact eigenstate, this can be improved by the median method, which uses $\log \delta^{-1}$ copies of the result from QPE without using ancilla qubits or increasing the circuit depth. When $|\psi\rangle$ is a linear combination of eigenstates, the problem of the aliasing effect becomes more difficult to handle. One possibility is to generalize the median method into the quantum median method [NWZ09], which uses classical arithmetics to evaluate the median using a quantum circuit. To reach success probability $1-\delta$, we still need $\log_2 \delta^{-1}$ ancilla qubits, but the maximal simulation time does not need to be increased. ◇

We summarize the preceding discussion as a theorem.

THEOREM 16.8. *Assume that an exact eigenstate $|\psi_0\rangle$ is available and $U|\psi_0\rangle = e^{i2\pi\varphi_0}|\psi_0\rangle$ for some $\varphi_0 \in [0,1)$. Let $\epsilon_0 \in (0,1/2)$ and $\delta \in (0,1)$, and apply the standard t-qubit QPE circuit with $T = 2^t$ to the input $|0^t\rangle|\psi_0\rangle$. If $T \geq (\epsilon_0\delta)^{-1}$ and we output the estimator $\widetilde{\varphi} = k'/T$ from the measurement outcome $k' \in [T]$, then*

$$\mathbb{P}\big(|\widetilde{\varphi} - \varphi_0|_1 \geq \epsilon_0\big) \leq \delta.$$

Moreover, in the standard implementation via controlled powers $U^{2^0}, \ldots, U^{2^{t-1}}$, the number of queries to U is $\mathcal{O}(T) = \mathcal{O}\big(1/(\delta\epsilon_0)\big)$.

PROOF. From the derivation above (using Eq. (16.40) and summing the tail), for $\epsilon_0 = \epsilon$ we have

$$\mathbb{P}\big(|\varphi_0 - \widetilde{\varphi}_{k'_0}|_1 \geq \epsilon_0\big) \leq \frac{1}{2T\epsilon_0} + \frac{1}{2(T\epsilon_0)^2}.$$

If $T\epsilon_0 \geq \delta^{-1}$, then the right-hand side is at most $(\delta+\delta^2)/2 \leq \delta$. Finally, implementing the controlled powers U^{2^i} (without fast-forwarding) uses $\sum_{i=0}^{t-1} 2^i = T-1$ calls to U, hence $\mathcal{O}(T)$ queries. □

We finish this section by stating a general proposition for the outcome of QPE, which relaxes both conditions (1) and (2) and generalizes the previous calculations.

Proposition 16.9. *Let U be a unitary with an eigendecomposition $U|\psi_k\rangle = e^{2\pi i\varphi_k}|\psi_k\rangle$, and let $|\phi\rangle = \sum_k c_k|\psi_k\rangle$ be a normalized initial state. Apply the standard t-qubit QPE circuit with $T = 2^t$ to the input $|0^t\rangle|\phi\rangle$. Then, upon measuring the first register in the computational basis and obtaining outcome $k' \in \{0, \ldots, T-1\}$, the probability of that outcome is*

$$\mathbb{P}(k') = \sum_k \big|c_k\big|^2 \, F_T\big(\varphi_k - k'/T\big).$$

where F_T is the Fejér kernel.

PROOF. We start with

$$|0^t\rangle|\phi\rangle \xrightarrow{U_{\mathrm{FT}}\otimes I} \sum_k c_k \frac{1}{\sqrt{T}} \sum_{j=0}^{T-1} |j\rangle|\psi_k\rangle \xrightarrow{\mathcal{U}} \sum_k c_k \frac{1}{\sqrt{T}} \sum_{j=0}^{T-1} |j\rangle e^{2\pi i\varphi_k j}|\psi_k\rangle .$$

After applying the inverse Fourier transform on the first register,

$$|j\rangle \xrightarrow{U_{\mathrm{FT}}^\dagger} \sum_{k'=0}^{T-1} \Big(\tfrac{1}{T}\sum_{j=0}^{T-1} e^{2\pi ij(\varphi_k - k'/T)}\Big)|k'\rangle ,$$

so the joint state becomes

$$\sum_{k'}\sum_k c_k \, \gamma_{k,k'} \; |k'\rangle|\psi_k\rangle ,$$

with

$$\gamma_{k,k'} = \frac{1}{T}\sum_{j=0}^{T-1} e^{2\pi ij(\varphi_k - k'/T)} = \frac{1}{T}\,\frac{1 - e^{2\pi iT(\varphi_k - k'/T)}}{1 - e^{2\pi i(\varphi_k - k'/T)}}.$$

Hence the probability of observing $|k'\rangle$ in the first register is

$$\mathbb{P}(k') = \Big\|\sum_k c_k \, \gamma_{k,k'} |\psi_k\rangle\Big\|^2 = \sum_k |c_k|^2 \, |\gamma_{k,k'}|^2 = \sum_k |c_k|^2 \, F_T\big(\varphi_k - k'/T\big),$$

which completes the proof. □

16.4. Eigenvalue transformation with quantum phase estimation

As an application of the preceding analysis, we show how QPE can be used as a coherent eigenvalue readout primitive for implementing matrix functions. Let H be a Hermitian matrix, and let $f(H)$ be a matrix function that we would like to apply to a quantum state. We assume access to a unitary U whose eigenphases encode the eigenvalues of H. For concreteness, and consistent with the QPE setup in Chapter 16, we take

$$U = e^{i2\pi H}, \tag{16.43}$$

which can be implemented, for instance, by Hamiltonian simulation. In this discussion we ignore the simulation error. The function $f : \mathbb{R} \to \mathbb{R}$ is assumed to satisfy $|f(x)| \leq 1$ for $x \in [-1, 1]$. Let H have the eigendecomposition

$$H\,|v_j\rangle = \lambda_j\,|v_j\rangle\,. \tag{16.44}$$

To simplify the discussion, we assume $0 < \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1} < 1$ and that all eigenvalues have an exact d-bit representation. More general spectra can be handled by shifting and rescaling H so that its spectrum lies in $[0, 1)$, but then the QPE output must be interpreted modulo 1. Our goal is to prepare a state $|\psi\rangle \propto f(H)\,|b\rangle$.

If the input state is an eigenvector, say $|b\rangle = |v_j\rangle$, then QPE implements the mapping

$$U_{\mathrm{QPE}}\,|0^d\rangle\,|v_j\rangle = |\lambda_j\rangle\,|v_j\rangle\,. \tag{16.45}$$

In general, write the input state as

$$|b\rangle = \sum_j \beta_j\,|v_j\rangle\,. \tag{16.46}$$

Then by linearity,

$$U_{\mathrm{QPE}}\,|0^d\rangle\,|b\rangle = \sum_j \beta_j\,|\lambda_j\rangle\,|v_j\rangle\,. \tag{16.47}$$

Note that

$$f(H)\,|b\rangle = \sum_j \beta_j f(\lambda_j)\,|v_j\rangle\,, \tag{16.48}$$

so it suffices to use the eigenvalue information stored in the ancilla register to multiply each coefficient β_j by the factor $f(\lambda_j)$. For this purpose the eigenvalues must be stored coherently, exactly as QPE provides. We therefore apply the following controlled rotation (see Proposition 5.7):

$$U_{\mathrm{CR}}\,|0\rangle\,|\lambda_j\rangle = \left(\sqrt{1 - \left|f(\widetilde{\lambda}_j)\right|^2}\,|0\rangle + f(\widetilde{\lambda}_j)\,|1\rangle\right)|\lambda_j\rangle\,. \tag{16.49}$$

Here $\widetilde{\lambda}_j$ denotes the value extracted from the QPE register; under the present exact-representation assumption, $\widetilde{\lambda}_j = \lambda_j$.

Finally, we uncompute by applying $U_{\mathrm{QPE}}^\dagger$, which returns the eigenvalue register from $|\lambda_j\rangle$ to $|0^d\rangle$. The overall algorithm is in Fig. 16.5.

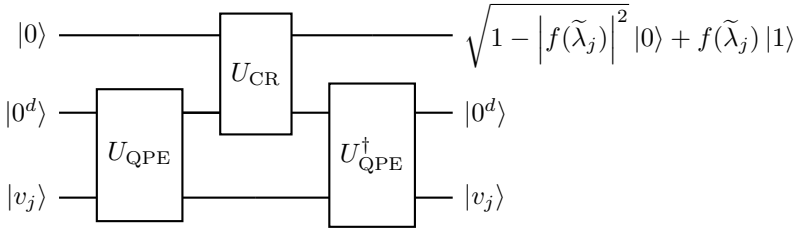


FIGURE 16.5. Circuit for the QPE based algorithm for implementing matrix functions.

After this uncomputation, the d ancilla qubits used for storing the eigenvalues become available again as workspace. Discarding all working registers, the resulting unitary, denoted by $\mathcal{U}$, satisfies

$$\mathcal{U}\,|0\rangle\,|b\rangle = \sum_j \left(\sqrt{1-\left|f(\widetilde{\lambda}_j)\right|^2}\,|0\rangle + f(\widetilde{\lambda}_j)\,|1\rangle\right)\beta_j\,|v_j\rangle\,. \tag{16.50}$$

Measuring the signal qubit, if the outcome is 1, the normalized vector stored in the system register is

$$|\psi\rangle \propto \sum_j \beta_j f(\widetilde{\lambda}_j)\,|v_j\rangle \approx f(H)\,|b\rangle\,. \tag{16.51}$$

The discussion in Section 16.3 shows that this picture is simplest when two idealized assumptions hold: the relevant eigenvalues admit exact d-bit representations, and the input is decomposed in an eigenbasis that QPE can resolve cleanly. Once these assumptions are relaxed, the bookkeeping becomes more involved because one must track both phase-estimation error and the distribution over different eigencomponents. In the applications below we will therefore work mainly in this idealized regime, while keeping in mind that a complete analysis requires the more general bounds developed above. On the other hand, QSVT-based eigenvalue transformation does not require such idealized assumptions, and the analysis is much more straightforward.

16.5. Heisenberg-limited scaling

A central theme in quantum metrology is the trade-off between the precision of an estimate and the quantum resources consumed. This trade-off leads to scaling laws that characterize the efficiency of an estimation protocol.

Consider estimating a parameter φ encoded by a quantum process, given R uses of that process. The most direct strategy uses the R resources independently: prepare R identical probe states, apply the process to each, measure them individually, and analyze the results. According to the quantum Cramér-Rao bound (see **??**), for any unbiased estimator the variance satisfies $\mathbb{V} = \Omega(1/R)$. Thus the precision scales as $\epsilon = \mathcal{O}(R^{-1/2})$, and achieving precision ϵ requires $R = \mathcal{O}(\epsilon^{-2})$. This is known as the shot-noise-limited scaling (SNL) or the standard quantum limit (SQL), and it is optimal when probes are used independently (see also **??**). Quantum mechanics also permits strategies that surpass the SNL. By utilizing entanglement across the R probes, or using the resources sequentially on a single probe, one can increase the quantum Fisher information and achieve **Heisenberg-limited scaling** (HL) [GLM04, GLM06]. At the Heisenberg limit the precision scales as $\epsilon = \mathcal{O}(R^{-1})$, corresponding to the quadratic improvement $R = \mathcal{O}(\epsilon^{-1})$.

In quantum algorithms involving phase estimation, the parameter of interest is often an eigenphase φ of a Hamiltonian H, accessed through a unitary oracle $U(t) = e^{-iHt}$. In this setting the relevant resource is the total evolution time T. The scaling laws can be stated in terms of T: the SNL corresponds to $\epsilon = \mathcal{O}(T^{-1/2})$, whereas the HL corresponds to $\epsilon = \mathcal{O}(T^{-1})$. The term "Heisenberg-limited" reflects the time–energy heuristic $\Delta E\, \Delta t \geq 1/2$ (in units where $\hbar = 1$), which suggests that resolving an energy to uncertainty ΔE requires time on the order of $1/\Delta E$, matching the $1/T$ scaling.

To connect these scaling laws to information-theoretic bounds, we relate the mean-square error and the variance via $\mathbb{V} \approx \epsilon^2$. Under this identification, the SNL corresponds to $\mathbb{V} = \mathcal{O}(T^{-1})$ (equivalently $T = \mathcal{O}(\epsilon^{-2})$), and the HL corresponds to $\mathbb{V} = \mathcal{O}(T^{-2})$ (equivalently $T = \mathcal{O}(\epsilon^{-1})$).

When estimating a circular quantity such as a phase $\varphi \in [0, 2\pi)$, the standard variance depends on an arbitrary choice of reference angle. A rotation-invariant alternative is the **Holevo variance** $\mathbb{V}_H$. Given an estimator $\hat{\varphi}$, it is defined by

$$\mathbb{V}_H(\hat{\varphi}) = \frac{1}{|\mathbb{E}(e^{i\hat{\varphi}})|^2} - 1. \tag{16.52}$$

Asymptotically, if the distribution of the estimator is sufficiently peaked around the true value, the Holevo variance coincides with the standard variance $\mathbb{V}(\hat{\varphi})$. Specifically, if we choose coordinates such that the mean error is zero, Taylor expansion shows [BHB+09]:

$$\begin{aligned} \mathbb{V}_H(\hat{\varphi}) &= \frac{1}{1 - \mathbb{E}(\hat{\varphi}^2) + \mathcal{O}(\mathbb{E}(\hat{\varphi}^4))} - 1 \\ &= \mathbb{V}(\hat{\varphi}) + \mathcal{O}\left(\mathbb{V}(\hat{\varphi})^2\right). \end{aligned} \tag{16.53}$$

Thus, for asymptotic scaling statements, the Holevo variance serves as a proxy for the phase uncertainty. We now formalize scaling in terms of how $\mathbb{V}_H$ behaves as a function of the total evolution time T.

Definition 16.10. *Let a quantum phase estimation protocol use a sequence of evolution times $\{t_1, \ldots, t_N\}$ to produce an estimator $\hat{\varphi}$ for an unknown phase φ^*. Let $T = \sum_{p=1}^{N} |t_p|$ be the total evolution time. Assume the estimator has bias at most $\widetilde{\mathcal{O}}(T^{-\beta/2})$. We characterize the scaling by the behavior of the Holevo variance as T increases:*

$$\mathbb{V}_H(\hat{\varphi}) = \widetilde{\mathcal{O}}\left(T^{-\beta}\right). \tag{16.54}$$

*For quantum phase estimation, $\beta = 1$ is referred to as **shot-noise-limited scaling**, and $\beta = 2$ is called **Heisenberg-limited scaling**.*

The Heisenberg limit ($\beta = 2$) is the best possible scaling in this model. It follows from the quantum Cramér–Rao bound (QCRB) [BC94], which lower bounds the variance of any locally unbiased estimator [BHB+09, WK97, WK98].

THEOREM 16.11 (Optimality of Heisenberg-limited scaling). *Consider a quantum protocol with the following properties:*

1. *It applies a controlled unitary U with total evolution time $T = \sum_{p=1}^{N} |t_p|$ to an eigenstate $|\psi\rangle$ with eigenvalue $e^{i\varphi^*}$.*
2. *It applies an arbitrary sequence of unitary operations $\{G_i\}$ on control qubits or on an arbitrary number of ancillary qubits, subject to the constraint that each G_i acts trivially on the target state, i.e., $G_i |\psi\rangle = |\psi\rangle$.*

Then for any estimator $\hat{\varphi}$ that is locally unbiased at φ^*, the variance satisfies

$$\mathbb{V}(\hat{\varphi}) \geq \frac{1}{4T^2}. \tag{16.55}$$

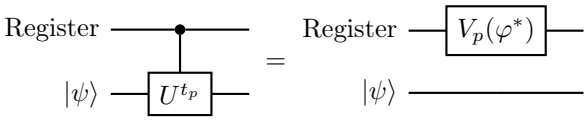
Moreover, if the estimator is sufficiently concentrated so that Eq. (16.53) *applies (in particular, $\mathbb{V}(\hat{\varphi}) = o(1)$), then*

$$\mathbb{V}_H(\hat{\varphi}) \geq \frac{1 - o(1)}{4T^2}, \tag{16.56}$$

and hence no protocol in this model can achieve $\beta > 2$ in Definition 16.10.

Proof. We apply the QCRB directly to the phase parameter. The constraints ensure that the target state $|\psi\rangle$ remains unchanged because $U|\psi\rangle = e^{i\varphi^*}|\psi\rangle$ and the gates G_i act trivially on it. Thus, we can focus on the evolution of the register (control and ancilla qubits). The application of the controlled unitary $C(U^{t_p})$ induces an effective phase rotation on the register:

(16.57)



The effective unitary can be written as $V_p(\varphi^*) = e^{it_p\varphi^* H_p}$, where H_p is the generator of the phase shift on the register. Its spectral norm is bounded as $\|H_p\| \leq 1$.

Let $|\Psi_0\rangle$ be the initial state of the register. The final state $|\Psi(\varphi^*)\rangle$, parameterized by φ^*, is

$$|\Psi(\varphi^*)\rangle = G_N V_N(\varphi^*) G_{N-1} \cdots V_1(\varphi^*) G_0 |\Psi_0\rangle . \tag{16.58}$$

For a pure state, the QFI is given by

$$F_Q(\varphi^*) = 4\left(\langle\dot{\Psi}|\dot{\Psi}\rangle - |\langle\Psi|\dot{\Psi}\rangle|^2\right) \leq 4\left\||\dot{\Psi}(\varphi^*)\rangle\right\|^2, \tag{16.59}$$

where $|\dot{\Psi}(\varphi^*)\rangle = \frac{\partial}{\partial\varphi^*}|\Psi(\varphi^*)\rangle$.

We compute the derivative using the product rule. Let $U_{\text{total}}(\varphi^*)$ be the total evolution operator. Then $|\dot{\Psi}(\varphi^*)\rangle = \frac{\partial U_{\text{total}}(\varphi^*)}{\partial\varphi^*}|\Psi_0\rangle$. We bound the norm using the triangle inequality:

$$\left\||\dot{\Psi}(\varphi^*)\rangle\right\| \leq \sum_{p=1}^{N} \left\| G_N V_N \cdots G_p \left(\frac{\partial V_p(\varphi^*)}{\partial\varphi^*}\right) G_{p-1} \cdots G_0 |\Psi_0\rangle \right\|. \tag{16.60}$$

Since G_i and $V_j(\varphi^*)$ are unitary, they preserve the norm. Using the submultiplicativity of the operator norm and $\||\Psi_0\rangle\| = 1$, we have:

$$\left\||\dot{\Psi}(\varphi^*)\rangle\right\| \leq \sum_{p=1}^{N} \left\| \frac{\partial V_p(\varphi^*)}{\partial\varphi^*} \right\|. \tag{16.61}$$

The derivative of $V_p(\varphi^*)$ is $\frac{\partial V_p(\varphi^*)}{\partial\varphi^*} = it_p H_p V_p(\varphi^*)$. The norm is bounded by

$$\left\| \frac{\partial V_p(\varphi^*)}{\partial\varphi^*} \right\| = |t_p| \, \|H_p V_p(\varphi^*)\| \leq |t_p| \, \|H_p\| \leq |t_p|. \tag{16.62}$$

Substituting this back, we obtain

$$\left\||\dot{\Psi}(\varphi^*)\rangle\right\| \leq \sum_{p=1}^{N} |t_p| = T. \tag{16.63}$$

Consequently, the QFI is bounded by $F_Q(\varphi^*) \le 4T^2$. The QCRB then implies

$$\mathbb{V}(\hat{\varphi}) \ge \frac{1}{F_Q(\varphi^*)} \ge \frac{1}{4T^2}. \tag{16.64}$$

If additionally $\mathbb{V}(\hat{\varphi}) = o(1)$ so that Eq. (16.53) applies, then $\mathbb{V}_H(\hat{\varphi}) = \mathbb{V}(\hat{\varphi})\,(1 + o(1))$, and the same T^{-2} lower bound holds for the Holevo variance. □

A minor subtlety in interpreting Theorem 16.11 is that the usual (linear) variance $\mathbb{V}(\hat{\varphi})$ depends on a choice of branch cut for the representative of $\hat{\varphi} \in [0, 2\pi)$, and can therefore be large even when $\hat{\varphi}$ is accurate modulo 2π. This issue is most relevant when proving **upper** bounds, where one wishes to compare an estimator to the true phase in a rotation-invariant manner (hence the use of the Holevo variance in Definition 16.10).

For the lower bound in Theorem 16.11, however, the argument is local: the QCRB constrains any estimator that is locally unbiased at φ^* through derivatives of the likelihood in a neighborhood of φ^*. Consequently, the branch-cut ambiguity does not affect the scaling conclusion. In particular, when the estimator is sufficiently concentrated so that Eq. (16.53) applies, the lower bound transfers directly to $\mathbb{V}_H(\hat{\varphi})$ up to $1 + o(1)$ factors, as stated in Theorem 16.11.

This result shows that $\beta = 2$ is the optimal scaling that can be achieved by any phase estimation protocol under these conditions, regardless of whether the protocol uses entanglement or intersperses the queries with additional unitaries (provided these operations act trivially on the eigenstate whose phase is being estimated). Subsequent work shows that, under additional regularity assumptions and for appropriate covariant measurements, one can optimize the constant factor in front of T^{-2}, with an optimal constant of order π^2 in this setting [GDDWB20].

We next show that Heisenberg-limited scaling is achievable. We illustrate this using the quantum Fourier transform based phase estimation scheme described above.

Proposition 16.12. *Quantum Fourier transform based phase estimation using the Fejér kernel, as described in Fig. 16.3, achieves Heisenberg-limited scaling according to Definition 16.10 as the number of bits of precision d increases, provided the register size is $t = d + 2$ and $\hat{\varphi}$ is taken to be the median of $\Theta(d)$ repetitions of the estimates.*

Proof. We analyze the performance in terms of the desired precision $\epsilon = 2^{-d}$. With register size $t = d + 2$, a single execution uses controlled powers up to $U^{2^{t-1}}$, so the evolution time is $\Theta(2^t) = \Theta(2^d)$. Repeating this procedure $\Theta(d)$ times gives total evolution time

$$T = \sum_i |t_i| = \Theta(d2^d). \tag{16.65}$$

With register size $t = d + 2$, a single run yields an estimate within error 2^{-d} of φ^* with probability at least $1/4$. By taking the median of $\Theta(d)$ repetitions, the failure probability, i.e., the probability that the resulting estimate $\hat{\varphi}$ is more than $\mathcal{O}(2^{-d})$ away from φ^* is suppressed exponentially. The Chernoff bound applied to the median gives $\mathbb{P}(\text{fail}) = e^{-\Theta(d)} = \mathcal{O}(2^{-d})$.

First, we analyze the bias. The maximum possible error is bounded by 2π.

$$|\mathbb{E}(\hat{\varphi} - \varphi^*)| \le \mathbb{E}(|\hat{\varphi} - \varphi^*|) \le \epsilon + 2\pi\,\mathbb{P}(\text{fail}) = \mathcal{O}(2^{-d}). \tag{16.66}$$

We relate this to the total time T. Since $T = \Theta(d2^d)$, we have $2^{-d} = \Theta(d/T)$. The bias is $\mathcal{O}(d/T)$. The requirement for $\beta = 2$ in Definition 16.10 is $\widetilde{\mathcal{O}}(T^{-1})$. Since d is logarithmic in T, this holds in the $\widetilde{\mathcal{O}}(\cdot)$ sense.

Next, we analyze the variance. We account for the contribution from both successful and failed estimations.

$$\mathbb{V}(\hat{\varphi}) \leq \mathbb{E}((\hat{\varphi} - \varphi^*)^2) \leq \epsilon^2 + (2\pi)^2 \, \mathbb{P}(\text{fail}). \tag{16.67}$$

Using statistical amplification in **??**, if we choose the constant implicit in $\Theta(d)$ repetitions large enough, we may ensure $\mathbb{P}(\text{fail}) = \mathcal{O}(2^{-2d})$, and hence $\mathbb{V}(\hat{\varphi}) = \mathcal{O}(2^{-2d})$. Using Eq. (16.53), we obtain

$$\mathbb{V}_H(\hat{\varphi}) = \mathcal{O}(2^{-2d}) = \mathcal{O}(d^2 T^{-2}) = \widetilde{\mathcal{O}}(T^{-2}). \tag{16.68}$$

This establishes Heisenberg-limited scaling. □

Theorem 16.11 gives an information-theoretic lower bound on the variance. While the QFT-based approach achieves the optimal scaling exponent, it does not saturate the constant factor in Theorem 16.11 because of the logarithmic overhead and the shape of the Fejér kernel. One can improve the constant by optimizing the input state. For example, using input states derived from the Kaiser window [GDDWB20, BSG+24] leads to a Holevo variance bound of the form

$$\mathbb{V}_H(\hat{\varphi}) \leq \tan^2\left(\frac{\pi}{T+2}\right) \approx \frac{\pi^2}{T^2}. \tag{16.69}$$

This demonstrates that the exact Heisenberg limit (including the preconstant) is achievable. Furthermore, practical optimizations can improve constant factors. For instance, using directionally controlled rotations $V(t) := |0\rangle\langle 0| \otimes U^\dagger(t) + |1\rangle\langle 1| \otimes U(t)$ instead of the standard controlled unitary $\mathrm{CU}(t)$ effectively doubles the phase kickback per query ($e^{\pm i\varphi t}$ versus $e^{i\varphi t}$), which can halve the required evolution time [BGB+18].

16.6. Amplitude estimation

Let $|\psi_0\rangle$ be prepared by an oracle U_{ψ_0}, i.e., $U_{\psi_0}|0^n\rangle = |\psi_0\rangle$. Assume that

$$|\psi_0\rangle = \sqrt{\mathbb{P}(0)}\,|\psi_{\text{good}}\rangle + \sqrt{1-\mathbb{P}(0)}\,|\psi_{\text{bad}}\rangle\,, \tag{16.70}$$

where $|\psi_{\text{bad}}\rangle$ is orthogonal to $|\psi_{\text{good}}\rangle$, and $\sqrt{\mathbb{P}(0)} = \sin\frac{\theta}{2}$. The goal of amplitude estimation is to estimate $\mathbb{P}(0)$ to additive precision ϵ. If $\mathbb{P}(0)$ is bounded away from 0 and 1, then estimating it by Monte Carlo sampling requires $\mathcal{N} = \mathcal{O}(\epsilon^{-2})$ samples.

Let $G = R_{\psi_0} R_{\text{good}}$ be the Grover operator as in Section 11.2. Then in the basis $\mathcal{B} = \{|\psi_{\text{good}}\rangle, |\psi_{\text{bad}}\rangle\}$, the subspace $\mathcal{H} = \operatorname{span}\mathcal{B}$ is an invariant subspace of G. Recall the computation of **??**, the matrix representation is

$$[G]_{\mathcal{B}} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}. \tag{16.71}$$

Its two eigenstates are

$$|\psi_\pm\rangle = \frac{1}{\sqrt{2}}\left(|\psi_{\text{good}}\rangle \pm i\,|\psi_{\text{bad}}\rangle\right), \tag{16.72}$$

with eigenvalues $e^{\pm i\theta}$, respectively.

Therefore the problem of estimating θ can be solved with phase estimation with an imperfect initial state. Note that

$$|\langle\psi_0|\psi_+\rangle|^2 = \frac{1}{2}\left|\sin\frac{\theta}{2} + i\cos\frac{\theta}{2}\right|^2 = \frac{1}{2} = |\langle\psi_0|\psi_-\rangle|^2\,. \tag{16.73}$$

Consider a QPE circuit with t ancilla qubits and (in the Fourier-based implementation) controlled powers up to $G^{2^{t-1}}$, so that the total "Grover time" is $T = \Theta(2^t)$. Then each execution with the system register will be in $|\psi_+\rangle$ or $|\psi_-\rangle$ states each with probability 1/2. This corresponds to the estimation of $\pm\theta$, respectively, and no postselection is needed.

Let

$$t = d + \lceil \log \delta^{-1} \rceil \tag{16.74}$$

be the number of ancilla qubits with $\epsilon' = 2^{-d}$. Then QPE obtains an estimate denoted by $\widetilde{\theta}$, which approximates θ to precision ϵ' with success probability $1-\delta$. Define the corresponding estimate $\widetilde{\mathbb{P}}(0) := \sin^2 \frac{\widetilde{\theta}}{2}$. Since $\mathbb{P}(0) = \sin^2 \frac{\theta}{2}$, we have

$$\begin{aligned} &\sin^2 \frac{\widetilde{\theta}}{2} - \sin^2 \frac{\theta}{2} \\ =& \sin^2 \frac{\widetilde{\theta}-\theta}{2} \cos^2 \frac{\theta}{2} + \cos^2 \frac{\widetilde{\theta}-\theta}{2} \sin^2 \frac{\theta}{2} + 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} \sin \frac{\widetilde{\theta}-\theta}{2} \cos \frac{\widetilde{\theta}-\theta}{2} - \sin^2 \frac{\theta}{2} \\ =& \sin \frac{\theta}{2} \cos \frac{\theta}{2} \sin(\widetilde{\theta}-\theta) + \left(1 - 2\sin^2 \frac{\theta}{2}\right) \sin^2 \frac{\widetilde{\theta}-\theta}{2}. \end{aligned} \tag{16.75}$$

Using the fact that $\left|\sin(\widetilde{\theta}-\theta)\right| \le \left|\widetilde{\theta}-\theta\right| \le \epsilon'$, we have

$$\left|\widetilde{\mathbb{P}}(0) - \mathbb{P}(0)\right| \le \sqrt{\mathbb{P}(0)(1-\mathbb{P}(0))}\,\epsilon' + |1-2\mathbb{P}(0)|\,\frac{\epsilon'^2}{4}. \tag{16.76}$$

Let ϵ' be sufficiently small. If $\mathbb{P}(0)(1-\mathbb{P}(0)) = \Omega(1)$, we can choose $\epsilon' = \mathcal{O}(\epsilon)$, and the total complexity of QPE is $\mathcal{O}(\epsilon^{-1})$.

If $\mathbb{P}(0)$ is small, it is natural to estimate $\mathbb{P}(0)$ to multiplicative accuracy ϵ instead. Using

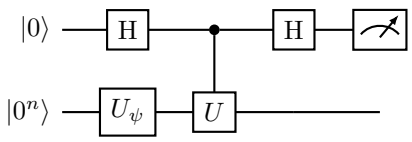
$$\left|\widetilde{\mathbb{P}}(0) - \mathbb{P}(0)\right| \approx \sqrt{\mathbb{P}(0)}\epsilon' < \mathbb{P}(0)\epsilon, \tag{16.77}$$

we take $\epsilon' = \sqrt{\mathbb{P}(0)}\,\epsilon$. Therefore the runtime of QPE is $\mathcal{O}(\mathbb{P}(0)^{-\frac{1}{2}}\epsilon^{-1})$. If $\mathbb{P}(0)$ is estimated to multiplicative accuracy ϵ using Monte Carlo sampling, the number of samples is $\mathcal{N} = \mathcal{O}(\mathbb{P}(0)^{-1}\epsilon^{-2})$.

In terms of the resource T used in Definition 16.10, phase estimation yields an estimate of θ with error $\mathcal{O}(1/T)$, which is Heisenberg-limited scaling for the eigenphase of G. Translating this into an estimate of $\mathbb{P}(0) = \sin^2(\theta/2)$ introduces a factor of $\sqrt{\mathbb{P}(0)(1-\mathbb{P}(0))}$ in the error, as shown above.

As for success probability, one may increase t so that a single run succeeds with probability $1-\delta$, which yields total Grover time polynomial in $1/\delta$ (for the standard Fourier-based QPE analysis, this is $\mathcal{O}(\epsilon'^{-1}\delta^{-1})$). A better method is to choose $t = d + \Theta(1)$ and repeat independently $\Theta(\log(1/\delta))$ times (taking the median), giving total Grover time $\widetilde{\mathcal{O}}(\epsilon'^{-1})$.

Example 16.13 (Amplitude estimation to accelerate Hadamard test)**.** Consider the circuit for the Hadamard test in Fig. 2.1 to estimate $\mathrm{Re}\,\langle\psi|U|\psi\rangle$. Let the initial state $|\psi\rangle$ be prepared by a unitary U_ψ, then the following combined circuit



maps $|0\rangle|0^n\rangle$ to

$$|\psi_0\rangle = \frac{1}{2}|0\rangle(|\psi\rangle + U|\psi\rangle) + \frac{1}{2}|1\rangle(|\psi\rangle - U|\psi\rangle) := \sqrt{\mathbb{P}(0)}\,|\psi_{\text{good}}\rangle + \sqrt{1-\mathbb{P}(0)}\,|\psi_{\text{bad}}\rangle\,, \tag{16.78}$$

and the goal is to estimate $\mathbb{P}(0)$. This also gives the implementation of the reflector R_{ψ_0}.

Note that R_{good} can be implemented by simply reflecting against the signal qubit, i.e.,

$$R_{\text{good}} = (I - 2|0\rangle\langle 0|) \otimes I^{\otimes n} = -Z \otimes I^{\otimes n}. \tag{16.79}$$

Then we can run QPE to the Grover unitary $G = R_{\psi_0} R_{\text{good}}$ to estimate $\mathbb{P}(0)$, and the circuit depth is $\mathcal{O}(\epsilon^{-1})$. ◇

Amplitude estimation is sometimes called Heisenberg-limited (despite the fact that it only measures a phase as an intermediate step) because it reduces to phase estimation of the Grover eigenphase and achieves $\mathcal{O}(1/T)$ scaling for estimating that phase as a function of Grover time T. This should be interpreted with care: the metrological Heisenberg limit in Section 16.5 concerns phase parameters generated by time evolution, whereas amplitude estimation reduces the problem to a phase-estimation task for a constructed unitary G.

NW: Make stuff connected to new amplitude amplification.

16.7. Iterative Phase Estimation and Cramér-Rao Bound

16.8. Kitaev's phase estimation algorithm

16.9. Eigenstate Projection for iterative phase estimation

Notes and further reading

In this chapter we have provided the foundation of quantum phase estimation including specific implementations for the most popular forms of phase estimation: Fourier-based QPE and Kitaev's Iterative QPE algorithm. However, the subfield of phase estimation (and more broadly quantum metrology) is much broader than the discussion that we can fit in here. We will now provide a guide to help the reader further explore other approaches to either coherent or iterative phase estimation methods.

In many works, Kitaev's phase estimation algorithm is often used synonymously with iterative phase estimation but this is certainly not true. As we discussed above, there are many ways that we can choose experiments to improve upon these results. Examples of this strategy include the work of [SHF13] which provides two iterative phase estimation algorithms that yield advantages over Kitaev. The first involves directly using Bayesian inference to learn the phase through a sequence of chosen with t taken uniformly on $[0, 2^n - 1]$. This result is shown to yield optimal scaling with the number of measurements. It requires, however, an inefficient inference algorithm that requires a uniform grid of 2^n potential phases. The second result in this work that is relevant for iterative phase estimation uses an efficient inference scheme for learning the phase at price of requiring a small constant factor more applications of the unitary than the measurement optimal Bayesian strategy [GFWC12, WG16].

Other approaches seeking to achieve the optimal constant factor for the variance for Heisenberg-limited scaling use adaptive experiment design to learn eigenphases in iterative frameworks [BWB01, HBB$^+$07, HS10, WG16]. The method explicitly tracks the prior distribution for the phase and then actively designs the experiment that minimizes the posterior variance. Heuristic optimization to this problem lead to scaling that approaches the optimal variance, often being within less than 10% deviation from the ultimate Heisenberg limit [RWS$^+$17].

A further technique that has been widely used for phase estimation is robust phase estimation [KLY15]. Robust Phase Estimation comes from a very different genealogy than other phase estimation protocols: it arises from the quantum characterization community rather than the algorithms or quantum optics communities. The approach has several advantages including the fact that it does not require any ancillary qubits (as opposed to the one needed for iterative phase estimation). It further can be used to characterize the unknown angles in the single qubit unitary $e^{-i(\alpha X+\beta Y+\gamma Z)}$ while achieving Heisenberg limited scaling. However, despite these strengths it is not known to achieve the ideal constant factor in the Heisenberg limit.

Further parallelized versions of phase estimation have been developed [KOS07, RWS+17, ACC+22]. The idea behind these approaches is inspired by NOON state methods developed in the quantum optics community [BHB+09] to prepare a state of the form $(|0\rangle |\psi\rangle^{\otimes N} + |1\rangle U^{\otimes N} |\psi\rangle^{\otimes N})/\sqrt{2}$ which will multiply the eigenphase observed by a factor of N. This not only allows phase estimation to be parallelized in cases where copies of the same eigenstate can be efficiently prepared, it also allows the simulation time to be shortened for cases where we are interested in estimating the eigenvalues of a Hamiltonian and in turn allow methods such as QDrift to allow phase estimation to be performed in constant depth [ACC+22]. Additionally, in cases where Bayesian methods are used to combine at least two independent phase estimation experiments it can be seen that even without an entangled state that N independent quantum computers can combine their data from phase estimation to achieve a variance of $\pi/2N$, which is the optimal scaling possible for non-entangled inputs [RWS+17].

There is the remaining issue about optimality of phase estimation. While iterative phase estimation can achieve the Heisenberg limit, the arguments given above do not actually show that this is the ultimate limit in precision estimation. This ultimate limit has been shown to coincide with the Heisenberg limit using more sophisticated arguments based on the quantum Cramèr-Rao bound, which provides a limit on the achievable variance for an unbiased estimator after optimizing over all possible quantum POVMs that could be used to learn the parameter [ZPDK10]. This shows that the central intuition first gleaned from the energy-time uncertainty principle in fact does represent the ultimate limits of learning a fixed phase. The quantum Cramèr-Rao bound can also be used to understand ultimate limits yielded also are understood for the case where we have a phase that stochastically varies over time. For example, in the case of Brownian motion scaling of $\mathcal{O}(\epsilon^{-2/3})$ is optimal and can be saturated by modifying the phase estimation algorithm to track the drift of phase [BHW13].

Subsequent work has also led to improvements in amplitude estimation. The use of priors has also been shown to provide a substantial advantage for estimation of expectation values of projectors. Using the fact that a projector can be easily square-rooted, it is possible to use LCU circuitry to subtract an apriori estimate of the variable to be estimated and then use amplitude estimation to estimate the smaller value. Since amplitude estimation is polynomially more efficient for small angle estimation, this can lead to a substantial improvement in such estimates and even surpass the Heisenberg-limit if such prior knowledge is provided [SDM+24]. Additionally, other works have used gradient estimation to obtain expectation of vector valued quantities to obtain quadratically better scaling than amplitude estimation allows in the number of variables [HWM+22]. These results illustrate that more work can be done to optimize quantum expectation value estimation beyond amplitude amplification.

Part IV

Application

CHAPTER 17

Quantum walks

Classical random walks and Markov chain Monte Carlo methods are powerful tools for designing randomized algorithms, with applications including sampling, optimization, and approximate counting. A motivating example comes from the early internet: to rank the importance of a website, one could start at a well-known site and walk randomly by following links with equal probability. The probability that this random walk visits a particular site reflects its importance within the network. This intuition underlies Google's PageRank algorithm.

Quantum walks provide a natural quantum generalization of classical random walks and have been employed to design quantum algorithms that outperform their classical counterparts in several scenarios. We focus on reversible Markov chains because they admit a Hermitian representation via the discriminant matrix. This allows tools from block encoding, qubitization, and phase estimation to act directly on the quantized version of the walk dynamics.

We then discuss continuous-time quantum walks, where the graph adjacency matrix defines a Hamiltonian and the quantum state evolves via Schrödinger dynamics. We introduce the glued trees problem which demonstrates an exponential query separation. In this example, classical random walks become trapped near the graph center with exponentially small probability of reaching the exit, while continuous-time quantum walks exhibit coherent transport through the column space and reach the exit in polynomial time.

17.1. Markov chains and classical random walks

We first review basic notions for Markov chains, using the column-stochastic convention to align with a quantum formulation.

Definition 17.1 (Markov chain)**.** *Let Σ be a state space. A Markov chain on Σ is a sequence of random variables $X_1, X_2, \ldots$ taking values in Σ, and the probability of moving to the next state depends only on the current state. More precisely,*

(17.1)
$$\mathbb{P}(X_{t+1} = i \mid X_t = j, X_{t-1} = i_{t-1}, \ldots, X_1 = i_1) = \mathbb{P}(X_{t+1} = i \mid X_t = j), \qquad \forall t \geq 1,\ i, j, i_1, \ldots, i_{t-1} \in \Sigma.$$

When the state space Σ is finite, we say the Markov chain is finite. In this case, we may identify Σ with $\{0, 1, \ldots, N-1\}$ where $N := |\Sigma|$, and denote

$$\mathbb{P}(X_{t+1} = i \mid X_t = j) = P_{ij}, \qquad \forall t \geq 1,\ i, j \in \Sigma. \tag{17.2}$$

Here P is a column-stochastic (left-stochastic) matrix: $\sum_i P_{ij} = 1$ for all j and $P_{ij} \geq 0$.

The **stationary distribution** of the Markov chain is an eigenvector π of the transition matrix P with eigenvalue 1:

$$P\pi = \pi, \qquad \pi_i \geq 0, \qquad \sum_i \pi_i = 1. \tag{17.3}$$

Given a stationary distribution π, the goal of quantum algorithms is to prepare the coherent version of π:

$$|\pi\rangle = \sum_i \sqrt{\pi_i}\,|i\rangle\,, \tag{17.4}$$

which is a normalized quantum state. For a classical observable O (i.e., O is diagonal in the computational basis), the expectation of O with respect to π can be computed as $\langle\pi|O|\pi\rangle$, which is the same as the classical expectation $\mathbb{E}_\pi(O) = \sum_i \pi_i O_{ii}$.

The stationary distribution need not be unique. Even if P has a unique stationary distribution, the direct sum $P' := P \oplus P$ has at least two linearly independent stationary distributions, corresponding to probability mass supported on the first or the second copy. This corresponds to a Markov chain on the disjoint union of two copies of the same state space.

A Markov chain is called **irreducible** if for any two states $i, j \in \Sigma$ there exists a $t \in \mathbb{N}$ such that $[P^t]_{ij} > 0$. Let $\mathcal{T}(i) := \{t \geq 1 : [P^t]_{ii} > 0\}$ be the set of return times to i. The **period** of i is defined as the greatest common divisor of $\mathcal{T}(i)$. A Markov chain is called **aperiodic** if the period of every state is one. For a finite, irreducible, and aperiodic Markov chain, there exists $t \in \mathbb{N}$ such that $[P^t]_{ij} > 0$ for all $i, j \in \Sigma$ [LP17, Proposition 1.7].

To proceed further, we first introduce the Perron theorem for matrices with positive entries [HJ91, Theorem 8.2.8]. This is a special case of the Perron–Frobenius theorem for nonnegative matrices.

THEOREM 17.2 (Perron). *Let $A \in \mathbb{R}^{N\times N}$ with all positive entries. Then it has a simple eigenvalue equal to its spectral radius $\rho(A)$. The corresponding eigenvector v can be chosen to have positive entries, i.e., $v_i > 0$ for all i, and is unique up to scaling. All other eigenvalues λ of A satisfy $|\lambda| < \rho(A)$.*

For stochastic matrices, the spectral radius is 1. Let us use Perron's theorem to show the existence of a **spectral gap**.

Proposition 17.3. *Let P be the transition matrix of a finite, irreducible and aperiodic Markov chain. Then it has a unique stationary distribution π with eigenvalue 1. Moreover, there exists $\gamma \in (0, 1]$ such that every eigenvalue $\lambda \neq 1$ of P satisfies $|\lambda| \leq 1 - \gamma$.*

PROOF. From the assumptions, there exists $t \in \mathbb{N}$ such that P^t is a positive matrix. Let π be the eigenvector of P^t corresponding to the unique maximal eigenvalue with positive entries. Since P^t is column-stochastic,

$$\sum_{i,j} (P^t)_{ij}\pi_j = \sum_j \pi_j. \tag{17.5}$$

Therefore the corresponding eigenvalue is 1. We may normalize π so that $\sum_i \pi_i = 1$. From

$$P^t(P\pi) = P(P^t\pi), \tag{17.6}$$

we find that $P\pi$ is also an eigenvector of P^t with eigenvalue 1. By the uniqueness of the eigenvector, we have $P\pi = \pi$, i.e., π is the unique stationary distribution of P. By the Perron theorem, all other eigenvalues of P^t have absolute value strictly less than 1, so there exists $\gamma' > 0$ such that they are all bounded in absolute value by $1 - \gamma'$. Therefore if there is an eigenvector v of P with eigenvalue $\lambda \neq 1$, then v is an eigenvector of P^t and $|\lambda|^t \leq 1 - \gamma'$. The result then follows once we define $1 - \gamma = (1 - \gamma')^{1/t}$. □

Throughout most of this chapter, we will study a narrower family of Markov chains known as **reversible Markov chains**, for which every allowed transition has a corresponding reverse transition.

Definition 17.4 (Reversibility, or detailed balance)**.** *A Markov chain is **reversible** if there is a stationary distribution π such that the transition matrix P satisfies the **detailed balance condition**:*

$$P_{ji}\pi_i = P_{ij}\pi_j, \qquad \forall i, j \in \Sigma. \tag{17.7}$$

One immediate consequence of reversibility is that π must be a stationary distribution, since summing over i yields $\sum_i P_{ji}\pi_i = \sum_i P_{ij}\pi_j = \pi_j$.

Example 17.5. Let Σ be the set of all n-bit strings $\{0,1\}^n$, and $E(i)$ be a real-valued function on Σ, which is called the energy of the state i. The stationary distribution of interest is the Gibbs distribution

$$\pi_i := \frac{e^{-\beta E(i)}}{Z}, \qquad Z := \sum_{i\in\Sigma} e^{-\beta E(i)}, \tag{17.8}$$

with inverse temperature $\beta > 0$.

The **Metropolis–Hastings Markov chain** is constructed as follows. Given a current state $i \in \Sigma$, pick a uniformly random bit position $\ell \in \{1, \dots, n\}$, and let j be the configuration obtained from i by flipping the ℓ-th bit. This defines a proposal kernel Q by

$$Q_{ji} := \frac{1}{n} \quad \text{if } i \text{ and } j \text{ differ in exactly one bit}, \qquad Q_{ji} := 0 \text{ otherwise}, \tag{17.9}$$

so that $Q_{ji} = Q_{ij}$. Accept the proposal with probability

$$\alpha_{ji} := \min\{1, e^{-\beta(E(j)-E(i))}\}. \tag{17.10}$$

The resulting transition matrix P is given by

$$P_{ji} := Q_{ji}\alpha_{ji} \quad (j \neq i), \qquad P_{ii} := 1 - \sum_{j\neq i} P_{ji}. \tag{17.11}$$

Then P is column-stochastic by construction, and π is stationary because P satisfies detailed balance: for $i \neq j$ with $Q_{ji} > 0$,

$$\begin{aligned} \pi_i P_{ji} &= \frac{e^{-\beta E(i)}}{Z} Q_{ji} \min\{1, e^{-\beta(E(j)-E(i))}\} \\ &= \frac{e^{-\beta E(j)}}{Z} Q_{ij} \min\{1, e^{-\beta(E(i)-E(j))}\} = \pi_j P_{ij}, \end{aligned} \tag{17.12}$$

and summing over i yields $\sum_i P_{ji}\pi_i = \pi_j$. ◇

We define the **discriminant matrix** associated with a Markov chain as

$$D := \sum_{i,j} \sqrt{P_{ij}P_{ji}}\, |i\rangle\langle j|, \tag{17.13}$$

which is real symmetric and hence Hermitian. For a reversible Markov chain, the stationary state can be encoded as an eigenvector of D. This is shown in the following proposition.

Proposition 17.6 (Discriminant matrix of a reversible Markov chain)**.** *For a finite, irreducible, aperiodic and reversible Markov chain, the coherent version of the stationary state in Eq.* (17.4) *is an eigenvector of the discriminant matrix D satisfying*

$$D\,|\pi\rangle = |\pi\rangle\,. \tag{17.14}$$

Furthermore, we have

$$D = \mathrm{diag}(\sqrt{\pi})P^\top \,\mathrm{diag}(\sqrt{\pi})^{-1} = \mathrm{diag}(\sqrt{\pi})^{-1} P\,\mathrm{diag}(\sqrt{\pi}). \tag{17.15}$$

Therefore the set of eigenvalues of P and the set of the eigenvalues of D are the same.

Proof. Direct computation shows

$$\langle i|\,D\,|\pi\rangle = \sum_j \sqrt{P_{ij}P_{ji}\pi_j} = \sum_j P_{ji}\sqrt{\pi_i} = \sqrt{\pi_i}. \tag{17.16}$$

In the second equality we use the detailed balance condition, and the third equality uses the column-stochasticity of P. Therefore $|\pi\rangle$ is an eigenvector of D with eigenvalue 1.

Next we will proceed to prove (17.15):

(17.17)

$$\mathrm{diag}(\sqrt{\pi})P^\top \,\mathrm{diag}(\sqrt{\pi})^{-1} = \sum_{i,j} \frac{\sqrt{\pi_i}}{\sqrt{\pi_j}} P_{ji}|i\rangle\langle j| = \sum_{i,j} \frac{\sqrt{P_{ji}\pi_i}}{\sqrt{P_{ij}\pi_j}} \sqrt{P_{ij}P_{ji}}|i\rangle\langle j| = \sum_{i,j} \sqrt{P_{ij}P_{ji}}|i\rangle\langle j| = D.$$

Here we again use the detailed balance condition. Similarly

(17.18)

$$\mathrm{diag}(\sqrt{\pi})^{-1} P\,\mathrm{diag}(\sqrt{\pi}) = \sum_{i,j} \frac{\sqrt{\pi_j}}{\sqrt{\pi_i}} P_{ij}|i\rangle\langle j| = \sum_{i,j} \frac{\sqrt{P_{ij}\pi_j}}{\sqrt{P_{ji}\pi_i}} \sqrt{P_{ij}P_{ji}}|i\rangle\langle j| = \sum_{i,j} \sqrt{P_{ij}P_{ji}}|i\rangle\langle j| = D.$$

□

Since the discriminant matrix is real and symmetric, D is diagonalizable with real eigenvalues and orthogonal eigenvectors.

$$D\,|v_j\rangle = \lambda_j\,|v_j\rangle\,, \quad \langle v_j|v_k\rangle = \delta_{jk}. \tag{17.19}$$

Proposition 17.6 immediately implies that the transition matrix P corresponding to a reversible Markov chain is also diagonalizable as $P|\lambda_j\rangle\!\rangle = \lambda_j|\lambda_j\rangle\!\rangle$. Here the (unnormalized) eigenvectors $|\lambda_j\rangle\!\rangle$ can be chosen as

$$|\lambda_j\rangle\!\rangle = \mathrm{diag}(\sqrt{\pi})\,|v_j\rangle\,. \tag{17.20}$$

We order the eigenvalues $\{\lambda_j\}$ in non-increasing order with $\lambda_0 = 1$ and $|v_0\rangle = |\pi\rangle$. Then $|\lambda_0\rangle\!\rangle = \pi$, viewed as a column vector.

The following is often referred to as the **convergence theorem** for Markov chains.

Theorem 17.7. *Let P be the transition matrix of a finite, irreducible and aperiodic Markov chain with stationary distribution π. Then there exists a constant $\gamma \in (0,1]$ and $C > 0$ such that for any initial probability distribution ρ,*

$$\left\|P^t\rho - \pi\right\|_1 \le C(1-\gamma)^t, \qquad t \in \mathbb{N}. \tag{17.21}$$

While we do not prove Theorem 17.7 directly (see e.g. [LP17, Theorem 4.9]), we will show a more refined result for reversible Markov chains.

Proposition 17.8. *Let P be the transition matrix of a finite, irreducible, aperiodic and reversible Markov chain on a state space Σ of size $N := |\Sigma|$, and let γ be its spectral gap. Let π be the unique stationary distribution. Then from any initial distribution ρ, for any $\delta > 0$, there exists a positive integer t^* such that*

$$\left\|P^t\rho - \pi\right\|_1 \leq \delta, \qquad t \geq t^*. \tag{17.22}$$

where

$$t^* = \left\lceil \log\left(\frac{\sqrt{N/\min_i \pi_i}}{\delta}\right) / \log\left(\frac{1}{1-\gamma}\right) \right\rceil. \tag{17.23}$$

Proof. Because P is diagonalizable with real eigenvalues $\{\lambda_j\}_{j=0}^{N-1}$, using Eq. (17.19), we can express the probability vector ρ in the eigenbasis of P as

$$\rho = \sum_{j=0}^{N-1} \alpha_j |\lambda_j\rangle\!\rangle = \sum_{j=0}^{N-1} \alpha_j \operatorname{diag}(\sqrt{\pi})\,|v_j\rangle\,. \tag{17.24}$$

Here we set $\lambda_0 = 1$ and $|\lambda_0\rangle\!\rangle = \pi$, and we define the spectral gap γ by $1 - \gamma := \max_{j\geq 1} |\lambda_j|$. We then have

$$\alpha_j = \langle v_j| \operatorname{diag}(\sqrt{\pi})^{-1}\rho\rangle\,. \tag{17.25}$$

In particular,

$$\alpha_0 = \sum_i \rho_i = 1, \tag{17.26}$$

For any positive integer t,

$$P^t\rho = \pi + \sum_{j=1}^{N-1} \alpha_j \lambda_j^t \operatorname{diag}(\sqrt{\pi})\,|v_j\rangle\,. \tag{17.27}$$

Then using the Cauchy-Schwarz inequality,

$$\left\|\operatorname{diag}(\sqrt{\pi})\,|v_j\rangle\right\|_1 \leq \sqrt{\sum_i \pi_i}\ \||v_j\rangle\| = 1, \tag{17.28}$$

and

$$\left\|P^t\rho - \pi\right\|_1 \leq \sum_{j=1}^{N-1} |\alpha_j|\,|\lambda_j|^t \leq (1-\gamma)^t \sum_{j=1}^{N-1} |\alpha_j| \leq (1-\gamma)^t\sqrt{N}\sqrt{\sum_{j=1}^{N-1} |\alpha_j|^2}. \tag{17.29}$$

Furthermore by the resolution of the identity and the fact that $\{|v_j\rangle\}$ is an orthonormal basis,

$$\sum_{j=1}^{N-1} |\alpha_j|^2 \leq \sum_{j=1}^{N-1} \langle \operatorname{diag}(\sqrt{\pi})^{-1}\rho|v_j\rangle\,\langle v_j|\operatorname{diag}(\sqrt{\pi})^{-1}\rho\rangle \leq \sum_i \rho_i^2\pi_i^{-1} \leq \frac{\sum_i \rho_i}{\min_i \pi_i} = \frac{1}{\min_i \pi_i}, \tag{17.30}$$

we obtain

$$\left\|P^t\rho - \pi\right\|_1 \leq (1-\gamma)^t\sqrt{\frac{N}{\min_i \pi_i}}. \tag{17.31}$$

For the ℓ_1 distance (or total variation distance) to be at most δ, it suffices to take

$$t \geq \frac{\log(\frac{\sqrt{N/\min_i \pi_i}}{\delta})}{\log\left(\frac{1}{1-\gamma}\right)}. \tag{17.32}$$

□

Using the fact that $\frac{1}{\log \frac{1}{1-\gamma}} \approx \frac{1}{\gamma}$ when γ is small, we find that the time t^* is inversely proportional to the spectral gap γ.

Example 17.9. Consider the simple random walk on the d-dimensional hypertorus $(\mathbb{Z}_L)^d$, viewed as a graph $G = (V, E)$ with vertex set $V = (\mathbb{Z}_L)^d$ and edges between vertices at graph distance one. Assume that $L \geq 3$ is odd, so that the walk is aperiodic, and note that $|V| = L^d$. The walker can move in each of the $2d$ cardinal directions.

For $x, y \in V$, let $\mathrm{dist}(x, y)$ denote the graph distance. In the column-stochastic convention,

$$P_{yx} := \mathbb{P}(X_{t+1} = y \mid X_t = x) = \begin{cases} \frac{1}{2d}, & \text{if } \mathrm{dist}(x,y) = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{17.33}$$

By symmetry the stationary distribution is uniform, i.e.,

$$\pi_x = \frac{1}{|V|}, \qquad x \in V. \tag{17.34}$$

It is stationary since $P\pi = \pi$. Moreover, for any $x, y \in V$ with $\mathrm{dist}(x, y) = 1$,

$$P_{yx}\pi_x = \frac{1}{2d\,|V|} = P_{xy}\pi_y. \tag{17.35}$$

Thus the detailed balance condition is satisfied and the walk is reversible.

Since P is symmetric, the discriminant matrix satisfies $D = P$. Let

$$T := \sum_{i \in \mathbb{Z}_L} |i+1\rangle\langle i|, \tag{17.36}$$

where addition is modulo L. Then

$$D = P = \frac{1}{2d} \sum_{\nu=1}^{d} I^{\otimes(\nu-1)} \otimes (T + T^\top) \otimes I^{\otimes(d-\nu)}. \tag{17.37}$$

As shown earlier during our discussion of the Fourier transform, such operators are diagonalized by the discrete Fourier transform on $\mathbb{Z}_L$. In particular,

$$\mathrm{DFT}^{\otimes d}\, D \left((\mathrm{DFT})^\dagger\right)^{\otimes d} = \sum_{k_1,\ldots,k_d \in \mathbb{Z}_L} |k_1 \ldots k_d\rangle\langle k_1 \ldots k_d| \frac{1}{d}\Big(\cos(2\pi k_1/L) + \cdots + \cos(2\pi k_d/L)\Big). \tag{17.38}$$

The largest eigenvalue occurs when $k_1 = k_2 = \cdots = k_d = 0$, and the second largest eigenvalue occurs when exactly one of the k_ν equals 1 or $L-1$. Thus

$$\gamma = \frac{1 - \cos(2\pi/L)}{d} = \frac{2\sin^2(\pi/L)}{d}. \tag{17.39}$$

The number of steps needed for the walk to approach the stationary distribution within ℓ_1 distance δ is therefore approximately

$$\frac{1}{\gamma}\log\left(\frac{\sqrt{|V|\,/\min_x \pi_x}}{\delta}\right) = \frac{1}{\gamma}\log\left(\frac{|V|}{\delta}\right) = \mathcal{O}\left(dL^2\log(|V|\,/\delta)\right). \tag{17.40}$$

◇

17.2. Block encoding of the discriminant matrix

Now that we have discussed the fundamentals of Markov Chains, we can examine how we would implement such a walk on a quantum computer. The simplest approach to do this is to build a block encoding of the discriminant matrix. This block-encoding can then be used to sample from the stationary distribution for the Markov chain. This allows us to represent the non-unitary process as a block of a larger unitary matrix.

Our first goal in the encoding of the elements of the Markov chain is to build a circuit that gives access to the entries of the transition matrix P. Assume that we have access to an oracle O_P that will construct a weighted superposition over all the neighbors of a vertex j in the graph as follows:

$$O_P\,|0^n\rangle\,|j\rangle = \sum_k \sqrt{P_{kj}}\,|k\rangle\,|j\rangle\,. \tag{17.41}$$

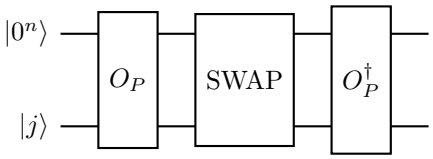
Since P is a left stochastic matrix, the right hand side is already a normalized vector. Hence the map defined by Eq. (17.41) is an isometry and can be extended to a unitary O_P without introducing an additional signal qubit. We also need the n-qubit SWAP operation:

$$\mathrm{SWAP}\,|i\rangle\,|j\rangle = |j\rangle\,|i\rangle\,, \tag{17.42}$$

which swaps the value of the two registers in the computational basis, and can be directly implemented using n SWAP operations between two qubits, and in turn $3n$ CNOT operations. The role of the SWAP operation is to easily prepare $D = \sum_{ij}\sqrt{P_{ij}P_{ji}}|i\rangle\langle j|$ from (17.13), since we need to ensure that the P_{ij} elements get paired with their transposes. This pairing is guaranteed by the use of a SWAP operation as seen in the following proposition.

Proposition 17.10. *Let D be a discriminant matrix associated with a transition matrix $P \in \mathbb{R}^{N\times N}$ with $N = 2^n$, then the following circuit provides a Hermitian block encoding of the matrix D via $(\langle 0^n|\otimes I)U_D(|0^n\rangle\otimes I) = D$ where*

$U_D :=$



Proof. Clearly U_D is unitary and Hermitian. Now we compute as before

$$|0^n\rangle\,|j\rangle \xrightarrow{O_P} \sum_k \sqrt{P_{kj}}\,|k\rangle\,|j\rangle \xrightarrow{\mathrm{SWAP}} \sum_k \sqrt{P_{kj}}\,|j\rangle\,|k\rangle\,. \tag{17.43}$$

Meanwhile

$$|0^n\rangle\,|i\rangle \xrightarrow{O_P} \sum_{k'} \sqrt{P_{k'i}}\,|k'\rangle\,|i\rangle\,. \tag{17.44}$$

So the inner product gives

$$\langle 0^n| \langle i| U_D |0^n\rangle |j\rangle = \sum_{k,k'} \sqrt{P_{k'i}P_{kj}}\delta_{j,k'}\delta_{i,k} = \sqrt{P_{ji}P_{ij}} = D_{ij}. \tag{17.45}$$

This proves the claim. □

How can we use this to solve a computational problem? Since D is a Hermitian matrix, it can be viewed as a Hamiltonian. This implies that we can use an algorithm such as quantum phase estimation to project onto the principal eigenvector corresponding to the equilibrium distribution π.

Proposition 17.11. *Let $P \in \mathbb{R}^{N\times N}$ be the transition matrix that corresponds to a finite, irreducible, aperiodic and reversible Markov chain, and let π be the stationary distribution and $|\pi\rangle$ be the coherent version of the distribution. Assume that we are provided a quantum state $|\psi\rangle \in \mathbb{C}^N$ such that $|\langle\psi|\pi\rangle| \geq \sqrt{1-\delta^2}$. Let $\gamma > 0$ be the spectral gap of P. Then for any $\epsilon > 0$ and $0 < \delta < 1$, there exists a quantum algorithm that can prepare a state $|\tilde{\pi}\rangle$ such that $|\langle\pi|\tilde{\pi}\rangle| \geq 1-\epsilon$ with success probability at least $1 - 3\delta^2/2$ using*

$$\mathcal{O}\left(\frac{\log(1/(\epsilon\gamma))\log(1/\delta)}{\gamma}\right) \tag{17.46}$$

queries to O_P and $O_P^\dagger$.

Proof. According to Proposition 17.6, the spectral gap of P is equal to the gap between the largest eigenvalue and the next largest eigenvalue of the discriminant matrix D.

As D is a Hermitian matrix with $\|D\| \leq 1$, we can implement a unitary W such that $\|W - e^{-iD\pi/2}\| \leq \epsilon_0$ using

$$\mathcal{C}_W = \mathcal{O}\left(1 + \frac{\log(1/\epsilon_0)}{\log\log(1/\epsilon_0)}\right) \tag{17.47}$$

queries to a block encoding of D. By Proposition 17.10, implementing such a block encoding uses $\mathcal{O}(1)$ calls to O_P and $O_P^\dagger$.

Phase estimation with precision $\mathcal{O}(\gamma)$ distinguishes the principal eigenphase (corresponding to the eigenvalue 1 of D) from the rest of the spectrum using $\mathcal{O}(1/\gamma)$ controlled applications of W. Using statistical amplification, achieving failure probability at most $\delta^2/2$ for this discrimination step incurs an additional factor $\mathcal{O}(\log(1/\delta))$.

Since $|\langle\psi|\pi\rangle|^2 \geq 1-\delta^2$, the probability of not projecting onto $|\pi\rangle$ due to the initial overlap is at most δ^2. Therefore, by the union bound, the overall probability of failure is at most $\delta^2 + \delta^2/2 = 3\delta^2/2$.

As a last step, we need to discuss the quality of the eigenstate that we prepare. Here we note that the eigenvalue 1 of D is simple for the equilibrium distribution. Using eigenvector perturbation bounds for a simple eigenvalue in Theorem 7.9, we find that

$$1 - |\langle\pi|\tilde{\pi}\rangle| = \mathcal{O}(\epsilon_0/\gamma). \tag{17.48}$$

Therefore it suffices to choose $\epsilon_0 = \Theta(\epsilon\gamma)$.

Combining the above bounds yields an overall query complexity of

$$\mathcal{O}\left(\frac{\log(1/(\epsilon\gamma))\log(1/\delta)}{\gamma}\right) \tag{17.49}$$

in calls to O_P and $O_P^\dagger$. □

This shows that we can prepare the stationary distribution using quantum phase estimation (QPE), providing an alternative to the classical approach of iteratively applying the transition matrix. However, as Proposition 17.8 indicates, the overall cost of this quantum method remains comparable to that of classical algorithm (with respect to the gap γ). This motivates a deeper question: can we truly accelerate convergence using quantum analogues of Markov chains? We will see that it is possible to quadratically amplify the spectral gap, thereby demonstrating that quantum effects can fundamentally speed up the mixing of Markov processes.

17.3. Szegedy's quantum walk and qubitization

For a Markov chain defined on a graph $G = (V, E)$ with $|V| = N = 2^n$, Szegedy constructed the following quantum walk operator [Sze04], which can be used to achieve quadratic speedup for a range of problems, using a strategy similar to that in Grover type algorithms in Chapter 11. For any input vertex j, we construct a state $O_P |0^n\rangle |j\rangle$, which is the coherent version of the probability distribution over the neighboring vertices of j according to the transition matrix P. It then swaps the role of the outgoing and incoming vertices: $\mathrm{SWAP} \cdot O_P |0^n\rangle |j\rangle$. These operators are exactly the same ones used above for block encoding the discriminant matrix D. However, as we will see below, after re-arranging the terms, we can quadratically increase the effective gap of the Markov chain. This means that we can achieve a fundamental advantage by using quantum as opposed to classical walks to solve problems.

Using the O_P oracle in Eq. (17.41) and the multi-qubit SWAP gate, we can define two sets of quantum states

$$\begin{aligned} |\psi_j^1\rangle &= O_P |0^n\rangle |j\rangle = \sum_k \sqrt{P_{kj}} |k\rangle |j\rangle , \\ |\psi_j^2\rangle &= \mathrm{SWAP}(O_P |0^n\rangle |j\rangle) = \sum_k \sqrt{P_{kj}} |j\rangle |k\rangle . \end{aligned} \tag{17.50}$$

This gives rise to two $2n$-qubit projection operators and reflection operators

$$\Pi_l = \sum_{j\in[N]} |\psi_j^l\rangle \langle\psi_j^l| , \quad R_{\Pi_l} = 2\Pi_l - I_{2n}, \quad l = 1, 2. \tag{17.51}$$

Using the resolution of identity, the reflection operators can also be written as

$$R_{\Pi_1} = O_P(Z_\Pi \otimes I_n)O_P^\dagger, \qquad Z_\Pi := 2|0^n\rangle\langle 0^n| - I_n. \tag{17.52}$$

Similarly

$$R_{\Pi_2} = \mathrm{SWAP}\, O_P(Z_\Pi \otimes I_n)O_P^\dagger \,\mathrm{SWAP}\, . \tag{17.53}$$

Szegedy's quantum walk operator is defined as the product of these two reflection operators

$$\mathcal{U}_Z = R_{\Pi_2} R_{\Pi_1} , \tag{17.54}$$

which is a rotation operator that resembles that in Grover's algorithm.

We first note that

$$O_P^\dagger \mathcal{U}_Z O_P = \Big(O_P^\dagger \,\mathrm{SWAP}\, O_P(Z_\Pi \otimes I_n)\Big)^2 =: O_Z^2, \tag{17.55}$$

where the circuit for $O_Z := O_P^\dagger \,\mathrm{SWAP}\, O_P(Z_\Pi \otimes I_n)$ is shown in Fig. 17.1 and is often called the **walk operator**. Let $U_D = O_P^\dagger \,\mathrm{SWAP}\, O_P$ be the Hermitian block encoding of D in Proposition 17.10. Compare this with Fig. 10.1, we find that the walk operator O_Z is exactly the qubitization circuit associated with the block encoding of D. Therefore, Szegedy's quantum walk operator is the same

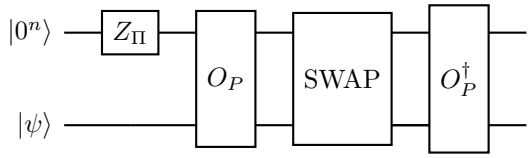


FIGURE 17.1. Circuit implementing one step of the O_Z operator which corresponds to the Szegedy walk. This circuit is precisely the same form of the circuit to that used in qubitization-based simulation algorithms.

as a block encoding of $T_2(D)$, up to a matrix similarity transformation, where $T_2(x) = 2x^2 - 1$ is the 2nd order Chebyshev polynomial. Furthermore, the matrix power O_Z^{2k} provides a block encoding of the Chebyshev matrix polynomial $T_{2k}(D)$.

From the eigendecomposition $D|v_i\rangle = \lambda_i |v_i\rangle$, for each $|v_i\rangle$, the associated basis in the 2-dimensional subspace is $\mathcal{B}_i = \{|0^n\rangle |v_i\rangle, |\perp_i\rangle\}$. Then the qubitization procedure gives

$$[O_Z]_{\mathcal{B}_i} = \begin{pmatrix} \lambda_i & -\sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & \lambda_i \end{pmatrix}. \tag{17.56}$$

The eigenvalues of O_Z in the 2×2 matrix block are

$$e^{\pm i \arccos(\lambda_i)}. \tag{17.57}$$

This relation is important for the following reasons. By Proposition 17.6, if a Markov chain is reversible and ergodic, the eigenvalues of D and P are the same. In particular, the largest eigenvalue of D is unique and is equal to 1, and the second largest eigenvalue of D is $1-\gamma$, where $\gamma > 0$ is called the spectral gap. Since $\arccos(1) = 0$, and $\arccos(1-\gamma) \approx \sqrt{2\gamma}$, we find that the spectral gap of O_Z on the unit circle is in fact $\mathcal{O}(\sqrt{\gamma})$ instead of $\mathcal{O}(\gamma)$. This is called the **spectral gap amplification**. Thus, applying quantum phase estimation to O_Z instead of $e^{-iD\pi/2}$ as in Proposition 17.11 yields a quadratic improvement in the dependence on the spectral gap γ for preparing the stationary distribution.

Here is an example illustrating how Szegedy's walk construction can be used to prepare a Gibbs state when the Hamiltonian is explicitly diagonalizable.

Example 17.12 (Preparing a Gibbs state for explicitly diagonalizable Hamiltonians)**.** Assume that the Hamiltonian $H \in \mathbb{C}^{2^n \times 2^n}$ can be written as

$$H = U\mathcal{E}U^\dagger, \tag{17.58}$$

where U is a unitary and $\mathcal{E} = \sum_{j=0}^{2^n-1} E(j)|j\rangle\langle j|$ is a diagonal matrix of energies. Assume furthermore that the map $|j\rangle \mapsto E(j)$ can be computed in poly(n) time on a quantum computer, and that U can be implemented in poly(n) time. For inverse temperature $\beta > 0$, the Gibbs state is

$$\rho_\beta := \frac{e^{-\beta H}}{\mathrm{Tr}\left[e^{-\beta H}\right]} = U\left(\frac{e^{-\beta \mathcal{E}}}{\mathrm{Tr}\left[e^{-\beta \mathcal{E}}\right]}\right)U^\dagger. \tag{17.59}$$

Thus preparing ρ_β reduces to preparing the classical Gibbs distribution over the eigenbasis of $\mathcal{E}$ and conjugating by U.

Let $\Sigma = \{0,1\}^n$. Consider the Metropolis–Hastings chain in Example 17.5, with symmetric proposal kernel Q given by single-bit flips and acceptance probability

$$\alpha_{ji} := \min\{1, e^{-\beta(E(j)-E(i))}\}. \tag{17.60}$$

Define the transition matrix P by

$$P_{ji} := Q_{ji}\alpha_{ji} \quad (j \neq i), \qquad P_{ii} := 1 - \sum_{j \neq i} P_{ji}. \tag{17.61}$$

Then P is column-stochastic and reversible with stationary distribution

$$\pi_i := \frac{e^{-\beta E(i)}}{Z}, \qquad Z := \sum_{i \in \Sigma} e^{-\beta E(i)}. \tag{17.62}$$

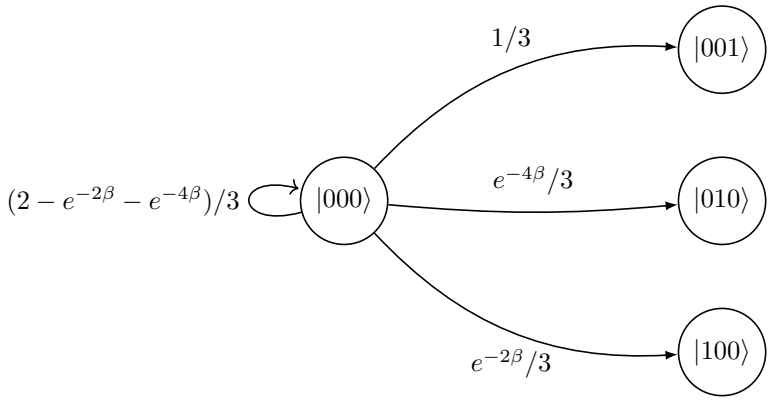
In order to visualize this process, let us assume that we have a Hamiltonian that is already diagonal in the computational basis:

$$H = (I - Z_0) + 2(I - Z_1) - 4(I - Z_2). \tag{17.63}$$

Note that $(I - Z_k)$ contributes 0 on $|0\rangle$ and 2 on $|1\rangle$. Therefore the energy of $|b_0 b_1 b_2\rangle$ is

$$E(b_0 b_1 b_2) = 2b_0 + 4b_1 - 8b_2. \tag{17.64}$$

From the state $|000\rangle$, the three single-bit-flip proposals lead to $|100\rangle$ (energy change $+2$), $|010\rangle$ (energy change $+4$), and $|001\rangle$ (energy change -8). Using the Metropolis acceptance rule, the corresponding acceptance probabilities are $e^{-2\beta}$, $e^{-4\beta}$, and 1, respectively. Since $Q_{ji} = 1/3$ for these proposals, the transition probabilities out of $|000\rangle$ are as shown in the following figure.



The figure shows the transition probabilities out of $i = 000$. Substituting the above energy changes into the Metropolis rule gives the displayed values. In the present example, single-bit flips connect all vertices of $\{0,1\}^3$, so the chain is irreducible. Moreover, $P_{ii} > 0$ for every i (because proposals can be rejected), so the chain is aperiodic. Therefore the stationary distribution is unique.

Let $\gamma > 0$ denote the spectral gap of P, meaning that the second-largest eigenvalue of P is at most $1-\gamma$ in absolute value. The associated Szegedy walk has eigenphases related to the eigenvalues of P: if $\lambda \in [-1,1]$ is an eigenvalue of P (equivalently, of the discriminant matrix), then the walk has eigenphases $\pm \arccos(\lambda)$. In particular, the smallest nonzero eigenphase is

$$\arccos(1-\gamma) = \Theta(\sqrt{\gamma}), \tag{17.65}$$

when γ is small. Consequently, given an initial state with non-negligible overlap with the coherent stationary state, one can prepare a state close to

$$|\pi\rangle = \sum_{i\in\Sigma} \sqrt{\pi_i}\,|i\rangle\,. \tag{17.66}$$

We then append $|0^n\rangle$ to this state, yielding

$$\sum_{i\in\Sigma} \sqrt{\pi_i}\,|i\rangle\,|0^n\rangle\,. \tag{17.67}$$

By applying n CNOT gates with the first register as control and the second register as target, and subsequently applying the unitary U to the first register, we prepare the state

$$\sum_{i\in\Sigma} \sqrt{\pi_i}(U\,|i\rangle)\,|i\rangle\,. \tag{17.68}$$

This is a purification of ρ_β. Indeed, tracing out the second register leads to

$$\sum_{i\in\Sigma} \pi_i\, U|i\rangle\langle i|U^\dagger = \frac{e^{-\beta H}}{\operatorname{Tr}\left[e^{-\beta H}\right]}. \tag{17.69}$$

Each step in the Szegedy walk can be implemented using a constant number of evaluations of $E(\cdot)$ (to compute the acceptance probabilities) together with elementary gates, and the total cost is $\mathrm{poly}(n)$. Suppressing logarithmic factors in the target precision and success probability, the number of walk steps required to resolve the eigenphase gap scales as $\mathcal{O}\big(1/\sqrt{\gamma}\big)$. Compared to Proposition 17.8, where the number of applications of the transition matrix scales as $\mathcal{O}(1/\gamma)$, this yields a quadratic improvement in the dependence on the spectral gap.

It is worth contrasting this with the results in Section 13.5, where a quadratic quantum speedup is obtained under the assumption that a block encoding of H is available via QSVT. The key difference lies in the input model. In the block-encoding oracle model for $H \in \mathbb{C}^{2^n\times 2^n}$, the cost typically includes a prefactor of $\sqrt{2^n/\operatorname{Tr}\left[e^{-\beta H}\right]}$, which is efficient only at sufficiently high temperatures. In contrast, the Markov chain-based cost model depends on the spectral gap γ rather than directly on the Hilbert space dimension, although γ itself may become small at low temperatures. ⋄

The discussion so far has focused on reversible Markov chains. However, Szegedy's quantum walk framework can also be applied to irreversible chains. As demonstrated in the example below, the quantum walk operator is constructed from the discriminant matrix of an irreversible chain. While the spectrum of the discriminant matrix generally differs from that of the original transition matrix, it still encodes useful information. In particular, by comparing the spectra of discriminant matrices for graphs with and without a marked vertex, one can determine the existence of a marked vertex.

Example 17.13 (Determining whether there is a marked vertex in a complete graph)**.** Let $G = (V, E)$ be a complete graph of $N = 2^n$ vertices. We would like to distinguish the following two scenarios:

(1) All vertices are the same, and the random walk is given by the transition matrix

$$P = \frac{1}{N} ee^\top, \quad e = (1, \ldots, 1)^\top. \tag{17.70}$$

(2) There is one *marked* vertex. Without loss of generality we may assume this is the 0-th vertex (of course we do not have access to this information). In this case, the transition matrix is

$$\widetilde{P}_{ij} = \begin{cases} \delta_{i0}, & j = 0, \\ P_{ij}, & j > 0. \end{cases} \tag{17.71}$$

In other words, in the case (2), the random walk will stop at the marked index. The transition matrix can also be written in the block partitioned form as

$$\widetilde{P} = \begin{pmatrix} 1 & \frac{1}{N}\widetilde{e}^\top \\ 0 & \frac{1}{N}\widetilde{e}\widetilde{e}^\top \end{pmatrix}. \tag{17.72}$$

Here $\widetilde{e}$ is an all 1 vector of length $N-1$.

For the random walk defined by P, the stationary state is $\pi = \frac{1}{N}e$, and the spectral gap is 1. For the random walk defined by $\widetilde{P}$, the stationary state is $\widetilde{\pi} = (1, 0, \ldots, 0)^\top$, and the spectral gap is $\gamma = N^{-1}$. Starting from the uniform state π (as a column vector), the probability distribution after k steps of random walk is $\widetilde{P}^k\pi$. This converges to the stationary state of $\widetilde{P}$, and hence reach the marked vertex after $\mathcal{O}(N)$ steps of walks.

Since P is symmetric, the discriminant matrix D is equal to P. Even though $\widetilde{P}$ is not reversible, the discriminant matrix $\widetilde{D}$ is still well-defined:

$$\widetilde{D} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{N}\widetilde{e}\widetilde{e}^\top \end{pmatrix}. \tag{17.73}$$

To distinguish the two cases, we are given a Szegedy quantum walk operator called O, which can be either O_Z or $\widetilde{O}_Z$, which is associated with $D, \widetilde{D}$, respectively. The initial state is

$$|\psi_0\rangle = |0^n\rangle \left(\mathrm{H}^{\otimes n} |0^n\rangle\right). \tag{17.74}$$

Our strategy is to measure the expectation

$$m_k = \langle\psi_0|O^k|\psi_0\rangle\,, \tag{17.75}$$

which can be obtained via the Hadamard test.

Before determining the value of k, first notice that if $O = O_Z$, then $O_Z |\psi_0\rangle = |\psi_0\rangle$. Hence $m_k = 1$ for all values of k.

On the other hand, if $O = \widetilde{O}_Z$, we use the fact that $\widetilde{D}$ only has two nonzero eigenvalues 1 and $(N-1)/N = 1-\gamma$, with associated eigenvectors denoted by $|\widetilde{\pi}\rangle$ and $|\widetilde{v}\rangle = \frac{1}{\sqrt{N-1}}(0, 1, 1 \ldots, 1)^\top$, respectively. Furthermore,

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} |0^n\rangle |\widetilde{\pi}\rangle + \sqrt{\frac{N-1}{N}} |0^n\rangle |\widetilde{v}\rangle\,. \tag{17.76}$$

Due to qubitization, we have

$$\widetilde{O}_Z^k |\psi_0\rangle = \frac{1}{\sqrt{N}} |0^n\rangle T_k(1) |\widetilde{\pi}\rangle + \sqrt{\frac{N-1}{N}} |0^n\rangle T_k(1-\gamma) |\widetilde{v}\rangle + |\perp\rangle\,, \tag{17.77}$$

where $|\perp\rangle$ is an unnormalized state satisfying $(|0^n\rangle\langle 0^n|) \otimes I_n |\perp\rangle = 0$. Then using $T_k(1) = 1$ for all k, we have

$$m_k = \frac{1}{N} + \left(1 - \frac{1}{N}\right) T_k(1-\gamma). \tag{17.78}$$

Use the fact that $T_k(1-\gamma) = \cos(k \arccos(1-\gamma))$, in order to have $T_k(1-\gamma) \approx 0$, the smallest k satisfies

$$k \approx \frac{\pi}{2\arccos(1-\gamma)} \approx \frac{\pi}{2\sqrt{2\gamma}} = \frac{\pi\sqrt{N}}{2\sqrt{2}}. \tag{17.79}$$

Therefore taking $k = \lceil \frac{\pi\sqrt{N}}{2\sqrt{2}} \rceil$, we have $m_k \approx 1/N$. Running Hadamard's test to constant accuracy allows us to distinguish the two scenarios.

Alternatively, we may evaluate the success probability of obtaining 0^n in the ancilla qubits, i.e.,

$$p(0^n) = \left\| (|0^n\rangle\langle 0^n| \otimes I_n) O^k |\psi_0\rangle \right\|^2. \tag{17.80}$$

When $O = O_Z$, we have $p(0^n) = 1$ with certainty. When $O = \widetilde{O}_Z$, according to Eq. (17.77),

$$p(0^n) = \frac{1}{N} + \left(1 - \frac{1}{N}\right) T_k^2(1-\gamma). \tag{17.81}$$

So running the problem with $k = \lceil \frac{\pi\sqrt{N}}{2\sqrt{2}} \rceil$, we can distinguish between the two cases.

It is natural to draw comparisons between Szegedy's quantum walk and Grover's search. The two algorithms make queries to different oracles, and both yield quadratic speedup compared to the classical algorithms. The quantum walk is slightly weaker, since it only tells whether there is one marked vertex or not. On the other hand, Grover's search also finds the location of the marked vertex. Both algorithms consist of repeated usage of the product of two reflectors. The number of iterations need to be carefully controlled. Indeed, choosing a polynomial degree four times as large as Eq. (17.79) would result in $m_k \approx 1$ for the case with a marked vertex.

Another possible solution of the problem of finding the marked vertex is to perform QPE on the Szegedy walk operator O (which can be O_Z or $\widetilde{O}_Z$). The effectiveness of the method rests on the spectral gap amplification discussed above. We refer to [Chi21, Chapter 17] for more details. ⋄

17.4. Glued tree problem and continuous time quantum walk

The continuous time quantum walk on the glued tree is one of the most important quantum algorithms. This is because it provides an example of an algorithmic task wherein there is a provable exponential separation in quantum and classical query complexity for solving the problem. Further, this algorithm was discovered first within the paradigm of continuous time quantum walks rather than using existing discrete paradigms. Note that this does not yet imply that $\mathsf{BQP} \neq \mathsf{BPP}$ because this exponential separation is relative to an oracle. Nonetheless, an exponential separation in query complexity strongly suggests that something is profoundly different between the quantum and classical computation.

17.4.1. Glued tree problem. The **glued tree problem** is a graph traversal problem that consists of two binary trees of depth n (our convention is that the root has depth 0) rooted at two vertices that we will label s and t drawn from an exponentially large set of labels. These two trees are then glued together by a random bipartite cycle graph that alternates between the leaf nodes of the two balanced binary trees (see Fig. 17.2).

Definition 17.14 (Glued Tree Graph)**.** *A glued tree graph $G(V, E)$ of depth $2n$ is constructed in the following manner.*

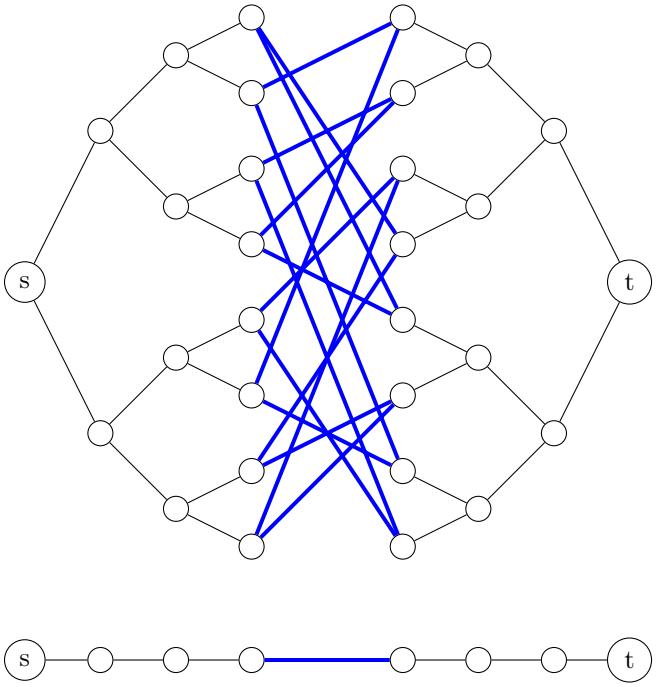


FIGURE 17.2. (Top) Glued-tree graph with parameter $n = 3$, with entrance node s on the left and exit node t on the right. The thick lines in blue color indicate the edges of the bipartite cycle graph. (Bottom): Column-space representation, where each vertex represents all nodes in the corresponding column of the original graph.

(1) *Divide the vertex set such that* $V = V_s \bigcup V_t$ *such that* V_s *and* V_t *are disjoint sets of vertices of size* $2^{n+1} - 1$*, and construct* $G_s = (V_s, E_s)$ *as a balanced binary tree of depth* n *rooted at a vertex* s*, and similarly construct* $G_t = (V_t, E_t)$ *as a balanced binary tree graph of depth* n *rooted at* t*.*
(2) *Construct a random bipartite cycle graph,* $C = ((L_s \bigcup L_t), E_C)$ *where* L_s *and* L_t *are the sets of leaf nodes in the tree* G_s *and* G_t*. Specifically, for the bipartite cycle graph we have that if* $x \in L_s$ *then the neighbors of* x *are only in* L_t *and vice-versa and every vertex in the graph has degree precisely* 2*.*
(3) *Construct the graph* $G = G_s \bigcup C \bigcup G_t$ *where the graph union is formed by constructing the union of the vertex and edge sets of the constituent graphs.*

The goal of the problem is to find the label of t using as few queries to an oracle that provides the labels of all vertices adjacent to any requested labeled vertex. This is an example of the **hitting problem**. The aim of the bipartite cycle graph is to create a maze in which a classical algorithm can easily get lost.

First, note that the exit label t here cannot be found by brute force due to the exponentially large size of the graph. Even Grover's algorithm would take an exponentially long time to find the exit vertex.

Algorithm 17.1 Classical Markov Chain for Glued Trees

Input: Oracle that yields the neighboring vertex of any vertex in the graph $G = (V, E)$ where V is the vertex set and E is the edge set and label of the starting node label(s).

1: Initialize $x \leftarrow \text{label}(s)$.
2: **while** $x \neq \text{label}(t)$ **do**
3: Query oracle to get the neighbor set of x $N(x) := \{y : (x, y) \in E\}$.
4: Draw vertex $x' \in N(x)$ uniformly over the set $N(x)$.
5: $x \leftarrow x'$.
6: **end while**

Next, intuitively we expect as n increases for any random walk algorithm to become increasingly trapped near the center of the graph. To see this, consider the following Markov chain Monte Carlo algorithm (Algorithm 17.1). At first, this algorithm can efficiently explore the graph. After starting at s, it moves to an adjacent vertex which will always be closer to the vertex t. In the second step, because the Markov Chain does not have memory, it will have a 1/3 probability of moving back to s and a 2/3 probability of moving closer to t. This situation reverses, however, after reaching the center of the graph. Now the probability of progressing towards the label t is only 1/3 and the probability of returning closer to the center is 2/3. This means that the walker tends to be trapped near the center for a long time without moving towards the label t.

We may analyze the long-time behavior of the Markov chain by examining its stationary distribution. The specific vertex occupied by the random walker is not essential. Rather, what matters is the **column** in which the walker resides. Since the walker begins in the first column and terminates in the last, the individual vertex labels in the intermediate steps carry no additional information. Consequently, it suffices to track the column-level transitions rather than individual vertex transitions. When we aggregate the graph by columns, the graph simplifies dramatically to a path graph, as shown in the lower panel of Fig. 17.2. Labeling the first $n+1$ columns as $0, \ldots, n$ and the remaining $n+1$ columns as $n+1, \ldots, 2n+1$, the transition probability from the column i to the column j is:

$$P_{ji} = \begin{cases} 1 & \text{if } i = 0 \text{ and } j = 1, \text{ or if } i = 2n+1 \text{ and } j = 2n \\ 2/3 & \text{if } 1 \leq i \leq n \text{ and } j = i+1, \text{ or if } n+1 \leq i \leq 2n \text{ and } j = i-1 \\ 1/3 & \text{if } 1 \leq i \leq n \text{ and } j = i-1, \text{ or if } n+1 \leq i \leq 2n \text{ and } j = i+1 \\ 0 & \text{otherwise} \end{cases}. \tag{17.82}$$

The stationary distribution π can be obtained from the detailed balance condition $P_{ji}\pi_i = P_{ij}\pi_j$. We have that $\pi_0 \cdot (1) = \pi_1 \cdot (1/3)$ and similarly, $\pi_1 \cdot (2/3) = \pi_2 \cdot (1/3)$ and so forth. We then see that $\pi_n(2/3) = \pi_{n+1}(2/3)$ which implies that the two columns in the middle of the graph must have the same probabilities (as anticipated from symmetry). From this, we see that the following is a stationary distribution that satisfies the detailed balance condition and further because the graph is irreducible we know that the state is unique.

$$\pi = \left[1/(3 \cdot 2^{n-1}), 1/2^{n-1}, \ldots, 1/2, 1, 1, 1/2, \ldots, 1/2^{n-1}, 1/(3 \cdot 2^{n-1})\right] / \|\pi\|_1. \tag{17.83}$$

As the stationary distribution has exponentially small probability at the vertices s and t, the walker sampled from the stationary distribution will have a vanishingly small probability of reaching the exit vertex t. Thus, even if we are able to efficiently prepare the stationary distribution, it does not

help us solve the hitting problem. This also implies that Szegedy's quantum walk, which accelerates the process of reaching the stationary distribution, cannot resolve this problem either.

In fact, it can be shown that actually **no** classical algorithm can find the label of the exit vertex t using a sub-exponential in n number of queries to the graph. The proof of this theorem involves a series of reductions and is omitted here.

THEOREM 17.15 ([CCD+03, Theorem 6]). *For the glued tree problem of depth $2n$, any classical algorithm that finds the exit vertex t with success probability at least $\Omega(2^{-n/6})$ must make at least $\Omega(2^{n/6})$ queries to the adjacency matrix, where the adjacency matrix serves as an oracle that returns the neighbors of a vertex $x \in V$.*

17.4.2. Continuous-time quantum walk. For an undirected graph $G = (V, E)$, its adjacency matrix A is a real symmetric matrix with

$$A_{x,y} = \begin{cases} 1 & \text{if } (x,y) \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{17.84}$$

Therefore A can be viewed as a Hamiltonian that defines a quantum dynamics. The **continuous-time quantum walk** refers to the following time-evolved state

$$|\psi(t)\rangle = e^{-iAt} |\psi(0)\rangle . \tag{17.85}$$

We first note that, unlike in the classical random walk, a continuous-time quantum walk cannot drive an arbitrary initial state to the stationary distribution. The only stationary states are the eigenvectors of the adjacency matrix A. To solve a hitting problem, that is, to reach a target state $|\phi\rangle$, we instead evolve the system for a random amount of time chosen uniformly from the interval $[0, T]$. The success probability is then given by the time-averaged transition probability, computed as

$$p = \frac{1}{T} \int_0^T |\langle \phi | \psi(t) \rangle|^2 \ \mathrm{d}t. \tag{17.86}$$

As noted earlier, the adjacency matrix generates transitions between the columns of the glued trees graph. Specifically, let $|C_j\rangle$ denote a quantum state in column j which is supported over the set of all vertices that are graph distance j away from s (where the graph distance gives the minimum number of edges that need to be traversed to go between two vertices). Specifically, we have that if x is in the second column of the glued tree and y is in the fourth column then $A_{x,y} = 0$ because the two vertices are not adjacent. Similarly, if x, y are in the same column then $A_{x,y} = 0$ because the glued tree has no edges between the same column of vertices. Thus the adjacency matrix only leads to non-trivial dynamics in the columns of the matrix as does the graph Laplacian.

This observation allows us to define a **column space** for the graph. In particular, if we let C_j be the set of vertices in column j then

$$|C_j\rangle = \frac{1}{\sqrt{|C_j|}} \sum_{x \in C_j} |x\rangle . \tag{17.87}$$

This notation also then directly implies that $|C_0\rangle = |s\rangle$ and $|C_{2n+1}\rangle = |t\rangle$. We then have that in this sub-space of column states

$$\begin{aligned}\langle C_{j+1}| A |C_j\rangle &= \frac{1}{\sqrt{|C_{j+1}||C_j|}} \sum_{x\in C_{j+1}} \sum_{y\in C_j} \langle x| A |y\rangle \\ &= \begin{cases} \frac{2|C_j|}{\sqrt{|C_{j+1}||C_j|}} & 0 \leq j \leq n, \\ \frac{|C_j|}{\sqrt{|C_{j+1}||C_j|}} & n+1 \leq j \leq 2n. \end{cases} \\ &= \begin{cases} \sqrt{2} & j \neq n, \\ 2 & j = n. \end{cases}\end{aligned} \tag{17.88}$$

This is true because if we consider all columns less than n then the subsequent column will contain twice as many vertices in it than the previous column. Past this point there will be half as many in every subsequent column. This implies that $|C_{j+1}| = 2|C_j|$ if $j < n$, $|C_{n+1}| = |C_n|$, and $|C_{j+1}| = |C_j|/2$ if $j > n$, and the final claim in Eq. (17.88) follows by substitution.

Furthermore the matrix A will only generate transitions between vectors within this space. The simplest way to see this is by considering the following complete basis whose elements are defined for any $p \in \mathbb{Z}_{|C_j|}$

$$|C_j^{(p)}\rangle := \sum_{x\in C_j} \frac{1}{\sqrt{|C_j|}} e^{-i2\pi p r_j(x)/|C_j|} |x\rangle , \tag{17.89}$$

where $r_j(x)$ gives the location of the vertex x in a sorted list of the vertices in the set C_j (the particular sorting order does not matter for the definition). As argued in our discussion of the quantum Fourier transform, these states form an orthonormal basis in each column C_j which implies that $\langle C_j^{(q)}|C_j^{(p)}\rangle = \delta_{pq}$. Also note that $|C_j^{(0)}\rangle = |C_j\rangle$. Similarly, we see that the inner product between any two vectors from different columns is zero because they contain disjoint sets of vertices. We then have that for any $j \leq n-1$ and $p > 0$

$$\begin{aligned}\langle C_{j+1}^{(p)}| A |C_j^{(0)}\rangle &= \frac{1}{\sqrt{|C_{j+1}||C_j|}} \sum_{x\in C_{j+1}} \sum_{y\in C_j} e^{-i2\pi p r_{j+1}(x)/|C_{j+1}|} \langle x| A |y\rangle \\ &= \frac{\sqrt{2}}{\sqrt{|C_{j+1}||C_j|}} \sum_{x\in C_{j+1}} e^{-i2\pi p r_{j+1}(x)/|C_{j+1}|} = 0.\end{aligned} \tag{17.90}$$

The same argument can be repeated for the remaining cases, which shows that the matrix A does not generate transitions between the $p = 0$ vectors of this complete basis and the remainder of the space. Thus as the vector space is complete, we then have that set $\{|C_j\rangle : j = 0, \ldots, 2n+1\}$ forms a basis for $A^k |C_0\rangle = A^k |s\rangle$ without needing to include $p > 0$. Thus by Taylor's theorem, it also forms a basis for every state of the form $e^{-iAt} |s\rangle$ for $t \in \mathbb{R}$.

From this perspective, we can see that the A matrix is the adjacency matrix for a path graph when represented in the column space, as illustrated by Figure 17.2. If we were to examine the dynamics in the continuum, A could be easily diagonalized with the eigenvectors being sine/cosine functions with appropriate periods. However, this intuition breaks down, apart from the obvious fact that we are focusing on a discrete problem, due to the defect at the center of the graph where $\langle C_{n+1}^{(p)}| A |C_n^{(p)}\rangle = 2$ rather than $\sqrt{2}$.

Now let us consider the reflection operator R such that for any $j < n$

$$R\,|C_j\rangle = |C_{2n+1-j}\rangle\,. \tag{17.91}$$

Hence the set $\{|C_j\rangle\}$ also remains closed under applications of R. We then further have that

$$AR\,|C_j\rangle = RA\,|C_j\rangle\,. \tag{17.92}$$

If we define $\hat{A}$ and $\hat{R}$ to be the restriction of the operators onto the subspace formed by the span of these column space vectors, then $[\hat{A}, \hat{R}] = 0$. Then the eigenvectors of $\hat{A}$ can be re-written in terms of simultaneous eigenvectors of $\hat{A}$ and $\hat{R}$. We can find eigenvectors by noting that due to the required symmetry (or anti-symmetry) under reflection imposed on any eigenvector of $\hat{R}$, the eigenvectors must also possess the same reflection parity. Thus, a reasonable guess to make is that the solution will be a linear combination of sine functions, with appropriate periodicity, and with the parity of the solution flipping at $j = n+1$ if the eigenvalue of $\hat{R}$ is negative for the eigenvector in question and positive otherwise. This guess can be written as [CCD+03]

$$|E_k^{\pm}\rangle := \sum_{j=0}^{n} \sin(p_k^{\pm}(j+1))\,|C_j\rangle \pm \sum_{j=n+1}^{2n+1} \sin(p_k^{\pm}(2n+2-j))\,|C_j\rangle \tag{17.93}$$

with eigenvalues $E_k^{\pm} = 2\sqrt{2}\cos(p_k^{\pm})$ where $p_k^{\pm}$ is a solution of the equation

$$\frac{\sin((n+2)p_k^{\pm})}{\sin((n+1)p_k^{\pm})} = \pm\sqrt{2}. \tag{17.94}$$

We will also prove these relations below.

Algorithm 17.2 describes a quantum algorithm for solving the glued tree problem using continuous-time quantum walk. The walker is initialized in the known state $|\text{label}(s)\rangle$, and the goal is to reach the target state $|\phi\rangle = |\text{label}(t)\rangle$. Crucially, this target state is not directly accessible; it can only be identified via a projective measurement onto $|\phi\rangle$. Similar to Grover's search, the ability to verify a candidate label t does not imply knowledge of how to construct or locate the state $|\text{label}(t)\rangle$.

Algorithm 17.2 Continuous-Time Quantum Walk for Glued Trees

Input: Maximum simulation time T; oracle that implements Hamiltonian evolution e^{-iAu} for any $u \in [0, T]$, where A is the adjacency matrix of the glued tree graph; oracle that performs projective measurement onto $P = |\text{label}(t)\rangle\langle\text{label}(t)|$.

1: Prepare the initial state $|\psi(0)\rangle = |\text{label}(s)\rangle$.
2: Sample a time u uniformly at random from the interval $[0, T]$, and evolve the state as $|\psi(u)\rangle = e^{-iAu}\,|\psi(0)\rangle$.
3: Perform a projective measurement onto P. If the outcome is successful, return label(t) in the computational basis. Otherwise, repeat the procedure.

To justify the efficiency of Algorithm 17.2 for solving the glued tree problem, we analyze the quantum dynamics of the continuous-time walk in the column space of the graph. In this reduced representation, the quantum walk exhibits coherent propagation from the entrance to the exit, in stark contrast to the classical random walk which becomes trapped near the central region. As a result, we can find the exit vertex label with high probability using only poly(n) quantum queries to the Hamiltonian and measurement oracles (Theorem 17.18).

Proposition 17.16. *For the glued tree problem of depth $2n$, let $\min|E-E'|$ be the minimum difference between any two eigenvalues of $\hat{A}$ (that is, A restricted to the column subspace). Then for any $T>0$, the time averaged probability of the continuous time quantum walk transitioning from the entrance, $|s\rangle$, to the exit, $|t\rangle$ is bounded below as*

$$\frac{1}{T}\int_0^T \left|\langle t|\, e^{-iAu}\,|s\rangle\right|^2 \mathrm{d}u \geq \frac{1}{2n+2} - \frac{2}{T\min|E-E'|}. \tag{17.95}$$

Proof. We will argue by symmetry and the Cauchy–Schwarz inequality that the time averaged probability of transitioning from the entrance to the exit is only polynomially small in the depth of the glued trees. As a unitary process cannot have a unique fixed point for all inputs, we compute this probability in terms of a time averaged probability of finding the exit node for times chosen uniformly over the interval $[0,T]$. Specifically we have that if we expand the time average in an eigenbasis $|E\rangle$ of $\hat{A}$,

$$\begin{aligned}\frac{1}{T}\int_0^T \langle t|\, e^{-iAu}|s\rangle\langle s|e^{iAu}\,|t\rangle \;\mathrm{d}u &= \frac{1}{T}\sum_{E,E'}\int_0^T \langle t|E\rangle\,\langle E'|t\rangle\,\langle E|s\rangle\,\langle s|E'\rangle\, e^{i(E-E')u}\,\mathrm{d}u \\ &= \sum_E |\langle t|E\rangle|^2\,|\langle E|s\rangle|^2 + \frac{1}{T}\sum_{E\neq E'}\frac{e^{i(E-E')T}-1}{i(E-E')}\,\langle t|E\rangle\,\langle E'|t\rangle\,\langle E|s\rangle\,\langle s|E'\rangle\,.\end{aligned} \tag{17.96}$$

Next using the symmetry condition set by the fact that $|E\rangle$ is a simultaneous eigenvector of the reflection operator $\hat{R}$, we know that $|\langle t|E\rangle|^4 = |\langle s|E\rangle|^4$. Thus the Cauchy–Schwarz inequality implies that

$$(2n+2)\sum_E |\langle t|E\rangle|^4 \geq \left(\sum_E |\langle t|E\rangle|^2\right)^2 = 1. \tag{17.97}$$

Hence

$$\sum_E |\langle t|E\rangle|^2\,|\langle E|s\rangle|^2 \geq 1/(2n+2). \tag{17.98}$$

Repeating a similar argument using the Cauchy–Schwarz inequality allows us to bound the term where $E\neq E'$ by

$$\left|\frac{1}{T}\sum_{E\neq E'}\frac{e^{i(E-E')T}-1}{i(E-E')}\,\langle t|E\rangle\,\langle E'|t\rangle\,\langle E|s\rangle\,\langle s|E'\rangle\right| \leq \frac{2}{T\min|E-E'|}. \tag{17.99}$$

Thus from the triangle inequality

$$\frac{1}{T}\int_0^T \langle t|\, e^{-iAu}|s\rangle\langle s|e^{iAu}\,|t\rangle\, du \geq \frac{1}{(2n+2)} - \frac{2}{T\min|E-E'|}. \tag{17.100}$$

□

The above result shows that if the eigenvalue gap is large enough then we can simply simulate the evolution under A for randomly chosen evolution times t chosen from $[0,T]$ and then if T is large enough then the probability of finding the exit is $\Omega(1/n)$. This means that the number of trials needed to find the exit vertex t is geometrically distributed with a mean of $O(n)$. Thus we

can show that the total evolution time needed to find the label of t is polynomial if the gap is at least polynomial. The following lemma demonstrates precisely such a claim.

Lemma 17.17. *Let $\hat{A}$ be the restriction of the adjacency matrix to the column subspace described in Proposition 17.16. The minimum eigenvalue gap between any two eigenvalues of $\hat{A}$ obeys*

$$\min|E - E'| > \frac{2\sqrt{2}\pi^2}{(1+\sqrt{2})(n+1)^3} + \mathcal{O}(1/n^4). \tag{17.101}$$

Proof. The eigenvalue equation implies that

$$\langle C_j| A |E_k^{\pm}\rangle = 2\sqrt{2}\cos(p)\langle C_j|E_k^{\pm}\rangle \tag{17.102}$$

Then by explicitly evaluating the action of the adjacency matrix on the column space vectors,

$$\sqrt{2}\langle C_{n+2}|E_k^{\pm}\rangle + \langle C_n|E_k^{\pm}\rangle = 2\cos(p)\langle C_{n+1}|E_k^{\pm}\rangle. \tag{17.103}$$

Similarly, the use of (17.93) further allows us to see that

$$\pm\sqrt{2}\sin((n+2)p_k^{\pm}) + \sin((n+1)p_k^{\pm}) = 2\sin((n+1)p_k^{\pm})\cos(p_k^{\pm}). \tag{17.104}$$

This yields the following non-linear expression for the quantization condition p,

$$\frac{\sin((n+2)p_k^{\pm})}{\sin((n+1)p_k^{\pm})} = \pm\sqrt{2}. \tag{17.105}$$

as alluded to previously.

Let us consider for simplicity the negative branch of the expression. An analytic solution to this expression is difficult to obtain, however, we can find an asymptotic solution for large n. In particular, let

$$p_k^- = \pi(k+1)/(n+1) - \delta_k \tag{17.106}$$

where $\delta_k = 0$ corresponds to the root of $\sin((n+1)p_k^-)$ so we are looking, in essence, for solutions that are perturbations away from the k^{th} root of the denominator which is justified because the function will vary rapidly in this vicinity as it approaches the singularity at $\delta_k = 0$. Expressing the eigenvalue relation in this limit gives

$$\sin((n+1)\delta_k - \frac{(k+1)\pi}{(n+1)} + \delta_k) = -\sqrt{2}\sin((n+1)\delta_k). \tag{17.107}$$

Expanding δ_k^- in powers of $1/(n+1)$ yields $\delta_k^{-1} = c_0 + c_1/(n+1) + O(1/n^2)$. Substituting this into (17.107) and eliminating any terms that are first order or higher in $1/(n+1)$

$$-\sqrt{2}\sin(c_0) = \sin(c_0). \tag{17.108}$$

This expression must hold for all n thus we must have that $c_0 = m\pi$ for integer π. For simplicity, we choose the trivial root of $c_0 = 0$. The expression for all first order terms in $1/(n+1)$ (neglecting all higher order terms) is

$$\begin{aligned} &-\sqrt{2}\sin\left(\frac{c_1}{n+1}\right) = \sin\left(\frac{c_1}{n+1} - \frac{(k+1)\pi}{(n+1)}\right) \\ \Rightarrow\quad &-\sqrt{2}\left(\frac{c_1}{n+1}\right) + O(1/n^3) = \left(\frac{c_1}{n+1} - \frac{(k+1)\pi}{(n+1)}\right) + O(1/n^3) \end{aligned} \tag{17.109}$$

This expression must be true for all n sufficiently large. As a result, the only way this expression can asymptotically hold is if

$$p_k^- = \frac{(k+1)\pi}{(n+1)} - \frac{(k+1)\pi}{(1+\sqrt{2})(n+1)^2} + O(1/n^3). \tag{17.110}$$

We can apply similar reasoning to find the positive roots are asymptotically

$$p_k^+ = \frac{(k+1)\pi}{(n+1)} + \frac{(k+1)\pi}{(1+\sqrt{2})(n+1)^2} + O(1/n^3). \tag{17.111}$$

From these expressions, we note that the two closest solutions to p_k^- will both be positive solutions corresponding to the same k and the subsequent one. This gives us that the gap between the two nearest $p_k^\pm$ to p_k^- is

$$\Delta_k := \min\left(\frac{(k+1)\pi\sqrt{2}}{(1+\sqrt{2})(n+1)^2}, \frac{(k+1)\pi}{(1+\sqrt{2})(n+1)^2}\right) + O(1/n^3). \tag{17.112}$$

The smallest gap corresponds to $k = 0$ which yields

$$\Delta_{\min} := \frac{\pi}{(1+\sqrt{2})(n+1)^2} + O(1/n^3). \tag{17.113}$$

The minimum eigenvalue gap then can be found by substituting this into the eigenvalue expression to find that the eigenvalue gap is minimal for $k' = 0$ and $k = 1$ and hence is of the form

$$\begin{aligned}
\min(|E_k - E_{k'}|) &= 2\sqrt{2}\left(\cos(p_k^+(k+1)) - \cos(p_{k'}^-(k'+1))\right) \\
&\geq 2\sqrt{2}\left(\cos(p_1^+) - \cos(p_0^-)\right) \\
&= 2\sqrt{2}\left|\int_0^{p_1^+ - p_0^-} \frac{\partial}{\partial_s}\cos(p_0^- + s)ds\right| \\
&= 2\sqrt{2}\left|\int_0^{p_1^+ - p_0^-} \sin(p_0^- + s)ds\right| \\
&= 2\sqrt{2}|p_1^+ - p_0^-||\sin(p_0^-(k+1))| + O(1/n^4) \\
&\geq \frac{2\sqrt{2}\pi}{(1+\sqrt{2})(n+1)^2}\sin\left(\frac{\pi}{n+1}\right) + O(1/n^4) \\
&\geq \frac{2\sqrt{2}\pi^2}{(1+\sqrt{2})(n+1)^3} + O(1/n^4)
\end{aligned} \tag{17.114}$$

□

From this result we have a bound on the minimum eigenvalue gap for the adjacency matrix. This allows us to then use this result to prove a bound on the total time needed to find the vertex label of t with inverse polynomial probability in n.

THEOREM 17.18. *For the glued tree problem of depth $2n$, it is sufficient to choose $T = \Theta(n^3)$ so that the time averaged probability of the continuous time quantum walk transitioning from the entrance, $|s\rangle$, to the exit, $|t\rangle$ is bounded below by $1/4(n+1)$.*

PROOF. Proof follows directly from substituting the result of Lemma 17.17 into Proposition 17.16 and requiring that the higher-order terms in the probability expansion add up to at most half of the probability that would be seen in the limit of $T \to \infty$. □

Notes and further reading

Classical Markov chains, including both discrete-time and continuous-time variants (the latter not discussed here), provide the foundation for quantum walks. Their convergence properties, mixing times, and spectral characteristics are treated in [Liu01, LP17], and they underpin the definition of the discriminant matrix used in Szegedy's quantum walk [Sze04]. Historically, Szegedy's construction directly inspired the development of block encodings for Hermitian matrices, with applications in Hamiltonian simulation [BC12] and quantum algorithms for solving linear systems of equations [CKS17]. As discussed in earlier chapters, the Szegedy walk operator is now best understood as a special case of qubitization; see also [AGJ21].

We introduced both discrete-time and continuous-time quantum walks. These two models differ significantly in their operational mechanisms; for a comparative overview, see [Chi10]. The exponential query advantage of the continuous-time quantum walk was demonstrated in the glued trees problem [CCD$^+$03], which subsequently motivated further research in quantum adiabatic algorithms [SNK12, GHV21].

CHAPTER 18

Solving eigenvalue problems

CHAPTER 19

Solving linear systems of equations

CHAPTER 20

Solving linear differential equations

CHAPTER 21

Solving open quantum systems

Open quantum systems interact with degrees of freedom that are not explicitly modeled. Consequently, their evolution is no longer described by unitary operators on the system register alone, but by quantum channels acting on density operators. In the Markovian regime, the natural continuous-time model is the Lindblad equation, which separates coherent Hamiltonian evolution from irreversible dissipative effects. This equation is standard in quantum statistical mechanics and quantum optics, and it also provides an algorithmic model for noise, equilibration, and engineered dissipation.

This shift from unitary dynamics to channel semigroups changes both the mathematical questions and the algorithmic ones. At the same time, dissipation is not merely a source of error: it can also be used to drive a system toward a desired fixed point, such as a thermal state or a ground state. We first develop the basic structural framework, and then discuss how these dynamics can be exploited and simulated using dilation-based, splitting-based, and truncated-Dyson methods.

21.1. Lindblad dynamics

The Gorini–Kossakowski–Sudarshan–Lindblad master equation, or **Lindblad equation** for short, has the form [Lin76, GKS76],

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = \underbrace{-i[H,\rho]}_{\mathcal{L}_H(\rho)} + \underbrace{\sum_{j=1}^{J}\left(V_j\rho V_j^\dagger - \frac{1}{2}\{V_j^\dagger V_j,\rho\}\right)}_{\mathcal{L}_V(\rho)} =: \mathcal{L}(\rho). \tag{21.1}$$

Here $H \in \mathbb{C}^{d\times d}$ is the system Hamiltonian, and $V_j \in \mathbb{C}^{d\times d}$ are the jump operators arising from the interaction with the environment. Because exchange with an environment is typically an irreversible process, Lindblad evolution is often called **dissipative dynamics**. Accordingly, $\mathcal{L}_H$ is called the coherent part of the Lindbladian, while $\mathcal{L}_V$ is called the dissipative part. The brackets $[A,B]$ and $\{A,B\}$ denote the commutator and anticommutator of A and B, respectively. The generator $\mathcal{L}$ is called the Lindbladian. The solution to Eq. (21.1) can be written formally as

$$\rho(t) = e^{\mathcal{L}t}\rho(0). \tag{21.2}$$

It is convenient to denote the propagator by

$$\Phi_t := e^{\mathcal{L}t}, \qquad t \geq 0. \tag{21.3}$$

When $V_j = 0$ for all j, Eq. (21.1) reduces to the Liouville–von Neumann equation. In this sense, the Lindblad equation generalizes closed-system Hamiltonian evolution to open quantum systems.

A family $\{\Phi_t\}_{t\geq 0}$ of quantum channels is called a **norm-continuous semigroup of quantum channels** (or **quantum dynamical semigroups**) if

$$\Phi_0 = \mathcal{I}, \qquad \Phi_{t+s} = \Phi_t\Phi_s, \qquad t,s \geq 0, \tag{21.4}$$

and

$$\|\Phi_t - \Phi_s\|_\diamond \to 0 \qquad \text{as } t \to s. \tag{21.5}$$

It describes a continuous-time analogue of the discrete-time channels in Section 3.2. In the classical setting of Section 17.1, one studies a semigroup of stochastic matrices acting on probability vectors. The Lindblad equation plays the same role for density operators, with stochastic matrices replaced by quantum channels.

Before turning to structural results, it is useful to note that classical continuous-time Markov chains arise as a special case.

Example 21.1 (Classical Markov chains as Lindblad dynamics)**.** Let $\Sigma = [N]$, and let $q_{ij} \geq 0$ for all $i \neq j$. We interpret q_{ij} as the transition rate from state j to state i, consistent with the column-vector convention in Section 17.1. Set $H = 0$ and define jump operators

$$V_{ij} := \sqrt{q_{ij}}|i\rangle\langle j|, \qquad i \neq j. \tag{21.6}$$

Consider a classical state

$$\rho = \sum_{j\in[N]} p_j |j\rangle\langle j|, \tag{21.7}$$

where $p = (p_0, \ldots, p_{N-1})^\top$ is a probability vector. Then

$$V_{ij}\rho V_{ij}^\dagger = q_{ij}p_j|i\rangle\langle i|, \qquad V_{ij}^\dagger V_{ij} = q_{ij}|j\rangle\langle j|. \tag{21.8}$$

Substituting these identities into the Lindblad equation gives

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = \sum_{i\neq j} q_{ij}p_j|i\rangle\langle i| - \sum_{j\in[N]} \left(\sum_{i\neq j} q_{ij}\right) p_j|j\rangle\langle j|. \tag{21.9}$$

Therefore the diagonal subspace is invariant, and the coefficients obey

$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = \sum_{j\neq i} q_{ij}p_j - \left(\sum_{k\neq i} q_{ki}\right) p_i. \tag{21.10}$$

Equivalently,

$$\frac{\mathrm{d}p}{\mathrm{d}t} = Qp, \tag{21.11}$$

where $Q \in \mathbb{R}^{N\times N}$ is the classical generator defined by

$$Q_{ij} = q_{ij} \quad (i \neq j), \qquad Q_{jj} = -\sum_{i\neq j} q_{ij}. \tag{21.12}$$

The matrix Q has nonnegative off-diagonal entries and each column sums to zero. Therefore it generates a classical continuous-time Markov semigroup, and $P_t := e^{Qt}$ is column stochastic for all $t \geq 0$. Hence, when restricted to diagonal density operators, the Lindblad semigroup $e^{\mathcal{L}t}$ is exactly the classical Markov semigroup P_t acting on probability distributions. ⋄

The Lindblad form is justified by two complementary statements. We first show constructively that every equation of the form Eq. (21.1) generates a semigroup of quantum channels. We then record the finite-dimensional characterization.

THEOREM 21.2. *For any $t \geq 0$, $e^{\mathcal{L}t} : \mathbb{C}^{d\times d} \to \mathbb{C}^{d\times d}$ is a completely positive trace preserving map, i.e., it is a quantum channel.*

PROOF. Let Δt be a small time step and write $t = N_t \Delta t$. Our strategy is to approximate the short-time propagator $e^{\mathcal{L}\Delta t}$ by a quantum channel Φ to precision $\mathcal{O}(\Delta t^2)$, that is,

$$\left\| e^{\mathcal{L}\Delta t} - \Phi \right\|_\diamond \leq C_1 \Delta t^2. \tag{21.13}$$

Since Φ is a quantum channel, we have $\|\Phi\|_\diamond = 1$. For any finite t, let $C_2 = \sup_{0\leq s\leq t} \left\| e^{\mathcal{L}s} \right\|_\diamond$. Since the diamond norm is an induced norm, we have $\|\mathcal{Q}_1\mathcal{Q}_2\|_\diamond \leq \|\mathcal{Q}_1\|_\diamond \|\mathcal{Q}_2\|_\diamond$, and can use the telescoping series

$$\begin{aligned} \left\| e^{\mathcal{L}t} - \Phi^{N_t} \right\|_\diamond \leq & \left\| e^{\mathcal{L}(N_t-1)\Delta t}(e^{\mathcal{L}\Delta t} - \Phi) \right\|_\diamond + \cdots + \left\| (e^{\mathcal{L}\Delta t} - \Phi)\Phi^{N_t-1} \right\|_\diamond \\ \leq & C_2 C_1 N_t \Delta t^2 = C_1 C_2 t \Delta t. \end{aligned} \tag{21.14}$$

Taking the limit $\Delta t \to 0$, we find that $e^{\mathcal{L}t}$ is the limit of a sequence of quantum channels as $N_t \to \infty$. Since the set of quantum channels is closed in the diamond norm, it follows that $e^{\mathcal{L}t}$ is itself a quantum channel.

One convenient construction of Φ starts from the dilated Hamiltonian

$$\widetilde{H} = |0\rangle\langle 0| \otimes H\sqrt{\Delta t} + \sum_j (|j\rangle\langle 0| \otimes V_j + \text{h.c.}) \tag{21.15}$$

whose matrix form is

$$\widetilde{H} = \begin{pmatrix} H\sqrt{\Delta t} & V_1^\dagger & \cdots & V_J^\dagger \\ V_1 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & 0 \\ V_J & 0 & \cdots & 0 \end{pmatrix}. \tag{21.16}$$

From this dilated Hamiltonian we define the channel Φ by

$$U = e^{-i\widetilde{H}\sqrt{\Delta t}}, \quad \Phi(\rho) = \mathrm{Tr}_a\, U\,(|0\rangle\langle 0| \otimes \rho)\, U^\dagger, \quad \forall \rho \in \mathcal{D}(\mathbb{C}^d). \tag{21.17}$$

Here Tr_a is the partial trace over the ancilla register with $a = \lceil \log_2(J+1) \rceil$ qubits.

We now prove Eq. (21.13) directly in diamond norm. Since the diamond norm is defined by taking a supremum over ancillary extensions and input operators of trace norm one, fix an ancilla register of dimension K and an operator $X \in L(\mathbb{C}^K \otimes \mathbb{C}^d)$ with $\|X\|_1 = 1$. Define

$$\widetilde{U} := I_K \otimes U. \tag{21.18}$$

Then, by linearity, the corresponding ancillary short-time channel is

$$(\mathcal{I}_K \otimes \Phi)(X) = \mathrm{Tr}_a \left[\widetilde{U}(|0\rangle\langle 0| \otimes X)\widetilde{U}^\dagger \right]. \tag{21.19}$$

The Taylor expansion of $e^{-i\widetilde{H}\sqrt{\Delta t}}$ to the third order gives

$$W = I - i\widetilde{H}(\Delta t)^{\frac{1}{2}} - \frac{1}{2}\widetilde{H}^2\Delta t + \frac{1}{6} i\widetilde{H}^3 (\Delta t)^{\frac{3}{2}}, \quad \|U - W\| = \mathcal{O}\left(\left\| \widetilde{H} \right\|^4 \Delta t^2 \right). \tag{21.20}$$

Set $\widetilde{W} := I_K \otimes W$. Using $\|AB\|_1 \le \|A\|\,\|B\|_1$, the fact that partial trace does not increase the Schatten 1-norm, and regarding $\left\|\widetilde{H}\right\|$ as a constant, we have

$$\begin{aligned}
&\left\|(\mathcal{I}_K \otimes \Phi)(X) - \mathrm{Tr}_a\, \widetilde{W}\,(|0\rangle\langle 0| \otimes X)\,\widetilde{W}^\dagger\right\|_1 \\
\le &\left\|\mathrm{Tr}_a(\widetilde{U} - \widetilde{W})\,(|0\rangle\langle 0| \otimes X)\,\widetilde{U}^\dagger\right\|_1 + \left\|\mathrm{Tr}_a\,\widetilde{U}\,(|0\rangle\langle 0| \otimes X)\,(\widetilde{W} - \widetilde{U})^\dagger\right\|_1 \\
\le &\left\|(\widetilde{U} - \widetilde{W})\,(|0\rangle\langle 0| \otimes X)\,\widetilde{U}^\dagger\right\|_1 + \left\|\widetilde{U}\,(|0\rangle\langle 0| \otimes X)\,(\widetilde{W} - \widetilde{U})^\dagger\right\|_1 \\
\le &2\left\|\widetilde{U} - \widetilde{W}\right\| = 2\,\|U - W\| = \mathcal{O}(\Delta t^2).
\end{aligned} \tag{21.21}$$

When calculating $\mathrm{Tr}_a\, \widetilde{W}\,(|0\rangle\langle 0| \otimes X)\,\widetilde{W}^\dagger$, we find that odd powers of $\sqrt{\Delta t}$ vanish after the partial trace. Then

$$\begin{aligned}
\mathrm{Tr}_a\, \widetilde{W}\,(|0\rangle\langle 0| \otimes X)\,\widetilde{W}^\dagger =& X + \Delta t(\mathcal{I}_K \otimes \mathcal{L})(X) + \mathcal{O}(\Delta t^2) \\
=& (\mathcal{I}_K \otimes e^{\mathcal{L}\Delta t})(X) + \mathcal{O}(\Delta t^2).
\end{aligned} \tag{21.22}$$

The $\mathcal{O}(\Delta t^2)$ term only involves products between bounded operators and X, with constants independent of K and X under the normalization $\|X\|_1 = 1$. Taking the supremum over all ancillary dimensions and all such X proves Eq. (21.13). □

We now record the converse structural statement: in finite dimensions, Lindblad generators are exactly the generators of norm-continuous semigroups of quantum channels.

Proposition 21.3 (Gorini–Kossakowski–Sudarshan–Lindblad characterization)**.** *Let $\{\Phi_t\}_{t\ge 0}$ be a norm-continuous semigroup of quantum channels on $L(\mathbb{C}^d)$, and let*

$$\mathcal{G}(\rho) := \lim_{t\to 0^+} \frac{\Phi_t(\rho) - \rho}{t} \tag{21.23}$$

be its generator. Then there exist a Hermitian matrix $H \in \mathbb{C}^{d\times d}$ and matrices $V_1, \dots, V_J \in \mathbb{C}^{d\times d}$ such that

$$\mathcal{G}(\rho) = -i[H, \rho] + \sum_{j=1}^{J} \left(V_j \rho V_j^\dagger - \frac{1}{2}\{V_j^\dagger V_j, \rho\} \right). \tag{21.24}$$

Proof sketch. For each sufficiently small $t > 0$, the map Φ_t is a quantum channel and hence admits a Kraus representation,

$$\Phi_t(\rho) = \sum_\alpha K_\alpha(t)\rho K_\alpha(t)^\dagger, \tag{21.25}$$

with $\sum_\alpha K_\alpha(t)^\dagger K_\alpha(t) = I$. A standard argument based on the first-order expansion of the Choi matrix near $t = 0$ allows one to choose the Kraus operators so that one term has the form

$$K_0(t) = I + tG + o(t), \tag{21.26}$$

while the remaining terms satisfy

$$K_j(t) = \sqrt{t}\, V_j + o(\sqrt{t}), \qquad j \ge 1. \tag{21.27}$$

Expanding the trace-preserving condition to first order shows that

$$G + G^\dagger + \sum_{j\ge 1} V_j^\dagger V_j = 0. \tag{21.28}$$

Therefore G can be written as

$$G = -iH - \frac{1}{2}\sum_{j\geq 1} V_j^\dagger V_j \tag{21.29}$$

for some Hermitian matrix H. Substituting these expansions into the Kraus form of Φ_t and collecting all first-order terms gives

$$\frac{\Phi_t(\rho)-\rho}{t} = -i[H,\rho] + \sum_{j\geq 1}\left(V_j\rho V_j^\dagger - \frac{1}{2}\{V_j^\dagger V_j,\rho\}\right) + o(1). \tag{21.30}$$

Taking $t \to 0^+$ yields the claim. □

If the short time propagator can be accurately approximated to precision ϵ', i.e., there is a quantum channel Φ such that

$$\left\|e^{\mathcal{L}\Delta t} - \Phi\right\|_\diamond \leq \epsilon', \tag{21.31}$$

then by the linear error growth property of quantum channels, we obtain the long time error estimate

$$\left\|e^{\mathcal{L}N_t\Delta t} - \Phi^{N_t}\right\|_\diamond \leq N_t \left\|e^{\mathcal{L}\Delta t} - \Phi\right\|_\diamond \leq N_t\epsilon'. \tag{21.32}$$

We next describe the dual evolution on observables.

For an observable O, the expectation value $\mathrm{Tr}[\rho(t)O]$ evolves under the Lindblad dynamics according to

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Tr}[\rho(t)O] = \mathrm{Tr}[\rho(t)\mathcal{L}^\dagger(O)], \tag{21.33}$$

where $\mathcal{L}^\dagger$, called the adjoint of the Lindbladian $\mathcal{L}$, is defined as

$$\mathcal{L}^\dagger(O) = \underbrace{i[H,O]}_{\mathcal{L}_H^\dagger(O)} + \underbrace{\sum_{j=1}^{J}\left(V_j^\dagger O V_j - \frac{1}{2}\{V_j^\dagger V_j, O\}\right)}_{\mathcal{L}_V^\dagger(O)}. \tag{21.34}$$

From Eq. (21.33), we may also define the Heisenberg evolution of the observable O as

$$O(t) := e^{\mathcal{L}^\dagger t}(O). \tag{21.35}$$

Then $\mathrm{Tr}[\rho(t)O] = \mathrm{Tr}[\rho(0)O(t)]$.

Example 21.4 (Photon loss in an optical cavity)**.** Consider a single mode of an optical cavity with bosonic annihilation operator a. Photon loss in the cavity is modeled by a single jump operator $V = \sqrt{\kappa}\,a$, where κ is the decay rate.

The density operator ρ then evolves according to

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -i[H,\rho] + \kappa\left(a\rho a^\dagger - \frac{1}{2}\{a^\dagger a,\rho\}\right), \tag{21.36}$$

where $H = \omega a^\dagger a$ is the Hamiltonian of the cavity mode with frequency ω. Using the adjoint Lindbladian introduced above, with number operator $n := a^\dagger a$, we compute

$$\begin{aligned}\mathcal{L}^\dagger(n) &= i[H,n] + \kappa\left(a^\dagger n a - \frac{1}{2}\{a^\dagger a, n\}\right)\\ &= \kappa\left(a^\dagger a^\dagger a a - n^2\right) = \kappa\left(n(n-1) - n^2\right) = -\kappa n.\end{aligned} \tag{21.37}$$

Here we used $[H, n] = 0$ and $a^\dagger a^\dagger aa = n(n-1)$. Therefore the mean photon number $\langle n\rangle = \mathrm{Tr}[n\rho]$ evolves according to

$$\frac{\mathrm{d}\langle n\rangle}{\mathrm{d}t} = -\kappa\langle n\rangle. \tag{21.38}$$

Therefore,

$$\langle n\rangle(t) = \langle n\rangle(0)e^{-\kappa t}, \tag{21.39}$$

so the mean photon number decays exponentially. As $t \to \infty$, the cavity loses all photons and the system relaxes to the vacuum state $|0\rangle$, which is the steady state. ◇

Exercise 21.1. Verify that $\mathcal{L}^\dagger$ is indeed the adjoint of $\mathcal{L}$ with respect to the Hilbert–Schmidt inner product $\langle X|Y\rangle := \mathrm{Tr}[X^\dagger Y]$.

The master equation describes the evolution of an ensemble state $\rho(t)$. For intuition and for numerical methods, it is often useful to represent the same semigroup as an average over stochastic pure-state trajectories, analogous to describing a classical Markov process through its sample paths. This representation is called an unraveling of the Lindblad equation. Consider the stochastic Schrödinger equation

$$\mathrm{d}\,|\psi_t\rangle = \left(-iH - \frac{1}{2}\sum_{j=1}^{J} V_j^\dagger V_j\right)|\psi_t\rangle\ \mathrm{d}t + \sum_{j=1}^{J} V_j\,|\psi_t\rangle\ \mathrm{d}W_t^j\,, \tag{21.40}$$

where $\{W_t^j\}_{j=1}^J$ are independent Wiener processes. This stochastic differential equation should be interpreted in the Itô sense. Applying Itô's formula to $|\psi_t\rangle\langle\psi_t|$ and taking expectations yields

$$\frac{\mathrm{d}\mathbb{E}\left[|\psi_t\rangle\langle\psi_t|\right]}{\mathrm{d}t} = -i\left[H, \mathbb{E}\left[|\psi_t\rangle\langle\psi_t|\right]\right] + \sum_{j=1}^{J} V_j \mathbb{E}\left[|\psi_t\rangle\langle\psi_t|\right] V_j^\dagger - \frac{1}{2}\sum_{j=1}^{J}\left\{V_j^\dagger V_j, \mathbb{E}\left[|\psi_t\rangle\langle\psi_t|\right]\right\}. \tag{21.41}$$

If the initial condition is $\mathbb{E}\left[|\psi_0\rangle\langle\psi_0|\right] = \rho(0)$, then Eq. (21.41) is equivalent to Eq. (21.1) with $\rho(t) = \mathbb{E}\left[|\psi_t\rangle\langle\psi_t|\right]$.

21.2. Example: Dissipative quantum thermal and ground state preparation

A useful feature of Lindblad dynamics is that certain states are stationary, or fixed points of the dynamics, satisfying

$$\partial_t\sigma = \mathcal{L}(\sigma) = 0. \tag{21.42}$$

Fixed points also appear in unitary dynamics, where a state σ satisfies $-i[H, \sigma] = 0$. However, in the unitary setting, any matrix that commutes with the Hamiltonian is a fixed point of the dynamics. This includes all eigenstates and convex combinations of eigenstates (as mixed states), making the set of fixed points too large to be algorithmically useful. In contrast, when dissipation is introduced through jump operators, certain Lindbladians can have a **unique** fixed point σ. In such cases, the dynamics is said to be **ergodic**. Ergodic Lindbladians thus offer a powerful framework for algorithm design, where the target state, such as a quantum thermal or ground state, is encoded as the unique fixed point of the dynamics. This approach is commonly referred to as **dissipative state engineering** or **dissipative state preparation**.

A common construction for dissipative state preparation starts from a family of coupling operators $\{A_a\}_{a\in\mathcal{A}}$, where each A_a may be simple, for example a Pauli operator, and does not depend explicitly on H. From these operators we define a family of jump operators by

$$V_a(\omega) := \int_{-\infty}^{\infty} f_\omega(s)\, e^{iHs} A_a e^{-iHs}\, \mathrm{d}s. \tag{21.43}$$

Because the jump operators are built from the Heisenberg evolution generated by H, they encode spectral information about the system. Here $f_\omega(s)$ is a filtering function, and ω may range over either a discrete or a continuous set. If $f_\omega(s)$ is smooth and decays rapidly as $|s| \to \infty$, then the integral in Eq. (21.43) can be truncated and discretized, and a block encoding of $V_a(\omega)$ can be constructed using the LCU method.

The resulting Lindbladian takes the form

$$\mathcal{L}(\rho) = -i[G,\rho] + \sum_a \int \left(V_a(\omega)\rho V_a(\omega)^\dagger - \frac{1}{2}\left\{ V_a(\omega)^\dagger V_a(\omega), \rho \right\} \right) \mathrm{d}\mu(\omega), \tag{21.44}$$

where G is the coherent term, constructed from H and the jump operators $V_a(\omega)$. The measure $\mu(\omega)$ determines how the parameter ω is sampled. If $\mu(\omega)$ is discrete, then the Lindbladian becomes

$$\mathcal{L}(\rho) = -i[G,\rho] + \sum_{a,\omega} \mu(\omega) \left(V_a(\omega)\rho V_a(\omega)^\dagger - \frac{1}{2}\left\{ V_a(\omega)^\dagger V_a(\omega), \rho \right\} \right), \tag{21.45}$$

which can be simulated using the methods discussed in previous chapters. If $\mu(\omega)$ is continuous, then the integral must first be discretized for simulation.

Despite this formal expression, the construction never requires explicitly diagonalizing H, which would be prohibitively expensive for large quantum systems.

To see how this framework can encode a target state, consider the simplest case of ground-state preparation. Let $\{\lambda_i, |\psi_i\rangle\}$ be the eigenpairs of H, ordered so that $\lambda_0 < \lambda_1 \leq \cdots$, and let $\Delta = \lambda_1 - \lambda_0$ be the spectral gap. For simplicity, take a single coupling operator A and define a single jump operator

$$V := \int_{-\infty}^{\infty} f(s)\, e^{iHs} A e^{-iHs}\, \mathrm{d}s. \tag{21.46}$$

This corresponds to choosing $\mu(\omega)$ as a discrete probability measure supported on a singleton set. Define the frequency-domain filter $\hat{f}(\omega)$ as the Fourier transform of the time-domain filter function:

$$\hat{f}(\nu) = \int_{\mathbb{R}} f(s) e^{i\nu s}\, \mathrm{d}s. \tag{21.47}$$

Inserting a resolution of the identity, we obtain

$$V = \sum_{i,j} \hat{f}(\lambda_i - \lambda_j)\, |\psi_i\rangle \langle\psi_i| A |\psi_j\rangle \langle\psi_j|. \tag{21.48}$$

As an idealized picture, suppose $\hat{f}$ satisfies

$$\hat{f}(\nu) = \begin{cases} 1, & \nu \leq -\Delta, \\ 0, & \nu \geq 0. \end{cases} \tag{21.49}$$

Then V only maps higher-energy states to lower-energy states, and never the reverse. In particular, $V|\psi_0\rangle = 0$. Moreover, for any $\lambda_j > \lambda_0$, if there exists some $\lambda_i \leq \lambda_j - \Delta$ such that $\langle\psi_i| A |\psi_j\rangle \neq 0$, then $V|\psi_j\rangle \neq 0$. This gives a mechanism by which the ground state can be selected as a dark state,

although uniqueness of the dark state and convergence to it require additional assumptions on the coupling structure.

Now choose $G = H$. The associated Lindblad dynamics is

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -i[H,\rho] + \left(V\rho V^\dagger - \frac{1}{2}\{V^\dagger V, \rho\}\right). \tag{21.50}$$

Then $|\psi_0\rangle\langle\psi_0|$ is a fixed point of the dynamics. Under further assumptions one can prove convergence to the ground state and obtain quantitative mixing bounds; see, for example, [ZDH+26] for rigorous rapid-mixing results in a ground-state preparation setting.

These dissipative algorithms differ from most algorithms discussed in this book. For a given problem, the choice of coupling operators $\{A_a\}$ and filters can be highly system-dependent and offers a broad design space. They can be tailored to specific problems rather than relying only on coarse-grained parameters such as the spectral gap or the initial overlap with the target state. Preparing thermal states involves additional considerations, and proving ergodicity or quantitative mixing bounds typically requires problem-specific analysis.

In dissipative algorithms, assuming the dynamics converges to a fixed point σ, the time to reach σ from an initial state ρ_0 is characterized by the **mixing time**, defined in terms of the trace distance as

$$\tau_{\mathrm{mix}}(\eta) = \inf\left\{t \mid \left\|e^{\mathcal{L}t}(\rho_0) - \sigma\right\|_1 \le \eta, \quad \text{for all initial states } \rho_0\right\}. \tag{21.51}$$

The mixing time is often studied at a fixed precision, for example $\eta = \frac{1}{2}$. In quantum many-body systems on n sites, such as spin, bosonic, or fermionic systems, the dynamics is said to exhibit **fast mixing** if $\tau_{\mathrm{mix}} = \mathcal{O}(\mathrm{poly}(n))$, and **rapid mixing** if $\tau_{\mathrm{mix}} = \mathcal{O}(\mathrm{polylog}(n))$. Establishing such mixing bounds for a given class of quantum Hamiltonians is an active area of research.

21.3. Simulating a dilated Hamiltonian

The proof of Theorem 21.2 directly yields an algorithm for simulating Lindblad dynamics by repeatedly simulating a dilated Hamiltonian $\widetilde{H}$ and tracing out the ancilla register; see Fig. 21.1. More generally, the Stinespring dilation theorem (Theorem 3.20) states that any quantum channel can be represented as a unitary evolution on an extended Hilbert space that includes both the system and an ancillary register. This scheme is often called repeated interactions. Consequently, Hamiltonian simulation algorithms can be deployed directly to simulate Lindblad dynamics.

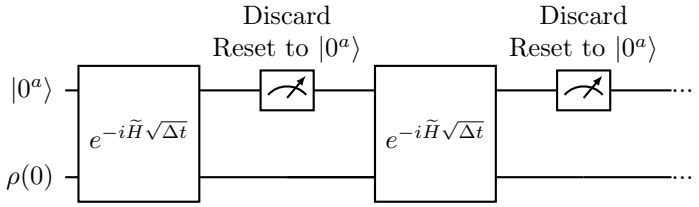


FIGURE 21.1. Simulating Lindblad dynamics by repeatedly simulating a dilated Hamiltonian $\widetilde{H}$ and tracing out the ancilla register.

Since $e^{\mathcal{L}t}$ is a quantum channel for every $t \ge 0$, we can refine Theorem 21.2 to obtain a more explicit error bound.

THEOREM 21.5. *Let* $\|\mathcal{L}\|_{\text{be}} := 1 + \|H\| + \sum_j \|V_j\|^2$, *and let* Φ *be defined in Eq.* (21.17). *Suppose* $\Delta t = \mathcal{O}(\|\mathcal{L}\|_{\text{be}}^{-1})$, *and write* $t = N_t \Delta t$. *Then*

$$\left\| e^{\mathcal{L}t} - \Phi^{N_t} \right\|_\diamond = \mathcal{O}\left(t \, \|\mathcal{L}\|_{\text{be}}^2 \, \Delta t \right). \tag{21.52}$$

PROOF. We refine the proof of Theorem 21.2 as follows. Define

$$B := \sum_{j=1}^{J} |j\rangle\langle 0| \otimes V_j. \tag{21.53}$$

Then

$$\widetilde{H} = |0\rangle\langle 0| \otimes H\sqrt{\Delta t} + B + B^\dagger. \tag{21.54}$$

Relative to the decomposition $\mathbb{C}^{J+1} = \text{span}\{|0\rangle\} \oplus \text{span}\{|1\rangle, \ldots, |J\rangle\}$, the operator $B + B^\dagger$ has block form

$$\begin{pmatrix} 0 & B^\dagger \\ B & 0 \end{pmatrix}, \tag{21.55}$$

so $\left\| B + B^\dagger \right\| = \|B\|$. Moreover,

$$B^\dagger B = |0\rangle\langle 0| \otimes \sum_{j=1}^{J} V_j^\dagger V_j, \tag{21.56}$$

which implies

$$\|B\|^2 = \left\| \sum_{j=1}^{J} V_j^\dagger V_j \right\| \leq \sum_{j=1}^{J} \|V_j\|^2 . \tag{21.57}$$

Therefore,

$$\left\| \widetilde{H} \right\| \leq \|H\| \sqrt{\Delta t} + \|B\| , \tag{21.58}$$

and hence

$$\left\| \widetilde{H} \right\|^2 \leq 2 \|H\|^2 \Delta t + 2 \|B\|^2 \leq 2 \|H\|^2 \Delta t + 2 \sum_{j=1}^{J} \|V_j\|^2 . \tag{21.59}$$

Since $\Delta t = \mathcal{O}(\|\mathcal{L}\|_{\text{be}}^{-1})$ and $\|\mathcal{L}\|_{\text{be}} \geq 1 + \|H\|$, the first term is $\mathcal{O}(\|\mathcal{L}\|_{\text{be}})$. The second term is also $\mathcal{O}(\|\mathcal{L}\|_{\text{be}})$ by definition. Thus

$$\left\| \widetilde{H} \right\|^2 = \mathcal{O}(\|\mathcal{L}\|_{\text{be}}). \tag{21.60}$$

Then by Eq. (21.20),

$$\|U - W\| = \mathcal{O}\left(\left\| \widetilde{H} \right\|^4 \Delta t^2 \right) = \mathcal{O}\left(\|\mathcal{L}\|_{\text{be}}^2 \, \Delta t^2 \right). \tag{21.61}$$

Hence

$$\left\| \Phi - \widetilde{\Phi} \right\|_\diamond = \mathcal{O}(\|\mathcal{L}\|_{\text{be}}^2 \, \Delta t^2), \tag{21.62}$$

where the quantum channel $\widetilde{\Phi}(\rho) = \text{Tr}_a \, W \left(|0\rangle\langle 0| \otimes \rho \right) W^\dagger$.

A direct expansion gives

$$\widetilde{\Phi}(\rho) = \rho + \Delta t \mathcal{L}(\rho) + \mathcal{O}(\|\mathcal{L}\|_{\text{be}}^2 \Delta t^2). \tag{21.63}$$

Taylor expansion also gives

$$e^{\mathcal{L}\Delta t}(\rho) = \rho + \Delta t \mathcal{L}(\rho) + \mathcal{O}(\|\mathcal{L}\|_{\text{be}}^2 \Delta t^2). \tag{21.64}$$

Putting all results above together, we obtain a refined estimation of the approximation error in Eq. (21.13), i.e.,

$$\left\| e^{\mathcal{L}\Delta t} - \Phi \right\|_\diamond = \mathcal{O}(\|\mathcal{L}\|_{\text{be}}^2 \Delta t^2). \tag{21.65}$$

The linear error growth property of quantum channels then yields the desired long-time estimate

$$\left\| e^{\mathcal{L}t} - \Phi^{N_t} \right\|_\diamond \leq N_t \left\| e^{\mathcal{L}\Delta t} - \Phi \right\|_\diamond = \mathcal{O}\left(t \|\mathcal{L}\|_{\text{be}}^2 \Delta t \right). \tag{21.66}$$

□

Example 21.6 (Hamiltonian evolution with dephasing)**.** Consider the one-dimensional transverse-field Ising model (TFIM) on n sites with nearest-neighbor interactions,

$$H = -\sum_{j=1}^{n-1} Z_j Z_{j+1} - g \sum_{j=1}^{n} X_j, \tag{21.67}$$

where g is the coupling constant. We now apply a dephasing operation to each spin. This can be implemented by associating a jump operator $V_j = \sqrt{\gamma} Z_j$ to each spin, where γ is called the dephasing rate.

The Lindblad equation describing the time evolution of the density matrix ρ for this system is

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -i[H, \rho] + \sum_{j=1}^{n} \left(V_j \rho V_j^\dagger - \frac{1}{2} \{ V_j^\dagger V_j, \rho \} \right). \tag{21.68}$$

Substituting the expressions for H and V_j gives

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = i \left[\sum_{j=1}^{n-1} Z_j Z_{j+1} + g \sum_{j=1}^{n} X_j, \rho \right] + \gamma \sum_{j=1}^{n} \left(Z_j \rho Z_j - \rho \right). \tag{21.69}$$

Let us now apply the method based on simulating a dilated Hamiltonian to solve this problem. The dilated Hamiltonian takes the form

$$\widetilde{H} = |0\rangle\langle 0| \otimes H\sqrt{\Delta t} + \sum_{j=1}^{n} (|j\rangle\langle 0| \otimes \sqrt{\gamma} Z_j + \text{h.c.}) \tag{21.70}$$

which uses $\lceil \log_2(n+1) \rceil$ ancilla qubits. The dilated Hamiltonian in its matrix form is

$$\widetilde{H} = \begin{pmatrix} H\sqrt{\Delta t} & \sqrt{\gamma} Z_1 & \cdots & \sqrt{\gamma} Z_n \\ \sqrt{\gamma} Z_1 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & 0 \\ \sqrt{\gamma} Z_n & 0 & \cdots & 0 \end{pmatrix}. \tag{21.71}$$

We can then simulate the Lindblad equation by repeatedly simulating this dilated Hamiltonian $U = e^{-i\widetilde{H}\sqrt{\Delta t}}$ and tracing out the ancilla qubits. Assuming $g, \gamma = \mathcal{O}(1)$, we have

$$\|\mathcal{L}\|_{\mathrm{be}} = 1 + \|H\| + \gamma \sum_{j=1}^{n} \|Z_j\|^2 = \mathcal{O}(n), \tag{21.72}$$

since $\|H\| = \mathcal{O}(n)$ and $\|Z_j\| = 1$. To obtain trace-distance error at most ϵ at time T, it suffices to choose

$$\Delta t = \mathcal{O}(\epsilon/(T\,\|\mathcal{L}\|_{\mathrm{be}}^2)), \tag{21.73}$$

provided this choice also satisfies the step-size condition $\Delta t = \mathcal{O}(\|\mathcal{L}\|_{\mathrm{be}}^{-1})$. The resulting number of channel steps is

$$N_t = T/\Delta t = \mathcal{O}(T^2\,\|\mathcal{L}\|_{\mathrm{be}}^2/\epsilon) = \mathcal{O}(T^2 n^2/\epsilon), \tag{21.74}$$

which is also the number of queries to U. Note that the unitary $U = e^{-i\widetilde{H}\sqrt{\Delta t}}$ still needs to be prepared using Hamiltonian evolution methods. ◇

21.4. Operator splitting method

Suppose the Lindbladian is decomposed as

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2, \tag{21.75}$$

where each semigroup $e^{\mathcal{L}_1 t}$ and $e^{\mathcal{L}_2 t}$ is a quantum channel for every $t \geq 0$. For example, one may take

$$\mathcal{L}_1 = \mathcal{L}_H, \qquad \mathcal{L}_2 = \mathcal{L}_V, \tag{21.76}$$

or refine the dissipative part further by splitting $\mathcal{L}_V$ into simpler channel generators.

If the factors $e^{\mathcal{L}_1 \Delta t}$ and $e^{\mathcal{L}_2 \Delta t}$ can be simulated efficiently, the natural first-order product formula is

$$\Phi(t) := e^{\mathcal{L}_1 t} e^{\mathcal{L}_2 t}. \tag{21.77}$$

As in the setting of Hamiltonian simulation, there are two useful ways to bound the error: a commutator bound, which captures near-commutativity, and a coarser operator norm bound that can be obtained from it as a corollary.

We begin with the commutator bound. Differentiating $\Phi(t)$ gives

$$\begin{aligned} \partial_t \Phi(t) &= \mathcal{L}_1 e^{t\mathcal{L}_1} e^{t\mathcal{L}_2} + e^{t\mathcal{L}_1} \mathcal{L}_2 e^{t\mathcal{L}_2} \\ &= (\mathcal{L}_1 + \mathcal{L}_2) e^{t\mathcal{L}_1} e^{t\mathcal{L}_2} + e^{t\mathcal{L}_1} \mathcal{L}_2 e^{t\mathcal{L}_2} - \mathcal{L}_2 e^{t\mathcal{L}_1} e^{t\mathcal{L}_2} \\ &= \mathcal{L}\Phi(t) + [e^{t\mathcal{L}_1}, \mathcal{L}_2] e^{t\mathcal{L}_2}, \end{aligned} \tag{21.78}$$

with initial condition $\Phi(0) = \mathcal{I}$. By the Duhamel principle,

$$\Phi(t) = e^{\mathcal{L}t} + \int_0^t e^{\mathcal{L}(t-s)} [e^{s\mathcal{L}_1}, \mathcal{L}_2] e^{s\mathcal{L}_2}\, \mathrm{d}s. \tag{21.79}$$

To rewrite the commutator term, define

$$G(s) := [e^{s\mathcal{L}_1}, \mathcal{L}_2] \tag{21.80}$$

and

$$\widetilde{G}(s) := \int_0^s e^{\tau\mathcal{L}_1} [\mathcal{L}_1, \mathcal{L}_2] e^{(s-\tau)\mathcal{L}_1}\, \mathrm{d}\tau. \tag{21.81}$$

Then $G(0) = \widetilde{G}(0) = 0$, and both satisfy the same inhomogeneous differential equation,

$$\partial_s G(s) = e^{s\mathcal{L}_1}[\mathcal{L}_1, \mathcal{L}_2] + G(s)\mathcal{L}_1, \tag{21.82}$$

$$\partial_s \widetilde{G}(s) = e^{s\mathcal{L}_1}[\mathcal{L}_1, \mathcal{L}_2] + \widetilde{G}(s)\mathcal{L}_1. \tag{21.83}$$

Therefore,

$$[e^{s\mathcal{L}_1}, \mathcal{L}_2] = \int_0^s e^{\tau\mathcal{L}_1}[\mathcal{L}_1, \mathcal{L}_2]e^{(s-\tau)\mathcal{L}_1}\,\mathrm{d}\tau. \tag{21.84}$$

Since $e^{\mathcal{L}(t-s)}$, $e^{\tau\mathcal{L}_1}$, $e^{(s-\tau)\mathcal{L}_1}$, and $e^{s\mathcal{L}_2}$ are all quantum channels, their diamond norms are at most 1. Hence

$$\left\|e^{\mathcal{L}t} - \Phi(t)\right\|_\diamond \le \int_0^t \int_0^s \left\|[\mathcal{L}_1, \mathcal{L}_2]\right\|_\diamond \,\mathrm{d}\tau\,\mathrm{d}s = \frac{t^2}{2}\left\|[\mathcal{L}_1, \mathcal{L}_2]\right\|_\diamond. \tag{21.85}$$

This is the direct analogue of the commutator Trotter bound in Section 15.4. In particular, if $[\mathcal{L}_1, \mathcal{L}_2] = 0$, then the first-order splitting is exact.

The commutator bound immediately implies a coarser norm bound. Indeed,

$$\left\|[\mathcal{L}_1, \mathcal{L}_2]\right\|_\diamond \le 2\left\|\mathcal{L}_1\right\|_\diamond \left\|\mathcal{L}_2\right\|_\diamond \le (\left\|\mathcal{L}_1\right\|_\diamond + \left\|\mathcal{L}_2\right\|_\diamond)^2. \tag{21.86}$$

To connect this with the Lindblad data, note first that

$$\left\|\mathcal{L}_H\right\|_\diamond \le 2\left\|H\right\|. \tag{21.87}$$

Also, for a single jump operator V_j, the dissipative generator

$$\mathcal{D}_j(\rho) := V_j\rho V_j^\dagger - \frac{1}{2}\{V_j^\dagger V_j, \rho\} \tag{21.88}$$

satisfies

$$\left\|\mathcal{D}_j\right\|_\diamond \le 2\left\|V_j\right\|^2. \tag{21.89}$$

Therefore, for any partition of the coherent part and dissipative terms into two groups defining $\mathcal{L}_1$ and $\mathcal{L}_2$, we have

$$\left\|\mathcal{L}_1\right\|_\diamond + \left\|\mathcal{L}_2\right\|_\diamond \le 2\left(\left\|H\right\| + \sum_j \left\|V_j\right\|^2\right) \le 2\left\|\mathcal{L}\right\|_{\mathrm{be}}. \tag{21.90}$$

Substituting into Eq. (21.85) yields the operator norm based error bound

$$\left\|e^{\mathcal{L}t} - \Phi(t)\right\|_\diamond \le 2t^2\left\|\mathcal{L}\right\|_{\mathrm{be}}^2. \tag{21.91}$$

For long-time simulation, let $t = N_t\Delta t$. The linear error growth property of quantum channels gives

$$\left\|e^{\mathcal{L}t} - \Phi(\Delta t)^{N_t}\right\|_\diamond \le N_t\left\|e^{\mathcal{L}\Delta t} - \Phi(\Delta t)\right\|_\diamond = \mathcal{O}\left(t\left\|[\mathcal{L}_1, \mathcal{L}_2]\right\|_\diamond \Delta t\right). \tag{21.92}$$

Using the coarse norm bound above, this further implies

$$\left\|e^{\mathcal{L}t} - \Phi(\Delta t)^{N_t}\right\|_\diamond = \mathcal{O}\left(t\left\|\mathcal{L}\right\|_{\mathrm{be}}^2 \Delta t\right). \tag{21.93}$$

As in the Hamiltonian setting, higher-order product formulas cannot be directly implemented since they typically require negative time steps, whereas $e^{-\mathcal{L}t}$ need not be a quantum channel.

21.5. Truncated Dyson method

To develop a truncated Dyson method for simulating the Lindblad dynamics, it is convenient to separate the Lindbladian into a no-jump part and a jump part. The Lindblad equation Eq. (21.1) can then be written as

$$\partial_t \rho(t) = \underbrace{-i(H_{\text{eff}}\rho(t) - \rho(t)H_{\text{eff}}^\dagger)}_{\text{Non-Hermitian, } \mathcal{L}_N[\rho(t)]} + \underbrace{\sum_{j=1}^J V_j \rho(t) V_j^\dagger}_{\text{Diffusion, } \mathcal{L}_D[\rho(t)]}. \tag{21.94}$$

Here

$$H_{\text{eff}} = H - \frac{i}{2}\sum_{j=1}^J V_j^\dagger V_j, \tag{21.95}$$

is a non-Hermitian effective Hamiltonian. The map generated by $\mathcal{L}_N$ is

$$e^{\mathcal{L}_N t}(\rho) = e^{-iH_{\text{eff}}t}\rho e^{iH_{\text{eff}}^\dagger t}. \tag{21.96}$$

Since H_{eff} is non-Hermitian, the operator $e^{-iH_{\text{eff}}t}$ is not unitary. However, **??** implies that $\left\|e^{-iH_{\text{eff}}t}\right\| \leq 1$. Therefore we may assume that there is a block encoding

$$\mathcal{V}_N(t) \in \mathrm{BE}_{1,a}(e^{-iH_{\text{eff}}t}). \tag{21.97}$$

Consequently,

$$e^{\mathcal{L}_N t}(\rho) = \mathrm{Tr}_a\left[\mathcal{V}_N(t)(|0^a\rangle\langle 0^a| \otimes \rho)\mathcal{V}_N^\dagger(t)\right]. \tag{21.98}$$

The non-Hermitian evolution $e^{-iH_{\text{eff}}t}$ can be simulated using the differential-equation solvers in Chapter 20. Indeed,

$$A = iH_{\text{eff}} = \frac{1}{2}\sum_{j=1}^J V_j^\dagger V_j + iH, \tag{21.99}$$

satisfies the positivity requirement because its Hermitian part is $\frac{1}{2}\sum_{j=1}^J V_j^\dagger V_j \succeq 0$. Therefore the LCHS solver in **??** applies directly to this non-Hermitian quantum dynamics.

The diffusion part $\mathcal{L}_D$ is already written in Kraus form. Although the normalization condition $\sum_j V_j^\dagger V_j = I$ is usually not satisfied, we can still construct a block encoding of the jump operators, denoted by $\mathcal{V}_D$, such that after tracing out the ancillas we obtain, for some subnormalization factor β,

$$\mathrm{Tr}_b[\mathcal{V}_D(|0^b\rangle\langle 0^b| \otimes \rho)\mathcal{V}_D^\dagger] = \frac{1}{\beta}\mathcal{L}_D(\rho). \tag{21.100}$$

Applying the Duhamel principle to the decomposition $\mathcal{L} = \mathcal{L}_N + \mathcal{L}_D$ gives

$$\begin{aligned}\rho(t) =& e^{\mathcal{L}_N t}\rho(0) + \int_0^t e^{\mathcal{L}_N(t-s)}\mathcal{L}_D[\rho(s)]\,\mathrm{d}s \\ =& e^{\mathcal{L}_N t}\rho(0) + \int_0^t e^{\mathcal{L}_N(t-s)}\beta\,\mathrm{Tr}_b\left[\mathcal{V}_D(|0^b\rangle\langle 0^b| \otimes \rho(s))\mathcal{V}_D^\dagger\right]\,\mathrm{d}s.\end{aligned} \tag{21.101}$$

For a single short step, replacing $\rho(s)$ by $\rho(0)$ inside the integral yields the first-order truncated Dyson approximation

$$\begin{aligned}\rho(\Delta t) =& e^{\mathcal{L}_N \Delta t}\rho(0) + \int_0^{\Delta t} e^{\mathcal{L}_N(\Delta t - s)}\mathcal{L}_D[\rho(0)]\, \mathrm{d}s + \mathcal{O}(\Delta t^2) \\ =& e^{\mathcal{L}_N \Delta t}\rho(0) + \Delta t\, e^{\mathcal{L}_N \Delta t}\mathcal{L}_D[\rho(0)] + \mathcal{O}(\Delta t^2).\end{aligned} \tag{21.102}$$

If we drop the $\mathcal{O}(\Delta t^2)$ term, the resulting short-time approximation remains completely positive. More concretely, it has Kraus operators

$$K_0 = e^{-iH_{\mathrm{eff}}\Delta t}, \qquad K_j = \sqrt{\Delta t}\, V_j, \qquad 1 \le j \le J. \tag{21.103}$$

Indeed, because $H_{\mathrm{eff}} = H - \frac{i}{2}\sum_j V_j^\dagger V_j$, these operators satisfy

$$K_0^\dagger K_0 + \sum_{j=1}^{J} K_j^\dagger K_j = I + \mathcal{O}(\Delta t^2). \tag{21.104}$$

Therefore a standard completion argument adds one more Kraus operator and yields a quantum channel Φ such that

$$\rho(\Delta t) = \Phi(\rho(0)), \quad \left\|e^{\mathcal{L}\Delta t} - \Phi\right\|_\diamond = \mathcal{O}(\Delta t^2). \tag{21.105}$$

Compared with the Hamiltonian-simulation-based method and the Trotter-based method, the truncated Dyson approach is more involved, but it extends naturally to arbitrary high order. We refer readers to [LW23] for the high-order construction.

Notes and further reading

For background on open quantum systems, see [BP02]. In finite dimensions, the characterization of generators of quantum dynamical semigroups was obtained independently by Lindblad and by Gorini, Kossakowski, and Sudarshan in 1976 [Lin76, GKS76]; see also the short review [Man20]. In physical derivations from weak coupling to a bath, the Davies construction is the standard reference [Dav74, Dav76]; for a pedagogical discussion of the approximations behind the Markovian master equation, see [Lid19]. The chapter works entirely in finite dimensions, so it avoids the domain issues that arise for unbounded operators in infinite-dimensional settings.

For quantum algorithms, an early simulation result based on product formulas was given in [KBG$^+$11], and sparse Lindbladian simulation with higher-order product formulas was developed in [CL17]. The technical limitation is that higher-order splitting formulas usually require negative time steps, whereas $e^{-\mathcal{L}t}$ need not be a quantum channel. This is one reason to pass to dilation-based and series-based constructions. The Stinespring-dilation viewpoint underlies the Hamiltonian-simulation approach in [CW17, DLL24b], while higher-order truncated Dyson constructions were developed in [LW23]. These methods are closely related to exponential-integrator ideas in numerical analysis, but the complete positivity constraint forces a different implementation strategy.

Ref. [VWC09] is a seminal paper on dissipative state preparation. Recent directions include dissipative preparation of structured many-body states [RCGG20, ZCL21, LLKH22, WSR$^+$23, Cub23], Lindbladian algorithms for thermal or ground-state preparation [TOV$^+$11, RWW23, SM21, GCDK24, DLL24a, DCL24, ZDH$^+$26], and rigorous analyses of mixing based on quantum logarithmic Sobolev inequalities and related tools [TKR$^+$10, KT13, BCG$^+$23, RFA24a, KACR25, RFA24b]. These mixing questions are the open-system analogue of convergence-rate questions for classical

Markov chains, and they are central when Lindblad dynamics is used as an algorithm rather than as a physical model.

Bibliography

[AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In **Proceedings of the forty-third annual ACM symposium on Theory of computing**, pages 333–342, 2011.

[Aar14] Scott Aaronson. Quantum machine learning algorithms: Read the fine print. **Nat. Phys.**, page 5, 2014.

[AAT24] Junaid Aftab, Dong An, and Konstantina Trivisa. Multi-product hamiltonian simulation with explicit commutator scaling. **arXiv preprint arXiv:2403.08922**, 2024.

[ABF16] F. Arrigo, M. Benzi, and C. Fenu. Computation of generalized matrix functions. **SIAM J. Matrix Anal. Appl.**, 37:836–860, 2016.

[ABO97] Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error. In **Proceedings of the twenty-ninth annual ACM symposium on Theory of computing**, pages 176–188, 1997.

[ACC+22] James Ang, Gabriella Carini, Yanzhu Chen, Isaac Chuang, Michael DeMarco, Sophia Economou, Alec Eickbusch, Andrei Faraon, Kai-Mei Fu, Steven Girvin, et al. Arquin: Architectures for multinode superconducting quantum computers. **ACM Transactions on Quantum Computing**, 2022.

[ACL26] Dong An, Andrew M Childs, and Lin Lin. Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters: D. an, am childs, l. lin. **Communications in Mathematical Physics**, 407(1):19, 2026.

[ACNR22] Simon Apers, Shantanav Chakraborty, Leonardo Novo, and Jérémie Roland. Quadratic speedup for spatial search by continuous-time quantum walk. **Phys. Rev. Lett.**, 129:160502, 2022.

[ADW17] Srinivasan Arunachalam and Ronald De Wolf. Guest column: A survey of quantum learning theory. **ACM Sigact News**, 48(2):41–67, 2017.

[AFL21] Dong An, Di Fang, and Lin Lin. Time-dependent unbounded hamiltonian simulation with vector norm scaling. **Quantum**, 5:459, 2021.

[AFL22] Dong An, Di Fang, and Lin Lin. Time-dependent hamiltonian simulation of highly oscillatory dynamics and superconvergence for schrödinger equation. **Quantum**, 6:690, 2022.

[AGJ21] Simon Apers, András Gilyén, and Stacey Jeffery. A unified framework of quantum walk search. In **38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)**, volume 187, pages 6:1–6:13, 2021.

[ALL23] Dong An, Jin-Peng Liu, and Lin Lin. Linear combination of hamiltonian simulation for nonunitary dynamics with optimal state preparation cost. **Phys. Rev. Lett.**, 131:150603, 2023.

[ALM+26] Michel Alexis, Lin Lin, Gevorg Mnatsakanyan, Christoph Thiele, and Jiasu Wang. Infinite quantum signal processing for arbitrary Szegő functions. **Commun. Pure**

Appl. Math., 79:123, 2026.

[ALWZ25] Dong An, Jin-Peng Liu, Daochen Wang, and Qi Zhao. Quantum differential equation solvers: limitations and fast-forwarding. **Commun. Math. Phys.**, 406:189, 2025.

[AMT24] Michel Alexis, Gevorg Mnatsakanyan, and Christoph Thiele. Quantum signal processing and nonlinear fourier analysis. **Revista Matemática Complutense**, pages 1–40, 2024.

[BACS07] Dominic W Berry, Graeme Ahokas, Richard Cleve, and Barry C Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. **Commun. Math. Phys.**, 270:359–371, 2007.

[BBC+95] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. **Phys. Rev. A**, 52:3457, 1995.

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald De Wolf. Quantum lower bounds by polynomials. **Journal of the ACM (JACM)**, 48:778–797, 2001.

[BBHT98] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. **Fortschritte der Physik: Progress of Physics**, 46(4-5):493–505, 1998.

[BBK+23] Ryan Babbush, Dominic W Berry, Robin Kothari, Rolando D Somma, and Nathan Wiebe. Exponential quantum speedup in simulating coupled classical oscillators. **Phys. Rev. X**, 13:041041, 2023.

[BC94] Samuel L Braunstein and Carlton M Caves. Statistical distance and the geometry of quantum states. **Phys. Rev. Lett.**, 72:3439, 1994.

[BC12] Dominic W Berry and Andrew M Childs. Black-box hamiltonian simulation and unitary implementation. **Quantum Information & Computation**, 12:29–62, 2012.

[BCC+14] Dominic W Berry, Andrew M Childs, Richard Cleve, Robin Kothari, and Rolando D Somma. Exponential improvement in precision for simulating sparse Hamiltonians. In **Proceedings of the forty-sixth annual ACM symposium on Theory of computing**, pages 283–292, 2014.

[BCG14] Dominic W Berry, Richard Cleve, and Sevag Gharibian. Gate-efficient discrete simulations of continuous-time quantum query algorithms. **Quantum Information and Computation**, 14:1–30, 2014.

[BCG+23] Ivan Bardet, Ángela Capel, Li Gao, Angelo Lucia, David Pérez-García, and Cambyse Rouzé. Rapid thermalization of spin chain commuting hamiltonians. **Phys. Rev. Lett.**, 130:060401, 2023.

[BCK15] D. W. Berry, A. M. Childs, and R. Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. **Proceedings of the 56th IEEE Symposium on Foundations of Computer Science**, pages 792–809, 2015.

[BCOR09] S. Blanes, F. Casas, J. A. Oteo, and J. Ros. The Magnus expansion and some of its applications. **Phys. Rep.**, 470:151–238, 2009.

[Ben87] Charles H Bennett. Demons, engines and the second law. **Scientific American**, 257:108–117, 1987.

[BGB+18] Ryan Babbush, Craig Gidney, Dominic W Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. Encoding electronic spectra in quantum circuits with linear t complexity. **Phys. Rev. X**, 8:041015, 2018.

[Bha97] Rajendra Bhatia. **Matrix Analysis**, volume 169. Springer, 1997.

[BHB+09] Dominic W Berry, Brendon L Higgins, Stephen D Bartlett, Morgan W Mitchell, Geoff J Pryde, and Howard M Wiseman. How to perform the most accurate possible phase measurements. **Phys. Rev. A**, 80:052114, 2009.

[BHMT02] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. **Contemp. Math.**, 305:53–74, 2002.

[BHW13] Dominic W Berry, Michael JW Hall, and Howard M Wiseman. Stochastic heisenberg limit: optimal estimation of a fluctuating phase. **Phys. Rev. Lett.**, 111:113601, 2013.

[BP02] Heinz-Peter Breuer and Francesco Petruccione. **The theory of open quantum systems**. OUP Oxford, 2002.

[Bri98] Matthew Edward Briggs. **An introduction to the general number field sieve**. PhD thesis, Virginia Tech, 1998.

[BS24] Bjorn K Berntson and Christoph Sünderhauf. Complementary polynomials in quantum signal processing. **arXiv preprint arXiv:2406.04246**, 2024.

[BSG+24] Dominic W Berry, Yuan Su, Casper Gyurik, Robbie King, Joao Basso, Alexander Del Toro Barba, Abhishek Rajput, Nathan Wiebe, Vedran Dunjko, and Ryan Babbush. Analyzing prospects for quantum advantage in topological data analysis. **PRX Quantum**, 5:010319, 2024.

[BWB01] Dominic W Berry, HM Wiseman, and JK Breslin. Optimal input states and feedback for interferometric phase estimation. **Phys. Rev. A**, 63:053804, 2001.

[BWM+18] Ryan Babbush, Nathan Wiebe, Jarrod McClean, James McClain, Hartmut Neven, and Garnet Kin-Lic Chan. Low-depth quantum simulation of materials. **Physical Review X**, 8(1):011044, 2018.

[Cam19] Earl Campbell. Random compiler for fast hamiltonian simulation. **Phys. Rev. Lett.**, 123:070503, 2019.

[CC02] Siu A Chin and CR Chen. Gradient symplectic algorithms for solving the schrödinger equation with time-dependent potentials. **J. Chem. Phys.**, 117:1409–1415, 2002.

[CCD+03] Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. Exponential algorithmic speedup by a quantum walk. In **Proceedings of the thirty-fifth annual ACM symposium on Theory of computing**, pages 59–68, 2003.

[CCHL22] Sitan Chen, Jordan Cotler, Hsin-Yuan Huang, and Jerry Li. Exponential separations between learning with and without quantum memory. In **2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)**, pages 574–585. IEEE, 2022.

[CDG+20] Rui Chao, Dawei Ding, András Gilyén, Cupjin Huang, and Mario Szegedy. Finding angles for quantum signal processing with machine precision. **arXiv preprint arXiv:2003.02831**, 2020.

[Chi10] Andrew M Childs. On the relationship between continuous-and discrete-time quantum walk. **Commun. Math. Phys.**, 294:581–603, 2010.

[Chi21] Andrew Childs. Lecture notes on quantum algorithms, 2021.

[CHKT21] Chi-Fang Chen, Hsin-Yuan Huang, Richard Kueng, and Joel A Tropp. Concentration for random product formulas. **PRX Quantum**, 2:040305, 2021.

[Cho75] Man-Duen Choi. Completely positive linear maps on complex matrices. **Linear algebra and its applications**, 10(3):285–290, 1975.

[CKS17] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. **SIAM J. Comput.**, 46:1920–1950, 2017.

[CL17] Andrew M Childs and Tongyang Li. Efficient simulation of sparse markovian quantum dynamics. **Quantum Inf Comput**, 17:0901–0947, 2017.

[CMN+18] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. **Proc. Nat. Acad. Sci.**, 115:9456–9461, 2018.

[COS19] Andrew M Childs, Aaron Ostrander, and Yuan Su. Faster quantum simulation by randomization. **Quantum**, 3:182, 2019.

[CS17] Anirban Narayan Chowdhury and Rolando D. Somma. Quantum algorithms for gibbs sampling and hitting-time estimation. **Quantum Inf. Comput.**, 17:41–64, 2017.

[CS19] Andrew M Childs and Yuan Su. Nearly optimal lattice simulation by product formulas. **Phys. Rev. Lett.**, 123:050503, 2019.

[CST+21] Andrew M Childs, Yuan Su, Minh C Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. **Phys. Rev. X**, 11:011020, 2021.

[Cub23] Toby S. Cubitt. Dissipative ground state preparation and the dissipative quantum eigensolver. **arXiv:2303.11962**, 2023.

[CW12] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. **Quantum Information and Computation**, 12:901–924, 2012.

[CW17] Richard Cleve and Chunhao Wang. Efficient quantum algorithms for simulating Lindblad evolution. In **ICALP 2017**, 2017.

[CZA24] Pablo Antonio Moreno Casares, Modjtaba Shokrian Zini, and Juan Miguel Arrazola. Quantum simulation of time-dependent hamiltonians via commutator-free quasi-magnus operators. **Quantum**, 8:1567, 2024.

[Dav74] E Brian Davies. Markovian master equations. **Commun. Math. Phys.**, 39:91–110, 1974.

[Dav76] Edward Brian Davies. Quantum theory of open systems. **(No Title)**, 1976.

[DB16] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. **J. Mach. Learn. Res.**, 17:1–5, 2016.

[DCL24] Zhiyan Ding, Chi-Fang Chen, and Lin Lin. Single-ancilla ground state preparation via Lindbladians. **Phys. Rev. Research**, 6:033147, 2024.

[Dem97] James W. Demmel. **Applied Numerical Linear Algebra**. SIAM, 1997.

[Die25] Reinhard Diestel. **Graph theory**. Springer Nature, 2025.

[DL21] Yulong Dong and Lin Lin. Random circuit block-encoded matrix and a proposal of quantum linpack benchmark. **Phys. Rev. A**, 103:062412, 2021.

[DLL24a] Zhiyan Ding, Bowen Li, and Lin Lin. Efficient quantum gibbs samplers with kubo–martin–schwinger detailed balance condition. **arXiv preprint arXiv:2404.05998**, 2024.

[DLL24b] Zhiyan Ding, Xiantao Li, and Lin Lin. Simulating open quantum systems using Hamiltonian simulations. **PRX Quantum**, 5:020332, 2024.

[DLNW24a] Yulong Dong, Lin Lin, Hongkang Ni, and Jiasu Wang. Infinite quantum signal processing. **Quantum**, 8:1558, 2024.

[DLNW24b] Yulong Dong, Lin Lin, Hongkang Ni, and Jiasu Wang. Robust iterative method for symmetric quantum signal processing in all parameter regimes. **SIAM J. Sci.**

Comput., 46:A2951–A2971, 2024.

[DLT22] Yulong Dong, Lin Lin, and Yu Tong. Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices. **PRX Quantum**, 3:040305, 2022.

[DMWL21] Yulong Dong, Xiang Meng, K Birgitta Whaley, and Lin Lin. Efficient phase factor evaluation in quantum signal processing. **Phys. Rev. A**, 103:042419, 2021.

[Don23] Yulong Dong. **Quantum Signal Processing Algorithm and Its Applications**. PhD thesis, 2023.

[DT10] Stéphane Descombes and Mechthild Thalhammer. An exact local error representation of exponential operator splitting methods for evolutionary problems and applications to linear Schrödinger equations in the semi-classical regime. **BIT Numer. Math.**, 50:729–749, 2010.

[Fey82] Richard P Feynman. Simulating physics with computers. **Int. J. Theor. Phys**, 21, 1982.

[FG98] Edward Farhi and Sam Gutmann. Quantum computation and decision trees. **Phys. Rev. A**, 58:915, 1998.

[FLS25] Di Fang, Diyi Liu, and Rahul Sarkar. Time-dependent hamiltonian simulation via magnus expansion: Algorithm and superconvergence. **Commun. Math. Phys.**, 406:1–36, 2025.

[FN09] Yu Farforovskaya and L Nikolskaya. Modulus of continuity of operator functions. **St. Petersburg Mathematical Journal**, 20:493–506, 2009.

[FVDG02] Christopher A Fuchs and Jeroen Van De Graaf. Cryptographic distinguishability measures for quantum-mechanical states. **IEEE T. Inform. Theory.**, 45:1216–1227, 2002.

[GB14] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, mar 2014.

[GCDK24] András Gilyén, Chi-Fang Chen, Joao F Doriguello, and Michael J Kastoryano. Quantum generalizations of Glauber and Metropolis dynamics. **arXiv preprint arXiv:2405.20322**, 2024.

[GDDWB20] Wojciech Górecki, Rafał Demkowicz-Dobrzański, Howard M Wiseman, and Dominic W Berry. π-corrected heisenberg limit. **Phys. Rev. Lett.**, 124:030501, 2020.

[GFWC12] Christopher E Granade, Christopher Ferrie, Nathan Wiebe, and David G Cory. Robust online hamiltonian learning. **New J. Phys.**, 14:103013, 2012.

[GHV21] András Gilyén, Matthew B Hastings, and Umesh Vazirani. (sub) exponential advantage of adiabatic quantum computation with no sign problem. In **Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing**, pages 1357–1369, 2021.

[Gid18] Craig Gidney. Halving the cost of quantum addition. **Quantum**, 2:74, 2018.

[GKS76] Vittorio Gorini, Andrzej Kossakowski, and Ennackal Chandy George Sudarshan. Completely positive dynamical semigroups of n-level systems. **J. Math. Phys.**, 17:821–825, 1976.

[GLM04] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum-enhanced measurements: beating the standard quantum limit. **Science**, 306:1330–1336, 2004.

[GLM06] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum metrology. **Phys. Rev. LEtt.**, 96:010401, 2006.

[GLM08] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. **Physical review letters**, 100(16):160501, 2008.

[Gro96] Lov K Grover. A fast quantum mechanical algorithm for database search. In **Proceedings of the twenty-eighth annual ACM symposium on Theory of computing**, pages 212–219, 1996.

[GSLW18] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. **arXiv:1806.01838**, 2018.

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In **Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing**, pages 193–204, 2019.

[GVL13] G. H. Golub and C. F. Van Loan. **Matrix computations**. Johns Hopkins Univ. Press, 2013.

[Haa19] J. Haah. Product decomposition of periodic functions in quantum signal processing. **Quantum**, 3:190, 2019.

[HBB+07] Brendon L Higgins, Dominic W Berry, Stephen D Bartlett, Howard M Wiseman, and Geoff J Pryde. Entanglement-free heisenberg-limited phase estimation. **Nature**, 450:393–396, 2007.

[HBC+22] Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, et al. Quantum advantage in learning from experiments. **Science**, 376(6598):1182–1186, 2022.

[HBI73] J. B. Hawkins and A. Ben-Israel. On generalized matrix functions. **Linear and Multilinear Algebra**, 1:163–171, 1973.

[Hel69] Carl W Helstrom. Quantum detection and estimation theory. **Journal of Statistical Physics**, 1(2):231–252, 1969.

[HHB+25] Alexander Hahn, Paul Hartung, Daniel Burgarth, Paolo Facchi, and Kazuya Yuasa. Lower bounds for the trotter error. **Phys. Rev. A**, 111:022417, 2025.

[Hig02] N. J. Higham. **Accuracy and stability of numerical algorithms**, volume 80. Siam, 2002.

[Hig08] N. Higham. **Functions of matrices: theory and computation**, volume 104. SIAM, 2008.

[HJ91] Roger A. Horn and Charles R. Johnson. **Topics in Matrix Analysis**. Cambridge University Press, 1991.

[HLS07] Peter Hoyer, Troy Lee, and Robert Spalek. Negative weights make adversaries stronger. In **Proceedings of the thirty-ninth annual ACM symposium on Theory of computing**, pages 526–535, 2007.

[HLW06] E. Hairer, C. Lubich, and G. Wanner. **Geometric numerical integration: structure-preserving algorithms for ordinary differential equations**, volume 31. Springer, 2006.

[HS10] Alexander Hentschel and Barry C Sanders. Machine learning for precise quantum measurement. **Phys. Rev. Lett.**, 104:063603, 2010.

[HWM+22] William J Huggins, Kianna Wan, Jarrod McClean, Thomas E O'Brien, Nathan Wiebe, and Ryan Babbush. Nearly optimal quantum algorithm for estimating multiple expectation values. **Phys. Rev. Lett.**, 129:240501, 2022.

[Jam72] Andrzej Jamiołkowski. Linear transformations which preserve trace and positive semidefiniteness of operators. **Reports on mathematical physics**, 3(4):275–278, 1972.

[JL00] Tobias Jahnke and Christian Lubich. Error bounds for exponential operator splittings. **BIT. Numerical Mathematics**, 40:735–744, 2000.

[Jor75] Camille Jordan. Essai sur la géométrie à n dimensions. **Bulletin de la Société mathématique de France**, 3:103–174, 1875.

[JSW+25] Stephen P Jordan, Noah Shutty, Mary Wootters, Adam Zalcman, Alexander Schmidhuber, Robbie King, Sergei V Isakov, Tanuj Khattar, and Ryan Babbush. Optimization by decoded quantum interferometry. **Nature**, 646(8086):831–836, 2025.

[KACR25] Jan Kochanowski, Alvaro M Alhambra, Angela Capel, and Cambyse Rouzé. Rapid thermalization of dissipative many-body dynamics of commuting hamiltonians. **Commun. Math. Phys.**, 406:176, 2025.

[Kat76] Tosio Kato. **Perturbation theory for linear operators; 2nd ed.** Springer, 1976.

[KBDW83] Karl Kraus, Arno Böhm, John D Dollard, and WH Wootters. **States, effects, and operations fundamental notions of quantum theory: Lectures in mathematical physics at the university of Texas at Austin**. Springer, 1983.

[KBG+11] Martin Kliesch, Thomas Barthel, Christian Gogolin, Michael J Kastoryano, and Jens Eisert. Dissipative quantum Church-Turing theorem. **Phys. Rev. Lett.**, 107, 2011.

[Kit97] A Yu Kitaev. Quantum computations: algorithms and error correction. **Russian Mathematical Surveys**, 52(6):1191, 1997.

[Kit03] A Yu Kitaev. Fault-tolerant quantum computation by anyons. **Annals of physics**, 303(1):2–30, 2003.

[KLY15] Shelby Kimmel, Guang Hao Low, and Theodore J Yoder. Robust calibration of a universal single-qubit gate set via robust phase estimation. **Phys. Rev. A**, 92:062315, 2015.

[KOS07] Emanuel Knill, Gerardo Ortiz, and Rolando D Somma. Optimal quantum measurements of expectation values of observables. **Phys. Rev. A**, 75:012328, 2007.

[KSB19] Mária Kieferová, Artur Scherer, and Dominic W Berry. Simulating the dynamics of time-dependent hamiltonians with a truncated dyson series. **Phys. Rev. A**, 99:042314, 2019.

[KT13] Michael J Kastoryano and Kristan Temme. Quantum logarithmic sobolev inequalities and rapid mixing. **J. Math. Phys.**, 54:1–34, 2013.

[Lan61] Rolf Landauer. Irreversibility and heat generation in the computing process. **IBM journal of research and development**, 5:183–191, 1961.

[LC17a] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by uniform spectral amplification. **arXiv:1707.05391**, 2017.

[LC17b] Guang Hao Low and Isaac L. Chuang. Optimal hamiltonian simulation by quantum signal processing. **Phys. Rev. Lett.**, 118:010501, 2017.

[Lid19] Daniel A Lidar. Lecture notes on the theory of open quantum systems. **arXiv preprint arXiv:1902.00967**, 2019.

[Lin76] Goran Lindblad. On the generators of quantum dynamical semigroups. **Commun. Math. Phys.**, 48:119–130, 1976.

[Lin25] Lin Lin. Mathematical and numerical analysis of quantum signal processing. **arXiv preprint arXiv:2510.00443**, 2025.

[Liu01] Jun S Liu. **Monte Carlo strategies in scientific computing**, volume 10. Springer, 2001.

[LKW19] Guang Hao Low, Vadym Kliuchnikov, and Nathan Wiebe. Well-conditioned multiproduct hamiltonian simulation. **arXiv preprint arXiv:1907.11679**, 2019.

[LLKH22] Tsung-Cheng Lu, Leonardo A Lessa, Isaac H Kim, and Timothy H Hsieh. Measurement as a shortcut to long-range entangled quantum matter. **PRX Quantum**, 3:040337, 2022.

[Llo96] Seth Lloyd. Universal quantum simulators. **Science**, pages 1073–1078, 1996.

[Low19] Guang Hao Low. Hamiltonian simulation with nearly optimal dependence on spectral norm. In **Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing**, pages 491–502, 2019.

[LP17] David A Levin and Yuval Peres. **Markov chains and mixing times**, volume 107. American Mathematical Soc., 2017.

[LS24] Guang Hao Low and Yuan Su. Quantum eigenvalue processing. In **2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)**, pages 1051–1062. IEEE, 2024.

[LW19] G. H. Low and N. Wiebe. Hamiltonian simulation in the interaction picture. **arXiv:1805.00675**, 2019.

[LW23] 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, 2023.

[LYC16] Guang Hao Low, Theodore J Yoder, and Isaac L Chuang. Methodology of resonant equiangular composite quantum gates. **Physical Review X**, 6(4):041067, 2016.

[Man80] Yu I Manin. Vychislimoe i nevychislimoe (computable and noncomputable), moscow: Sov, 1980.

[Man20] Daniel Manzano. A short introduction to the lindblad master equation. **AIP Advances**, 10, 2020.

[McL95] Robert I McLachlan. On the numerical integration of ordinary differential equations by symmetric composition methods. **SIAM J. Sci. Comput.**, 16:151–168, 1995.

[MRTC21] John M Martyn, Zane M Rossi, Andrew K Tan, and Isaac L Chuang. Grand unification of quantum algorithms. **PRX Quantum**, 2:040203, 2021.

[MW24] Danial Motlagh and Nathan Wiebe. Generalized quantum signal processing. **PRX Quantum**, 5:020368, 2024.

[NC00] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2000.

[NSYL25] Hongkang Ni, Rahul Sarkar, Lexing Ying, and Lin Lin. Inverse nonlinear fast fourier transform on su (2) with applications to quantum signal processing. **arXiv preprint arXiv:2505.12615**, 2025.

[NWZ09] Daniel Nagaj, Pawel Wocjan, and Yong Zhang. Fast amplification of QMA. **Quantum Inf. Comput.**, 9:1053–1068, 2009.

[NY24] Hongkang Ni and Lexing Ying. Fast phase factor finding for quantum signal processing. **arXiv preprint arXiv:2410.06409**, 2024.

[Pat92] Ramamohan Paturi. On the degree of polynomials that approximate symmetric boolean functions (preliminary version). In **Proceedings of the twenty-fourth annual ACM symposium on Theory of computing**, pages 468–474, 1992.

[RCGG20] Sthitadhi Roy, JT Chalker, IV Gornyi, and Yuval Gefen. Measurement induced steering of quantum systems. **Phys. Rev. Research**, 2:033347, 2020.

[RFA24a] Cambyse Rouzé, Daniel Stilck Franca, and Álvaro M. Alhambra. Efficient thermalization and universal quantum computing with quantum Gibbs samplers. **arXiv preprint arXiv:2403.12691**, 2024.

[RFA24b] Cambyse Rouzé, Daniel Stilck França, and Álvaro M Alhambra. Optimal quantum algorithm for Gibbs state preparation. **arXiv:2411.04885**, 2024.

[RP11] Eleanor G Rieffel and Wolfgang H Polak. **Quantum computing: A gentle introduction**. MIT Press, 2011.

[RWS+17] Markus Reiher, Nathan Wiebe, Krysta M Svore, Dave Wecker, and Matthias Troyer. Elucidating reaction mechanisms on quantum computers. **Proc. Nat. Acad. Sci.**, 114:7555–7560, 2017.

[RWW23] Patrick Rall, Chunhao Wang, and Pawel Wocjan. Thermal state preparation via rounding promises. **Quantum**, 7:1132, 2023.

[RWW24] Gumaro Rendon, Jacob Watkins, and Nathan Wiebe. Improved accuracy for trotter simulations using chebyshev interpolation. **Quantum**, 8:1266, 2024.

[SBW+21] Yuan Su, Dominic W Berry, Nathan Wiebe, Nicholas Rubin, and Ryan Babbush. Fault-tolerant quantum simulations of chemistry in first quantization. **PRX Quantum**, 2:040332, 2021.

[SDM+24] Sophia Simon, Matthias Degroote, Nikolaj Moll, Raffaele Santagati, Michael Streif, and Nathan Wiebe. Amplified amplitude estimation: Exploiting prior knowledge to improve estimates of expectation values. **arXiv preprint arXiv:2402.14791**, 2024.

[SHF13] Krysta M Svore, Matthew Hastings, and Michael Freedman. Faster phase estimation. **Quantum Information and Computation**, 14:306–328, 2013.

[Sho94] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In **Proceedings 35th annual symposium on foundations of computer science**, pages 124–134. Ieee, 1994.

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. **SIAM review**, 41:303–332, 1999.

[SM21] Oles Shtanko and Ramis Movassagh. Preparing thermal states on noiseless and noisy programmable quantum processors. **arXiv:2112.14688**, 2021.

[SNK12] Rolando D Somma, Daniel Nagaj, and Mária Kieferová. Quantum speedup by quantum annealing. **Phys. Rev. Lett.**, 109:050501, 2012.

[SS90] G. W. Stewart and Ji-Guang Sun. **Matrix Perturbation Theory**. Academic Press, 1990.

[SS04] Robert Spalek and Mario Szegedy. All quantum adversary methods are equivalent. **arXiv preprint quant-ph/0409116**, 2004.

[ŞS20] Burak Şahinoğlu and Rolando D Somma. Hamiltonian simulation in the low energy subspace. **arXiv:2006.02660**, 2020.

[Sti55] W Forrest Stinespring. Positive functions on c^*-algebras. **Proceedings of the American Mathematical Society**, 6(2):211–216, 1955.

[Suz90] Masuo Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations. **Phys. Lett. A**, 146:319–323, 1990.

[Suz91] Masuo Suzuki. General theory of fractal path integrals with applications to many-body theories and statistical physics. **J. Math. Phys.**, 32:400–407, 1991.

[Sze04] Mario Szegedy. Quantum speed-up of markov chain based algorithms. In **45th Annual IEEE symposium on foundations of computer science**, pages 32–41, 2004.

[TB97] Lloyd N. Trefethen and David Bau. **Numerical Linear Algebra**. SIAM, 1997.

[Tha08] Mechthild Thalhammer. High-order exponential operator splitting methods for time-dependent Schrödinger equations. **SIAM J. Numer. Anal.**, 46:2022–2038, 2008.

[TKR+10] Kristan Temme, Michael James Kastoryano, Mary Beth Ruskai, Michael Marc Wolf, and Frank Verstraete. The χ^2-divergence and mixing times of quantum markov processes. **J. Math. Phys.**, 51, 2010.

[TM71] Myron Tribus and Edward C McIrvine. Energy and information. **Scientific American**, 225:179–190, 1971.

[TOV+11] Kristan Temme, Tobias J. Osborne, Karl G. Vollbrecht, David Poulin, and Frank Verstraete. Quantum Metropolis sampling. **Nature**, 471:87–90, 2011.

[TT24] Ewin Tang and Kevin Tian. A CS guide to the quantum singular value transformation. In **2024 Symposium on Simplicity in Algorithms (SOSA)**, pages 121–143, 2024.

[Tur36] Alan Turing. On computable numbers, with an application to the entscheidungsproblem. **J. Math**, 58:5, 1936.

[Uhl76] Armin Uhlmann. The transition probability in the state space of a c^* algebra. **Reports on Mathematical Physics**, 9(2):273–279, 1976.

[VN93] John Von Neumann. First draft of a report on the edvac. **IEEE Ann. Hist. Comput.**, 15:27–75, 1993.

[VWC09] Frank Verstraete, Michael M. Wolf, and I. Cirac. Quantum computation and quantum-state engineering driven by dissipation. **Nat. Phys.**, 5:633–636, 2009.

[Wat18] John Watrous. **The theory of quantum information**. Cambridge Univ. Pr., 2018.

[WBAG11] James D Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. Simulation of electronic structure hamiltonians using quantum computers. **Mol. Phys.**, 109:735–750, 2011.

[WBHS10] N. Wiebe, D. Berry, P. Høyer, and B. C. Sanders. Higher order decompositions of ordered operator exponentials. **J. Phys. A**, 43:065203, 2010.

[WDL22] Jiasu Wang, Yulong Dong, and Lin Lin. On the energy landscape of symmetric quantum signal processing. **Quantum**, 6:850, 2022.

[WG16] Nathan Wiebe and Chris Granade. Efficient bayesian phase estimation. **Phys. Rev. Lett.**, 117:010503, 2016.

[WK97] Howard M Wiseman and Rowan B Killip. Adaptive single-shot phase measurements: A semiclassical approach. **Phys. Rev. A**, 56:944, 1997.

[WK98] Howard M Wiseman and Rowan B Killip. Adaptive single-shot phase measurements: The full quantum theory. **Phys. Rev. A**, 57:2169, 1998.

[WKAG09] Nicholas J Ward, Ivan Kassal, and Alán Aspuru-Guzik. Preparation of many-body states for quantum simulation. **J. Chem. Phys.**, 130, 2009.

[WSR+23] Yunzhao Wang, Kyrylo Snizhko, Alessandro Romito, Yuval Gefen, and Kater Murch. Dissipative preparation and stabilization of many-body quantum states in a superconducting qutrit array. **Phys. Rev. A**, 108:013712, 2023.

[Yin22] Lexing Ying. Stable factorization for phase factors of quantum signal processing. **Quantum**, 6:842, 2022.

[Yos90] Haruo Yoshida. Construction of higher order symplectic integrators. **Phys. Lett. A**, 150:262–268, 1990.

[ZCL21] Leo Zhou, Soonwon Choi, and Mikhail D Lukin. Symmetry-protected dissipative preparation of matrix product states. **Phys. Rev. A**, 104:032418, 2021.

[ZDH+26] Yongtao Zhan, Zhiyan Ding, Jakob Huhn, Johnnie Gray, John Preskill, Garnet Kin-Lic Chan, and Lin Lin. Rapid quantum ground state preparation via dissipative dynamics. **Phys. Rev. X**, 16:011004, 2026.

[ZPDK10] Marcin Zwierz, Carlos A Pérez-Delgado, and Pieter Kok. General optimality of the heisenberg limit for quantum metrology. **Phys. Rev. Lett.**, 105:180402, 2010.

Index