

NEW QUANTUM ALGORITHMS AND ANALYSES FOR HAMILTONIAN SIMULATION

By

Jacob Watkins

A DISSERTATION

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

Physics—Doctor of Philosophy

2024

ABSTRACT

Digital quantum Hamiltonian simulation is, by now, a relatively mature field of study; however, new investigations are justified by the importance of quantum simulation for scientific and societal applications. In this dissertation, we discuss several advances in circuit-based Hamiltonian simulation.

First, following two introductory chapters, we consider the mitigation of Trotter errors using Chebyshev interpolation, a standard yet powerful function approximation technique. Implications for estimating time-evolved expectation values are discussed, and a rigorous analysis of errors and complexity show near optimal estimation of dynamical expectation values using only Trotter and constant overhead. We supplement our theoretical findings with numerical demonstrations on a 1D random Heisenberg model.

Next, we introduce a computational reduction from time dependent to time independent Hamiltonian simulation based on the standard (t, t') technique. Our approach achieves two advances. First, we provide an algorithm for simulating time dependent Hamiltonians using qubitization, an optimal algorithm that cannot handle time-ordering directly. Second, we provide an algorithm for time dependent simulation using a natural generalization of multiproduct formulas, achieving higher accuracies than product formulas while retaining commutator scaling. Rigorous performance analyses are performed for both algorithms, and simple numerics demonstrate the effectiveness of the multiproduct formulas procedure at reducing Trotter error.

Finally, we consider several practical methods for near-term quantum simulation. First, we consider the analog quantum simulation of bound systems with discrete scale invariance using trapped-ion systems, with applications to Efimov physics. Next, we discuss the Projected Cooling Algorithm, a method for preparing bound states of non-relativistic quantum systems with localized interactions based on the dispersion of unbound states. Lastly, we discuss the Rodeo Algorithm, a probabilistic, iterative, phase-estimation-like protocol which is resource-frugal and effective at measuring and preparing eigenstates. Concluding remarks and possible future directions of research are given in a brief final chapter.

Dedicated to my late grandmother, Elizabeth,
who swore to be the first to call me "Dr. Watkins."

(I told her my committee would beat her to it.
She insisted otherwise.)

ACKNOWLEDGEMENTS

This work is the culmination of six years of graduate study in physics and quantum computing, which itself builds upon a long journey of educational and personal development. I could not have done this without the many people and institutions that have supported me along the way.

First, I thank the Earth for supporting lifeforms who do silly things like study quantum computing. May these pages not be a waste of your resources.

I thank my family, who I've missed dearly, for their encouragement and support. In particular: my mother Diana, for setting an example of good character, for being tender, and for knowing how to handle a son who was hard on himself; my grandfather Rob, for contributing to my love for the outdoors and showing me how to cut the Thanksgiving turkey; my Aunt Dawn, for being the only one who enjoys board games and karaoke as much as I do; my Uncle Gene, for teaching me to like the taste of oyster and pause to appreciate the flowers; and my cousins Ethel and Elizabeth, for being like siblings to me.

Friends, past and present, give warmth and meaning to my life. I thank my "camp friends" (you know who you are), and Camp Orkila itself for letting me sing silly songs and instilling values I retain today. I thank friends I made at the University of Washington, especially Brian, Frances, Katrina, and Mark. My visits to you, and yours to me, have been a source of strength during my PhD. Thanks to Keigan and Elliott for uplifting phone calls and for bringing Rosemary into this world. To Kyndra and Josh; I will always appreciate our adventures into the woods.

A long list of mentors and educators have guided me since deep into my childhood. I thank my Taekwondo instructor Eric Shields for teaching me about perseverance and indomitable spirit. I thank the wonderful educators of Shaw, Kalles, and Puyallup High who have given me a well-rounded liberal arts education. Particular thanks go to Ms. Kreiger, for calling my mother to tell her I did well on an assignment, and to Ms. Kooser, for an engaging, hands-on social studies curriculum. Thanks to Mr. Ryan, for facilitating most of my social life via the PHS band program. Of course, a heartfelt thanks to Mr. Segers, my high school physics teacher, for tolerating my (frequent) class-entry skits and my (occasional) anxieties about grades.

More recently, thanks to Ben Hall, my first office mate and friend in graduate school, for showing me how cool quantum computing is; Maria Violaris for singing Taylor Swift parodies with me at the premier quantum information conference and showing me that "punting" meant more than kicking an American football. Thanks to Kirtimaan Mohan, Katie Hinko, Nick Ivanov, Rachel Barnard, Matt Oney and others who have helped me explore my interests in education while being supportive mentors and collaborators.

Despite being housed in a major nuclear science lab for most of my studies, the reader may notice that my research has strayed quite far from nuclear physics proper. I thank Dean Lee for giving me ample freedom and encouragement to explore my interests. Thanks as well to Nathan Wiebe for hosting me for a summer in Toronto, and for mentoring me in the kind of rigorous quantum algorithms research exhibited in much of this thesis. I also thank Ale Roggero for being a supportive and patient mentor and collaborator, and for being there for a young graduate student struggling through his first real research talk.

Finally, I thank my partner, Brita, for her excellent pies and shakshuka, introducing me to Margaret Atwood, and converting me to coffee. We have supported each other through the difficulties of the pandemic and come out stronger people. I love you like a rock.

TABLE OF CONTENTS

CHAPTER 1	HOUSEKEEPING	1
CHAPTER 2	BACKGROUND	3
	2.1 Quantum Mechanics and Challenges to Realism	3
	2.2 Computer Science and the Role of Physics	6
	2.3 Quantum Computing, an Overview	8
	2.4 The Case for Quantum-Based Quantum Simulation	12
	2.5 Quantum Hamiltonians: A Closer Look	16
	2.6 Classical Simulation Algorithms	19
	2.7 Quantum Simulation Algorithms	20
	2.8 Mathematical Reference	31
CHAPTER 3	TROTTER ERROR MITIGATION	36
	3.1 Introduction and Motivation	36
	3.2 Polynomial Interpolation	38
	3.3 Stability Analysis	41
	3.4 The Effective Hamiltonian	47
	3.5 Application to Dynamical Observables	55
	3.6 Numerical Demonstration	59
	3.7 Discussion	63
	3.8 Proofs	65
CHAPTER 4	TIME DEPENDENT HAMILTONIAN SIMULATION THROUGH DISCRETE CLOCK CONSTRUCTIONS	71
	4.1 Introduction and Motivation	71
	4.2 The Clock Space	74
	4.3 Finite Clock Spaces	81
	4.4 Time Dependent Qubitization	93
	4.5 Discussion	102
CHAPTER 5	MULTIPRODUCT FORMULAS FOR TIME DEPENDENT SIMULATION	103
	5.1 Introduction and Background	104
	5.2 Definition and Effectiveness	106
	5.3 Time Dependent Multiproduct Simulation	111
	5.4 Error Analysis	112
	5.5 Time Step Analysis	125
	5.6 Query Complexity	129
	5.7 Numerical Demonstrations	134
	5.8 Discussion	143
	5.9 Algorithm for Time Mesh	143
CHAPTER 6	A SIMULATION MIXED BAG	146
	6.1 Discrete Scale Invariance on Trapped-Ion Systems	146

6.2	Projected Cooling Algorithm	153
6.3	Rodeo Algorithm	159
CHAPTER 7	CONCLUSION AND OUTLOOK	173
	BIBLIOGRAPHY	175

CHAPTER 1

HOUSEKEEPING

This dissertation is concerned primarily with the task of simulating a Hamiltonian on a quantum computer. What is a Hamiltonian? Why do we want to simulate it? What is a quantum computer?

Those already in-the-know, who want to skip to the technical advances of this thesis, are welcome to survey the chapters to find what they are looking for. I expect they are already inclined to do so! Those who want a little more context for this thesis, and to learn some of the big ideas underlying the field, are encouraged to continue on to the next chapter. There I will provide a sweeping, but necessarily brief, survey of the big ideas in the field of digital quantum Hamiltonian simulation, steadily working towards the more technical advances of this work.

The Abstract of this dissertation provides, as expected, a synopsis of all topics covered. All chapters are essentially independent of each other, with the partial exception of Chapter 5, which relies some ideas from Chapter 4 to support a conjecture. Moreover, all but Chapter 6 correspond to a single, self-contained research project. The penultimate Chapter 6 covers a mixed bag of projects to which I contributed.

All of the projects discussed in this thesis have corresponding publications or preprints, and references to these works are provided near the beginning of the relevant chapter or section. When deciding what to include here, and what to leave to those works, I use several criteria. First, since this thesis serves as a compendium of my work, I focused chiefly on my contributions to a given project. Sometimes though, for the sake of a self-contained document, I include results and derivations that are primarily due to my collaborators. I will indicate clearly when this is the case. Finally, for various reasons, there are results from these projects that did not make their way into the corresponding publication, especially in the miscellaneous Chapter 6. By including those here, I hope to complement the publications by supporting and extending their findings through numerics or derivations.

The flavor of this thesis is analytical, in at least two senses. First, in the mathematical sense. To approach a problem "analytically" means to utilize tools of mathematical proof and derivation, in

contrast to numerical calculation. The central results are proofs and analytical bounds on error and computational complexity. Numerics, however, are used to provide assurance and to see the "actual" performance in a way that complexities cannot showcase. These benchmarks are usually far from complete, suggesting an obvious path for additional research. Second, this thesis is analytical in that it is primarily concerned with analyzing something, namely quantum algorithms. Although we propose novel simulation methods, they are typically variations on existing tools. The performance and errors analyses are likely the major technical advancement of this dissertation. I believe that such careful analyses provide firm guideposts for those who wish to apply algorithms to specific use cases. Hopefully, our methods for estimating algorithmic resources can be useful for the analysis of quantum algorithms developed in the future.

Often in practice, analytical error bounds fall short of representing the typical error of a simulation method [25, 66]. This is mostly good news, meaning performance is often much better than expected. What, then, is the value of such bounds if they fail to capture the "actual" behavior of the method? Worst-case error and resource bounds represent a first important step towards understanding the behavior and capability of a method, providing us the most robust guarantees of how well an algorithm will perform. This is only part of the picture, and while numerical experiments can provide more insight, there is additional work for theorists as well. For example, recent work on average-case hardness for Trotter simulations likely represents a step towards a fuller understanding of the "typical" hardness [23].

Without further ado, please enjoy what this dissertation has to offer. I hope you find these chapters helpful not only for technical content, but for inspiration and ideas for your own pursuits.

CHAPTER 2

BACKGROUND

I expect there will be as many readers of this thesis who are newcomers to quantum computing as those who are quantum algorithms experts. Thus, I am motivated to dedicate a chapter to provide both background and inspiration for the technical results that follow. We will start with broad scope and little detail, gradually narrowing our focus to the new stuff. More detailed background on specific research is given at the beginning of each chapter.

Quantum information science, which includes quantum computing, is a relatively young discipline which overlaps several technical fields, particularly physics and computer science. My approach to this chapter is to discuss each of these domains separately, then their surprising interconnection made most obvious by (but not reliant on) quantum computing. With these ingredients in place, we then introduce Hamiltonian simulation, which relates to the computation of either closed, naturally occurring quantum systems or problems with equivalent mathematical structure. I hope the reader finds these short surveys valuable to understand the more particular and technical work of later chapters.

2.1 Quantum Mechanics and Challenges to Realism

The developments in physics which began at the turn of the 20th century, were, in many respects, parallel with those found in the arts in that same period. As the modernists eschewed accurate portrayals for abstract figures and geometries, the physicists grappled with phenomena increasingly removed from regular experience. And, like the modernists, these new ways of doing physics were met with some backlash. Despite this, the resulting theories, namely relativity and quantum mechanics, were better at explaining the world around them than the earlier "classical" physics. Yet their character was so strange that it led some physicists, notably Dirac¹, to emphasize mathematics over the senses in formulating physical theories.

This is especially true for quantum mechanics, which on the surface said a number of very

¹"I learned to distrust all physical concepts as the basis for a theory. Instead one should put one's trust in a mathematical scheme, even if the scheme does not appear at first sight to be connected with physics." [92]

strange things. Particles were neither here nor there until measured, the story goes, seemingly defying the scientific tenet of realism. Particles may be waves, waves may be particles. More truthful than such common refrains is that quantum mechanics provides a relatively well-defined framework for accurate calculations of particles, atoms, molecules, and nuclei, even as the theory appeared unintuitive or even nonsensical. The mathematics of quantum mechanics nicely captured a variety of phenomena which eluded classical treatment, but the classical theory made more sense. Yet even today, the meaning of quantum mechanics remains mainly unresolved. In response to this absurdity, the prevailing attitude amongst quantum physics practitioners is captured in the pithy mandate: "Shut up and calculate." The meaning: don't worry about what the theory *means*, per se, just worry about what it predicts. While it's easy to criticize this point of view, delaying thorny questions of interpretation arguably allowed for more rapid understanding of physical phenomena in the decades following the invention of quantum mechanics.

One unsettling aspect of quantum mechanics is its intrinsic nondeterminism. The theory only predicts *probabilities* of certain outcomes in a physical experiment, where "experiment" is interpreted broadly as any means by which observers (such as people) experience the world around them. While probabilities had appeared earlier in statistical mechanics and its connection to thermodynamic entropy, their appearance here in a *fundamental* physical theory was a notable break from the past, and carried unsettling philosophical implications. It led Einstein, Podolsky, and Rosen to argue that quantum mechanics was actually an incomplete theory of reality [42], and a more complete understanding, even if impossible to achieve by mere mortals, would reveal an underlying determinism. Some alternate theories, most notably Bohmian mechanics [55], purport to restore determinism to quantum by relegating all chance to inaccessible knowledge of the particle trajectories. It was shown in a groundbreaking work by John Bell that certain reasonable "hidden variable theories" make predictions distinct from those of quantum mechanics [12]. Experiments on the matter came out in favor of quantum theory [7, 49], and these contributions were rewarded with the 2022 Nobel Prize in Physics. The bottom line is that it is hard to restore determinism, or even conventional probability, to quantum mechanics without violating other cherished physical

principles such as locality. In contrast to hidden-variable theories, which seem to deny quantum mechanics as it is, the many worlds interpretation [38] asks us to take quantum theory at face value, including the reality of the quantum wavefunction as a description of all phenomena.

So far, I have emphasized that quantum physics is a radical departure, conceptually, from classical physics. Yet this belies the fact that, in formulating quantum mechanical models, classical models are often used as a starting points. It is easiest to start with existing tools when trying to create new ones. The process of taking a classical theory and tweaking it to describe a quantum system is known as "quantization." It turns out that not all ideas from classical physics are equally suitable for quantization. For example, the Newtonian framework, in which changes in motion are generated by forces, does not have a great correspondence to quantum mechanics principles. Rather, the most natural jumping off point for quantum mechanics is the *Hamiltonian* formulation, named after Irish mathematician, astronomer, and physicist William Rowan Hamilton. In this framework, a classical system has physical configurations given by a number of coordinates q . For example, the location and orientation of an airplane may be exactly represented by a set of 6 numbers. Each coordinate q has a corresponding conjugate momentum p , which in simple cases may be seen as expressing a "velocity" for q . Specifying all coordinates q and momenta p gives a complete specification of the system in the sense that any "observable quantity" O is a function $O(q, p)$ of the coordinates and momenta.

One uniquely special observable is the Hamiltonian $H(q, p)$, which provides the total energy of the system as a function of its coordinates and momenta. It also contains all information about future states of the physical system. That is, the classical Hamiltonian defines a set of differential equations

$$\frac{dq}{dt} = \frac{\partial H}{\partial p}, \quad \frac{dp}{dt} = -\frac{\partial H}{\partial q} \quad (2.1)$$

which, when solved, provide the state of the system at any subsequent time. More succinctly, H encodes the *dynamics* of the physical system in question. The importance to physics is immediate. One of the primary goals of physics is to understand a phenomena well enough to make future predictions given current data. Prediction is more powerful, and impressive, than retroactive

explanation. The Hamiltonian provides all the information needed to make these predictions.

In an exactly analogous manner, the quantum Hamiltonian H (we use the same symbol) encodes all of the dynamics of closed quantum system. Given a system described by wavefunction $|\psi\rangle$ the dynamics are found by solving the famous Schrödinger equation.

$$i\partial_t |\psi_t\rangle = H |\psi_t\rangle \quad (2.2)$$

Formally, H in the quantum setting is a Hermitian operator on a Hilbert space, and $|\psi_t\rangle$ is a vector-valued function on this space. The main point is that solving (2.2) is a fundamentally important task for understanding the dynamics of physical phenomena. Solving this equation allows for understanding the formation of the elements, the properties of molecules and materials, and the fundamental constituents of nature.

In practice, solving (2.2) is far too difficult with even today's best computational devices and cleverest tricks. Instead, people come up with a number of clever partial solutions and approximation schemes, to various degrees of success. In talking about what can be computed efficiently and what cannot, however, we have already begun to enter a different domain worth discussion: computer science.

2.2 Computer Science and the Role of Physics

In Scott Aaronson's *Quantum Computing Since Democritus*, he writes that computer science "is a bit of a misnomer." Rather than being about computers, in the particular sense of desktops, servers, and smart phones, he views it as "the study of the capacity of finite being such as us to learn mathematical truths." [2] Aaronson understands that "mathematical truths" encompasses more than what is sought by professional mathematicians. It could involve finding the shortest route to work (Dijkstra's Algorithm), or predicting protein structure from an amino acid sequence [74]. Such tasks, in light of modern science, are likely viewed as being intrinsically mathematical and thus within the domain of computation. However, with recent developments in artificial intelligence, particularly Large Language Models (e.g., chatGPT), even more domains of human activity have been made amenable to computational treatment.

The diversity of computational problems is paralleled by the diversity of entities which can serve as a computational medium. Indeed, prior to the development of general-purpose, digital computers in the mid-20th century (and even following that), the term referred to human computers who performed much of the calculations for scientific, industrial, and governmental applications (see, e.g., [117]). While standard, modern computers are electronic, computers could in principle be made from billiard balls [48] or even water [3]. In some sense, however, all of these computers are equivalent to a Turing machine, an idealized computer consisting of a tape for manipulating symbols, an input program, and several internal states. One way to express this equivalence is that a Turing machine may be used to simulate the calculations performed by any of these other models.

The hypothesis that "computable by Turing machine" captures the notion of what is computable is referred to as the Church-Turing thesis [32]. No serious challenge to this thesis has been sustained in the near-century since it was proposed. If the thesis holds, it seems to suggest that details about the physical system performing the computation may be abstracted away. If so, one shouldn't expect questions in physics to have much bearing on computer science, besides the practicalities of engineering an effective computational device.

Although understanding computability, i.e. what problems may be solved by computation, is important, we are left without an understanding of which problems are "practically" solvable on real world computers. For example, let's return to the Schrödinger equation (2.2). This equation can be solved straightforwardly, to arbitrary accuracy, given enough time, space, and energy. However, the amount of these resources needed is prohibitively large for interesting instances. No one wants to wait 1000 years for a single result. Questions of what may be computed *efficiently* falls under the purview of the subfield of *computational complexity*. Nowadays, most aspects of theoretical computer science are concerned with complexity, and computability is a relatively closed subject.

If we are actually interested in computational complexity, rather than computability, it makes sense to consider a modified Church-Turing thesis which deals more with the former than the latter. In particular, we may ask: Is "efficiently computable by Turing machine" equivalent to "efficiently computable by actual, physical computers"? The so-called *strong* Church-Turing thesis is the

assertion that this is the case, and because of the new word "efficiently" in the above question, the claim is indeed stronger than what either Church or Turing originally proposed. Though the claim is stronger, the evidence for its truth is correspondingly weaker. In fact, existing evidence suggests that *quantum* computers, if constructed, could solve problems efficiently that Turing machines could not [118, 120]. The potential for quantum computers to solve problems relevant to society has driven major investment from government and industry [53]. From a more fundamental perspective, the power of quantum computing suggests a greater interplay between physics and computer science than has been historically explored. Are there other physically realizable models of computation even more powerful than classical or quantum computers? In this direction, work by Aaronson has shown how computers based on hidden variable theories would be slightly more powerful than standard quantum computers [1]. The upshot of these developments is that physics seems to play an essential role in a fundamental computer science question: what computational tasks may be efficiently performed?

2.3 Quantum Computing, an Overview

In the last section, we rapidly converged on the notion of a quantum computer, and here we discuss in more detail what this means. As we approach our primary topic, Hamiltonian simulation on a quantum computer, I will use increasingly precise and technical language, and no longer avoid mathematics. Readers with background in linear algebra and complex numbers are encouraged to consult standard resources for more thorough introductions to quantum computing [101].

At a high level, a quantum computer is nothing more than a computer based on the laws of quantum mechanics. Any computation requires, abstractly, the encoding of information and its manipulation by certain operations to achieve a result. For example, a classical computer may perform addition by storing two numbers in binary registers, then manipulating these registers in a specified way to get the sum on one of the two registers. A quantum computer, by contrast, stores its information as quantum states, and manipulates these states. As a caution, although any laptop is describable, in principle, in quantum mechanical terms, the way they store information and perform operations is most aptly described as "classical." This is true of essentially any computational device,

save the handful of quantum computers under development today.

How do we model the workings of a quantum computer concretely? Just like with classical computers, many possible computational models exist. Quantum Turing machines [37], measurement-based [111], and adiabatic quantum computation [4] are several well-explored examples. But the most popular approach by far is the circuit-based model of quantum computation, which we will now explain in detail. The reader will benefit in having some background in the classical circuit model of computation, or experience with real digital logic circuits.

Figure 2.1 provides an example of a quantum circuit. As with the classical circuit model, quantum circuits have wires and gates that feed forward (no feedback loops), but here the wires contain quantum information. Instead of well-defined bits in the 0 or 1 state, wires carry quantum bits, or *qubits*. In isolation, a qubit can have a state in the form

$$c_0|0\rangle + c_1|1\rangle. \tag{2.3}$$

Here, the symbols $|0\rangle$ and $|1\rangle$ are "kets" which take on a precise meaning as orthonormal vectors in a two-dimensional complex inner product space, but can be thought of informally as the "definite" states that the qubit can take. The coefficients c_i are complex numbers that are often called "amplitudes." The fact that c_i does not have to be a positive real number is the crucial difference between quantum computing and probabilistic, a.k.a. Monte Carlo, computing. The probability of measuring 0 or 1 is given by $|c_0|^2$ and $|c_1|^2$, respectively, but before a measurement is performed the amplitudes can exhibit *interference*. In order to have total probability one, we must have the following *normalization condition*.

$$|c_0|^2 + |c_1|^2 = 1 \tag{2.4}$$

For vectors $|\psi\rangle$ and $|\phi\rangle$, their inner product is denoted $\langle\phi|\psi\rangle$. In this language, the normalization condition for a qubit in state $|\psi\rangle$ can be written as $\langle\psi|\psi\rangle = 1$. In this thesis, the term *state vector* will mean the normalized vector used to mathematically represent the state of our quantum system. Many readers will be familiar with the more general *density matrix* representation of quantum states, but because of our focus on closed-system dynamics we will not have much need for this more complicated formalism.

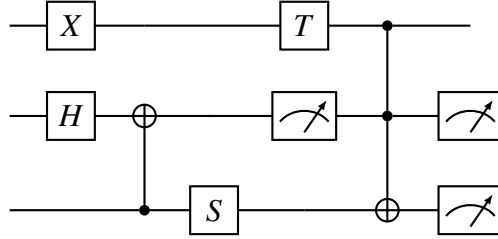


Figure 2.1 Example of a quantum circuit on three qubits. Information is carried on the wires as a collective quantum state, which is a superposition of possible values of the bitstrings on the register. This information is manipulated by gates which act on one or more of the qubits. Partial information about the quantum state is obtained by measurements, here represented by meters. The measurements also affect the state in accordance with quantum mechanics.

For any interesting computation, combining multiple qubits together will be necessary. Such a collection will be referred to as a quantum register. As with combining any two (distinguishable) quantum mechanical systems, joint qubit states are described formally through the *tensor product* of the individual state spaces. Any state vector $|\Psi\rangle$ on the joint system is a linear combination of *product vectors* of the form

$$|\phi\rangle \otimes |\chi\rangle \tag{2.5}$$

where $|\phi\rangle$ and $|\chi\rangle$ are state vectors on each individual space. As an important example, a state vector on a collection of n qubits may be generally expressed as a sum

$$|\psi\rangle = \sum_{b \in \{0,1\}^n} c_b |b\rangle \tag{2.6}$$

over all bitstrings $b = b_1 b_2 \dots b_n$. There is an underlying tensor product $|b\rangle \equiv |b_1\rangle \otimes |b_2\rangle \otimes \dots \otimes |b_n\rangle \equiv |b_1 b_2 \dots b_n\rangle$ that is often convenient to leave implicit. Whenever the joint state $|\Psi\rangle$ cannot be written as a product vector, it is said to be *entangled*. Generically speaking, almost all quantum states are entangled, in the sense that choosing a random state has vanishingly small probability of being a product state for systems with more than a small number of states [144]. Entanglement is a necessary condition for quantum computers to exhibit superior performance over classical computers, though identifying the source of "power" quantum computation is a somewhat subtle issue [69].

To summarize, the objects of our quantum computer are qubits, and their collective state is

given by quantum state vector, i.e. a normalized vector in the tensor product over each qubit vector space. More concisely, this is just a normalized vector in \mathbb{C}^{2^n} for n qubits. Measuring the qubits returns a bitstring with probability given by the squared amplitude.

We must now establish the appropriate operations for our quantum computer. Naturally, our operations should not take us outside the set of allowed states, namely the quantum state vectors described above. Moreover, empirical evidence suggests that quantum mechanical operations are all *linear*, so that operations on the full state may be understood by considering the operations on each component of the superposition. We are therefore led to consider gates as *unitary* operations acting on a subset of qubits. Unitary operations can be defined in a number of equivalent ways, all of which relate to the idea of preserving the norm of the state vector. More precisely, a unitary U is a linear operator on an inner product space such that

$$\|U|v\rangle\| = \||v\rangle\| \tag{2.7}$$

for any vector v in the space, where $\|\cdot\|$ represents the Euclidean, or L^2 , norm. A quantum gate, or simply "gate" in this context, is a unitary operation acting on a small number of qubits in a circuit. We generally represent gates as boxes with wires passing through.

Today, computer programmers rarely work at the level of digital logic and gates on their laptop. Instead, programmers work at higher levels of abstraction to skirt minute details, accomplishing more as a result. In this thesis, we will seek to analyze quantum algorithms, and it will be easier and more insightful if we consider larger chunks, or subroutines, used to carry out the method. In the analysis, we will call these subroutines *oracles*: "black boxes" that are used as part of the algorithm, whose inner workings we either don't know or delay considering. For the quantum context, the oracle O will be a unitary operation on some number of qubits. As an example of an oracle, consider a function f from m -bit to n -bit strings. From this classical operation, we can define a unitary oracle U_f that computes f . Given two quantum registers of length n and m in state $|x\rangle \otimes |a\rangle \equiv |x\rangle |a\rangle$, define U_f as

$$U_f |x\rangle |a\rangle = |x\rangle |f(x) \oplus a\rangle, \tag{2.8}$$

where \oplus is addition modulo 2, and extend the definition to general vectors by linearity. It can be checked that U_f is unitary, and when $a = 0$ it is clear that U_f computes $f(x)$. Such oracles are used, for example, in Shor's order-finding algorithm, where $f(x) = a \cdot x$ is multiplication by some integer a . In the context of this dissertation, we will define oracles that compute the parameters of a Hamiltonian or evolve a quantum register according to some Hamiltonian. Oracles may also encode specific observables to be measured during a quantum algorithm. Without knowing the computational cost of implementing an oracle O , it is impossible to know the cost of any algorithm utilizing O as a subroutine. One of the benefits of using oracles is to break down the problem into smaller pieces: first implementing the oracle, then implementing the algorithm given the oracle. Another benefit is abstraction: we can analyze the general oracle problem, then apply those results to any particular instance of that oracle. In the example above, this might entail different functions f , some of which are easy to compute, others of which are uncomputable!

2.4 The Case for Quantum-Based Quantum Simulation

The advent of quantum mechanics carried great promise to better understand a range of physical phenomena, but challenges remained that were more about computational feasibility than theoretical understanding. As expressed by Paul Dirac [39] regarding the invention of quantum mechanics,

The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble.

Though there appears "only" one barrier to solving all of chemistry, it is indeed a very large one. Massive supercomputing resources are needed to solve, starting from quantum mechanics, even relatively small molecules. As a result, heuristic and phenomenological techniques are employed in practice, which are less expensive but also less reliable and general. Examples of computational techniques for quantum many-body problems include a slew of variational techniques, such as Hartree-Fock and Coupled Cluster, and dynamical methods such as Molecular Dynamics [33].

Despite these clever and well-established approaches, we emphasize that there remains a lack of general, efficient methods for classical simulation of quantum dynamics.

Although efficient quantum simulation would likely not obviate the need for high-level, domain-specific concepts in chemistry, materials science, etc., such simulations would nevertheless be valuable to science and human knowledge. Unfortunately, it is believed that classical computers could never simulate quantum physics efficiently, meaning they couldn't solve the Schrödinger equation (2.2) in interesting instances without intractable amounts of resources. After a hundred years of effort, effective classical methods for computing general quantum properties have simply not been found, although discovering such methods would be both useful and philosophically profound.

Some of the earliest explorations into quantum computing were motivated by the desire for efficient quantum simulation [46, 37]. As it happens, not only *can* quantum computers simulate physical Hamiltonians efficiently, as we shall see in Section 2.7, but also this task fully captures the power of quantum computing. More precisely, it turns out that k -local Hamiltonian simulation is BQP-complete. Here k -local refers to Hamiltonians of the form

$$H = \sum_{\gamma=1}^{\Gamma} H_{\gamma} \tag{2.9}$$

where each H_{γ} only operates on at most k qubits. In words, interactions across the system are built up from interactions involving only a small number of constituents. BQP is, loosely speaking, the class of problems which may be efficiently solved by a quantum computer. "BQP-complete" refers to the fact that k -local Hamiltonian simulation is both in BQP (quantum computers can do it) and also that any problem in BQP may be encoded as a k -local simulation problem (in fact, 2-local)². Thus, we may view digital quantum circuits and local Hamiltonian simulations as two sides of the same coin. This may not come as a surprise to many physicists, who are used to viewing all quantum operations as arising from an underlying Hamiltonian. The importance is that, if quantum computers can do anything interesting at all compared to classical computers, then Hamiltonian simulation should be one of those things.

²For experts: this encoding may require at most polynomial overhead in spatial and temporal resources.

A heuristic, imperfect argument that quantum computers can simulate quantum systems effectively is that they also are quantum mechanical. By having the computer mimic the quantum transformations occurring in the physical system, one might hope to achieve what the classical computer cannot replicate efficiently. This argument provides some insight but also, by itself, is unconvincing. What is the essential difference between quantum simulation and simply "watching" a quantum system of interest "do its thing"? There are, in fact, major differences:

1. You may not have much access to the system of interest. For example, it is challenging, expensive, or impossible to send a probe to the sun's core to gain information on nuclear processes. A protein under study may behave differently in a test tube than *in situ*. Essentially, there are many phenomena that are hard to measure directly under the desired circumstances.
2. There may be very little control over the *parameters* of the system of interest, or little ability to measure a wide variety of properties. Digital quantum simulation gives an enormous (though not limitless) degree of control over the model parameters and read out of the desired results. Identical trials of the same simulation could, in principle, be prepared as desired. The ability to tweak aspects of the simulation leads to a greater understanding of the phenomenon under study. By contrast, barring a highly tunable experimental setup, a system "just is" and doesn't necessarily provide insight.

In short, simulation is more than just experiment, both classically and quantumly. In Section 2.7, we will see how Hamiltonian simulation algorithms can be quite abstracted from any particular information about the system, thus earning designations such as "computation" and "algorithm".

The classical intractability of quantum simulation is a more subtle issue than is often let on in popular, or even research, presentations, and we take a moment to challenge these oversimplifications. The common argument starts and ends with the exponential growth in the number of elementary states as the quantum system size increases. Recall the expression (2.6) for an arbitrary n -qubit state. Without additional structure, it appears the state $|\psi\rangle$ requires 2^n complex numbers to describe, one for each bitstring b . Each additional qubit doubles the number of states, and 2^{100} is

already an enormous number for just 100 particles. Normalization conditions do not help much, nor does the irrelevance of global phase. This amount of data is inefficient to store in memory, let alone perform operations on.

The problem with the above argument is that *any* physical system, quantum or not, has such a scaling: a system of n particles with d single-particle states has a total of d^n states. Why is there no objections to exponentiality in this context? To explore this, we look at standard probability theory, which is the closest jumping-off point for quantum mechanics. We may express the state $|p\rangle$ of n probabilistic bits³ in the following suggestive form.

$$|p\rangle = \sum_{b \in \{0,1\}^n} p_b |b\rangle \quad (2.10)$$

Here the collection of 2^n p_b 's forms a probability distribution, where p_b is the probability of observing bitstring b . Compare with equation (2.6), and we seem to be facing the same conundrum. And indeed we are, if the goal is to represent the full distribution $|p\rangle$. However, in applications, we would usually rather *sample* from the distribution rather than express it. The difference is the same as flipping a fair coin vs. writing a list $(1/2, 1/2)$. Sampling typically requires much less spatial overhead, and can be performed using probabilistic bits and operations. The situation in quantum computing is very, very similar. We cannot "see" the output state $|\psi\rangle$ in entirety, but measure and achieve some result with some probability. Thus, the source of quantum computing's power appears to come not just from an exponential number of amplitudes, but also the way these amplitudes interfere under unitary operations.

To summarize, although we have a comprehensive theory of quantum phenomena, it is currently very difficult to compute the consequences of this theory. Further, we should not believe generic, efficient classical methods will ever be found. On the other hand, the use of quantum computers for quantum simulation is an intuitive and promising solution to this challenge. We shall see in Section 2.7 that there are several efficient quantum algorithms for Hamiltonian simulation. For the most part, the only remaining obstacle is a very difficult engineering problem: building better

³Note that the notation $|p\rangle$ is not suggestive of anything quantum. I use it here in this context to emphasize that the ket is just notation.

quantum computers⁴. Building effective quantum hardware is, in its own right, a fascinating and difficult problem. At the time of writing, many different platforms for quantum computing are being actively developed or researched [19, 17, 65, 47], and after enormous investments [53] the technology is improving at a steady clip. However, robust error-corrected quantum computing remains out of reach. Nevertheless, this dissertation will, for sake of analysis, typically assume the kind of noiseless (or error corrected) quantum computing capabilities that we hope for in the future. Occasional comments on the likely effects of hardware noise are made, but detailed analysis is left for future work.

2.5 Quantum Hamiltonians: A Closer Look

Having motivated digital quantum Hamiltonian simulation, let's now elaborate on what this entails. First, strictly speaking, Hamiltonians describe *closed* quantum systems. Closed systems are idealizations of any real physical system, where the system of interest is perfectly isolated from any external influence. Though this is a useful concept in all areas of physics, closed systems are in some sense far from typical. Indeed, the science of decoherence suggests that the apparent classicality of our world emerges from open system dynamics [143]. For open systems, a more general framework of quantum states and operations needs to be invoked, but for our purposes the mathematical representations we've discussed thus far (state vectors, unitary operations, etc.) will suffice.

Taking physical time t to be continuous, we imagine the state $|\psi_t\rangle$ at any time t to be related to the state at a previous time $s \leq t$ via some unknown unitary operation, which we denote $U(t, s)$.

$$|\psi_t\rangle = U(t, s) |\psi_s\rangle \quad (2.11)$$

What properties should U possess? Tacit in our notation is the assumption that U does not depend on the states $|\psi_s\rangle$ and $|\psi_t\rangle$, meaning that the dynamical laws themselves do not care about the specific states involved. This is also the situation we find in classical mechanics. An another reasonable property, we might expect various U to chain together naturally when applied in succession.

$$U(t, s) = U(t, r)U(r, s) \quad s \leq r \leq t \quad (2.12)$$

⁴Part of the solution likely entail better protocols for correcting errors.

As a reasonable corollary, we have $U(t, t) = I$. It is sensible to define $U(s, t) \equiv U(t, s)^\dagger$ for $s < t$, and with this the transitive property above generalizes to any $s, r, t \in \mathbb{R}$.

We might reasonably impose some degree of continuity to the wavefunction $|\psi_t\rangle$, hence to the unitary U . Let's assume, at the very least, that $U(t, s)$ is differentiable in t (and by symmetry, s). Taking a derivative of equation (2.11) with respect to t , we obtain

$$\begin{aligned} |\psi'_t\rangle &= U'(t, s) |\psi_s\rangle \\ &= U'(t, s)U(s, t) |\psi_t\rangle. \end{aligned} \tag{2.13}$$

Now we have a differential equation in $|\psi_t\rangle$. For consistency, we must have that $U'(t, s)U(s, t)$ is independent of s (this can be verified explicitly). Moreover, by using some basic properties of U and the product rule, it is easy to check that it is also anti-Hermitian. Thus, if

$$-iH(t) \equiv U'(t, s)U(s, t), \tag{2.14}$$

then H is a Hermitian operator. We call H the *Hamiltonian*, and with this notation we recover the famous Schrödinger equation

$$i\partial_t |\psi_t\rangle = H(t) |\psi_t\rangle. \tag{2.15}$$

More generally, using relation (2.11), we obtain an operator Schrödinger equation

$$i\partial_t U(t, s) = H(t)U(t, s) \tag{2.16}$$

which, in conjunction with (2.11), provides an expression for a quantum state at any time t following an initial time s .

To the practitioner of quantum physics, this might seem like an odd approach to the foundations of quantum dynamics. We started by postulating a unitary evolution operator U with certain reasonable properties, then derived a Hamiltonian which generates U via the Schrödinger equation. In physics, one typically starts with a Hamiltonian, then attempts to solve for U . It is insightful to ponder the reasons for this. As discussed in Section 2.1, the Hamiltonian concept is closely linked to classical mechanics, and hence serves as a more natural starting point for humans to model quantum phenomena. Perhaps more fundamentally, a compact, elementary description of

U instead of H would be like having the cheat code for solving any quantum physics problem. By analogy, while the forces on classical systems may be more or less easy to describe, predicting resulting trajectories is a straightforward but expensive computational task. In short, having U instead of H feels "too good to be true."

The primary problem of quantum simulation, then, is to solve (2.16), or (2.15) more specifically, given a description of $H(t)$. As needed for sensible physics, the solution for $U(t, s)$ exists and is unique with the initial condition $U(s, s) = I$. More interesting is that a succinct description of the solution exists. We may write it as a so-called time ordered operator exponential

$$U(t, s) = \exp_{\mathcal{T}} \left\{ -i \int_s^t H(\tau) d\tau \right\} \quad (2.17)$$

which may be understood in a number of ways. One that is particularly relevant to this thesis is the product integration approach. Given a family of partitions $\{t_j\}_{j=1}^n$ of the interval $[s, t]$, with maximum width δ_n tending to zero as $n \rightarrow \infty$, a solution is given by the product integral [73]

$$U(t, s) = \lim_{n \rightarrow \infty} \prod_{j=1}^{n-1} e^{-iH(t_j)\delta t_j} \quad (2.18)$$

where $\delta t_j = t_{j+1} - t_j$. One feature of this approach is that, for sufficiently large but finite n , the product represents an approximation that is also *unitary*, in contrast with the more common Dyson series representation. Solution (2.18) is closely linked to the idea of product formulas, which will be discussed in subsequent chapters. An even simpler expression for U can be found when H is independent of time. Philosophically, this condition amounts to the notion that the laws of physics should not change over time. In this case, (2.18) simplifies to a simple operator exponential

$$U(t, s) = e^{-iH\Delta t} \quad (2.19)$$

where $\Delta t = t - s$. This expression can be understood through a power series expansion or the spectral theorem for normal operators. It is remarkable that such a simple and succinct expression can be written for the solution to essentially all closed quantum dynamics. If H is over a finite-dimensional space, i.e., a matrix, computing a partial sum

$$U(t, s) \approx \sum_{j=0}^N \frac{(-iH\Delta t)^j}{j!} \quad (2.20)$$

for a sufficiently large N will yield an arbitrarily accurate approximation of $U(t, s)$. Hence, given the matrix H , the computation of U reduces to "just" matrix multiplication and addition. Similarly, equation (2.18) can be approximated by taking n sufficiently large (but finite) and calculating the product of matrix exponentials. For systems that aren't finite dimensional, they may nevertheless be approximated to arbitrary accuracy by a sufficiently large, but finite, quantum system via discretization. Consistent with our previous discussion, we have shown that quantum dynamics may be computed using standard computational techniques, however, the exponentially large matrices involved ensure that the above approach will not be efficient.

In light of the elegant solutions to the Schrödinger equation given by expressions such (2.19), and more generally (2.18), a mathematically inclined reader may conclude that closed-system quantum dynamics is, at the broadest level, a solved problem. However, without computational methods, we cannot extract useful information from these solutions, such as what's needed to make concrete predictions about the behavior of a physical system. In this section, our goal is to clearly state a high-level procedure for carrying out such computations on both classical and quantum computers.

2.6 Classical Simulation Algorithms

We've already discussed the obstacles to classical simulation of quantum mechanics in prior sections. Despite these, the importance of the problem to physics, and more broadly natural science, has led practitioners to develop very clever methods that provide insight in limited but interesting cases. Exact diagonalization [86] of the Hamiltonian, expressed as a matrix in a suitable basis, is a guaranteed approach in principle but intractable for large systems. Quantum Monte Carlo [8] methods refer to a broad range of techniques which utilize random sampling, and are especially effective at calculating the low lying energies of a Hamiltonian. However, these cannot be used directly for general calculations of quantum dynamics. Moreover, Monte Carlo techniques suffer a notorious sign problem, in which large quantities of various signs need to be added together to get a relatively small result. This leads to severe round-off errors from floating-point arithmetic. Indeed, it would be both interesting and surprising if Monte Carlo methods were more successful

at computing quantum properties. This would suggest that, computationally, the randomness of quantum theory could be reduced to mere coin flips. For this author, the sign problem is an indication that there is more to quantum probability theory than normal randomness.

More recently, tensor network methods, based on Matrix Product States (MPSs) or Projected Entangled Pairs (PEPs), represent the state of the art for dynamical simulation. As the name suggests, such methods involve the representation of quantum states as tensor networks rather than a vector of amplitudes. For systems with low entanglement, the "rank" of the tensors involved is not too large. This allows for efficient representation and manipulation of quantum states. Area-law bounds on entanglement, as found in lattice systems with geometrically local interactions, aid the effectiveness of such methods [43]. Moreover, noise in imperfectly isolated computers can reduce the coherence and entanglement, which can be further exploited. Thus, tensor networks have brought more quantum systems within the capabilities of classical computers, and have become a standard benchmark for testing the classical feasibility of quantum circuits.

Despite the aforementioned techniques, efficient simulation by classical means is likely unobtainable. This comes from both complexity theoretic arguments as well as the sheer effort towards making effective simulation methods.

2.7 Quantum Simulation Algorithms

As we've anticipated, quantum computers provide a platform for efficient quantum simulation given reasonable assumptions on the input Hamiltonian. Figure 2.2, reprinted from [52], gives a high-level overview of the Hamiltonian simulation workflow on quantum devices. Before any computation can be performed, the problem of interest must be encoded onto a quantum computer. This could be accomplished in several ways depending on the nature of the problem. For example, effective mappings from fermionic [94] and bosonic [116] systems to qubits have been extensively studied in the literature. We will not say much more on how to properly map a problem of interest onto a collection of qubits, though this is a crucial starting step for any attempt at quantum Hamiltonian simulation.

Having identified a mapping, one can turn to the simulation proper. An initial state $|\psi_0\rangle$

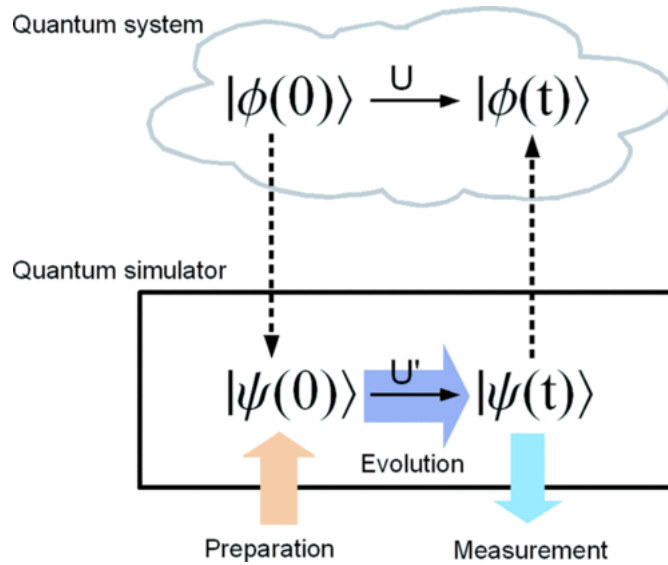


Figure 2.2 Schematic of the quantum simulation workflow. The system of interest, represented by the upper "cloud," is first represented on the quantum computer through some correspondence given by the dashed lines. The three main tasks for simulation are state preparation $|\psi(0)\rangle$, time evolution by $U(t, 0)$, and a measurement procedure. Each of these steps has its own set of tools and methods. Reprinted from [52] with permission. Copyright 2014 by the American Physical Society.

of the simulation must be prepared by a quantum circuit given an initial "fiducial state" of the device, typically $|0\rangle^{\otimes n}$. The difficulty of preparing $|\psi_0\rangle$ depends on the nature of the problem and the qubit encoding used. Entire subdomains of digital Hamiltonian simulation are devoted to preparing effective initial states in various contexts [121, 40, 9, 99]. Although special states such as ground states and wavepackets are often sought for, the arbitrariness of initial state choice makes it difficult to do a general analysis of such methods. In contrast, given a specific input model for the Hamiltonian $H(t)$, one can express general algorithms for generating a quantum circuit V which approximates the time evolution operator U . Often, this step alone is what is referred to as "Hamiltonian simulation", though it is necessarily one part of the full process needed to extract useful information. Essentially, the problem is, given an input Hamiltonian $H(t)$, a desired simulation interval $[0, t]$ and a desired tolerance $\epsilon > 0$ construct a quantum circuit V such that the output $V|\psi_0\rangle$ is within ϵ of $U(t, 0)|\psi_0\rangle$.

Once the state is evolved in time, it needs to be measured. Simply having the evolved state on a quantum register is not enough, and is nothing like having the state vector written on paper. Learning

the full output state $|\psi_t\rangle$, known as full-state tomography, is extremely inefficient. However, in actual quantum calculations we are typically interested in a handful $\{O_i\}$ of observables of interest. Learning these observables given $|\psi_t\rangle$ turns out to be a much more approachable task; standard routines such as phase estimation and amplitude estimation are sufficient [56, 67]. In Section 6.3, we will discuss the Rodeo Algorithm, a new addition to the suite of phase estimation algorithms which can perform resource-efficient measurements in the eigenbasis of a Hamiltonian using time evolution.

Despite the utility of the of the Preparation-Evolution-Measurement schematic, actual simulation protocols may not be purely sequential. For example, optimal protocols for eigenvalue and expectation value measurement protocols, particularly quantum phase estimation protocols, incorporate the Evolution as a subroutine. We will in fact see this in the Iterative Amplitude Estimation protocol of Chapter 3. Even so, the framework is still useful in providing a program for developing simulation algorithms. In any conceivable case where a simulation is needed, we will need a way to (a) prepare a state, (b) evolve the state and (c) measure the state.

Without additional assumptions, even quantum computers cannot solve Hamiltonian simulation efficiently in all instances. A simple counting argument shows that an ϵ -approximation to an n qubit unitary U requires, in general, an exponentially large number of elementary quantum gates in n [101]. Any such unitary may, if desired, be viewed as a time evolution operator for *some* Hamiltonian H .

$$U = e^{-iH} \tag{2.21}$$

Evidently, there are some Hamiltonians H whose simulation is requires exponential quantum resources, and is hence intractable.

This point aside, most Hamiltonians of interest are not of this nature. Physical Hamiltonians, as described previously, are naturally k -local, and we shall leverage this in the product formula algorithms below. In contexts removed from physics, such as solving linear systems, the Hamiltonian H is a sparse matrix (with efficiently computable and locatable nonzero entries). Such assumptions ensure efficient simulations are possible, and are typically baked into the definition of "Hamiltonian

simulation" and not mentioned. Over the past several decades, enormous progress has been made such that there are multiple good quantum algorithms for approximating $U(t, s)$. We now outline the three categories which, broadly speaking, classify all such methods.

2.7.1 Product Formulas

Early thinkers like Feynman and Deutsch had long claimed that quantum computers could efficiently simulate quantum mechanics, but it was Seth Lloyd's seminal algorithm, based on Trotterization, that first proved this was the case [87]. Lloyd considered k -local Hamiltonians of the form (2.9), and in this case the exponentials $\exp\{-iH_\gamma\Delta t\}$ of each term were unitary operations on only k qubits. Since k remains fixed as the number of qubits n grew, these exponentials can be implemented as quantum circuits of fixed depth. Moreover, these unitaries can be combined in sequence to approximate the full time evolution $\exp\{-iHt\}$. In his paper, Lloyd used the so-called first-order Trotter formula S_1 , defined as

$$S_1(t) := e^{-iH_1t} e^{-iH_2t} \dots e^{-iH_\Gamma t}. \quad (2.22)$$

This is a unitary operator which for small t , accurately approximates $U(t)$. In particular,

$$S_1(t) - e^{-iHt} \in O(t^2), \quad (2.23)$$

where t is taken asymptotically to zero. Longer simulation times can be achieved by dividing the full interval into r steps.

$$S_1(t/r)^r = \left(e^{-iH_1t/r} e^{-iH_2t/r} \dots e^{-iH_\Gamma t/r} \right)^r = e^{-iHt} + O(t^2/r) \quad (2.24)$$

By taking r sufficiently large, the error can be arbitrarily diminished.

More generally, *product formulas* are unitary approximations to e^{-iHt} made by splitting the exponential along the terms H_γ in a specified sequence. The *order* of a product formula \mathcal{P} characterizes the degree of approximation to U , and is the largest integer $p \in \mathbb{Z}_+$ such that

$$U(t) - \mathcal{P}(t) \in O(t^{p+1}). \quad (2.25)$$

In other words, the order is the largest p for which $U(t)$ and $\mathcal{P}(t)$ share the same p th Taylor polynomial. This definition justifies our referring to (2.22) as 1st order. Product formulae exist for

all orders p , and in fact there is a recursive procedure for generating higher order formulas from lower ones [123]. The standard candle is the so-called Suzuki-Trotter formulas. To define these let S_1 be given by (2.22), and define

$$S_2(t) := \text{rev}[S_1(t/2)]S_1(t/2) \quad (2.26)$$

where rev is the reverse of the terms in the product of S_1 . It can be quickly verified via Taylor expansion that S_2 is second order. It is also *symmetric*, both in the sense of the ordering of its terms and in a time reversal sense.

$$S_2(-t) = S_2(t)^\dagger \quad (2.27)$$

Any product formula \mathcal{P} satisfying the condition (2.27) will be termed "symmetric." Symmetric product formulas have the useful property that their error series, namely the power series of $U(t) - \mathcal{P}(t)$, is an odd function. Thus, any procedure which seeks to eliminate errors term by term can skip all even powers.

For any $k \in \mathbb{Z}_+$, with $k > 1$, we define S_{2k} recursively as

$$S_{2k}(t) = S_{2(k-1)}^2(u_k t) S_{2(k-1)}((1 - 4u_k)t) S_{2(k-1)}^2(u_k t), \quad (2.28)$$

where $u_k = (4 - 4^{(k-1)})^{-1}$. This formula is symmetric and order $2k$. Thus, product formulas of arbitrary order exist. However, we observe that our recursive procedure generates an exponentially increasing number of unitaries as a function of k , leading to impractically high costs for modest accuracy gains. It is possible to show that this feature is present for any high order formula, by considering the number of terms needed to eliminate errors term by term. Thus, in practice, only the lowest orders formulas are used. Nevertheless, the existence of arbitrary order formulas is valuable theoretically to understand asymptotic scaling of simulation costs. Moreover, the forward and backward evolutions present in (2.28) give rise to fractal behavior for large k that is itself interesting [122].

Product formulas satisfy the useful property that, in the case where all the H_γ commute, the error in the Trotterization vanishes, for the same reason that $e^w e^z = e^{w+z}$ for $w, z \in \mathbb{C}$. More

generally, one should expect the simulation errors to be small even when the commutator

$$[H_i, H_j] := H_i H_j - H_j H_i \quad (2.29)$$

is small but nonzero. We term this feature *commutator scaling*. It took a surprisingly long time to show this rigorously, but was finally done using the calculus of matrix-valued functions [27]. These results applied to a general class of *staged* product formulas of the form

$$\mathcal{P}(t) = \prod_{j=1}^{\Upsilon} \prod_{\gamma=1}^{\Gamma} e^{-t\tau_{j\gamma} H_{\pi_j(\gamma)}}. \quad (2.30)$$

Here, Υ is the number of "stages", and π_j is a permutation of the first Γ positive integers. Moreover, τ_{jk} are real numbers. The Suzuki-Trotter formulas are examples of staged formulas. It was shown that the additive and multiplicative errors \mathcal{A}, \mathcal{M} in the estimation of e^{-iHt} for a p th order staged product formula \mathcal{P}_p scale as

$$\mathcal{A}, \mathcal{M} \in O(\tilde{\alpha}_{\text{comm}} t^{p+1}) \quad (2.31)$$

where

$$\tilde{\alpha}_{\text{comm}} = \sum_{\gamma_1=1}^{\Gamma} \sum_{\gamma_2=1}^{\Gamma} \cdots \sum_{\gamma_{p+1}=1}^{\Gamma} \|[H_{\gamma_{p+1}}, \dots, [H_{\gamma_2}, H_{\gamma_1}], \dots]\| \quad (2.32)$$

is a sum of the norms of all nested commutators of $p + 1$ terms of H [27]. Though $\tilde{\alpha}_{\text{comm}}$ may be practically difficult to compute, this result gives a tighter characterization of the kinds of errors to expect from a Trotter simulation. Moreover, these commutators can be computed once for specific classes of Hamiltonians, such as lattice systems, and the results can be applied thereafter.

Despite these sophisticated theoretical characterizations, product formulas have been observed to perform even better than expected [25, 66]. The relative simplicity and flexibility of Trotter methods, as compared to the more recent and theoretically improved methods described in subsequent sections, makes this approach the current frontrunner in practical quantum simulation. However, as quantum hardware continues to improve, we may find that so-called post-Trotter methods become increasingly attractive as the overhead costs become less burdensome.

Using the error bounds (2.31), we can derive a rigorous upper bound on the number of Trotter steps r needed to simulate a k -local H for time T to accuracy ϵ using a p th order product formula.

$$r \in O\left(\frac{\tilde{\alpha}_{\text{comm}}^{1/p} T^{1+1/p}}{\epsilon^{1/p}}\right) \quad (2.33)$$

For fixed order formula p the total number of exponentials $e^{-iH_j\delta t}$ scales as r up to constant factors. Since each exponential requires at most a constant number of quantum gates, the above formula also gives the scaling of the number of two-qubit gates. What we see is that we have achieved efficient simulation in terms of T and ϵ . Moreover, the number of qubits needed is only those n which generate the state space of the system. Contrast this with the 2^n needed, naively, to write the full state classically.

Product formulas will recur throughout this thesis. One of the deficiencies of this approach to simulation is their relatively low accuracy, especially to post-Trotter methods with $O(\log 1/\epsilon)$ scaling in the accuracy. In Chapter 3, we explore the use of polynomial interpolation to improve this accuracy without employing additional quantum resources. Trotterization also arises in the time dependent setting, where each $H_\gamma(t)$ depends on time, by using (2.18) for finite n and Trotterizing each $H(t_j)$. The use of an auxiliary "clock space" connects the notions time independent and time dependent simulation (and Trotterization), which we use to propose and analyze several approaches to time dependent Hamiltonian simulation. This is considered in Chapter 4.

2.7.2 Linear Combination of Unitaries

For a couple decades (a long time relative to the field), product formulas were the only algorithm in town for Hamiltonian simulation on quantum computers. Then, in 2012, Childs and Wiebe introduced [24] a new primitive for quantum computation: applying a linear combination of unitaries

$$\sum_{j=1}^L \alpha_j U_j. \quad (2.34)$$

to a quantum register. Figure 2.3 gives a schematic circuit for implementing this sum. Because a sum of unitaries is not unitary, we should expect that some measurements are required for the successful implementation of the operation, and indeed, it is conditioned on measuring all 0's on

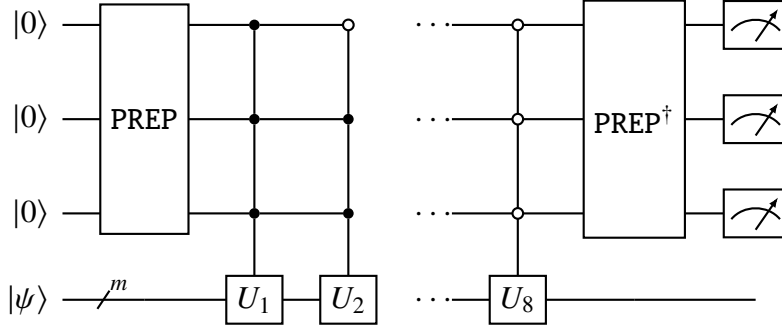


Figure 2.3 A schematic of the linear combination of unitaries (LCU) circuit. Conditioned upon measuring 0 on every measurement, the result applied is $\sum_j \alpha_j U_j$. In this case, there are 3 auxiliary qubits, hence $2^3 = 8$ possible unitaries to add. The PREP circuits are any circuit satisfying $\text{PREP } |0\rangle^n = \sum_j \sqrt{\alpha_j} |j\rangle$.

the auxiliary register. Measuring "success" is more likely the closer (2.34) is to a unitary operator. Success can be achieved through repeated trials or, more efficiently, through quantum amplitude amplification.

The linear combination of unitaries (LCU) circuit consists of two subroutines. The first is a PREP ("prepare") unitary which acts on a quantum register initialized to $|0\rangle^{\otimes k}$ as

$$\text{PREP } |0\rangle^{\otimes k} = \sum_{j=1}^L \sqrt{\frac{\alpha_j}{\|a\|_1}} |j\rangle \quad (2.35)$$

with $\|a\|_1 := \sum_j |a_j|$. The second is a SEL ("select") unitary which applies U_j to the main register controlled on state j of the auxiliary.

$$\text{SEL } |j\rangle |\psi\rangle = |j\rangle U_j |\psi\rangle \quad (2.36)$$

By applying the operation $(\text{PREP}^\dagger \otimes I)\text{SEL}(\text{PREP} \otimes I)$, then measuring the appropriate outcome on the auxiliary (all zeros), the operation (2.34) may be implemented on the main register up to normalization.

The LCU primitive led to several distinct Hamiltonian simulation algorithms, the first post-Trotter methods. Here we discuss the one most relative to this thesis, and the one considered in the original Childs and Wiebe paper. This is the quantum implementation of the multiproduct formula (MPF), which is a linear combination of product formulas producing a more accurate approximation to e^{-iHt} . The best way to understand multiproduct formulas is as an instance of

Richardson extrapolation, and we will elaborate on this point in Chapter 5. The major point is that we wish to extrapolate $r \rightarrow \infty$, or equivalently $1/r \rightarrow 0$, to achieve increasing accuracies. While higher-order Trotter formulas require an exponential number of terms, it is much less demanding to cancel error terms using summation. The upshot is that MPFs, and other well-known LCU algorithms, achieve an exponential improvement in accuracy over product formulas alone.

2.7.3 Qubitization

In our discussion of the Linear Combination of Unitaries (LCU) primitive above, we saw that the unitary

$$U = (\text{PREP}^\dagger \otimes I)\text{SEL}(\text{PREP} \otimes I) \quad (2.37)$$

encodes the desired operation (a sum of unitaries) provide a certain measurement result is achieved on a part of the full quantum register. This turns out to point towards a more general phenomenon. We say that U *block encodes* the desired operation $\sum_j \alpha_j U_j$ via the state $|0\rangle^{\otimes k}$, in the sense that

$$\sum_j \alpha_j U_j = (\langle 0|^{\otimes k} \otimes I)U(|0\rangle^{\otimes k} \otimes I) \quad (2.38)$$

up to normalization. In our context, this block-encoded operator may be the Hamiltonian of interest. Remarkably, this simple requirement, the encoding of H in a subblock of a larger unitary matrix, is all that is necessary for qubitization [88], the first asymptotically optimal approach to Hamiltonian simulation. By this, we mean that the simulation cost for simulating for time T and accuracy ϵ scales as

$$O\left(T + \frac{\log 1/\epsilon}{\log \log 1/\epsilon}\right) \quad (2.39)$$

which saturates the known lower bounds [14] for T and $1/\epsilon$ and is *additive* rather than multiplicative.

Besides the concept of block encoding, the other major ingredient to qubitization is Quantum Signal Processing (QSP), which generates polynomial functions of the eigenvalues of the block encoded operator. It does so by interleaving unitary rotations that act independently on a two-dimensional subspace corresponding to each eigenvalue. Since $e^{-i\lambda t}$ can be approximated by polynomials $P(\lambda t)$ uniformly over an interval $[0, \lambda T]$, constructing these polynomial transformations provide a means to Hamiltonian simulation. Qubitization was later generalized to the

Quantum Singular Value Transformation (QSVT) [54], which performs polynomial transformation of the singular values of a block encoded matrix. The method is effective, general, conceptually rich and, unfortunately, rather complicated to understand fully.

One notable drawback to qubitization is that it only directly works for time independent Hamiltonian simulations. Of course, one could Trotterize via (2.18) and perform qubitization on each factor, but the error from the Trotterization would overshadow any gains from qubitization. In Chapter 4, we will embed $H(t)$ in a system with time dependent Hamiltonian, then apply qubitization "directly." We construct a suitable block encoding of the augmented Hamiltonian, then derive bounds on the query complexity. While our analysis does not show improvements compared to other time dependent schemes, we believe this is from imperfections in the analysis rather than a features of the method.

2.7.4 Analog Quantum Simulation

In order to anticipate the material of Section 6.1, we digress momentarily from our discussion of quantum computers and consider an alternative simulation platform: *analog quantum simulators*. Recall from Figure 2.2 the need to map the problem of interest onto a simulator. Instead of mapping to a "digital" setting, in which a universal set of discrete, unitary quantum gates are applied in sequence to construct the time evolution operator, we could instead map to a system which, though not a quantum computer, nevertheless allows for a great deal of control and emulation of a system of interest. This is potentially less technologically demanding, because we no longer need to perform arbitrary unitary operations, but rather specific Hamiltonians of interest.

For excellent discussions of analog quantum simulators, see e.g. [63, 52]. For our purposes, we can roughly model analog simulators as devices which can implement a class of Hamiltonians, with terms

$$H = \sum_j J_j(t) H_j \tag{2.40}$$

such that the time dependence $J_j(t)$ is controllable to some degree. Depending on the collection of H_j and the degree of time dependence J_j , we might be able to achieve a universal device. Yet this may not be necessary for the application we are interested in.

Analog simulators are often considered a more achievable route to quantum Hamiltonian simulation. In Section 6.1 we will explore some interesting phenomena accessible to trapped-ion quantum simulators, with implications in nuclear physics.

2.7.5 Measurement and Hamiltonian Evolution

As already emphasized, time evolution represents only one piece of a full quantum simulation algorithm. Although the time evolved state $|\psi_t\rangle$ is prepared on a quantum register, no information is gained without measurement. By analogy, a random variable that is not sampled leaves nothing gained. There is nontrivial work to be done in the extraction of information, as sampling in the quantum case (i.e., measurement) is more nuanced than simple probability sampling. This is because multiple different bases can be measured, choosing the right basis is important for extracting valuable information.

One basic fact we might like to know about a Hamiltonian are its eigenvalues, which correspond to the "allowed" energies of a physical system. Additionally, general observables O can be often simulated as if they were Hamiltonians assuming they are, say, sparse or k -local. Suppose we are able to prepare an (approximate) eigenvector $|E\rangle$ of H on a quantum register. Performing Hamiltonian simulation will produce an output state that picks up merely a phase.

$$U(t) |E\rangle = e^{-iEt} |E\rangle \tag{2.41}$$

This phase cannot be measured as is, because only relative phase shifts, not overall phases, are measurable in general wavelike phenomena. Our strategy will be to introduce a reference unshifted state and produce interference, such that the phase $\varphi = Et$ is measurable. One way to do this in the setting of quantum computing is to apply a controlled- U gate. By putting an auxiliary register in a superposition $|0\rangle + |1\rangle$ and applying U to the main register conditioned on $|1\rangle$, we've introduced a relative phase shift into the auxiliary qubit. This can be measured through a change of basis.

What we've heuristically described above is a general framework for performing *phase estimation* on a quantum computer, which allows for the extraction of eigenvalues. Figure 2.4 gives the basic circuit for the simplest phase estimation algorithm. Physically, the circuit is essentially a

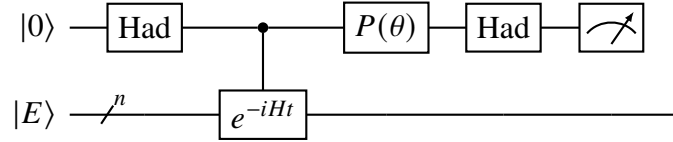


Figure 2.4 Schematic of the simplest quantum phase estimation algorithm, which is useful for measuring in the eigenbasis of a Hamiltonian. The measurement outcomes associated with the top auxiliary qubit are directly related to the eigenvalues of H , and by repeated trials these eigenvalues can be estimated. The phase rotation $P(\theta)$ has an angle parameter θ which may be varied to resolve certain ambiguities due to \cos^2 not being one-to-one on $[0, \pi]$.

Mach-Zehnder interferometer [67], with Hadamards

$$\text{Had} := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (2.42)$$

acting as beam splitters and the controlled time evolution introducing a phase shift. The family of phase estimation algorithms is, by now, enormous [67, 77, 125, 136]. This thesis will discuss a recent addition to this family known as the Rodeo Algorithm, a resource-efficient, randomized procedure which performs a selective search over the space of possible eigenvalues. Like all phase estimation protocols, it works well provided the initial state has reasonable overlap with the eigenstates of interest (formally, if the overlap decreases polynomially with problem size).

Perhaps even more foundational than phase estimation, in a sense, is the estimation of an *amplitude* on a quantum computer. In fact, the simple scheme above encodes the phase in an amplitude to be measured. Slightly more generally, we suppose an operator V acting on an initial state $|0\rangle$ of a single qubit as $V|0\rangle = a_0|0\rangle + a_1|1\rangle$, and wish to estimate a_0 (without loss of generality we may take $a_0 \geq 0$). This estimation may be done through repeated computational basis measurements, but more clever approaches using amplitude amplification lead to a quadratic improvement in scaling. See [59] for details on an efficient iterative procedure for amplitude estimation.

2.8 Mathematical Reference

The following section includes important technical definitions and results that are used throughout the dissertation. The reader may, if interested, browse these mathematical tools now, or they may come back and reference as needed as they read subsequent chapters.

2.8.1 Combinatorics

The simple factorial $n!$ counts the number of permutations of n objects, and is usefully approximated by *Stirling's approximation*. In the paper, we always make use of a version of the approximation which gives strict bounds for $n \in \mathbb{Z}_+$.

$$\sqrt{2\pi n} \left(\frac{n}{e}\right)^n < n! < \sqrt{2\pi n} \left(\frac{n}{e}\right)^n e^{1/(12n)} \quad (2.43)$$

These bounds are extremely tight, even for small n .

The *multinomial coefficient* is a generalization of the more common binomial coefficient, and it arises in several combinatorial situations. It is defined by

$$\binom{n}{n_1, \dots, n_k} := \frac{n!}{n_1! n_2! \dots n_k!} \quad (2.44)$$

where $n \in \mathbb{Z}_+$ and the $(n_\ell)_{\ell=1}^k$ are nonnegative integers which sum to n . It is a positive integer corresponding to the number of distinct ways of placing n distinguishable items into k boxes, where each box has a fixed number n_ℓ of items. In this work, we will find occasion to make use of the multinomial when evaluating high-order derivatives of a product.

$$\left(\frac{d}{dt}\right)^n f_1(t) f_2(t) \dots f_k(t) \quad (2.45)$$

Here, $(f_\ell)_{\ell=1}^k$ are n -differentiable functions of $t \in \mathbb{R}$. Employing the product rule, one is left to count all the possible combinations of derivatives of each f_ℓ . It turns out that the multinomial is suited for this.

$$\left(\frac{d}{dt}\right)^n \prod_{\ell=1}^k f_\ell(t) = \sum_N \binom{n}{n_1, \dots, n_k} \prod_{\ell=1}^k \left(\frac{d}{dt}\right)^{n_\ell} f_\ell(t) \quad (2.46)$$

The sum is taken over the set N of sequences of nonnegative integers $(n_\ell)_{\ell=1}^k$ summing to n . A useful property is that

$$\sum_N \binom{n}{n_1, \dots, n_k} = k^n \quad (2.47)$$

for nonnegative integers k, n (with convention $0^0 = \lim_{x \rightarrow 0} x^x = 1$).

Besides derivatives of products, we will also need to bound derivatives of ordinary exponentials of a time dependent matrix. Useful for this purpose is an expression for derivatives of exponentials

of a scalar function $a(t)$.

$$\left(\frac{d}{dt}\right)^n e^{a(t)} \quad (2.48)$$

The solution we rely on is *Faà di Bruno's formula*, which asserts that

$$\left(\frac{d}{dt}\right)^n e^{a(t)} = e^{a(t)} Y_n(a'(t), a''(t), \dots, a^{(n)}(t)) \quad (2.49)$$

where Y_n is the *complete exponential Bell polynomial* [31]. An explicit formula is given by

$$Y_n(x_1, x_2, \dots, x_n) = \sum_C \frac{n!}{c_1! c_2! \dots c_n!} \prod_{j=1}^n \left(\frac{x_j}{j!}\right)^{c_j} \quad (2.50)$$

where the sum is taken over the set C of all sequences $(c_j)_{j=1}^n$ such that $c_j \geq 0$ and

$$c_1 + 2c_2 + \dots + nc_n = n. \quad (2.51)$$

Essentially, each coefficient in Y_n counts the ways one can partition a set of fixed size n into subsets of given sizes and number. When one simply wants to count the total number of possible partitions, one is led to the *Bell numbers* b_n . These are related to the Y_n by evaluating all arguments to 1.

$$b_n = Y_n(1, 1, \dots, 1) \quad (2.52)$$

More generally, for any $x \in \mathbb{R}$,

$$Y_n(x, x^2, \dots, x^n) = x^n b_n, \quad (2.53)$$

which can be seen directly from (2.50) along with the sum rule (2.51). The Bell numbers b_n grow combinatorially; in particular, the following upper bound [13] is useful.

$$b_n < \left(\frac{.792n}{\log(n+1)}\right)^n, \quad \forall n \in \mathbb{Z}_+ \quad (2.54)$$

More generally, the single-variable *Bell polynomial*, or *Touchard polynomial* $B_n(x)$, is simply Y_n with all arguments evaluated to x .

$$B_n(x) = Y_n(x, x, \dots, x) \quad (2.55)$$

Of course, $b_n = B_n(1)$. The n th Bell polynomial $B_n(x)$ is also the value of the n th moment of the Poisson distribution with mean x . From [5] we have the following upper bound on B_n

$$B_n(x) \leq \left(\frac{n}{\log(1 + \frac{n}{x})}\right)^n, \quad \forall x \geq 0 \quad (2.56)$$

which we observe is very close to that for the Bell numbers ($x = 1$) in equation (2.54). From their definitions, Y_n , B_n and b_n all grow monotonically, both in their functional arguments and their index n . This is intuitive from being combinatorial functions whose coefficients count something according to the size of n .

2.8.2 Norms

Norms are used widely throughout the paper to characterize the size of mathematical objects and quantify simulation costs. For finite-dimensional vectors $v = (v_1, \dots, v_n)$, real valued or complex, the *Schatten p -norm*, is defined as

$$\|v\|_p := \left(\sum_{j=1}^n |v_j|^p \right)^{1/p} \quad (2.57)$$

for any $p \in [1, \infty)$, and for $p = \infty$ as $\|v\|_\infty = \max_j |v_j|$. We make particular use of the 1 and ∞ norm in our paper, to express our results or quote previous ones.

We also make use of functional norms, which are defined analogously. Given a scalar function $f(t)$ with scalar input over an interval $[0, T]$, the p -norm, or L^p norm, for $p \in [1, \infty)$ is given by

$$\|f(t)\|_p := \left(\int_0^T |f(\tau)|^p d\tau \right)^{1/p} \quad (2.58)$$

for functions such that it is defined. Analogously, the ∞ -norm is given by the supremum $\sup|f(t)|$ over $[0, T]$. For the piecewise smooth functions we consider, this is just the maximum value on the interval, so we might write $\|f(t)\|_{\max}$.

In Table 1, we use notation $\|\cdot\|_{p,q}$ to denote nested norms whenever our objects have both a (finite) vector and functional character. Specifically, this notation means take the vector p -norm first, then take the q -norm of the resulting scalar function. For example, if $\alpha(t) = (\alpha_1(t), \alpha_2(t), \dots, \alpha_n(t))$, then

$$\|\alpha\|_{1,1} = \left\| \sum_{j=1}^n |\alpha_j(t)| \right\|_1 = \int_0^T \sum_{j=1}^n |\alpha_j(\tau)| d\tau \quad (2.59)$$

and

$$\|\alpha\|_{1,\max} = \max_{\tau \in [0, T]} \sum_{j=1}^n |\alpha_j(\tau)|. \quad (2.60)$$

In the main paper we claim our Hamiltonian simulation algorithm exhibits L^1 -norm scaling. This means it has complexity $O(\|f\|_1)$, where f is a function whose value is some measure of the size of the Hamiltonian and its derivatives at each $t \in [0, T]$.

Finally, our paper makes use of the *spectral norm* for linear operators, also known as the *induced 2-norm*. It is defined for any bounded operator A on a Hilbert space \mathcal{H} by

$$\|A\| := \sup_{v \in \mathcal{H} \setminus \{0\}} \frac{\|Av\|_2}{\|v\|_2}. \quad (2.61)$$

In our case, A will always be finite dimensional, and $\|A\|$ is the largest singular value of A . This norm is invariant under left or right multiplication by a unitary operator U , and $\|U\| = 1$. The spectral norm is *submultiplicative*, a property we make frequent use of.

$$\|AB\| \leq \|A\| \|B\| \quad (2.62)$$

CHAPTER 3

TROTTER ERROR MITIGATION

This chapter is based on recent work [113] concerning the reduction of Trotter error by use of standard Chebyshev interpolation, and is outlined as follows. After providing some background and motivation, we spend two sections describing the interpolation procedure and proving some results on its robustness to noisy data. We then apply the framework to the important task of measuring expectation values of time-evolved observables. Numerical demonstrations are provided for a random 1D Heisenberg model, and show the expected behavior. Discussion on the implications of this work is given, and technical results are proven at the end of the chapter. With the exception of the numerics of Section 5.7, which are new, additional details, applications, and numerics can be found in the main publication [113].

3.1 Introduction and Motivation

In Section 2.7, we gave an overview of product formulas as a means for Hamiltonian simulation. Efficient, versatile, and simple, product formulas will perhaps remain the preferred method of quantum simulation on quantum computers for the foreseeable future. This motivates the search for techniques to further bolster the method, particularly by mitigating its biggest flaws. Chief among these, perhaps, is their relative inaccuracy compared to post-Trotter methods such as qubitization. As an example, the 1st order Trotter formula S_1 , which splits the exponential e^{-iHt} in the simplest imaginable way,

$$S_1(t) = e^{-iH_1t} e^{-iH_2t} \dots e^{-iH_r t} \quad (3.1)$$

has an error scaling given by the $O(t^2/r)$. From this we deduce that the simulation cost scales with the error ϵ as $O(1/\epsilon)$. Contrast this with, say, qubitization, which scales slightly better than $O(\log 1/\epsilon)$, an exponential improvement. More generally, every major post-Trotter method scales polylogarithmically in $1/\epsilon$ (that is, $O(p(\log 1/\epsilon))$ for some polynomial p). Higher order product formulas have better accuracy, but are generally impractical due to exponential scaling of cost with the order k of the formula.

Can the accuracy of product formulas be improved with additional techniques? Indeed, multi-

product formulas, mentioned in Section 2.7.2 and elaborated in Chapter 5, achieve exponentially improved accuracy compared to product formulas alone by summing formulas of different Trotter step size. Unfortunately, the additional quantum overhead and controlled operations required for implementing the required LCU procedure are noteworthy, and introduce barriers which are especially burdensome in the current era of noisy hardware.

Faced with these limitations, we might look beyond multiproduct formulas based on LCU and ask if there is a way forward using only *classical* resources, such as randomness. Indeed, within the past few years there have been claims of using multiproduct formulas without LCU-type procedures [45, 130]. These schemes don't produce a true multiproduct formula, in the sense of applying the operation

$$\text{MPF}(t) = \sum_{j=1}^L c_j \mathcal{P}(t/r_j)^{r_j} \quad (3.2)$$

to a quantum register, where \mathcal{P} is a product formula, $r_j \in \mathbb{Z}_+$, and $c_j \in \mathbb{R}$. But in any case, this is not strictly necessary. After all, preparing $e^{-iHt} |\psi\rangle$ is not a full algorithm, but a possible intermediate step, after which measurements must be performed to extract the desired information. For example, one might be interested in estimating computing the dynamics of observables via

$$\langle O(t) \rangle = \langle \psi_t | O | \psi_t \rangle. \quad (3.3)$$

This suggests the possibility of summing not operations, but classical data coming from measurement schemes. This amounts to a standard Richardson extrapolation of the expectation values [44]. Let $s = 1/r$ be the "normalized Trotter step." For a given product formula \mathcal{P} in r steps, the expectation value

$$\langle O_s(t) \rangle := \langle \psi_0 | \mathcal{P}^\dagger(st)^{1/s} O \mathcal{P}(st)^{1/s} | \psi_0 \rangle \quad (3.4)$$

is a real-valued, smooth function $f(s)$. By estimating $f(s_j)$ for various s_j , an extrapolation can be performed to the desired $s = 0$. What is attractive about this scheme is that no additional quantum resources, beyond the product formulas simulation and expectation value measurements, are needed.

There are several directions in which these ideas can be extended. First, the function $f(s)$ could represent a variety of kinds of measurements obtained from product simulations, beyond expectation values. For example, they could represent eigenvalue estimates from phase estimation. Second, we might consider other means of estimating $f(0)$ besides Richardson extrapolation. One possibility is to produce a uniform approximation to f in a neighborhood of $s = 0$. Function approximation is an old science, and many numerical techniques are available for the task. Modern machine learning approaches such as neural networks might achieve excellent approximation for f , but finding the simplest effective tool for the task is desirable, and moreover it is difficult to provide rigorous guarantees for machine learning.

3.2 Polynomial Interpolation

Among the collection of function approximation methods available, we choose one that is simple to implement and easy to analyze: polynomial interpolation. Essentially, our goal is to use interpolation to "extrapolate" the Trotter step size s to the ideal of $s = 0$.¹ There are many quantities that we could be interested in extrapolating. For this thesis, we will be primarily concerned with expectation value estimation.

$$\begin{aligned}\langle O_s(t) \rangle &= \text{Tr } \rho O_s(t) \\ O_s(t) &:= \tilde{U}_s(t)^\dagger O \tilde{U}_s(t)\end{aligned}\tag{3.5}$$

For purposes of analysis, we'll assume these expectation values are estimated on a quantum computer using one of the Suzuki-Trotter (ST) formulas of equation (2.28). However, in principle our approach should work for any product formula simulation, not just ST.

While the interpolation is classical and independent of the method in which the data is generated, we will assume a quantum simulation is used when considering the computational cost. We assume all quantum operations are executed perfectly, including the exponentials $\exp(-iH_j t)$ for simulation. This is not to say that the interpolation method could not be applied to noisy quantum systems, but rather that our cost analysis does not account for it. Consequently, the only sources of error

¹We occasionally interchange between the terminology "extra-" and "interpolation." We view our method as an extrapolation beyond the data using a numerical technique commonly known as polynomial interpolation.

considered are the interpolation error and error in the calculation of the data points (e.g. the Hamiltonian energies or expectation values at various points s_i). Error in the data points may arise from hardware noise, but even in its absence, a measurement protocol such as phase estimation induces a systematic error that cannot be removed.

Without further ado, we now describe the interpolation framework. Let $f \in C^\infty([-a, a])$ be a smooth, real-valued function of a single variable $s \in [-a, a]$ and suppose we have calculated f (perfectly) for n distinct points $s_1, s_2 \dots s_n \in [-a, a]$. That is, we have data in the form of a set of tuples $D = \{(s_i, f_i)\}_{i=1}^n$, where $f_i = f(s_i)$. Let $P_{n-1}f$ be the unique $(n-1)$ -degree polynomial interpolating D , i.e. $P_{n-1}f(s_i) = f_i$ for each $i = 1, \dots, n$. For any $s \in [-a, a]$, standard results in polynomial interpolation [107] tell us that the signed error is given by

$$E_{n-1}(s) := f(s) - P_{n-1}f(s) = \frac{f^{(n)}(\xi)}{n!} \omega_n(s) \quad (3.6)$$

for some $\xi \in I_s$, where $I_s \subset [-a, a]$ is the smallest interval containing s and the interpolation points $\{s_i\}$. Throughout this work, superscripts such as in $f^{(n)}$ will refer to n th-order derivatives. The n th degree nodal polynomial $\omega_n(s)$ is defined as the unique monic polynomial with zeros at the interpolation points.

$$\omega_n(s) := \prod_{i=1}^n (s - s_i) \quad (3.7)$$

Our estimate for $f(0)$ is $P_{n-1}f(0)$. Since we are interested in $s = 0$, ω_n becomes a (signed) product of the interpolation points. We can bound the interpolation error $E_n(0)$ in a way that is independent of the precise value of ξ (which is unknown and difficult to find) by maximizing over $\xi \in I_s$.

$$|E_{n-1}(0)| \leq \max_{s \in I_s} \frac{|f^{(n)}(s)|}{n!} \prod_{i=1}^n |s_i| \quad (3.8)$$

Much of the technical work in this dissertation involves finding suitable bounds on the size of the derivatives $f^{(n)}$. In particular, in the expectation values of equation (3.5), $f(s) = \langle O_s(t) \rangle$.

For reasons which we discuss in the following Section, we choose the Chebyshev nodes on $[-a, a]$ as our interpolation points.

$$s_i = a \cos\left(\frac{2i-1}{2n}\pi\right) \quad (3.9)$$

This allows us to specialize our interpolation error in the manner described in the following lemma.

Lemma 3.2.1. *Let s_i , $i = 1, 2, \dots, n$ be the collection of Chebyshev interpolation points on the interval $[-a, a]$. In the notation above, we have*

$$|E_{n-1}(0)| < \max_{s \in [-a, a]} |f^{(n)}(s)| \left(\frac{a}{2n}\right)^n.$$

Proof. For n odd, $s = 0$ is one of the interpolation points, so the error is zero and the bound holds automatically. Hereafter, we only consider n even (which will be the case of practical interest).

Using the generic bound (3.8) with the Chebyshev nodes,

$$|E_{n-1}(0)| \leq \max_{\xi \in [-a, a]} |f^{(n)}(\xi)| \frac{1}{n!} a^n \prod_{i=1}^n \left| \cos\left(\frac{2i-1}{2n}\pi\right) \right|. \quad (3.10)$$

To obtain the lemma, we just need to appropriately bound the product of cosines. Since n is even, $n = 2m$ for some $m \in \mathbb{Z}_+$. Moreover, we have a reflectional symmetry about m , in the sense that

$$\left| \cos\left(\frac{2i-1}{2n}\pi\right) \right| = \left| \cos\left(\frac{2(n-i+1)-1}{2n}\pi\right) \right|. \quad (3.11)$$

Hence, we only need to take the product over $i = 1, \dots, m$ and square it.

$$\prod_{i=1}^n \left| \cos\left(\frac{2i-1}{2n}\pi\right) \right| = \left(\prod_{i=1}^m \cos\left(\frac{2i-1}{4m}\pi\right) \right)^2 \quad (3.12)$$

To proceed further, let's reindex the remaining product by $i \rightarrow m - i + 1$. This gives

$$\begin{aligned} \prod_{i=1}^m \cos\left(\frac{2i-1}{4m}\pi\right) &= \prod_{i=1}^m \cos\left(\frac{\pi}{2} - \frac{2i-1}{4m}\pi\right) \\ &= \prod_{i=1}^m \sin\left(\frac{2i-1}{4m}\pi\right) \\ &< \prod_{i=1}^m \frac{2i-1}{4m} \end{aligned} \quad (3.13)$$

where we used the fact that $\sin(x) < x$ for all $x > 0$. Factoring out the denominator from the product, the remaining terms become a double factorial.

$$\prod_{i=1}^m \frac{2i-1}{4m} = \frac{(2m-1)!!}{(4m)^m} \quad (3.14)$$

The double factorial can be bounded as

$$(2m - 1)!!^2 < (2m - 1)!!(2m)!! = 2m! , \quad (3.15)$$

so that $(2m - 1)!! < \sqrt{(2m)!}$. Returning to the original product of equation (3.12), and reintroducing $n = 2m$, the resulting bound is

$$\prod_{i=1}^n \left| \cos \left(\frac{2i - 1}{2n} \pi \right) \right| < \left(\frac{\sqrt{n!}}{(2n)^{n/2}} \right)^2 = \frac{n!}{(2n)^n}. \quad (3.16)$$

Reinserting this result into the last line of equation (3.10) gives the bound stated in the lemma. \square

Though Chebyshev interpolation enjoys nice mathematical properties, it presents a challenge for Trotter simulation because of the need for noninteger time steps in equation (3.44). In the face of this obstacle, there are several options one could take: rounding to integer time steps, or perform fractional queries using, say, the Quantum Singular Value Transformation (QSVT).

First, consider rounding to integer time steps, i.e., gathering data at the nearest reciprocal integer $1/\tilde{r}$ to the Chebyshev node s . For symmetrical interval $[-a, a]$, the rounding error $|s - 1/\tilde{r}|$ goes as $O(a^2)$ as $a \rightarrow 0$. From here, one could either (a) take the estimate for $f(1/\tilde{r})$ as the estimate for $f(s)$, accruing some error in the process, or (b) perform the interpolation at the *approximate* Chebyshev nodes given by the collection of points $1/r_i$. Unfortunately, for our purposes, option (a) leads to unacceptable errors of order $O(a)$ in the data, eliminating accuracy gains. As for option (b), it is possible to use robustness results on Chebyshev interpolation [132] to argue that almost-Chebyshev nodes should be almost as well-conditioned. Again, however, we find that our scaling of the number of nodes is such that the node displacements must be quite small, leading again to poorer scaling. Because of this, for most of this work we choose to invoke access to fractional queries using the QSVT [54]. While fractional queries increase the overhead compared to Trotter alone, this overhead is a constant. In practice, it may be that taking approximate Chebyshev nodes is perfectly acceptable and stable, and if so would likely be the preferred method.

3.3 Stability Analysis

Polynomial interpolation is a valuable numerical tool, but some implementations can lead to numerical instability [36]. However, the situation is not as bad as often presented in textbooks [126].

While linear algebraic approaches involving Vandermonde matrices suffer instability for high degree polynomials [51], methods such as barycentric formulas are provably stable with respect to floating point arithmetic [68].

A particularly important consideration is the choice of interpolation nodes. It is well known that equally spaced nodes can lead to the Runge phenomenon: rapid oscillations near the ends of the interval that grow with polynomial degree [107]. These oscillations can be overcome with a superior choice of nodes, such as the zeros of the Chebyshev polynomials. Interpolations done with this set of nodes are guaranteed to converge to functions that are Lipschitz continuous as $n \rightarrow \infty$. Moreover, they are well-conditioned in the sense of small errors in the data values. Finally, because they anti-cluster around $s = 0$, they are relatively cheap to compute with Trotter formulas. In this work, we will always interpolate at the n th-degree Chebyshev nodes, or approximations thereof, on a symmetric interval $[-a, a]$ about the origin, defined in (3.9). We choose even n so as to avoid the origin (which has infinite cost to compute), and also utilize the reflectional symmetry of $f(s)$.

To compute the interpolant $P_{n-1}f$ linear algebraically, we overcome the limitations of the standard Vandermonde approach by expanding in terms of orthonormal Chebyshev polynomials rather than monomials x^j .

$$P_{n-1}f(s) = \sum_{j=0}^{n-1} c_j p_j(s) \quad (3.17)$$

Here, p_j is defined by

$$p_j(s) := \begin{cases} \sqrt{\frac{1}{n}} T_0(s), & j = 0 \\ \sqrt{\frac{2}{n}} T_j(s), & j = 1, 2, \dots \end{cases} \quad (3.18)$$

where T_j is the standard j th Chebyshev polynomial.

$$T_j(x) := \cos(j \cos^{-1} x) \quad (3.19)$$

By orthonormality, we are referring to the condition [93]

$$\sum_{k=1}^n p_i(s_k) p_j(s_k) = \delta_{ij} \quad (3.20)$$

for all $0 \leq i, j < n$, with s_k being the zeros of T_n given in (3.9). This immediately implies the matrix

$$\mathbf{V} := \begin{pmatrix} p_0(s_1) & p_1(s_1) & \dots & p_{n-1}(s_1) \\ p_0(s_2) & p_1(s_2) & \dots & p_{n-1}(s_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_0(s_n) & p_1(s_n) & \dots & p_{n-1}(s_n) \end{pmatrix} \quad (3.21)$$

is orthogonal, and therefore has condition number $\kappa(\mathbf{V}) := \|\mathbf{V}\| \|\mathbf{V}^{-1}\|$ equal to one. This is the source of well-conditioning in our approach. The coefficients $c = (c_0, c_1, \dots, c_{n-1})$ in equation (3.17) satisfy

$$y = \mathbf{V}c \quad (3.22)$$

for the vector of values $y = (f(s_1), f(s_2), \dots, f(s_n))$, since $P_{n-1}f$ is an interpolant. Hence, $c = \mathbf{V}^T y$ gives the vector of coefficients.

We now develop our argument for well-conditioning. Unless otherwise subscripted, all logarithms are natural.

Lemma 3.3.1. *Let s_1, s_2, \dots, s_n be the standard Chebyshev nodes on $[-a, a]$ (3.9) with n even. Then the nodes satisfy*

$$\sum_{k=1}^n \frac{1}{|s_k|} \leq \frac{4n}{\pi a} (\gamma + \log(2n + 2)),$$

where $\gamma \approx 0.577$ is the Euler-Mascheroni constant.

Proof. We focus on the case $a = 1$, since the general result follows by a simple rescaling. Because sine and cosine are phase shifted by $\pi/2$,

$$\sum_{k=1}^n \frac{1}{|s_k|} = \sum_{k=1}^n \frac{1}{\left| \cos\left(\frac{2k-1}{2n}\pi\right) \right|} = \sum_{k=1}^n \frac{1}{\left| \sin\left(\frac{n-2k+1}{2n}\pi\right) \right|}. \quad (3.23)$$

Taking advantage of the symmetry about $s = 0$,

$$\sum_{k=1}^n \frac{1}{\left| \sin\left(\frac{n-2k+1}{2n}\pi\right) \right|} = 2 \sum_{k=1}^{n/2} \frac{1}{\sin\left(\frac{2k-1}{2n}\pi\right)}. \quad (3.24)$$

Next, we use the lower bound

$$\sin x \geq x/2 \quad (0 \leq x \leq \pi/2) \quad (3.25)$$

in order to bound the terms of the sum.

$$\begin{aligned} 2 \sum_{k=1}^{n/2} \frac{1}{\sin\left(\frac{2k-1}{2n}\pi\right)} &\leq \frac{8n}{\pi} \sum_{k=1}^{n/2} \frac{1}{2k-1} \\ &= \frac{8n}{\pi} \left(H_n - \frac{1}{2} H_{n/2} \right) \end{aligned} \quad (3.26)$$

Here, H_n denotes the n th harmonic number. From the relation $H_n = \gamma + \psi(n+1)$, where ψ is the digamma function,

$$H_n - \frac{1}{2} H_{n/2} = \gamma/2 + \psi(n+1) - \frac{1}{2} \psi(n/2+1). \quad (3.27)$$

Moreover, since $\psi(x) \in (\log(x-1/2), \log(x))$ for any $x > 1/2$, this is upper bounded by

$$H_n - \frac{1}{2} H_{n/2} < \gamma/2 + \log(n+1) - \frac{1}{2} \log\left(\frac{n+1}{2}\right) = \frac{\gamma + \log 2}{2} + \frac{\log(n+1)}{2}. \quad (3.28)$$

Reinserting this into (3.26), one obtains the bound

$$\sum_{k=1}^n \frac{1}{|s_k|} \leq \frac{4n}{\pi} (\gamma + \log(2n+2)). \quad (3.29)$$

The general lemma follows from a rescaling by $1/a$. □

Observe that $1/|s_k|$ is essentially the number of Trotter steps to compute the k th interpolation point. Thus, Lemma 3.3.1 amounts to a bound on the total number of Trotter steps, and we see this grows as $O(a^{-1}n \log n)$.

Lemma 3.3.2. *Let $p(s) = (p_0(s), p_1(s), \dots, p_{n-1}(s))$ be a vector of (normalized) Chebyshev polynomials on $[-a, a]$. Then,*

$$\|\mathbf{V}p(0)\|_1 < \frac{2}{\pi} \log(n+1) + 1$$

where $\|\cdot\|_1$ denotes the vector 1-norm.

Proof. Let $d(s) = \mathbf{V}p(s)$. For each $k = 1, 2, \dots, n$ we have

$$\begin{aligned} d_k(s) &= \sum_{j=0}^{n-1} \mathbf{V}_{kj} p_j(s) = \sum_{j=0}^{n-1} p_j(s_k) p_j(s) \\ &= \frac{1}{n} + \frac{2}{n} \sum_{j=1}^{n-1} \cos\left(j \left(\frac{2k-1}{2n} \pi\right)\right) \cos(j \cos^{-1}(s)). \end{aligned} \quad (3.30)$$

At $s = 0$, $\cos(j \cos^{-1}(0)) = \cos(j\pi/2)$, which is zero for odd j . Hence,

$$\begin{aligned} d_k(0) &= \frac{1}{n} + \frac{2}{n} \sum_{j=2, \text{even}}^{n-2} \cos\left(j \left(\frac{2k-1}{2n} \pi\right)\right) (-1)^{j/2} \\ &= \frac{1}{n} + \frac{2}{n} \sum_{j'=1}^{n/2-1} (-1)^{j'} \cos\left(\pi j' \frac{2k-1}{n}\right). \end{aligned} \quad (3.31)$$

The sum can be evaluated exactly (we used Mathematica), yielding

$$d_k(0) = \frac{1}{n} - \frac{2}{n} \left(\frac{1 - \cos((k+n/2)\pi) \tan(\pi \frac{2k-1}{2n})}{2} \right) \quad (3.32)$$

$$= \frac{1}{n} - \frac{1}{n} \left(1 - (-1)^{k+n/2} \tan\left(\frac{2k-1}{2n} \pi\right) \right) \quad (3.33)$$

$$= \frac{1}{n} (-1)^{k+n/2} \tan\left(\frac{2k-1}{2n} \pi\right). \quad (3.34)$$

With coefficients in hand, we now compute the one norm of $d(0)$.

$$\|d(0)\|_1 = \frac{1}{n} \sum_{k=1}^n \left| \tan\left(\frac{2k-1}{2n} \pi\right) \right| \quad (3.35)$$

We have a reflectional symmetry about $k \rightarrow n-k+1$, allowing us to cut the sum in half and remove the absolute value sign.

$$\begin{aligned} \|d(0)\|_1 &= \frac{2}{n} \sum_{k=1}^{n/2} \tan\left(\frac{2k-1}{2n} \pi\right) \\ &= \frac{1}{m} \sum_{k=1}^m \tan\left(\frac{2k-1}{2m} \frac{\pi}{2}\right) \end{aligned} \quad (3.36)$$

Here, $m \equiv n/2$. We observe that the sum increases as k approaches m due to the first order pole at $\pi/2$. We can upper bound $\tan(x)$, and therefore the sum above, as follows.

$$\begin{aligned} \frac{1}{m} \sum_{k=1}^m \tan\left(\frac{2k-1}{2m} \frac{\pi}{2}\right) &\leq \frac{1}{m} \sum_{k=1}^m \frac{1}{\frac{\pi}{2} - \frac{\pi}{2} \left(\frac{2k-1}{2m}\right)} = \frac{4}{\pi} \sum_{k=1}^m \frac{1}{2(m-k)+1} \\ &= \frac{4}{\pi} \sum_{j=1}^m \frac{1}{2j-1} \end{aligned} \quad (3.37)$$

In the last line, we reindexed by $j = m - k + 1$. Borrowing the reasoning from the prior lemma,

$$\frac{4}{\pi} \sum_{j=1}^m \frac{1}{2j-1} < \frac{2}{\pi} (\gamma + \log(2n+2)). \quad (3.38)$$

Tracing back, this is an upper bound on $\|d(0)\|_1$. Hence,

$$\|d(0)\|_1 < \frac{2}{\pi} \log(n+1) + \frac{2(\gamma + \log(2))}{\pi} < \frac{2}{\pi} \log(n+1) + 1. \quad (3.39)$$

□

The benefit of well-conditioning comes from relaxing the need to have exquisitely precise data to achieve good interpolations. This property is captured by the following Theorem.

Theorem 3.3.3. *Let $y = (f(s_1), f(s_2), \dots, f(s_n))^T$, and let $\tilde{y} \in \mathbb{R}^n$ be an approximation of y in the sense that, for all $1 \leq j \leq n$, $|f(s_j) - \tilde{y}_j| \leq \epsilon / (\frac{2}{\pi} \log(n+1) + 1)$ with probability at least $1 - \delta/n$. Let $p(s) = (p_0(s), \dots, p_{n-1}(s))^T$ be the vector of orthonormal Chebyshev polynomials. Then $\tilde{y}^T \mathbf{V} p(s)$ is an estimate of the interpolant $P_{n-1} f(s)$ at $s = 0$ to precision*

$$|P_{n-1} f(0) - \tilde{y}^T \mathbf{V} p(0)| \leq \epsilon$$

with probability at least $1 - \delta$.

Proof. First, observe that $P_{n-1} f(s) = p(s)^T c = p(s)^T \mathbf{V}^T y$ by the discussion surrounding (3.22).

Hence,

$$|P_{n-1} f(0) - p(0)^T \mathbf{V}^T \tilde{y}| = |(\mathbf{V} p(0))^T (y - \tilde{y})|. \quad (3.40)$$

By Hölder's inequality,

$$|(\mathbf{V} p(0))^T (y - \tilde{y})| \leq \|\mathbf{V} p(0)\|_1 \|y - \tilde{y}\|_\infty. \quad (3.41)$$

From Lemma 3.3.2, and from the assumptions on the distance between y and \tilde{y} ,

$$\|\mathbf{V} p(s)\|_1 \|y - \tilde{y}\|_\infty \leq \left(\frac{2}{\pi} \log(n+1) + 1 \right) \frac{\epsilon}{\frac{2}{\pi} \log(n+1) + 1} = \epsilon \quad (3.42)$$

with probability $\Pr = (1 - \delta/n)^n$. In fact, since the probability of each component \tilde{y} exceeding the specified distance is δ/n , by the union bound the total probability of at least one component

exceeding this distance is less than $n \times (\delta/n) = \delta$. Thus, the inequality is satisfied with probability $\Pr \geq 1 - \delta$. This completes the proof. \square

Theorem 3.3.3 is what suggests that our interpolation approach may have the potential to achieve accuracy improvements without increasing costs compared to standard Trotter. It tells us that the error in Trotter data can be as large as the error of the final estimate up to a factor which is logarithmically small in the number of interpolation points, and therefore these data \tilde{y}_i can be computed "cheaply enough." Thus, Theorem 3.3.3 plays an important role in the proofs of Lemma 3.5.1, presented in Section 3.5.

3.4 The Effective Hamiltonian

Trotter formulas approximate $U(t)$ only in a neighborhood around $t = 0$; thus the standard procedure for product formula simulations, as described in Section 2.7.1, is to divide the simulation interval $[0, t]$ into r subintervals, such that each interval is sufficiently small that the Trotter approximation is valid. For the simple case of a uniform mesh of r subintervals, this becomes

$$S_{2k}(t/r)^r = U(t) + O(t^{2k+1}/r^{2k}) \quad (3.43)$$

where big O is understood as taking r large. However, it is simpler for our subsequent analysis to consider $s = 1/r$ as a "dimensionless step size," and instead think about $s \rightarrow 0$. In terms of s , we define

$$\tilde{U}_s(t) := S_{2k}(st)^{1/s} \quad (3.44)$$

as the approximate evolution operator for $s \neq 0$. The discontinuity at $s = 0$ in (3.44) may be filled by the exact evolution $\tilde{U}_0(t) := U(t)$. Though we defined s as a reciprocal integer, definition (3.44) suggests an extension to allow s to be real-valued. In fact, the resulting function \tilde{U}_s is smooth on a neighborhood of $s = 0$, a fact that will allow us to precisely characterize the interpolation error. For our purposes, we will only consider $|s| \leq 1$. When $1/s$ is not an integer, we may implement \tilde{U}_s using fractional queries [54] by splitting $1/s$ into integer and fractional parts.

$$1/s = r + f \quad (3.45)$$

Here, $r = \text{rnd}(1/s) \in \mathbb{Z}$ is $1/s$ rounded to the nearest integer, and $f \in [-1/2, 1/2]$. Finally, we note that \tilde{U}_s is an even function of s , which we will make use of to cut the number of interpolation points in half by reflecting across $s = 0$.

Prior work has demonstrated the value of considering the effective Trotter Hamiltonian in the analysis of Trotter formulas [139]. This approach is also helps us calculate high order derivatives of \tilde{U}_s as needed for our error bounds. We define an effective Hamiltonian

$$\tilde{H}_s := \frac{i}{st} \log S_{2k}(st) \quad (3.46)$$

so that

$$\tilde{U}_s(t) = e^{-i\tilde{H}_s t}. \quad (3.47)$$

Note that \tilde{H}_s depends on t as well, though this dependence will be left implicit. For the purposes of bounding the interpolation error, we require a bound on the norm of \tilde{H}_s . This is supplied by the following lemma.

Lemma 3.4.1. *In the notation introduced above, let s be chosen such that*

$$k(5/3)^k m \max_{l \in [1, m]} \|H_l\| |s| t \leq \pi/20.$$

Then the following bound on the derivatives of \tilde{H}_s with respect to s holds.

$$\|\partial_s^n \tilde{H}_s\| \leq 2t^{-1} n^n (e^2 k (5/3)^k m \max_{l \in [1, m]} \|H_l\| t)^{n+1}$$

Note that our bounds are uniformly worse for larger k , i.e., higher order ST formulas. Assuming that this is not an artifact of our mathematical treatment, this suggests low order formulas are unconditionally preferred over high order ones for interpolation. Numerical studies could help determine the true impact of higher order formulas on the interpolation procedure.

We conclude this section with the proof of the above Lemma, which will be essential to our subsequent error analysis. The upper bound will prove useful because, as we will see, the error in polynomial interpolation can be expressed using a formula akin to the Taylor remainder, which involves a high-order derivative.

Proof of Lemma 3.4.1. Recall the definition of the effective Hamiltonian (3.46), defined for $s \in \mathbb{R} \setminus \{0\}$ and for $s = 0$ by $\tilde{H}_0 := \lim_{s \rightarrow 0} \tilde{H}_s = H$. We will understand $\log S_{2k}(st)$ through a power series expansion about the identity.

$$\log S_{2k}(st) = \sum_{j=0}^{\infty} \frac{(-1)^j}{j+1} (S_{2k}(st) - I)^{j+1} \quad (3.48)$$

This series converges precisely when

$$\|S_{2k}(st) - I\| \leq 1. \quad (3.49)$$

Using the fundamental theorem of calculus, we can derive a suitable condition for convergence as a neighborhood about $s = 0$. The condition above implies

$$\left\| \int_0^{st} S_{2k}(x) dx \right\| \leq 1 \quad (3.50)$$

which is satisfied provided

$$|st| \max_{x \in [0, st]} \left\| \frac{d}{dx} S_{2k}(x) \right\| \leq 1. \quad (3.51)$$

Writing out $S_{2k}(x) = \prod_{l=1}^{N_k} \exp(-iH_{j_l} \tau_l x)$ where H_{j_l} is some Hamiltonian piece H_j indexed by l , the derivative can be upper bounded as

$$\begin{aligned} \max_{x \in [0, st]} \left\| \frac{d}{dx} S_{2k}(x) \right\| &\leq \sum_{l=1}^{N_k} \|H_{j_l}\| |\tau_l| \\ &\leq \max_j \|H_j\| \|\tau\|_1 \end{aligned} \quad (3.52)$$

where $\tau = (\tau_l)_{l=1}^{N_k}$ is the vector of ST coefficients, and in going to the second line we used a Hölder inequality. We have $\|\tau\|_1 \leq N_k \max_l |\tau_l|$, and from Appendix A of [137] we have

$$\max_l |\tau_l| \leq 2k/3^k. \quad (3.53)$$

Thus, the requirement for convergence of the logarithm becomes

$$\frac{4}{3} k (5/3)^{k-1} m |st| \max_j \|H_j\| \leq 1 \quad (3.54)$$

where we used the expression $N_k = (2m)5^{k-1}$ for the number of ST exponentials.

We now assume s is within the symmetric interval defined by (3.54), such that (3.48) is convergent. Since $\log S_{2k}(0) = 0$, $s = 0$ is a zero of order at least one. We want to absorb the diverging $1/s$ term and better understanding the leading dependence in s . To facilitate this, we write

$$\tilde{H}_s = -\frac{1}{it} \sum_{j=0}^{\infty} \frac{(-1)^j}{j+1} s^j \Delta S_{2k}(st)^{j+1} \quad (3.55)$$

where we defined

$$\Delta S_{2k}(st) := \frac{S_{2k}(st) - I}{s}. \quad (3.56)$$

Note that ΔS_{2k} is analytic in s , and is a finite difference around $s = 0$, such that

$$\lim_{s \rightarrow 0} \Delta S_{2k}(st) = -iHt. \quad (3.57)$$

Through the series expansion (3.55) we may bound derivatives of \tilde{H}_s via bounds on derivatives of ΔS_{2k} . We first obtain a power series of ΔS_{2k} by Taylor expanding every term in the product formula S_{2k} . Regrouping in powers of st , the result is

$$\Delta S_{2k}(st) = \sum_{j=1}^{\infty} \frac{s^{j-1} (-it)^j}{j!} \sum_J \binom{j}{j_1 \dots j_{N_k}} \prod_{l=1}^{N_k} (H_l \tau_l)^{j_l} \quad (3.58)$$

where the parenthetical symbol is the multinomial coefficient, and the sum \sum_J is over all values of $J = (j_1, \dots, j_{N_k})$ such that $\sum_k j_k = j$. The derivatives with respect to s are now easy to compute.

Using the fact that

$$\partial_s^n s^{j-1} = \frac{(j-1)!}{(j-1-n)!} s^{j-n-1} \quad (3.59)$$

for $j > n$ (and zero otherwise), we have

$$\begin{aligned} \partial_s^n \Delta S_{2k}(st) &= \sum_{j=n+1}^{\infty} \frac{s^{j-n-1} (-it)^j}{j!} \frac{(j-1)!}{(j-1-n)!} \sum_J \binom{j}{j_1 \dots j_{N_k}} \prod_{l=1}^{N_k} (H_l \tau_l)^{j_l} \\ \|\partial_s^n \Delta S_{2k}(st)\| &\leq \sum_{j=n+1}^{\infty} \frac{t^j}{(j-n-1)!} s^{j-n-1} (\tau_{\max} N_k \Lambda)^j \end{aligned} \quad (3.60)$$

where $\Lambda := \max_j \|H_j\|$ and $\tau_{\max} = \max_l |\tau_l|$. Factoring out powers of $n + 1$ and reindexing, we are left with the following bound on derivatives of ΔS_{2k} .

$$\|\partial_s^n \Delta S_{2k}(st)\| \leq (\tau_{\max} N_k \Lambda t)^{n+1} e^{s\tau_{\max} N_k \Lambda t} \quad (3.61)$$

This expression is quite elegant; it is as if we were taking $n + 1$ derivatives of the exponential e^{cs} with

$$\begin{aligned} c &:= \tau_{\max} N_k \Lambda t \\ &\leq k(5/3)^k m \Lambda t. \end{aligned} \quad (3.62)$$

Factors of c will occur frequently in what follows, so we find it convenient to adopt this symbol as shorthand.

We return to bounding the derivatives of powers of $\Delta S_{2k}(st)$ as in equation (3.55).

$$\partial_s^n [\Delta S_{2k}(st)^{j+1}] \quad (3.63)$$

We reduce this to the previous case by performing a multinomial expansion.

$$\partial_s^n \Delta S_{2k}(st)^{j+1} = \sum_N \binom{n}{n_0 \dots n_j} \prod_{l=0}^j \partial_s^{n_l} \Delta S_{2k}(st) \quad (3.64)$$

As usual, the capital letter N denotes the set of all nonnegative indices n_0, \dots, n_j summing to n .

Applying the triangle inequality and submultiplicativity, and employing the bound (3.61),

$$\begin{aligned} \|\partial_s^n \Delta S_{2k}(st)^{j+1}\| &\leq \sum_N \binom{n}{n_0 \dots n_j} \prod_{l=0}^j \|\partial_s^{n_l} \Delta S_{2k}(st)\| \\ &\leq \sum_N \binom{n}{n_0 \dots n_j} \prod_{l=0}^j c^{n_l+1} e^{cs} \\ &= e^{(j+1)cs} c^{n+j+1} \sum_N \binom{n}{n_0 \dots n_j}, \end{aligned} \quad (3.65)$$

where we've used the sum property of the n_l where appropriate. The remaining sum over the multinomial coefficient is given by $(j + 1)^n$. Hence,

$$\|\partial_s^n \Delta S_{2k}(st)^{j+1}\| \leq ((j + 1)c)^n (ce^{cs})^{j+1}. \quad (3.66)$$

Notice that, when $j = 0$, this is consistent with equation (3.61).

With result (3.66) in hand, we return to the power series (3.55). Differentiating term by term

$$\partial_s^n \tilde{H}_s = -\frac{1}{it} \sum_{j=0}^{\infty} \frac{(-1)^j}{j+1} \partial_s^n \left(s^j \Delta S_{2k}(st)^{j+1} \right) \quad (3.67)$$

and performing a binomial expansion for each term

$$\partial_s^n \left(s^j \Delta S_{2k}(st)^{j+1} \right) = \sum_{q=0}^n \binom{n}{q} (\partial_s^q s^j) \left(\partial_s^{n-q} \Delta S_{2k}(st)^{j+1} \right) \quad (3.68)$$

will allow us to apply our previous results. It will be helpful to consider two cases separately: $j \leq n$ and $j > n$. These regimes are somewhat qualitatively different, since the derivatives of s^j may or may not vanish depending on the number of derivatives. Focusing on the case $j \leq n$, we have

$$\partial_s^n \left(s^j \Delta S_{2k}(st)^{j+1} \right) = \sum_{q=0}^j \binom{n}{q} \frac{j!}{(j-q)!} s^{j-q} \left(\partial_s^{n-q} \Delta S_{2k}(st)^{j+1} \right). \quad (3.69)$$

Note that the sum runs only to j , not n . Taking a triangle inequality upper bound using (3.66), we may upper bound (3.69) as

$$\begin{aligned} & \sum_{q=0}^j \binom{n}{q} \frac{j!}{(j-q)!} s^{j-q} ((j+1)c)^{n-q} (ce^{cs})^{j+1} \\ &= (ce^{cs})^{j+1} \sum_{q=0}^j \binom{j}{q} \frac{n!}{(n-q)!} s^{j-q} ((j+1)c)^{n-q} \end{aligned} \quad (3.70)$$

where we have factored out terms not involving q from the sum, and manipulated the factorials for reasons which will be seen presently. Taking the upper bound $n!/(n-q)! < n^q$, and factoring out $n-j$ powers of $(j+1)c$, we may upper bound the above expression by

$$\begin{aligned} & (ce^{cs})^{j+1} ((j+1)c)^{n-j} \sum_{q=0}^j \binom{j}{q} n^q ((j+1)cs)^{j-q} \\ &= (ce^{cs})^{j+1} ((j+1)c)^{n-j} (n + (j+1)cs)^j. \end{aligned} \quad (3.71)$$

Thus, with some minor polishing, we may express the bound on (3.68) for $j \leq n$ as

$$\left\| \partial_s^n \left(s^j \Delta S_{2k}(st)^{j+1} \right) \right\| \leq e^{(j+1)cs} c^{n+1} (j+1)^n \left(\frac{n}{j+1} + cs \right)^j. \quad (3.72)$$

Now let's move on to the $j > n$ case. Here, there are not enough derivatives to kill off the s^j term, so the binomial sum in (3.69) will run from $q = 0$ to n .

$$\partial_s^n \left(s^j \Delta S_{2k}(st)^{j+1} \right) = \sum_{q=0}^n \binom{n}{q} \frac{j!}{(j-q)!} s^{j-q} \left(\partial_s^{n-q} \Delta S_{2k}(st)^{j+1} \right) \quad (3.73)$$

Similar to before, we use the bound (3.66), to obtain

$$\begin{aligned} \|\partial_s^n \left(s^j \Delta S_{2k}(st)^{j+1} \right)\| &\leq \sum_{q=0}^n \binom{n}{q} \frac{j!}{(j-q)!} s^{j-q} ((j+1)c)^{n-q} (ce^{cs})^{j+1} \\ &= (ce^{cs})^{j+1} s^{j-n} \sum_{q=0}^n \binom{n}{q} \frac{j!}{(j-q)!} ((j+1)cs)^{n-q}. \end{aligned} \quad (3.74)$$

Taking $j!/(j-q)! < j^q$, a simpler upper bound is given by

$$(ce^{cs})^{j+1} s^{j-n} \sum_{q=0}^n \binom{n}{q} j^q ((j+1)cs)^{n-q} = (ce^{cs})^{j+1} s^{j-n} (j + (j+1)cs)^n. \quad (3.75)$$

With some minor rearrangements, this gives the following upper bound for $j > n$.

$$\|\partial_s^n \left(s^j \Delta S_{2k}(st)^{j+1} \right)\| \leq e^{(j+1)cs} c^{n+1} (j+1)^n (cs)^{j-n} \left(\frac{j}{j+1} + cs \right)^n \quad (3.76)$$

With the bounds (3.72) and (3.76), we can return to bounding $\partial_s^n \tilde{H}_s$. Still separating the two cases $j \leq n$ and $j > n$, we can write

$$\begin{aligned} \|\partial_s^n \tilde{H}_s\|_t &\leq \sum_{j=0}^n \frac{1}{j+1} \|\partial_s^n \left(s^j \Delta S_{2k}(st)^{j+1} \right)\| + \sum_{j=n+1}^{\infty} \frac{1}{j+1} \|\partial_s^n \left(s^j \Delta S_{2k}(st)^{j+1} \right)\| \\ &= B_l + B_h \end{aligned} \quad (3.77)$$

where B_l and B_h refer to bounds on the "low" and "high" parts of the series. Employing the bounds from equations (3.72) and (3.76), we have

$$\begin{aligned} B_l &\leq \sum_{j=0}^n \frac{1}{j+1} e^{(j+1)cs} c^{n+1} (j+1)^n \left(\frac{n}{j+1} + cs \right)^j \\ &= c^{n+1} \sum_{j=0}^n e^{(j+1)cs} (j+1)^{n-1} \left(cs + \frac{n}{j+1} \right)^j \end{aligned} \quad (3.78)$$

and

$$\begin{aligned}
B_h &\leq \sum_{j=n+1}^{\infty} \frac{1}{j+1} e^{(j+1)cs} c^{n+1} (j+1)^n (cs)^{j-n} \left(\frac{j}{j+1} + cs \right)^n \\
&\leq c^{n+1} \sum_{j=n+1}^{\infty} e^{(j+1)cs} (j+1)^{n-1} (cs)^{j-n} (1+cs)^n \\
&= c^{n+1} (1+cs)^n \sum_{j=n+1}^{\infty} e^{(j+1)cs} (j+1)^{n-1} (cs)^{j-n}.
\end{aligned} \tag{3.79}$$

Let's begin by simplifying the bound on B_l . We will at this point make the assumption that s is sufficiently small such that $cs < 1$. This will necessarily factor into the cost later. This simplification yields

$$\begin{aligned}
B_l &\leq c^{n+1} \sum_{j=0}^n e^{j+1} (j+1)^{n-1} \left(1 + \frac{n}{j+1} \right)^j \\
&\leq c^{n+1} \sum_{j=0}^n e^{j+1} (j+1)^{n-1} e^n \\
&\leq \frac{1}{e} (e^2 c)^{n+1} \sum_{j=1}^{n+1} j^{n-1}.
\end{aligned} \tag{3.80}$$

The remaining sum can be bounded by $(n+1)^n$, hence,

$$B_l \leq \frac{(e^2(n+1)c)^{n+1}}{e(n+1)} \leq (e^2 c)^{n+1} n^n, \tag{3.81}$$

where the definition that $0^0 = 1$ handles the edge case. Let's turn our attention to B_h . We will start by reindexing so that the series begins at $j = 0$ in (3.79).

$$B_h \leq c^{n+1} (1+cs)^n \sum_{j=0}^{\infty} e^{(j+n+2)cs} (j+n+2)^{n-1} (cs)^{j+1} \tag{3.82}$$

$$= (ce^{cs})^{n+1} (1+cs)^n \sum_{j=0}^{\infty} (ce^{cs})^{j+1} (j+n+2)^{n-1} \tag{3.83}$$

The series converges if and only if

$$cse^{cs} < 1. \tag{3.84}$$

This condition is slightly stronger than the condition (3.54) that we need for convergence of the logarithm (3.48), and is equivalent to $cs < W(1) \approx 0.567$, where W is the principal branch of the

Lambert W function. Returning to (3.82), we have the bound

$$(j+n+2)^{n-1} = (n+2)^{n-1} \left(1 + \frac{j}{n+2}\right)^{n-1} \leq (n+2)^{n-1} e^j. \quad (3.85)$$

Thus, we have

$$\begin{aligned} B_h &\leq (ce^{cs})^{n+1} (1+cs)^n (n+2)^{n-1} \sum_{j=0}^{\infty} (ecse^{cs})^j \\ &= (ce^{cs})^{n+1} (1+cs)^n (n+2)^{n-1} \frac{1}{1-ecse^{cs}}. \end{aligned} \quad (3.86)$$

To be concrete, let's take $ecse^{cs} < 1/2$, which is implied by $cs < \pi/20$. Coupled with the inequality in (3.62), this condition can be met provided that

$$k(5/3)^k m \Lambda s t \leq \pi/20, \quad (3.87)$$

which is exactly the assumption of Lemma 3.4.1. This allows us to upper bound B_h further as

$$B_h \leq 2e^{\pi/20} (cs)^{n+1} (3e^{\pi/20}/2)^n (n+2)^{n-1} \leq 4(cs)^{n+1} (9/5)^n (n+2)^{n-1}. \quad (3.88)$$

Since $(n+2)^{n-1} \leq e^2 n^n / 2$ (using $0^0 := 1$ for the edge case $n = 0$), we have

$$B_h \leq 2e^2 (cs)^{n+1} (9/5)^n n^n. \quad (3.89)$$

Altogether, using $s \leq 1$

$$\begin{aligned} \|\partial_s^n \tilde{H}_s\|_t &\leq n^n (e^2 c)^{n+1} \left(1 + 2(9/5e^2)^n\right) \\ &\leq 2n^n (e^2 c)^{n+1}. \end{aligned} \quad (3.90)$$

The final result then follows from substituting for c and noting that the duration of each time step is at most $2k/3^{k-1}$ using the results of [138]. \square

3.5 Application to Dynamical Observables

We now consider the application of Chebyshev interpolation to estimate expectation values, a fundamental task in quantum computation. The setting is as follows: given a quantum state ρ and observable O , the expectation value is given by $\langle O \rangle = \text{Tr}(\rho O)$. We evolve our system according

to a $2k$ -th order ST formula \tilde{U}_s given by (3.44). The time evolved expectation values of interest is captured by the function

$$f(s) := \frac{\text{Tr}(\rho O_s(t))}{\|O\|} \quad (3.91)$$

where $O_s(t)$ is given by equation (3.5). We've normalized the expectation values by $\|O\|$ because the relative error is a useful and natural metric, and also the normalized operators may be block encoded for amplitude estimation. Alternatively, we simply restrict our attention to normalized observables with $\|O\| = 1$. The interpolation algorithm we propose can be summarized as follows.

1. Given Hamiltonian H , simulation time t , and tolerance ϵ for the estimate of $\langle O(t) \rangle / \|O\|$, choose the appropriate interpolation interval $[-a, a]$ and an even number n of Chebyshev nodes. We neglect the cost of this step. The error analysis we will perform subsequently will inform the choices of a and n .
2. Compute estimates \tilde{y}_i of the expectation values $\langle O_{s_i}(t) \rangle$ for each s_i with $i = 1, \dots, n/2$, to an accuracy depending on ϵ and n . We will assume this step is done with Iterative Quantum Amplitude Estimation (IQAE) [56], a recent approach to amplitude estimation that exhibits low quantum overhead. Our metric of cost is the number of H_j exponentials executed on a quantum circuit, where $H = \sum_j H_j$. Note that by symmetry, we need not compute \tilde{y}_i for $i > n/2$. We have $f(s_i) = f(s_{n-i+1})$ for all $i \in \{1, \dots, n\}$.
3. Perform the polynomial fit $\tilde{P}_{n-1}f$ through the points (s_i, \tilde{y}_i) using a Chebyshev expansion (3.17). Note that $\tilde{P}_{n-1}f$ will automatically be even. This fit is well-conditioned, and we neglect the cost of this step.
4. Evaluate the $\tilde{P}_{n-1}f(0)$ to be our estimate of $\langle O(t) \rangle$.

To summarize, one performs amplitude estimation to acquire the time evolved expectation value at each Chebyshev node, then performs a polynomial interpolation of the data. The estimate is the value at $s = 0$.

Given an even set of Chebyshev nodes $\{s_1, \dots, s_n\}$, and making use of Lemma 3.2.1, the interpolation error E_{n-1} assuming perfect data points is given by

$$|E_{n-1}(0)| \leq \frac{|\text{Tr} \rho \partial_s^n O_s(t)|}{\|O\| n!} \prod_{i=1}^n |s_i| \leq \max_{s \in [-a, a]} \frac{\|\partial_s^n O_s(t)\|}{\|O\|} \left(\frac{a}{2n}\right)^n. \quad (3.92)$$

With a suitable bound on $\partial_s^n O(t)$, we can provide an upper bound on the interpolation error at $s = 0$. This bound is provided by the following lemma. In what follows, it will be helpful to define the parameter

$$c := k(5/3)^k m \max_{l \in [1, m]} \|H_l\| t \quad (3.93)$$

for ease of notation. This parameter is proportional to the the Hamiltonian size and the "total Trotter time," meaning the sum of all the forward and backward time steps, in absolute value, for a $2k$ -th ST formula.

Lemma 3.5.1 (Extrapolation Error Bound for Time-Evolved Observables.). *Under the conditions of Lemma 3.4.1 ($ca \leq \pi/20$), the following bounds holds on the Trotterized evolution $O_s(t)$ with step parameter $s \in (0, a]$:*

1. *for $c > n$ we have that*

$$\frac{\|\partial_s^n O_s(t)\|}{\|O\|} < \left(c \sqrt{e^3(1 + \sqrt{8/\pi} e^2)}\right)^{2n}$$

which gives an interpolation error

$$|E_{n-1}(0)| < \left(129 \frac{c^2 a}{n}\right)^n.$$

2. *For $c \leq n$, we have*

$$\frac{\|\partial_s^n O_s(t)\|}{\|O\|} \leq \sqrt{\frac{2n}{\pi}} \left(\frac{e^4 c}{2}\right)^n n! e^{4ce^2 \sqrt{2/\pi}}$$

giving an interpolation error

$$|E_{n-1}(0)| \leq 2\sqrt{2n} (6ca)^n e^{24c}.$$

The proof of this lemma is a tedious exercise in repeated in the combinatorics of large derivatives and the triangle inequality, and is left to the end of this chapter. Note that once the derivative bound holds, the interpolation error bound follows immediately from Lemma 3.2.1.

One motivation for these bounds is deriving asymptotic expressions for the algorithmic complexity. The following theorem gives an asymptotic query complexity for the number N_{exp} of Trotter exponentials $\exp(-iH_j\tau)$.

Theorem 3.5.2. *Let $O(t) = U^\dagger(t)OU(t)$ be a time-evolved observable under a Hamiltonian $H = \sum_{l=1}^m H_l$ on n qubits, so that $U(t) = e^{-iHt}$. Suppose there exists a $\gamma \in \mathbb{R}_+$ such that O/γ can be block encoded via a unitary U_{enc} by a state $|G\rangle$ on a set of L auxiliary qubits. Let ρ be a quantum state on n qubits, and suppose $\gamma/\|O\| \in O(1)$. Then, the number of exponentials N_{exp} required to estimate $\text{Tr}(\rho O(t))/\|O\|$ to precision ϵ with confidence $1 - \delta$ using a $2k$ order Suzuki Trotter formula satisfies*

$$N_{\text{exp}} \in \tilde{O}\left(c \max\{c, \log(1/\epsilon)\} \epsilon^{-1} \log(1/\delta)\right).$$

Here, \tilde{O} is big- O with multiplicative terms suppressed which are logarithmically smaller in $1/\epsilon$ and c . Moreover, the number of auxiliary qubits needed is $O(L)$.

We give a sketch of the proof. Given a choice of interval $[-a, a]$ and (even) number of interpolation points n , we have from Lemma 3.3.1 that the number of exponentials to perform evolutions for all Chebyshev nodes goes as

$$O\left(\frac{n \log n}{a}\right). \tag{3.94}$$

However, this is not the total cost since these circuits need to be repeated to perform the appropriate measurement protocols. Since O can be block encoded, the expectation value can be obtained via an amplitude estimation protocol. By the well-conditioning of Lemma 3.3.3, each data point needs to be within ϵ of the exact Trotter value, up to a logarithmic factor in n . This robustness is why our result maintains a $\tilde{O}(\epsilon^{-1})$ scaling.

In our proof, we assume the IQAE protocol is used, requiring only a single qubit overhead. The fractional queries for the noninteger time step also require $O(1)$ overhead, meaning the total

overhead is $O(L)$ due to the block encoding. To relate n and a to the required precision ϵ , simulation time t and Hamiltonian H , Lemma 3.5.1 can be used. Thus, we can relate N_{exp} to these basic parameters. We carry out the formal proof in Section 3.8.

As advertised, we see there is a "near-Heisenberg" scaling of $1/\epsilon$, up to logarithmic factors. However, there is an unsavory quadratic scaling in the simulation time in cases without high accuracy demands. I believe this can be improved, because our approach forces us to have n scale linearly in T , which seems overly pessimistic. Finally, our results suggest the best performance for using low order formulas, since our bounds are strictly worse for increasing ST order k .

3.6 Numerical Demonstration

To support the theoretical findings of this chapter, we numerically emulate the polynomial interpolation procedure on the 1D random Heisenberg model. I thank James Watson for his collaboration on these numerics. The Hamiltonian of interest is

$$H = \sum_{i=0}^{n-1} \sigma_i \cdot \sigma_{i+1} + h_i Z_i \quad (3.95)$$

where each $h_i \in [-h, h]$ is sampled randomly and uniformly, with $h > 0$ setting the "disorder strength." Moreover, $\sigma_i \cdot \sigma_{i+1} = X_i X_{i+1} + Y_i Y_{i+1} + Z_i Z_{i+1}$ is the dot product of the vector of Paulis $\sigma_i = (X_i, Y_i, Z_i)$. We take the chain to be a circle, so the n th qubit is adjacent to the 1st ($\sigma_{n+1} = \sigma_1$). We choose this system because it is simple, yet sufficiently interesting from the perspective of condensed matter physics, providing a model for the often-studied phenomenon of many-body localization and closed-system thermalization [25].

Although the product formula simulations are meant to be done on a quantum computer, here we instead perform all computations classically. Specifically, we compute the product of matrix exponentials for each product formula. Although not a true quantum simulation, this still provides accurate information about the Trotter error mitigated by the polynomial interpolation procedure.

To proceed, we first need to decide on a decomposition of the Hamiltonian (3.95), i.e. a partition of the various terms such that each partition is easy to simulate. To this end, it is helpful to observe that this system can be represented as a circular graph of n nodes, with links representing the hopping interaction $\sigma_i \cdot \sigma_{i+1}$. The interactions commute if the links don't meet at a vertex. Thus,

we might seek to *color* the edges of our graph, such that two edges of the same color don't meet. If n is even, only two colors are necessary for this, by coloring alternate edges. Taking n even for simplicity, we thus partition H as

$$H = H_{\text{even}} + H_{\text{odd}} + H_{\text{pot}} \quad (3.96)$$

where

$$\begin{aligned} H_{\text{even}} &= \sum_{i \text{ even}} \sigma_i \cdot \sigma_{i+1} \\ H_{\text{odd}} &= \sum_{i \text{ odd}} \sigma_i \cdot \sigma_{i+1} \\ H_{\text{pot}} &= \sum_i h_i Z_i. \end{aligned} \quad (3.97)$$

Within each partition, the terms commute, and therefore the exponential can be split without error. As a circuit, each term can be implemented in parallel. Exact methods for computing a two-qubit unitary can be used [129]. We use the decomposition (3.97) for all of the results in this Section.

Multiple different observables and initial states could be potentially considered. Here we choose the following:

$$\begin{aligned} O &= Z_{n-1} Z_0 X_1 \\ |\psi_0\rangle &= |1\rangle_{n/2} \end{aligned} \quad (3.98)$$

where $|1\rangle_{n/2}$ is the computational basis state on n qubits with a single 1 on qubit $n/2$. Intuitively, we imagine the 1 representing an excitation, and across the circle is the observable of interest O . The hopping terms will cause the 1 to "travel" to the other side, but the potential H_{pot} will cause more nontrivial behavior. To be concrete, we take the following parameter values in all of the data presented below: $T = 1$, $h = 1$, $n = 6$ and 2nd order Trotter.

We first wish to observe the error mitigation in action, as a proof of concept. To do so, need an "exact" estimate of $\langle O(t) \rangle$, meaning an estimate far more accurate than the simulation methods employed. We use direct matrix exponentiation with 20 digits of precision to accommodate this need, and all errors are measured with respect to this "exact" calculation. Next, we fix the

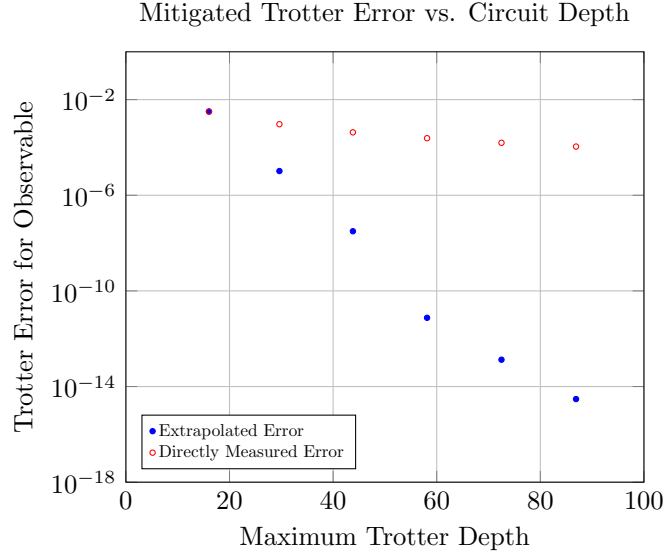


Figure 3.1 Additive error of the time-evolved expectation value $\langle O(T) \rangle$ plotted against maximum Trotter depth. The seven blue points represent seven distinct numbers of Chebyshev nodes N ranging from 2 to 14 in even increments. All data points are calculated to high precision using matrix multiplication, and thus the dominant error is due to Trotter. The orange denotes the best single data point gathered for the interpolation, representing the best estimate without classical post-processing. We see that the polynomial procedure, as the theory suggests, provides an exponential reduction in Trotter error with each additional data point. Meanwhile, the orange line appears to, as expected, follow only a polynomial trend.

interpolation interval $[-a, a]$, with $a = (\|H\|T)^{-1}$ a reasonable choice for the largest step size s . We then vary the number N of interpolation points, and for each N (from 2 to 14 in even increments) we perform the interpolation procedure and calculate the error. By comparing this error with the error of the "best" data point, i.e., the data point with the smallest Trotter step size, we can get a sense of how much the Trotter error is mitigated.

Figure 3.1 shows the results of this numerical experiment. Each blue point, going left to right, represents the Chebyshev interpolation estimate of $\langle O(t) \rangle$ with increasing number of interpolation points N , while the corresponding orange point is the best single data point. We see that the interpolation post-processing dramatically improves the accuracy of the expectation value calculation. The downward trend continues until flatlining where numerical roundoff takes over as the dominant error.

One respect in which the results of Figure 3.1 are limited is that data points will not be computed

to near perfection (or be limited by digital round off) even in the case of perfect time evolution. Expectation values must be estimated through a quantum measurement protocol which can achieve, at best, a Heisenberg limited scaling of $O(1/\epsilon_{\text{data}})$ for the number of operations needed to reach precision ϵ_{data} . And while Chebyshev interpolation provides robustness to data errors, a numerical demonstration of this is desirable.

To work within our classical setup, we model data imperfection as Gaussian noise of fixed width on top of the numerically computed value at each s . We change this noise parameter ϵ_{data} and observe the effect on the true error ϵ in the final interpolation estimate. Figure 3.2 gives the results for various degrees N of Chebyshev interpolation. We plot in terms of inverse errors, so that moving up along either axis corresponds to increased precision in the data or final estimate. We observe, for each N , two regimes: a linearly-sloped regime at low data precision and a plateau for sufficiently precise Trotter data. The first regime corresponds to the error in the final estimate being dominated by errors in the measurement itself, rather than the systematic Trotter error. In this regime, the Trotter error has been effectively mitigated, and only the data error remains. However, for each N there is a crossover point where the Trotter error can no longer be brought smaller than ϵ_{data} . At this point, further increases in data precision no longer improve the final estimate, because the dominant source is fundamentally the Trotter error. Unsurprisingly, larger N increases the mitigation of Trotter error, delaying this crossover point and improving the final estimate. Beyond verifying our interpolation approach, this graph illustrates how, for measurement data gathered on Trotter simulations, there is little value in achieving measurement accuracy beyond the crossover point. Absent hardware error, Trotter mitigation appears to remove barriers to higher accuracies through more precise data acquisition.

We haven't said much about the effects of simulation time, and we conclude with a brief investigation into this point. Figure 3.3 shows the error in interpolation calculations across four orders of magnitude for the simulation time. Other fundamental simulation parameters are held fixed, but following our definitions above, our interpolation interval $[-a, a]$ will change and hence the simulation cost. For all values of T considered, we see the exponential decay in error leading,

Total Error Scaling vs Measurement Error Scaling

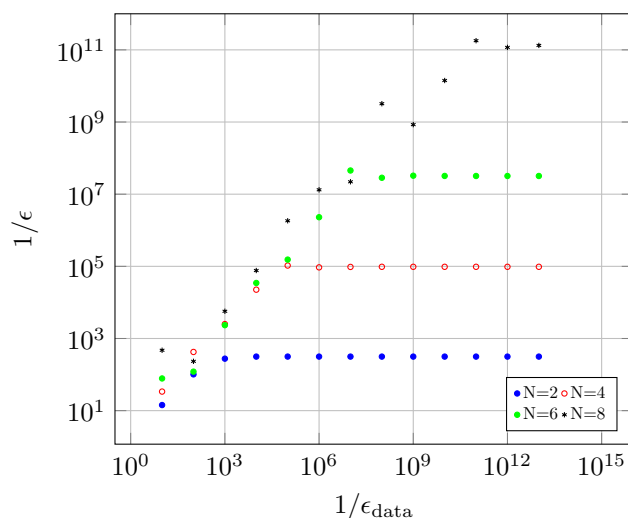


Figure 3.2 Error in the expectation value plotted against data noise for several different degrees $N - 1$ of Chebyshev interpolation. Going up along the y axis indicates improved performance. For small values of $1/\epsilon_{\text{data}}$ (large data errors), the final estimate error essentially tracks the data error, and the Trotter error is negligible. Once ϵ_{data} is brought under a certain threshold, the final error flatlines to some plateau, indicating that the final error is dominated by Trotter (interpolation) error. As N increases, the crossover point happens at smaller ϵ , and the final estimate is more accurate.

ultimately, to the plateau of machine precision. As T increases, the rate remains exponential but decreases in rate.

3.7 Discussion

In this chapter, we considered the mitigation of Trotter error by use of a standard numerical technique: polynomial interpolation. This approach is inspired by multiproduct formulas, which systematically cancel errors due to nonzero Trotter step. Here, however, we are cancelling the errors "offline", i.e. following the measurements of dynamical observables. This offers accuracy improvements without enormous quantum overhead, which is especially important as near-term quantum hardware is noisy and limited. Classical resources, though perhaps less powerful, are relatively abundant and cheap.

It is interesting to consider to what extent classical resources can chip away at the advantages of post-Trotter methods over product formulas. The fact that time evolution is only a piece of a full simulation has allowed us to achieve an exponential reduction in the Trotter error for measuring

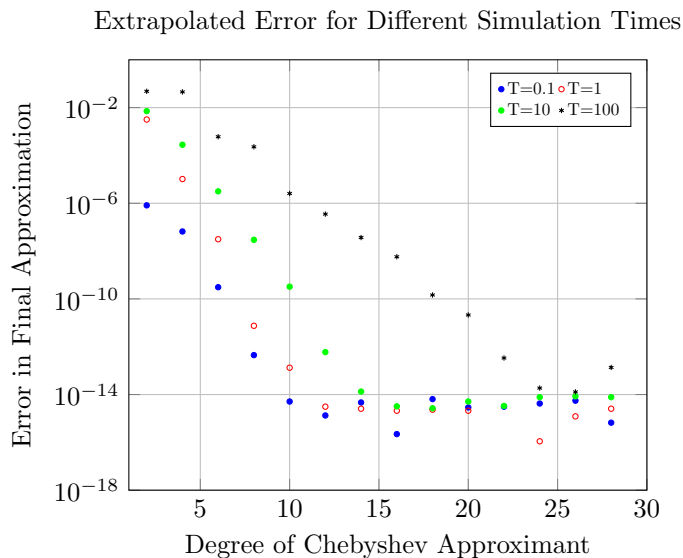


Figure 3.3 Chebyshev approximation error for the observable (3.98), plotted against Chebyshev degree for various total simulation times T . Across several orders of magnitude for T we observe an exponential decrease in the algorithmic error with respect to the number of data points until floating-point precision is reached. However, the decay is slower for larger times.

dynamical observables. A natural follow up question is whether classical techniques, coupled with first-order Trotter, can reduce the dependent on simulation time T to near-linear. An intuitive argument against this may be that such a scheme would require evolutions much less than T , and extrapolating the behavior to later times would be infeasible. This question is likely intimately related to the $O(T^2)$ dependence we've derived for our method. Numerical studies may ultimately shed light as to whether the T^2 scaling is "real" and whether it can be improved.

Polynomial interpolation is by no means the only approach to understanding the functional relation $f(s)$ between the Trotter step s and the quantity of interest. We could also apply Richardson extrapolation to f to estimate $f(0)$ given nearby points. Alternatively, rational approximations or machine learning techniques could be used to uniformly approximate $f(s)$. Extending our tests with the randomized Heisenberg model to include, say, Richardson extrapolation would be beneficial to facilitate this comparison.

3.8 Proofs

We now provide proofs for Lemma 3.5.1 and Theorem 3.5.2, whose statements were given in Section 3.5.

Proof of Lemma 3.5.1. For scalar functions $f(s)$, derivatives of $\exp(f(s))$ can be expressed through the complete Bell polynomials via Faà di Bruno's formula.

$$\partial_s^n e^{f(s)} = Y_n(f'(s), f''(s), \dots, f^{(n)}(s))e^{f(s)} \quad (3.99)$$

For operator exponentials such as $\exp(-i\tilde{H}_s t)$, derivatives can be expressed via repeated application of Duhamel's formula. Yet these expressions are always upper bounded by the commuting (scalar) case [133], so that

$$\|\partial_s^n e^{-i\tilde{H}_s t}\| \leq Y_n\left(t\|\partial_s \tilde{H}_s\|, t\|\partial_s^2 \tilde{H}_s\|, \dots, t\|\partial_s^n \tilde{H}_s\|\right). \quad (3.100)$$

Note that the exponential disappeared in the bound since it has norm one. Applying Lemma 3.4.1 and invoking the fact that Y_n is monotonic in each argument, this is upper bounded by

$$Y_n\left((2j^j (e^2 c)^{j+1})_{j=1}^n\right). \quad (3.101)$$

An explicit formula for this is given by

$$Y_n\left((2j^j (e^2 c)^{j+1})_{j=1}^n\right) = \sum_D \frac{n!}{d_1! \dots d_n!} \prod_{j=1}^n \left(\frac{2j^j (e^2 c)^{j+1}}{j!}\right)^{d_j} \quad (3.102)$$

where D is a sum over all indices $(d_j)_{j=1}^n$ such that $d_j \geq 0$ and

$$\sum_{j=1}^n d_j j = n. \quad (3.103)$$

Using a Stirling-type bound

$$\frac{1}{j!} \leq \left(\frac{e}{j}\right)^j \frac{1}{\sqrt{2\pi}} \quad (3.104)$$

allows us to write

$$\begin{aligned}
Y_n \left((2j^j (e^2 c)^{j+1})_{j=1}^n \right) &\leq \sum_D \frac{n!}{d_1! \dots d_n!} \prod_{j=1}^n \left(\sqrt{\frac{2}{\pi}} e^j (e^2 c)^{j+1} \right)^{d_j} \\
&= (e^3 c)^n \sum_D \frac{n!}{d_1! \dots d_n!} \prod_{j=1}^n \left(\sqrt{\frac{2}{\pi}} e^2 c \right)^{d_j} \\
&= (e^3 c)^n Y_n(\sqrt{2/\pi} e^2 c, \sqrt{2/\pi} e^2 c, \dots, \sqrt{2/\pi} e^2 c) \\
&= (e^3 c)^n B_n(\sqrt{2/\pi} e^2 c).
\end{aligned} \tag{3.105}$$

In the second line we brought out n factors of ec using the condition on the indices D , and we identified Y_n evaluated the same at every argument to be the single-variable Bell (or Touchard) polynomial B_n . We can bound the size of $B_n(\sqrt{2/\pi} e^2 c)$ [5] by

$$B_n(\sqrt{2/\pi} e^2 c) \leq \left(\frac{n}{\log(1 + \sqrt{\pi/2} n / (e^2 c))} \right)^n \tag{3.106}$$

for all $n > 0$, with $n = 0$ defined by the limit (which is 1). With this,

$$\|\partial_s^n e^{-i\tilde{H}_s t}\| \leq \left(\frac{e^3 cn}{\log(1 + \sqrt{\pi/2} n / (e^2 c))} \right)^n \leq \left(\frac{e^3 cn}{2} \right)^n \left(1 + \sqrt{\frac{8}{\pi}} \frac{e^2 c}{n} \right)^n \tag{3.107}$$

where we've used the bound $1/\log(1+x) \leq 1/2 + 1/x$. Again, this inequality is valid for $n = 0$ via the limit, which is always one.

With this bound on the ST formula derivatives, we now turn to bounding $\partial_s^n O_s(t)$. Applying the binomial theorem and triangle inequality to (3.5),

$$\begin{aligned}
\frac{\|\partial_s^n O_s(t)\|}{\|O\|} &\leq \sum_{p=0}^n \binom{n}{p} \|\partial_s^p e^{it\tilde{H}_s}\| \|\partial_s^{n-p} e^{-it\tilde{H}_s}\| \\
&\leq \left(\frac{e^3 c}{2} \right)^n \sum_{p=0}^n \binom{n}{p} p^p (n-p)^{n-p} \left(1 + \sqrt{\frac{8}{\pi}} \frac{e^2 c}{p} \right)^p \left(1 + \sqrt{\frac{8}{\pi}} \frac{e^2 c}{n-p} \right)^{n-p}.
\end{aligned} \tag{3.108}$$

At this point, it will be fruitful to consider two regimes. Recall that c encodes information about the simulation time.

In the case where $c > n$, we have

$$\begin{aligned}
\frac{\|\partial_s^n O_s\|}{\|O\|} &\leq \left(\frac{e^3 c}{2}\right)^n \sum_{p=0}^n \binom{n}{p} (c + \sqrt{8/\pi} e^2 c)^p (c + \sqrt{8/\pi} e^2 c)^{n-p} \\
&\leq \left(\frac{e^3 c}{2}\right)^n c^n \left(1 + \sqrt{8/\pi} e^2\right)^n \sum_{p=0}^n \binom{n}{p} \\
&= \left(c \sqrt{e^3 (1 + \sqrt{8/\pi} e^2)}\right)^{2n}.
\end{aligned} \tag{3.109}$$

This implies a relative error in the polynomial fit bounded by

$$|E_{n-1}(0)| < \left(129 \frac{c^2 a}{n}\right)^n. \tag{3.110}$$

In the case where $c \leq n$, the approximation

$$\left(1 + e^2 \sqrt{\frac{8c}{\pi p}}\right)^p < e^{ce^2 \sqrt{8/\pi}} \tag{3.111}$$

holds and is not so crude. Applying this to (3.108),

$$\frac{\|\partial_s^n O_s\|}{\|O\|} \leq \left(\frac{e^3 c}{2}\right)^n n! \sum_{p=0}^n \frac{p^p (n-p)^{n-p}}{p! (n-p)!} e^{4ce^2 \sqrt{2/\pi}}. \tag{3.112}$$

Regrouping and employing a Stirling bound where appropriate,

$$\begin{aligned}
\frac{\|\partial_s^n O_s\|}{\|O\|} &\leq e^{4ce^2 \sqrt{2/\pi}} \left(\frac{e^3 c}{2}\right)^n n! \left(2 \frac{e^n}{\sqrt{2\pi n}} + \frac{e^n}{2\pi} \sum_{p=1}^{n-1} \frac{1}{\sqrt{p(n-p)}}\right) \\
&\leq e^{4ce^2 \sqrt{2/\pi}} \left(\frac{e^4 c}{2}\right)^n n! \left(\frac{2}{\sqrt{2\pi n}} + \frac{\sqrt{n-1}}{2\pi}\right) \\
&\leq \frac{1}{2\pi} (\sqrt{8\pi} + \sqrt{n-1}) \left(\frac{e^4 c}{2}\right)^n n! e^{4ce^2 \sqrt{2/\pi}} \\
&\leq \sqrt{\frac{2n}{\pi}} \left(\frac{e^4 c}{2}\right)^n n! e^{4ce^2 \sqrt{2/\pi}}.
\end{aligned} \tag{3.113}$$

After another Stirling bound, this gives a corresponding interpolation error of

$$E_{n-1}(0) < 2\sqrt{2n} (6ca)^n e^{24c}. \tag{3.114}$$

□

Proof of Theorem 3.5.2. Let $f(s) = \langle O_s(t) \rangle / \|O\|$ be the normalized expectation value under Trotter evolution. Our interpolation algorithm produces an estimate \tilde{f} of $f(0)$ which we require to be accurate within ϵ .

$$|f(0) - \tilde{f}| \leq \epsilon \quad (3.115)$$

There is the interpolation error from the polynomial $P_{n-1}f$ fitting f assuming perfect interpolation points $(s_i, f(s_i))$. But $f(s_i)$ can only be estimated; let's call \tilde{y}_i this estimate. The error in \tilde{y}_i in our analysis comes from the statistical error inherent in the estimation protocol as well as the error in the fractional query procedure for a $1/s$ evolution. We can independently consider the interpolation error and the data error via the triangle inequality.

$$\begin{aligned} |f(0) - \tilde{f}| &\leq |f(0) - P_{n-1}f(0)| + |P_{n-1}f(0) - \tilde{P}_{n-1}f(0)| \\ &\leq \epsilon_{\text{int}} + L_n \epsilon_{\text{data}} \end{aligned} \quad (3.116)$$

Here L_n is the Lebesgue constant of the interpolation, essentially a condition number, and ϵ_{data} is an upper bound on the error in the data. $\tilde{P}_{n-1}f$ is the fit to the imperfect data and $P_{n-1}f$ the fit to the perfect data $(s_i, f(s_i))$. For generic interpolation nodes, L_n can grow rapidly; however, for the set of Chebyshev nodes we obtain a near-optimal value [114].

$$L_n \leq \frac{2}{\pi} \log(n+1) + 1 \quad (3.117)$$

Since we want the total error to be within a threshold ϵ , we can require

$$\epsilon_{\text{data}} \leq \frac{\epsilon}{2L_n}, \quad \epsilon_{\text{int}} \leq \frac{\epsilon}{2}. \quad (3.118)$$

Given these error bounds, we can now turn to the cost of acquiring the data points. Because O/γ can be block encoded, the expectation value calculation can be encoded as an amplitude estimation problem. Specifically, a Hadamard test circuit gives the amplitude

$$\frac{1 + \langle O_{s_i}(t) \rangle / \gamma}{2}. \quad (3.119)$$

If we estimate this amplitude to within accuracy $\epsilon_{\text{data}} \|O\| / 2\gamma$, we can estimate $f(s_i)$ within ϵ_{data} . Using Iterative Quantum Amplitude Estimation [56], we can obtain this estimate using a Grover

iterate G constructed from two Hadamard test oracles. The number of Grover oracles N_G required is given by

$$N_G \leq \frac{200\gamma L_n}{\|O\|\epsilon_{\text{data}}} \log \left(\frac{2n}{\delta} \log_2 \left(\frac{\gamma L_n \pi}{\|O\|\epsilon_{\text{data}}} \right) \right) \quad (3.120)$$

to ensure probability $1 - \delta$ of all data being within ϵ_{data} of the true value. Each G requires two Hadamard tests, and each Hadamard oracle calls a (controlled) ST evolution once. The number of controlled exponentials needed for a single data point at value s_i is in

$$O \left(\frac{N_k}{|s_i|} \log 1/\epsilon_{\text{data}} \right) \quad (3.121)$$

where $N_k = 2m5^{k-1}$, and where the logarithm comes from the need for fractional queries with QSVT. There is also a $O(1)$ overhead associated with the fractional queries and IQAE. Altogether, the number of exponentials for a single data point is in

$$O \left(N_G \times 2 \times \frac{N_k}{|s_i|} \log 1/\epsilon_{\text{data}} \right). \quad (3.122)$$

Therefore, the total number N_{exp} of generating all $n/2$ data points (we only need half due to symmetry) is in

$$N_{\text{exp}} \in O \left(N_G N_k \sum_{i=1}^{n/2} \frac{1}{s_i} \log(1/\epsilon_{\text{data}}) \right). \quad (3.123)$$

Plugging in (3.120) for N_G above and summing over $1/s_i$ using Lemma 3.3.1,

$$\begin{aligned} N_{\text{exp}} &\in O \left(\frac{\gamma N_k L_n n}{\|O\|\epsilon a} (\log n) \log \left(\frac{2n}{\delta} \log_2 \left(\frac{\gamma L_n \pi}{\|O\|\epsilon} \right) \right) \log(1/\epsilon_{\text{data}}) \right) \\ &\subset \tilde{O} \left(\frac{n}{a\epsilon} \log 1/\delta \right) \end{aligned} \quad (3.124)$$

where \tilde{O} suppresses factors logarithmic in n and ϵ . We also employed our assumption that $\gamma/\|O\| \in O(1)$. The number of nodes n and the interpolation interval $[-a, a]$ will be determined by ϵ_{int} , the interpolation error assuming perfect data. To apply our error bounds from the previous subsection, choose a to satisfy Lemma 3.4.1, i.e. $ca < \pi/20$, while also taking $1/a \in O(c)$.

Choose $n \geq \lceil c \rceil$. Then the second bound of Lemma 3.5.1 holds. From the interpolation error, we must satisfy

$$2\sqrt{2}n(6ca)^n e^{24c} < \epsilon/2 \quad (3.125)$$

which in turn can be satisfied provided that

$$4\sqrt{2}n \left(6e^{24}ca\right)^n < \epsilon \quad (3.126)$$

since $n \geq c$. Choose a such that $6e^{24}ca = 1/2$, which is consistent with our previous conditions on a . Then, to satisfy the error bound, n can satisfy

$$4\sqrt{2}n2^{-n} < \epsilon. \quad (3.127)$$

This can be solved using the -1 branch of the LambertW function.

$$n > -\frac{\text{LambertW}_{-1}\left(-\epsilon \log 2 / (4\sqrt{2})\right)}{\log 2} \quad (3.128)$$

The appropriate asymptotics is $n \in O(\log(1/\epsilon))$. By taking $n = n_*$ where

$$n_* = \max \left\{ \lceil c \rceil, \left\lceil -\frac{\text{LambertW}_{-1}(-\epsilon \log 2 / (4\sqrt{2}))}{\log 2} \right\rceil \right\} \quad (3.129)$$

$$\in O(\max\{c, \log(1/\epsilon)\}) \quad (3.130)$$

we satisfy all required constraints and arrive at our final asymptotic scaling.

$$N_{\text{exp}} \in \tilde{O}\left(\max\{c, \log(1/\epsilon)\}c\epsilon^{-1} \log(1/\delta)\right) \quad (3.131)$$

□

In the proof above, we set $n > c$ from the beginning, in order to use the second of the two bounds from Lemma 3.5.1. Together with $1/a \in O(c)$ this condemns us to a suboptimal c^2 scaling in the large c limit. However, using the first bound instead of the second would not help us, since the nc^2/a term in that bound must be order one.

CHAPTER 4

TIME DEPENDENT HAMILTONIAN SIMULATION THROUGH DISCRETE CLOCK CONSTRUCTIONS

The previous chapter was entirely concerned with time independent Hamiltonians. However, this is only a special (though important) case of general time dependent Hamiltonians that can occur in systems of interest. Interestingly, a reduction from time dependence to time independence is always possible, though this has not been properly utilized in the Hamiltonian simulation community.

This chapter concerns itself with a discretized version of the reduction, known as the (t, t') method, that is finite dimensional and therefore amenable to computation. We will consider, as an application of our formalism, a simulation by qubitization of the encoded time dependent Hamiltonian. After a brief introduction and motivation for time dependent simulations, we review the standard (t, t') formalism. Then we discretize the clock variable suitably and prove asymptotic error bounds on the accuracy compared to the time ordered operation. Finally, we describe a possible simulation by qubitization of the full clock system, before ending with some discussion.

The subsequent Chapter 5 makes use of the clock space construction to argue in favor of the conjecture that certain multiproduct formulas serve as good approximants to U . However, in that chapter, the clock space is only used for a theoretical proof, and plays no role in the simulation. Here, we take the clock space quite seriously and consider what it would take to fully implement it using the powerful method of qubitization.

Both chapter and the next are subjects of ongoing research. An early preprint has been posted [133] which will be updated as these projects reach a close.

4.1 Introduction and Motivation

When a fundamental physical law is expressed as a Hamiltonian, it should generally be expected to be time independent. This conveys the expected constancy of the theory under consideration. Such expectations are validated, for example, by experiments which seek to measure time variations in fundamental physical constants, such as \hbar [127]. To date, no such variations have been measured, though we must remember there is no theoretical reason, beyond elegance, to disallow them.

In any case, there are many *practical* situations in which a quantum system is naturally modeled as closed (hence unitary), while still exhibiting time-varying laws. For example, we may imagine impinging a molecule with electromagnetic pulses in a laboratory setting. These pulses may be considered large enough to be unaffected by the state of the molecule, yet the molecule is certainly influenced by the pulse train. Since the amplitude of this pulse may vary with time, so will the Hamiltonian describing the system. If the state of the quantum system *did* have an effect on the incoming electrical pulse, a distinct formalism of open system dynamics should be invoked. Even if the dynamics aren't inherently time-varying, it is often useful in the mathematics to shift to an interaction picture. When a Hamiltonian $H = H_0 + H_1$ has two pieces as shown, where H_0 is some baseline, "trivial" Hamiltonian that we know how to solve, moving into a frame of reference "rotating" with respect to H_0 generates a new time dependent $H_I(t)$ that encapsulates the nontrivial part. Such as splitting in H is common in perturbation theory, where we imagine H_1 small yet important. Thus, the study of time independent H can benefit from understanding how to simulate time dependent Hamiltonians.

As often occurs in physics, symmetries (in this case related to time translation invariance) lead to useful simplifications. We have seen in Section 2.5 how for time independent H , the time evolution operator $U(t) = e^{-iHt}$ is a simple matrix exponential, rather than the more nuanced time-ordered exponential of equation (2.18). Moreover, the existence of stationary states in the time independent setting provides a useful characterization of the "allowed energies" of a system, and a set of states whose dynamics are trivial. Despite these simplification, the study of quantum systems with time independent Hamiltonians is sufficiently rich as to warrant its own focused study.

In Chapter 2, we provided examples of time independent Hamiltonian simulation being used in generic quantum algorithms applications, particularly linear systems solvers. There are also applications for time dependent Hamiltonian simulations, such as adiabatic evolution. Generally speaking, a process is 'adiabatic' if it involves the slow transformation of parameters from one value to another. Here the meaning of "slow" depends on the nature of the problem at hand. Specifying to Hamiltonians, an adiabatic evolution is a dynamical evolution under a Hamiltonian $H(t)$ that

changes slowly with time. A standard example is the linear adiabatic evolution

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_1 \quad (4.1)$$

which, from $t = 0$ to $t = T$, changes the Hamiltonian from H_0 to H_1 . By "slow" we mean T is much larger than the smallest gap $\min_t 1/\delta E(t)$ between two eigenenergies which overlap the state of interest. Interestingly, such adiabatic evolutions approximately preserve eigenstates, in the sense that, starting and initial state as an eigenstate $|E_0\rangle$ of H_0 , the final state evolved under $H(t)$ will be approximately $|E_1\rangle$, and eigenstate of H_1 . This is especially useful when trying to prepare ground states of a nontrivial Hamiltonian H_1 given a simpler one H_0 . While the adiabatic approach is valuable for physical applications [128, 131], it can also enable solutions to optimization problems when we think of the ground state energy as minimizing, or "optimizing," a function. One can imagine the ground state of H_1 exhibiting information related to an optimization problem, unrelated to a physics context. For example, take Quadratic Unconstrained Binary Optimization (QUBO), which seeks a binary vector $x \in \{-1, 1\}^n$ that minimizes

$$\sum_{ij} x_i Q_{ij} x_j \quad (4.2)$$

for some real symmetric matrix Q . The solution can be encoded as a computational basis state $|x\rangle$ that is the ground state of the Hamiltonian

$$H_Q = \sum_{ij} Z_i Q_{ij} Z_j. \quad (4.3)$$

While this highlights the utility of ground state preparation for general optimization, recent literature points to other methods, such as Quantum Imaginary Time Evolution, as preferred for these sorts of tasks as opposed to adiabatic preparation [6, 142]. Nevertheless, it highlights how time dependent simulations can manifest in broader algorithmic settings.

Because time independent Hamiltonians are special cases of time dependent ones, any algorithms for general $H(t)$, and analysis thereof, immediately specializes to an algorithm and analysis for time independent case. While true, in practice our grasp of time independent simulation algorithms outstrips knowledge of the time dependent case. For example, qubitization is only viable for

simulating time independent Hamiltonians, as its basis of operation is a polynomial approximation to $f(\lambda) = e^{-i\lambda T}$ on the interval $[0, T]$ [88]. One could proceed by approximating $U(T, 0)$ via evaluating the expression (2.18) for fixed, large n . However, the resulting algorithm is, in general, quite inferior to its time independent version, since the errors from the truncation wash out any gains in precision. As of writing, there is no time dependent simulation algorithm which saturates known lower bounds.

Product formulas can be used in a similar manner, applying the formula to each term in the truncation. This is perhaps the simplest approach to time dependent simulation on a quantum computer. Because largely fluctuating Hamiltonians will require finer time meshes, the final cost will depend on the smoothness of the Hamiltonian and the size of derivatives within the mesh points [137]. More general time dependencies can be handled, without smoothness requirements, by a certain generalization of the Trotter scheme in which the time integrals (without time ordering) are retained [105]. The dropping of the smoothness requirement demonstrates that a much broader class of Hamiltonians is feasibly simulatable. Within their arguments, the authors use a generalization of product formulas in which the time ordered exponential is split, but the integrals are retained [70]. In this chapter, we will consider an entirely distinct generalization of the product formula in the time dependent setting. In fact, we will show that it relates to a standard product formula in an augmented clock space.

4.2 The Clock Space

Mathematically, and more broadly than the Hamiltonian setting, the distinction between time dependent vs independent systems can be cast as a distinction between autonomous and nonautonomous dynamical systems. Dynamical are differential equations in a single evolution parameter t , which can always be cast as a first-order initial value problem

$$\dot{x} = f(x, t), \quad x(0) = x_0 \tag{4.4}$$

possibly by standard reduction-of-order techniques. Here $x \in \mathbb{R}^n$ consists of n evolution parameters which implicitly depend on time t . Reasonable smoothness conditions on f may be imposed. The autonomous case corresponds to f being independent of t . Such equations are valuable because they

admit a geometric description in terms of phase space and flows. Classically, time independent Hamiltonian dynamics are a special case of autonomous systems, and the relevant initial value problem is given by *Hamilton's equations*.

$$(\dot{q}, \dot{p}) = \left(\frac{\partial H}{\partial p}, -\frac{\partial H}{\partial q} \right), \quad (q(0), p(0)) = (q_0, p_0) \quad (4.5)$$

This system is autonomous when H is independent of the evolution parameter t .

It has long been recognized that a simple transformation allows for the reduction of nonautonomous systems to autonomous ones [60]. The trick is to promote t to a coordinate, thereby making $f(x, t)$ satisfy the requirement of only depending on coordinates. Letting s take the place of the evolution parameter (time), we still want t and s to be essentially the same. This is supplied by the simple equation

$$\dot{t} = \frac{dt}{ds} = 1, \quad t(0) = 0. \quad (4.6)$$

With this, we have the following autonomous system

$$(\dot{x}, \dot{t}) = (f(x, t), 1), \quad (x(0), t(0)) = (x_0, 0) \quad (4.7)$$

whose solution encodes the solution to the original (4.4). While it seems that not much has been gained, this framework finds much application in the consideration of periodically driven systems. In this case, a cylindrical phase space, with t representing the angle, allows for interesting geometric understanding of the dynamics.

When a classical Hamiltonian $H(t)$ is time dependent (nonautonomous), can we perform a similar trick to reduce to the time independent (autonomous) case? Promoting t to a coordinate, we must formally introduce a conjugate momentum $-E$, choosing notation suggestive of the time-energy correspondence. We call the full Hamiltonian $K(q, p; t, E)$, to distinguish from the original Hamiltonian $H(q, p, t)$, and we wish to determine if a suitable K exists. As before, we want $dt/ds = 1$, which because of Hamilton's equations imply

$$\frac{\partial H}{\partial(-E)} = 1. \quad (4.8)$$

We conclude that $K = -E + F(q, p, t)$ for some F . But we need K to reproduce the same dynamics for q, p as H . This implies $F = H$. Interestingly, the final Hamilton equation

$$\frac{dE}{ds} = \frac{\partial H}{\partial t} \quad (4.9)$$

corresponds with the expected equation for energy change, up to a sign. This explains our choice of a minus sign in the conjugate momentum $-E$.

In summary, we have a prescription for converting nonautonomous Hamiltonians into autonomous ones depending only on coordinates. In hindsight, the way this is accomplished is rather silly. The coordinate t has dynamics completely independent of the values of any other coordinate or momentum, including E , and gets pulled in a straight line at a constant velocity. Time marches forward. As t changes, so does H , which affects all of the other coordinates in the desired way. This is much like a reel of movie tape moving through the machine at a constant rate to change what's on the screen, mimicking the forward flow of time.

What about *quantum* Hamiltonians? We could try to quantize the above, imposing the usual commutation relation

$$[\hat{t}, -\hat{E}] = iI. \quad (4.10)$$

Absent periodicity, a natural choice of Hilbert space of t is $L^2([0, T])$: square integrable functions on the interval of simulation. We then get a representation of t and E as t multiplication and t -derivatives, respectively.

$$(\hat{t}\psi)(t) = t\psi(t), \quad (\hat{E}\psi)(t) = i\partial_t\psi \quad (4.11)$$

The augmented "clock" Hamiltonian, so named because of the time coordinate t , has the form

$$H_c = H - i\partial_t \quad (4.12)$$

which looks eerily similar to a rearranged Schrödinger operator. Indeed, if $\psi_{\text{sol}}(t, q)$ is a solution to the Schrödinger equation encoded on the full "clock" Hilbert space, then we see that the state

$$\psi_\alpha(t, q) = e^{-i\alpha t}\psi_{\text{sol}}, \quad \alpha \in \mathbb{R} \quad (4.13)$$

is formally an eigenstate of H_c with eigenvalue α . As a caution, this state is not properly normalized, nor is it normalizable unless the simulation interval $[0, T]$ is finite.

The above manipulations are purely formal, helpful only for calculational or interpretational purposes. From a physical point of view, H_c is not bounded from below, allowing for potentially infinite energy extraction if such a system existed and could be coupled with. In particular, this system transitions to lower energies as wavepackets travel faster and faster to the left of the clock space, a seemingly senseless possibility.

The quantum mechanical version of this trick, sometimes called the (t, t') -formalism because of the two distinct "times", finds use in periodically-driven quantum systems [21]. But our purposes, the elimination of explicit dependence on the evolution parameter is most exciting, because it implies the time evolution operator requires no time-ordering, while still encoding the full time dynamics [103]. Thus, while time independent H is a special case of time dependence, time dependent simulation H reduces, formally, to time independent simulation on an augmented space.

To understand the nature of the encoding better, we have to be somewhat more careful. Let's provide a more concrete description of the situation. The full Hilbert space is given by

$$\mathcal{H} = \mathcal{H}_s \otimes \mathcal{H}_c \quad (4.14)$$

where $\mathcal{H}_c \cong L^2(\mathcal{M})$, and \mathcal{M} is the (connected) one-dimensional smooth manifold representing t . We have considered $\mathcal{M} \cong [0, T]$ here, but we might also consider a circle (periodic dynamics) or the real line, where translations are a bit more natural. On \mathcal{H}_c , E acts as a generator of translations, but is an unbounded operator. Nevertheless, the exponentials of E above are well defined through the spectral theorem and functional calculus for unbounded operators [61].

States $\psi \in \mathcal{H}$ can then be expressed as certain functions on \mathcal{M} whose value $\psi(t)$ is a state on \mathcal{H}_s . The inner product on \mathcal{H} is the natural one.

$$\langle \phi | \psi \rangle := \int_{\mathcal{M}} \langle \phi(t) | \psi(t) \rangle_s dt \quad (4.15)$$

Here, $\langle \cdot | \cdot \rangle_s$ denotes the inner product on \mathcal{H}_s . From now on, we use τ to denote the evolution parameter in order to avoid confusion with the clock coordinate t .

The interval I over which the dynamics of the quantum system take place needs to be embedded within \mathcal{M} . If I is not exactly \mathcal{M} , then the definition a time dependent observable $A(t)$ will need to be extended to cover the entire clock space. Once done, $A(t)$ is promoted to an (time independent) observable on \mathcal{H} , denoted \mathbf{A} , by acting on $\psi \in \mathcal{H}$ in a manner corresponding with the original space.

$$(\mathbf{A}\psi)(t) := A(t)\psi(t) \quad (4.16)$$

All such operators \mathbf{A} are seen to be *local* in \mathcal{H}_c , since they act with simple multiplication.

Having laid the above groundwork, we can return to the question of dynamics. Let \mathbf{H} be the promoted Hamiltonian operator as discussed in the previous paragraph. Let $\mathbf{U}(\tau)$ be the unitary operator given by

$$\mathbf{U}(\tau) = e^{iE\tau} e^{-i(\mathbf{H}-E)\tau}. \quad (4.17)$$

One can verify that \mathbf{U} solves the following Schrödinger equation.

$$\begin{aligned} i\partial_\tau \mathbf{U}(\tau) &= \mathbf{H}(\tau)\mathbf{U}(\tau) \\ \mathbf{U}(0) &= I \end{aligned} \quad (4.18)$$

Here,

$$\mathbf{H}(\tau) \equiv e^{iE\tau} \mathbf{H} e^{-iE\tau} \quad (4.19)$$

is a τ -dependent Hamiltonian corresponding to simple, uniform translation along the clock space.

For any state $\Psi_0 \in \mathcal{H}$, the function

$$\Psi(\tau) := \mathbf{U}(\tau)\Psi_0 \quad (4.20)$$

solves the Schrödinger equation generated by $\mathbf{H}(\tau)$, but more importantly, it encodes solutions to the dynamics under $H(t)$. Indeed, for any $t \in \mathcal{M}$, we have a state $\psi(\tau, t) \in \mathcal{H}_s$ defined by

$$\psi(\tau, t) := [\Psi(\tau)](t) = [\mathbf{U}(\tau)\Psi_0](t) \quad (4.21)$$

which solves the Schrödinger equation of interest.

$$\begin{aligned}
i\partial_\tau\psi(\tau, t) &= i\partial_\tau[\mathbf{U}(\tau)\Psi_0](t) \\
&= [\mathbf{H}(\tau)\Psi_0](t) \\
&= H(\tau + t)\psi(\tau, t)
\end{aligned} \tag{4.22}$$

The interpretation is that each t constitutes an initial time for performing the simulation, so we have a family of solutions parametrized by t with initial state $\psi(0, t)$. The evolution parameter acts, as expected, as the total time elapsed in the simulation.

Finally, we can obtain a collection of induced time-evolution operators, $U(t' + \Delta t, t')$ on \mathcal{H}_s for each $t' \in \mathcal{M}$. It acts on states $\psi_0 \in \mathcal{H}_s$ as follows

$$U(t + \tau, t)\psi_0 = [\mathbf{U}(\tau)\Psi_0](t) \tag{4.23}$$

where $\Psi_0 \in \mathcal{H}$ is any state for which $\Psi_0(t) = \psi_0$. This operator is unitary and solves the Schrödinger equation in the usual sense. Therefore it is exactly equivalent to the time-ordered exponential of expression (2.17).

We've shown now that the clock space encodes a collection of solutions to the dynamics under $H(\tau)$, one for each initial time t with initial state $\Psi_0(t) \in \mathcal{H}_s$. However, one might only care about *one* solution, say, at $t = 0$, and the ability to extract that solution from the encoding. Suppose the desired initial state is $|\psi_0\rangle \in \mathcal{H}_s$, and our initial time is $t_0 \in \mathcal{M}$. The idea is to prepare an initial product state $\Psi_0 = \psi_0 \otimes \phi_0 \in \mathcal{H}$, where $\phi_0 \in \mathcal{H}_c$ has overlap $1 - \delta$ in a ϵ -neighborhood of t_0 . After performing the evolution under \mathbf{U} for the desired length τ , perform a measurement of t . The probability of measuring t within ϵ of $t_0 + \tau$ is $1 - \delta$. Moreover, provided that the variation of $H(t)$ around $t = t_0$ is small for variations of $t \pm \epsilon$, any value within the ϵ -neighborhood will suffice. Of course, any real measurement of t will have an uncertainty width, and this must be brought within the size of the variation of H . The actual state will be slightly mixed, but very close to pure.

Preparing the initial state $\psi_0 \otimes \phi_0$ is just as hard as preparing each separately, so we focus on ϕ_0 . One way to prepare a sharp peaked state is an initial t measurement of accuracy ϵ_{meas} according to the requirements above. If $H(t)$ can be shifted appropriately so that the measurement result

aligns with the desired start time, or the state can be shifted, then we've successfully prepared the desired state, and the clock Hamiltonian $\mathbf{H} - E$ can be turned on. However, no serious attempt towards an implementation on physical quantum hardware has been made. In terms of digital quantum computing, a proposal for simulating the clock system will be supplied in Section 4.4, which follows our construction of a discrete clock space.

To summarize, we have shown how the propagator U generated by a time dependent H can be cast as an ordinary operator exponential, via the inclusion of a 1-dimensional clock space. Interesting in its own right, this framework also allows for a natural unification of ideas regarding "Trotterization." This term is used to refer to both (a) the splitting up of an (ordinary) operator exponential of $H = \sum_j H_j$ into exponentials of the various H_j , or (b) the simulation of a time dependent Hamiltonian by time independent simulations over small time intervals, indicated in (2.18). These can, in fact be viewed as manifestations of the same phenomenon: a splitting of operator exponentials. To illustrate this with an example relevant to this paper, let's consider a simple symmetric Trotterization of equation (4.17).

$$\begin{aligned} U_2(t_0 + \Delta t, t_0) &\equiv e^{iE\Delta t} \left(e^{-iE\Delta t/2} e^{-iH(t_0)\Delta t} e^{-iE\Delta t/2} \right) \\ &= e^{-iH(t_0+\Delta t/2)\Delta t} \end{aligned} \quad (4.24)$$

We have just derived the *midpoint formula* [137, 124] from scratch. The Trotter product theorem says that

$$\lim_{k \rightarrow \infty} e^{iE\Delta t} \left(e^{-iE\Delta t/2k} e^{-iH(t_0)\Delta t/k} e^{-iE\Delta t/2k} \right)^k = U(t_0 + \Delta t, t_0) \quad (4.25)$$

even though E is unbounded [97]. Thus, we can expect that

$$U_2^{(k)}(t_0 + \Delta t, t_0) \equiv e^{iE\Delta t} \left(e^{-iE\Delta t/2k} e^{-iH(t_0)\Delta t/k} e^{-iE\Delta t/2k} \right)^k \quad (4.26)$$

constitutes a good approximation to U for sufficiently large $k \in \mathbb{Z}_+$ and small Δt .

This opens up the possibility of a more unified approach to Hamiltonian simulation algorithms, which has not yet been properly considered. For example, a natural generalization of product formulas to time dependent H could be a regular product formula of H_c on the enlarged space. For simplicity, let's take $H(t)$ in whole as a single term, so that $H_c = H - E$ has two terms. We can

define time dependent product formulas by taking product formulas along these two terms. This allows us to define, in particular time dependent generalizations of the recursive Suzuki-Trotter formulas of (2.28). Mathematical difficulties emerge in classifying the approximation order of these formulas, arising from the unboundedness of E . We seek to ameliorate this in our current work, focusing our attention on the 2nd order symmetric formula.

4.3 Finite Clock Spaces

There are several reasons we are motivated to consider a discretization of the clock space introduced in the previous section. First, any real computation performed using the clock space will require a finite number of states. A natural choice for discretization is along the time ("position") basis, and we will consider that here.

There are also formal reasons to consider a finite clock space. We've already seen how the clock space aids in the understanding of time dependent generalization of product formulas. Taking this idea further, we might consider time dependent *multiproduct* formulas, i.e. how to construct MPFs for time dependent problems. To ensure these MPFs work, we would like to show that an error series exists whose terms can be cancelled order by order through linear combinations of time dependent product formulas. However, such error series are more difficult to show on generic separable Hilbert spaces, and moreover the operator E is unbounded. By making the clock space finite, performing the relevant analysis, then taking the limit, we can potentially avoid these. This programme is described in more detail in Chapter 5, but a full proof of this is an ongoing research project.

Without further ado, we introduce our finite dimensional clock space, which we will sometimes call the "clock register." We discretize the clock variable t into $N_c = N_p \times N_q$ basis states, where $N_p \in \mathbb{Z}_+$ will represent the number of Trotter steps used in the simulation. Each "Trotter step" is further divided into $N_q \in \mathbb{Z}_+$ steps for reasons that will be discussed shortly. We label these orthonormal basis states $|j\rangle$ for $j \in [0, N_c - 1] \cap \mathbb{Z}$. We will find it useful to consider, for our purposes, only periodic Hamiltonians. This is natural to understand, since translation operators like E act most naturally on \mathbb{R} or the circle (periodic boundary conditions), and the circle

is bounded. Nonperiodic Hamiltonians can be accommodated by a simple reflection, defining $H(T + t) := H(T - t)$ for $t \in [0, T]$. In our work below, we will want $H(t)$ to be a differentiable bounded function within the grid points, and although the reflection introduces nonsmoothness, we can simply take one of the grid points to be the midpoint of simulation.

For simplicity, and for lack of a reason otherwise, we will take these grid points $(t_j)_{j=0}^{N_c-1}$ to be uniformly spaced over the interval $[0, T]$: $t_j = Tj/N_c$ (taking N_c to be an even integer, so that the midpoint requirement discussed directly above is satisfied). We let $\delta t := T/N_c$ denote the grid width. We also take the natural discretization of $H(t)$ onto the clock space

$$H(t) \mapsto \sum_{j=0}^{N_c-1} H_j \otimes |j\rangle\langle j| \equiv C(H) \quad (4.27)$$

where $H_j \equiv H(t_j)$. Observe that $C(H)$ has no dependence on the evolution parameter, i.e., it is autonomous. The notation $C(H)$ is used to suggest a controlled operation, where the control is on the clock register.

Choosing the appropriate discretization of E is somewhat more tricky, though the choice appears obvious in hindsight. Since E acts as a derivative, it makes sense to take the discretized version to be a finite difference operator. For example,

$$\Delta := -i \frac{U_+ - U_-}{2\delta t} \quad (4.28)$$

where U_+ is the shift operator defined by $U_+|j\rangle = |j + 1\rangle$ and $U_- = U_+^\dagger$ is the backwards shift (all increments taken mod N_c). This is the approach we ultimately take. However, we note that author of this dissertation, and collaborators, began by considering a distinct approach via the logarithm of the translation operator

$$\tilde{\Delta} = i \log U_+. \quad (4.29)$$

While apparently sensible, given the analogous relation between E and shifts on the clock space, this operator is not nicely behaved. For example, its commutator with the "position operator" $\sum_j t_j |j\rangle\langle j|$, rather than being near-identity, has long off-diagonal tails. This behavior may be of independent interest, but from now on we will concern ourselves with Δ as the discrete version of E .

With these choices, our full clock Hamiltonian becomes

$$H_c := C(H) - \Delta. \quad (4.30)$$

Already, we can show some reasonable properties carry over to this setting.

Lemma 4.3.1. *In the notation above, let $H : [0, T] \rightarrow \text{Herm}(\mathcal{H})$ be a time dependent Hamiltonian on a finite-dimensional vector space \mathcal{H} . Then*

$$[\Delta, C(H)] = i \text{Re} \left(U_+ \sum_j \frac{H_{j+1} - H_j}{\delta t} \otimes |j\rangle\langle j| \right)$$

where $\text{Re}(A) := (A + A^\dagger)/2$ denotes the Hermitian part of A . If H is differentiable in each subinterval with bounded derivative, then we further have

$$\|[\Delta, C(H)]\| \leq \max_{t \in [0, T]} \|\dot{H}(t)\|.$$

We remark here the connections to the canonical commutation relation $[f(x), p] = if'(x)$. The additional shift by U_+ is a relatively small deviation from a finite difference approximation being performed on the Hamiltonian.

Proof. We proceed in several steps, first by computing $[U_+, C(H)]$. We have

$$\begin{aligned} [U_+, C(H)] &= \sum_{j=0}^{N_c-1} H_j \otimes [U_+, |j\rangle\langle j|] \\ &= \sum_{j=0}^{N_c-1} H_j \otimes (|j+1\rangle\langle j| - |j\rangle\langle j-1|). \end{aligned} \quad (4.31)$$

By splitting the sum and reindexing (all increments modulo N_c), we can move the difference to the H_j , giving

$$\begin{aligned} [U_+, C(H)] &= \sum_j (H_j - H_{j+1}) \otimes |j+1\rangle\langle j| \\ &= -U_+ \sum_j (H_{j+1} - H_j) \otimes |j\rangle\langle j|. \end{aligned} \quad (4.32)$$

Next, we have that $[U_-, C(H)] = -[U_+, C(H)]^\dagger$. Thus,

$$[U_+ - U_-, C(H)] = -2 \operatorname{Re} \left(U_+ \sum_j (H_{j+1} - H_j) \otimes |j\rangle\langle j| \right) \quad (4.33)$$

and the full result follows almost immediately from the definition of Δ given in equation (4.28).

As for the upper bound, we note that $\|\operatorname{Re}(A)\| \leq \|A\|$ for any finite-dimensional A , and by unitary invariance of the spectral norm we have

$$\|[\Delta, C(H)]\| \leq \left\| \sum_j \frac{H_{j+1} - H_j}{\delta t} \otimes |j\rangle\langle j| \right\| = \max_j \left\| \frac{H_{j+1} - H_j}{\delta t} \right\|. \quad (4.34)$$

The upper bound then follows from the claim

$$\left\| \frac{H_{j+1} - H_j}{\delta t} \right\| \leq \max_{t \in [t_j, t_{j+1}]} \|\dot{H}(t)\| \quad (4.35)$$

coming from a the fundamental theorem of calculus and the triangle inequality. \square

Having defined the clock space and Hamiltonian, we wish to prepare a suitable initial state. A seemingly adequate and natural choice is to take $|\psi_0\rangle \otimes |0\rangle$, where $|\psi_0\rangle$ is the initial state of the system of interest and $|0\rangle$ is the clock state at the initial time $t = 0$. However, problems immediately arise which can be traced to the fact that the continuous version of $|0\rangle$ is $\delta(t)$, which is not a normalizable state vector. This formal problem finds its way into the discrete setting, in that the finite difference Δ does not properly compute a derivative of $|0\rangle$. Thus, Δ fails to translate $|0\rangle$ properly into later times, and the time dependent simulation fails.

To fix this issue, we take a cue from the continuous setting, where the best we can do is take a wavepacket of small enough width to suit our purposes. For simplicity, this wavepacket may as well be Gaussian, with some width σ to be chosen with care. Thus, we introduce Gaussian functions

$$\phi_\mu(t; \sigma) = \frac{1}{\sqrt{\mathcal{N}}} e^{-|t-\mu|_c^2/\sigma^2} \quad (4.36)$$

of width $\sigma \in \mathbb{R}_+$ and center $\mu \in [0, T)$. Here $|\cdot|_c$ is the shortest distance to 0 modulo T ,

$$|t|_c = \min \{|t|, |T - t|\} \quad (4.37)$$

so that, with 0 and T identified, ϕ_μ is smooth everywhere except $\mu + T/2 \pmod T$. Moreover, $\mathcal{N} \in \mathbb{R}_+$ is chosen such that the discretized vector

$$|\phi_\mu\rangle = \sum_j \phi_\mu(t_j; \sigma) |j\rangle \quad (4.38)$$

is normalized in the Euclidean sense (i.e., a quantum state vector). Technically, \mathcal{N} has some dependence on μ , but in our case we will only consider $\mu = t_j$ for some j , in which case \mathcal{N} only depends on parameters such as N_c and σ . Because of this choice, we will more simply write $|\phi_j\rangle \equiv |\phi_{t_j}\rangle$.

We are now ready to more clearly elucidate the overall strategy of the clock space construction. Figure 4.1 gives a schematic of the relevant components. We imagine N_p chunks of time steps, each containing N_q subdivisions. As stated above, each of the N_p should be thought of as a single Trotter step in the evolution under $H(t)$. The N_q substates ensure that δt is sufficiently small such that the approximation of Δ to a derivative of ϕ_j holds. In particular, we will desire $\sigma \gg \delta t$. On the other hand, we want the variation of H within the envelope of ϕ_j to be small. That is, we want $\sigma < T/N_p$. Because, presumably, we've chosen each Trotter step sufficiently small, this ensures that H is approximately constant over the bulk of $|\phi_j\rangle$. Of course, we will want to ensure all of the above conditions are met using as few resources, such as clock register states, as possible to get an accurate simulation.

Let's now characterize the accuracy of this construction. First, it will be helpful to have a characterization of the size of the normalization \mathcal{N} .

Lemma 4.3.2. *In the notation above, the normalization constant $\mathcal{N} \in \mathbb{R}_+$ for Gaussian states $|\phi_j\rangle$ peaked at $\mu = t_j$ satisfies*

$$\frac{1}{\sqrt{\mathcal{N}}} \in O(\sqrt{\delta t/\sigma}).$$

Proof. By cyclicity, the normalization \mathcal{N} is the same for all $|\phi_j\rangle$, so we consider $j = 0$. Because

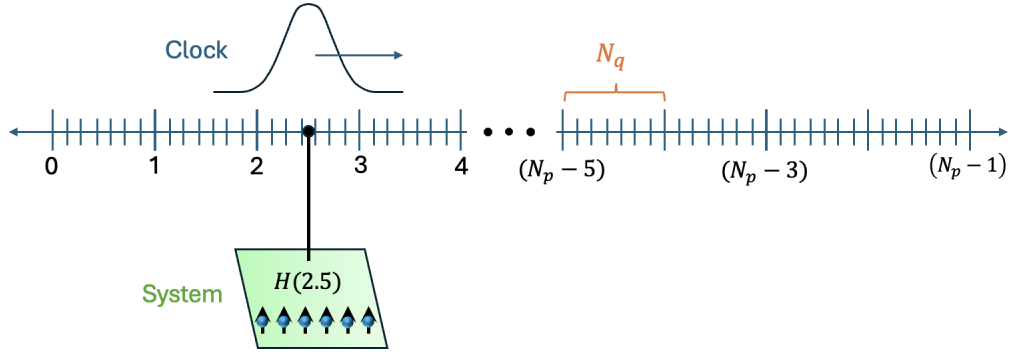


Figure 4.1 Schematic of the discrete clock Hilbert space. The clock register has an initially prepared Gaussian state which is translated uniformly under the clock Hamiltonian. Its location controls the Hamiltonian applied to the system of interest. The Hamiltonian varies little over each of the N_p large steps, and the Gaussian is wide compared to the N_q subdivisions within each large step.

$|\phi_0\rangle$ is normalized in the Euclidean norm, we have

$$\begin{aligned}
\mathcal{N} &= \sum_{j=0}^{N_c-1} e^{-2|\delta t^2 j|_c^2 / \sigma^2} \\
&= \sum_{j=0}^{N_c/2-1} e^{-2j^2 \delta t^2 / \sigma^2} + \sum_{j=N_c/2}^{N_c-1} e^{-2(N_c-j)^2 \delta t^2 / \sigma^2} \\
&= 1 + \sum_{j=1}^{N_c/2-1} e^{-2j^2 \delta t^2 / \sigma^2} + \sum_{j=1}^{N_c/2} e^{-2j^2 \delta t^2 / \sigma^2} \\
&= \sum_{j=0}^{\frac{N_c}{2}-1} e^{-2j^2 \delta t^2 / \sigma^2} + \sum_{j=0}^{\frac{N_c}{2}} e^{-2j^2 \delta t^2 / \sigma^2} - 1.
\end{aligned} \tag{4.39}$$

We may lower bound the sums as Riemann approximations to a Gaussian integral, giving error function erf

$$\mathcal{N} \geq \sqrt{\frac{\pi}{8}} \left(\operatorname{erf} \left(\frac{T+2\delta t}{\sqrt{2}\sigma} \right) + \operatorname{erf} \left(\frac{T}{\sqrt{2}\sigma} \right) \right) - 1 > \sqrt{\frac{\pi}{2}} \frac{\sigma}{\delta t} \operatorname{erf} \left(\frac{T}{\sqrt{2}\sigma} \right) - 1, \tag{4.40}$$

which then implies

$$\begin{aligned}
\frac{1}{\mathcal{N}} &\leq \sqrt{2/\pi} (\delta t / \sigma) \frac{1}{\operatorname{erf} \left(\frac{T}{\sqrt{2}\sigma} \right) - \sqrt{\frac{2}{\pi}} \frac{\delta t}{\sigma}} = \sqrt{\frac{2}{\pi}} (\delta t / \sigma) + O \left((\delta t / \sigma) \left(\frac{\delta t}{\sigma} + e^{-\frac{T^2}{2\sigma^2}} \right) \right) \\
&\in O(\delta t / \sigma).
\end{aligned} \tag{4.41}$$

The result follows simply from taking a square root. \square

With this technical lemma in hand, we turn to showing that Δ indeed acts as a generator of translations on the clock space for $|\phi_j\rangle$, provided σ is large relative to δt and that the Gaussian is not truncated by small T .

Lemma 4.3.3. *In the notation introduced in this section, for any $m \in \mathbb{Z}_+$ we have*

$$e^{i\Delta m\delta t}|\phi_j\rangle = |\phi_{j+m}\rangle + O\left(m(\delta t/\sigma)^2 + m\sqrt{\delta t/\sigma}e^{-(T/2\sigma)^2}\right)$$

where the asymptotics O are understood to be taken as $\delta t/\sigma \rightarrow 0$ and $\sigma/T \rightarrow 0$.

Proof. Performing a 1st order Taylor expansion of the exponential,

$$e^{i\Delta\delta t}|\phi_j\rangle = |\phi_j\rangle + i\delta t\Delta|\phi_j\rangle + R_1(\delta t)|\phi_j\rangle, \quad (4.42)$$

where R_1 is the Taylor remainder operator

$$R_1(\delta t) = \delta t \int_0^{\delta t} \frac{\partial^2}{\partial \tau^2} e^{i\Delta\tau} \delta t d\tau = - \int_0^{\delta t} e^{i\Delta\tau} d\tau (\delta t \Delta^2). \quad (4.43)$$

The Taylor error can be bounded, via the triangle inequality for integrals, as

$$\|R_1(\delta t)|\phi_j\rangle\| \leq \delta t^2 \|\Delta^2|\phi_j\rangle\|. \quad (4.44)$$

The action of Δ on discretized functions $|g\rangle$ of the clock space is given by

$$\begin{aligned} \Delta |g\rangle &= -i \sum_{j=0}^{N_c-1} g(t_j) \left(\frac{|j+1\rangle - |j-1\rangle}{2\delta t} \right) \\ &= i \sum_j \frac{g(t_{j+1}) - g(t_{j-1})}{2\delta t} |j\rangle \\ &= i |D_{\delta t}g\rangle. \end{aligned} \quad (4.45)$$

Here $D_{\delta t}f(x) := \frac{f(x+\delta t) - f(x-\delta t)}{2\delta t}$ is the symmetric finite difference of halfwidth δt at point x . Thus, $\Delta^2|\phi_j\rangle = -|D_{\delta t}^2\phi_j\rangle$. We consider the error of this finite difference in terms of an approximation to the derivative for values of t within $T/2 - 2\delta t$ of t_j in circle distance. On this part of the domain, $\phi_j(t \pm 2\delta t)$ is smooth, hence

$$|D_{\delta t}^2\phi_j\rangle = |\partial_t^2\phi_j\rangle + O(\delta t^2\phi_j^{(4)}) \quad (4.46)$$

where the superscript (4) indicates a fourth derivative. Near the edge of the Gaussian, the second-derivative property does not hold; however, these parts of the state vector have amplitude which is on the order $O(\mathcal{N}^{-1/2}e^{-(T/2\sigma)^2})$, which by Lemma 4.3.2 is $O(\sqrt{\delta t/\sigma}e^{-(T/2\sigma)^2})$. This gets multiplied by δt^{-2} due to the second finite difference $D_{\delta t}$ being taken. Taking the two sources independently as an upper bound, we have

$$\|\Delta^2|\phi_j\rangle\| \in O\left(\delta t^2/\sigma^4 + (\sigma\delta t^3)^{-1/2}e^{-(T/2\sigma)^2}\right) \quad (4.47)$$

where σ^{-4} comes from the four derivatives of the Gaussians. Thus, the total Taylor remainder may be upper bounded using (4.43) as

$$\|R_1(\delta t)|\phi_j\rangle\| \in O\left((\delta t/\sigma)^4 + \sqrt{\delta t/\sigma}e^{-(T/2\sigma)^2}\right). \quad (4.48)$$

To complete the proof we return to the linear Taylor expansion in (4.42). Using similar reasoning to above,

$$\begin{aligned} |\phi_j\rangle + i\delta t\Delta|\phi_j\rangle &= |\phi_j\rangle - \delta t|D_{\delta t}\phi_j\rangle \\ &= |\phi_j\rangle - \delta t|\partial_t\phi_j\rangle + O\left((\delta t/\sigma)^2 + \sqrt{\delta t/\sigma}e^{-(T/2\sigma)^2}\right). \end{aligned} \quad (4.49)$$

Finally, what remains is a linear approximation to $|\phi_{j+1}\rangle$, with error also $(\delta t/\sigma)^2$. Keeping only the leading terms, notice that the Taylor remainder error is subdominant. Altogether,

$$e^{i\Delta\delta t}|\phi_j\rangle = |\phi_{j+1}\rangle + O\left((\delta t/\sigma)^2 + \sqrt{\delta t/\sigma}e^{-(T/2\sigma)^2}\right). \quad (4.50)$$

So far, we've proved the result for $m = 1$. The full result follows by noting that $e^{i\Delta m\delta t} = (e^{i\Delta\delta t})^m$ and taking, as upper bound, m times the error of a single step. \square

We note that the error in Δ generating translations comes from two sources: the discretization at small scales and the boundary effects at large scales. We might name these, in the language of the lattice field theory, ultraviolet and infrared truncation effects, respectively.

Our next intermediate result will be concerned with the time evolution of the system under $C(H)$ controlled on the Gaussian state $|\phi_j\rangle$. We want the result to be, approximately, an evolution under $H(t_j)$ on the main register of interest. In what follows, it will be convenient to take $\tau := T/N_p$ as the time duration of a larger subdivision of steps.

Lemma 4.3.4. *Let $H : [0, T] \rightarrow \text{Herm}(\mathcal{H})$ be a bounded differentiable function with bounded derivative. For any $\eta \in \mathbb{R}$, we have*

$$e^{-iC(H)\eta}|\psi\rangle|\phi_j\rangle = e^{-iH(t_j)\eta}|\psi\rangle|\phi_j\rangle + O\left(\eta\tau \max_{t \in [0, T]} \|\dot{H}(t)\| + (1 + \eta \max_{t \in [0, T]} \|H(t)\|)e^{-\tau^2/4\sigma^2}\right)$$

where $\tau := T/N_p$.

Proof. We begin by grouping the terms of $C(H)$ into two chunks: one with significant overlap with the Gaussian, the other with small overlap. Specifically, we take $C(H) = H_{\text{av}} + H_{\perp}$, with

$$H_{\text{av}} := \sum_{k=j-N_q/2}^{j+N_q/2-1} H_k \otimes |k\rangle\langle k| \quad (4.51)$$

$$H_{\perp} := C(H) - H_{\text{av}}.$$

Because H_{av} and H_{\perp} commute, we can Trotterize with no error.

$$e^{-iC(H)\eta}|\psi\rangle|\phi_j\rangle = e^{-iH_{\perp}\eta}e^{-iH_{\text{av}}\eta}|\psi\rangle|\phi_j\rangle \quad (4.52)$$

We will show that the H_{av} term gives approximately $H(t_j)$, while H_{\perp} acts as approximately the identity (with the right parameter values).

First, consider $e^{-iH_{\text{av}}\eta}$. Define $P_j = \sum_k |k\rangle\langle k|$ as the projector onto the clock states on which H_{av} has support ($k \in \mathbb{Z} \cap [j - N_q/2, j + N_q/2 - 1]$). We have

$$\|e^{-iH_{\text{av}}\eta} - e^{-iH_j \otimes P_j \eta}\| \leq \eta \|H_{\text{av}} - H_j \otimes P_j\|. \quad (4.53)$$

Meanwhile,

$$\|H_{\text{av}} - H_j \otimes P_j\| = \left\| \sum_{k=j-N_q/2}^{j+N_q/2-1} (H_k - H_j) \otimes |k\rangle\langle k| \right\| = \max_k \|H_k - H_j\|. \quad (4.54)$$

By a simple Taylor bound, $\|H_k - H_j\| \leq (\tau/2) \max_t \|\dot{H}(t)\|$, where the max is over $[t_k, t_j]$ (taking the appropriate ordering of t_j, t_k if needed). We can therefore say

$$\|e^{-iH_{\text{av}}\eta} - e^{-iH_j \otimes P_j \eta}\| \leq \eta\tau \max_{t \in [0, T]} \|\dot{H}\| \quad (4.55)$$

so that, up to this error, we can replace a simulation by H_{av} with $H_j \otimes P_0$. Moving on to this situation, we have

$$e^{-iH_j \otimes P_0 \eta} |\psi\rangle \otimes |\phi_j\rangle = e^{-iH_j \eta} |\psi\rangle P_j |\phi_j\rangle + |\psi\rangle (I - P_j) |\phi_j\rangle. \quad (4.56)$$

Thinking of $\sigma < \tau$ and taking τ/σ increasing, we have $P_0 |\phi_j\rangle = |\phi_j\rangle + O\left(e^{-\tau^2/4\sigma^2}\right)$. Thus,

$$e^{-iH_j \otimes P_0 \eta} |\psi\rangle |\phi_j\rangle = e^{-iH_j \eta} |\psi\rangle |\phi_j\rangle + O\left(\eta \tau \max_{t \in [0, T]} \|\dot{H}\| + e^{-\tau^2/4\sigma^2}\right). \quad (4.57)$$

For the remainder of the proof, take $|\psi'\rangle = e^{-iH_j \eta} |\psi\rangle$ for notational convenience. We now consider the action of H_\perp on the remaining state, which we anticipate to be small. First,

$$\|e^{-iH_\perp \eta} |\psi'\rangle |\phi_j\rangle - |\psi'\rangle |\phi_j\rangle\| \leq \eta \|H_\perp |\psi'\rangle |\phi_j\rangle\|. \quad (4.58)$$

Let \mathcal{J} be an index set for all the time steps included in the summation H_{av} . We have

$$\begin{aligned} \|H_\perp |\psi'\rangle |\phi_j\rangle\| &= \left\| \sum_{k \notin \mathcal{J}} H_k |\psi'\rangle |k\rangle \langle k | \phi_j \rangle \right\| \\ &\leq \sqrt{\sum_{k \notin \mathcal{J}} \frac{1}{\mathcal{N}} e^{-2|t_j - t_k|^2/\sigma^2} \|H_k\|^2}. \end{aligned} \quad (4.59)$$

Employing a Hölder inequality on the inner product, followed by Lemma 4.3.2,

$$\begin{aligned} \sqrt{\sum_{k \notin \mathcal{J}} \frac{1}{\mathcal{N}} e^{-2|t_j - t_k|^2/\sigma^2} \|H_k\|^2} &\leq \max_{k \notin \mathcal{J}} \|H_k\| \sum_{k \notin \mathcal{J}} \frac{e^{-|t_j - t_k|^2/\sigma^2}}{\mathcal{N}} \\ &\in O\left(\max_t \|H(t)\| (\delta t / \sigma) \sum_{k=N_q/2}^{\infty} e^{-k^2 \delta t^2 / \sigma^2}\right). \end{aligned} \quad (4.60)$$

Following a similar procedure to before, we convert to an error function erf and take an exponential upper bound. Doing so gives

$$\|H_\perp |\psi'\rangle |\phi_j\rangle\| \in O\left(\max_{t \in [0, T]} \|H(t)\| e^{-\tau^2/2\sigma^2}\right). \quad (4.61)$$

Thus, $e^{-iH_\perp \eta}$ acts trivially on this state up to $O\left(\eta \max_t \|H(t)\| e^{-(\tau/2\sigma)^2}\right)$.

Combining the errors together, we take the widest exponential $e^{-\tau^2/4\sigma^2}$ as a simple upper bound for all exponentials that appear. Putting all the error sources together gets us the result of the Lemma statement. \square

With the previous two lemmas, we have the ingredients needed for a clock space simulation: controlled operations and time shifts. We combine them to show that our clock space indeed encodes time dependent dynamics.

Theorem 4.3.5. *Let $H[0, T] \rightarrow \text{Herm}(\mathcal{H})$ be a time dependent Hamiltonian on a finite dimensional vector space \mathcal{H} , such that $H(t)$ as a function is bounded and differentiable with bounded derivative. Then, the clock Hamiltonian, with Gaussian input $|\phi_0\rangle$, approximately applies the time evolution operator $U(T, 0)$ to an initial state $|\psi_0\rangle \in \mathcal{H}$. Precisely,*

$$e^{-iH_c T} |\psi_0\rangle |\phi_0\rangle = (U(T, 0) |\psi_0\rangle) |\phi_0\rangle + O\left(T\delta t/\sigma^2 + \sqrt{N_c T/\sigma} e^{-T^2/4\sigma^2} + \max_t \|\dot{H}\| \frac{T^2}{N_p} + e^{-\tau^2/4\sigma^2} (N_p + \max_t \|H\|T)\right).$$

Proof. Let $\tau := T/N_p$. We begin with a first-order Trotterization of H_c into N_p steps.

$$e^{-iH_c T} = \left(e^{i\Delta\tau} e^{-iC(H)\tau}\right)^{N_p} + O\left(\max_{t \in [0, T]} \|\dot{H}(t)\| \frac{T^2}{N_p}\right) \quad (4.62)$$

With initial state $|\psi_0\rangle |\phi_0\rangle$, combining Lemmas 4.3.4 and 4.3.3 gives the following error for a single Trotter step.

$$e^{i\Delta\tau} e^{-iC(H)\tau} |\psi_0\rangle |\phi_0\rangle = e^{-iH_0\tau} |\psi_0\rangle |\phi_{N_q}\rangle + O\left(\tau\delta t/\sigma^2 + \sqrt{N_q\tau/\sigma} e^{-T^2/4\sigma^2} + \tau^2 \max_t \|\dot{H}\| + (1 + \tau \max_t \|H\|) e^{-\tau^2/4\sigma^2}\right) \quad (4.63)$$

Thus, after all N_p steps, we can multiply the single step error above to get an upper bound of

$$\left(e^{i\Delta\tau} e^{-iC(H)\tau}\right)^{N_p} |\psi_0\rangle |\phi_0\rangle = e^{-iH_{N_q(N_p-1)\tau} \dots e^{-iH_{N_q}\tau} e^{-iH_0\tau} |\psi_0\rangle |\phi_0\rangle + O\left(T\delta t/\sigma^2 + \sqrt{N_c T/\sigma} e^{-T^2/4\sigma^2} + \max_t \|\dot{H}\| \frac{T^2}{N_p} + e^{-\tau^2/4\sigma^2} (N_p + \max_t \|H\|T)\right). \quad (4.64)$$

The right side, without the error, is a 1st order Suzuki Trotter splitting, which approximates $U(T, 0)$ to order $\max_{t \in [0, T]} \|H(t)\| T^2/N_p$. This can be absorbed into the third term of the big- O . This gives the result stated in the Theorem. \square

With this result in hand, we now show that the parameters (N_p, N_q, σ) of the clock can be chosen such that any desired degree of approximation to $U(T, 0)$ can be achieved.

Theorem 4.3.6. *In the context of the previous theorem, for any $\epsilon \in \mathbb{R}_+$, there exists clock parameters (N_p, N_q, σ) such that*

$$\|e^{-iH_c T} |\psi_0\rangle |\phi_0\rangle - U(T, 0) |\psi_0\rangle |\phi_0\rangle\| < \epsilon$$

with (N_p, N_q) scaling as

$$N_p \in \Theta\left(\max_{t \in [0, T]} \|\dot{H}\| \frac{T^2}{\epsilon}\right), \quad N_q \in \Theta\left(\frac{\max_t \|\dot{H}\| T^2}{\epsilon^2} x^2\right), \quad \sigma \in \Theta\left(\frac{\epsilon}{\max_t \|\dot{H}\| T x}\right).$$

Here,

$$x := \sqrt{\log\left(\frac{\Gamma T}{\epsilon}\right)}$$

$$\Gamma := \max\left\{\max_{t \in [0, T]} \|\dot{H}\| T, \epsilon \max_{t \in [0, T]} \|H\|\right\}.$$

In particular, there exists a sequence $(N_p(j), N_q(j), \sigma(j))$ of clock space parameters, such that

$$\lim_{j \rightarrow \infty} \text{Tr}_c(e^{-iH_c T} |\psi_0\rangle |\phi_0\rangle) = U(T, 0) |\psi_0\rangle$$

where Tr_c is a partial trace over the clock register, and $\text{Tr}_c(|\Psi\rangle) \equiv \text{Tr}_c(|\Psi\rangle\langle\Psi|)$.

Proof. To ensure a total error within ϵ is achievable, it suffices to ensure that each of the five terms constituting the error in Theorem 4.3.5 is within $O(\epsilon)$ independently. From the onset, we will choose $N_p \in \Theta(\max_{t \in [0, T]} \|\dot{H}\| T^2 / \epsilon)$ to satisfy the third term.

We next move to understand the necessary σ scaling. We parametrize it as

$$\sigma = \tau/x \tag{4.65}$$

with the hope that x can be chosen to increase slowly (i.e., that the Gaussian states have width only slightly smaller than the Trotter step size). For this, we focus on the last two terms, since they have no N_q dependence (which will set the smallest scales). We seek

$$\frac{\max_t \|\dot{H}\| T^2}{\epsilon} e^{-x^2/4} \in O(\epsilon), \quad \max_t \|H\| T e^{-x^2/4} \in O(\epsilon) \tag{4.66}$$

which can be satisfied provided that x is asymptotically lower bounded as

$$x^2 \in \Omega\left(\log \max\left\{\frac{\max_t \|\dot{H}\| T^2}{\epsilon^2}, \frac{\max_t \|H\| T}{\epsilon}\right\}\right)$$

$$= \Omega(\log(\Gamma T / \epsilon)). \tag{4.67}$$

This sets the scaling for σ .

We move next to the first term to fix N_q , since the 2nd term is expected to be quite small. We require $T\delta t/\sigma^2 \in O(\epsilon)$, which is equivalent to

$$\frac{N_p x^2}{N_q} \in O(\epsilon). \quad (4.68)$$

Therefore, there exists an $N_q \in \Theta(N_p x^2/\epsilon)$, satisfying the bound. All that remains is the second term, whose contribution can be easily shown to be subdominant compared to the other sources. Therefore, the choice of parameter scaling suffice to achieve the desired error ϵ .

We have shown that any desired precision ϵ for dynamical simulation can be accommodated by appropriate choice of clock space parameters. Taking a sequence $\epsilon_j \rightarrow 0$, we see there exists a sequence of clock space evolutions whose limit, restricted to the main register, is $U(T, 0)$. \square

The asymptotic scalings provided in the above theorem will be important in our discussion of qubitization, where we will use them to derive a query complexity.

4.4 Time Dependent Qubitization

In the previous section, we developed a clock space construction which encoded a time dependent Hamiltonian as a time independent one on an augmented, finite-dimensional space. The removal of time-ordering using a clock register opens the door for quantum algorithms for time independent Hamiltonian simulation to simulate the full clock-system dynamics directly. In particular, qubitization is an asymptotically optimal [88] simulation method that can only be applied to time independent H . In this section, we propose the simulation of time dependent Hamiltonians through qubitization using finite clock registers. To be concrete, we will work with an input model in which $H(t)$ is a linear combination of fixed unitaries with time-varying coefficients. This describes, for example, Pauli matrices on n qubits with fluctuating coefficients.

4.4.1 Pseudo-Algorithm

We take n_c qubits to provide a clock register of size $N_c = 2^{n_c}$, which are included with the main register of interest. The initial state $|\psi_0\rangle |\phi_0\rangle$ must be prepared on this joint register. We take $|\psi_0\rangle$ of the main register as given, since this is necessarily application dependent. We must, however,

prepare a Gaussian state $|\phi_0\rangle$ on the clock register of n_c qubits with width σ . Unsurprisingly, much effort has been devoted to this task. [80, 110, 109, 71, 57]. For our purposes, we will simply refer to the approach by Kitaev and Webb [79, 10] as efficient enough for our purposes. The Gaussian will nonnegligible support over $O(N_q)$ clock states, and their algorithm scales polynomially in the number of qubits $n_q = \log N_q$ over the Gaussian. This cost is negligible compared to the simulation costs that we are about to discuss.

Once the initial state is prepared, we employ qubitization to approximate $e^{-iH_c T}$ on the full register. Given $H(t)$ in LCU form, we need to express H_c in LCU form as well, which is not immediate. This is done through several applications of the Signature Matrix Decomposition. We also truncate Δ at high frequencies to reduce computational cost, with little loss in accuracy. Details of the LCU decomposition are provided in the next subsection.

Once H_c is in LCU form, select SEL and prepare PREP circuits may be constructed to block encode H_c as

$$H_c/\|c\|_1 = (\langle 0| \text{PREP}^\dagger \otimes I) \text{SEL}(\text{PREP} |0\rangle \otimes I) \quad (4.69)$$

where $\|c\|_1$ is the one-norm of the LCU coefficients. Standard qubitization can now be done on this block encoded Hamiltonian [88]. The PREP circuit must create a "quasi-uniform" distribution over some number N of states, in the sense that, on the LCU auxiliary register,

$$|\text{PREP}\rangle = \sum_{j=1}^{K-1} \sqrt{\delta} |j\rangle + \sum_{j=K}^N \sqrt{\delta'} |j\rangle \quad (4.70)$$

with δ, δ', K and N determined by parameters of simulation. Meanwhile the SEL circuit will need to apply controlled U_i operations, where U_i is a unitary in the $H(t)$ decomposition, and controlled signature matrices. These second operations can be done with classical comparator circuits. Each SEL will also require a Quantum Fourier Transform and its inverse on the clock register.

4.4.2 Block Encoding

To make progress, it seems $H(t)$ itself should be expressible neatly in terms of unitaries. Thus, we assume $H(t)$ is of the form

$$H(t) = \sum_{i=1}^L \alpha_i(t) U_i \quad (4.71)$$

where U_j are Hermitian and unitary (e.g., n -qubit signed Pauli operators) and $\alpha_j(t)$ are nonnegative real-valued functions on $[0, T]$. When we discretize, the coefficients $\alpha_{ij} \equiv \alpha_i(t_j)$ will be particularly important. Expanding out $C(H)$ from equation (4.27) using (4.71),

$$\begin{aligned} C(H) &= \sum_{j=0}^{N_c-1} \left(\sum_{i=0}^{L-1} \alpha_{ij} U_i \right) \otimes |j\rangle\langle j| \\ &= \sum_{i=0}^{L-1} U_i \otimes D_i \end{aligned} \tag{4.72}$$

where

$$D_i := \sum_{j=0}^{N_c-1} \alpha_{ij} |j\rangle\langle j| \tag{4.73}$$

is a diagonal operator on the clock register. There is a general technique for LCU constructions of diagonal, or easily diagonalized, operators via a Signature Matrix Decomposition, which we digress to discuss.

4.4.3 Interlude: Signature Matrix Decomposition or "Alternating Sign Trick"

In our manipulations of H_c , we have come across the problem of expressing a diagonal, Hermitian matrix D in LCU form. We will handle this in the present section. Although I did not invent this technique, I had a sufficiently hard time finding a clear reference to it in the literature, such that I felt an overview would be appropriate and possibly helpful to future researchers.

There are many unitary bases that exist, but in our case we have quite stringent requirements. We ultimately want our decomposition to consist of *Hermitian* operations as well, meaning they should be reflection operators ($U^2 = I$). Moreover, it is sensible to look for diagonal unitaries, because our operator D is diagonal. These requirements alone enforce that our unitaries are, in fact, *signature matrices*: $U = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ where $\lambda_j = \pm 1$.

For the moment, let's imagine the entries of D are all positive integers, and we allow ourselves to add unitaries only in integer amounts. Think of each entry as a bucket of size λ_j . We want to add to the bucket, and we can only do so in units of +1 (by unitarity) or -1. Each time we add a unitary, say, the identity, we are adding a unit to each bucket. Some of the buckets will fill up faster than others because they are smaller, but we are not allowed to stop adding, per se. We can only

add or remove one unit, not zero, by unitarity. The next best thing we can do is, while the other buckets are getting filled, add and remove 1 unit in alternating sequence. We need to do this until the largest bucket has been filled, at which point we can stop.

Let's put this all more formally. We want a sequence of unitaries that keep track of whether the entries are above, or below, the number k of additions that have already occurred. To this end, define

$$U_k := \sum_{j=1}^n (-1)^{k \lfloor k > \lambda_j \rfloor} |j\rangle\langle j| \quad (4.74)$$

where $\lfloor P \rfloor$ is the boolean function for proposition P assigning 1 to true, 0 to false. We see that, for k even, $U_k = I$ is the identity operator. while for odd k U_k has eigenvalue -1 whenever j is such that $k > \lambda_j$. Then, if we take a sum "until the largest bucket" $\|D\|$ has been filled, we should obtain D . In fact,

$$\sum_{k=1}^{\|D\|} U_k = \sum_{j=1}^n |j\rangle\langle j| = \sum_{j=1}^n |j\rangle\langle j| \sum_{k=1}^{\|D\|} (-1)^{k \lfloor k > \lambda_j \rfloor} \quad (4.75)$$

and the inner sum can be written as

$$\sum_{k=1}^{\lambda_j} 1 + \sum_{k=\lambda_j+1}^{\|D\|} (-1)^k = \lambda_j + \epsilon \quad (4.76)$$

where $\epsilon \in \{-1, 0, 1\}$ is an $O(1)$ -error. We will see shortly how to boost the precision, but first we note that, generalizing to positive real-valued entries, the same procedure for U_k (taking $\lceil \|D\| \rceil$ as the upper sum limit) generates $\lfloor \lambda_j \rfloor$ on the entries to accuracy ± 1 at worst. Thus, for real values the error is less than 2, which is still $O(1)$.

This might not seem like a good approximation, especially when λ_j is small. But we can artificially increase the size of λ_j by performing the same procedure on D/δ for suitably small $\delta > 0$, then multiplying by δ . Let $L_\delta := \lceil \|D\|/\delta \rceil$. Then

$$D/\delta = \sum_{k=1}^{L_\delta} U_k + O(1) \quad (4.77)$$

so

$$D = \sum_{k=1}^{L_\delta} \delta U_k + O(\delta) \quad (4.78)$$

with U_k the same as (4.74) but with the replacement $\lambda_j \rightarrow \lambda_j/\delta$. We've succeeded at expressing D in LCU form to accuracy $O(\delta)$ using L_δ terms.

We still haven't handled negative eigenvalues. This is accomplished by adding the appropriate sign. Altogether, the L_δ matrices

$$U_k := \sum_{j=1}^n \text{sgn}(\lambda_j) (-1)^{k \lceil |\lambda_j|/\delta \rceil} \quad (4.79)$$

are sufficient to approximate D to within 2δ in each entry.

Assuming the eigenvalues λ_j are known and classically computable, unitaries such as (4.79) can be implemented on a quantum computer using comparator circuits. Observe that this procedure can also generate an LCU type expansion, more generally, when the Hermitian operator H is easily diagonalizable. Moreover, the fact that the coefficients in the LCU are the same allows for simple implementation in a select-and-prepare block encoding.

4.4.4 Block Encoding (cont.)

Let $\Lambda_i(\delta) \equiv \lceil \max_j |\alpha_{ij}|/\delta \rceil$. Using a signature matrix decomposition, we can write

$$D_i = \sum_{k=1}^{\Lambda_i(\delta)} \delta S_{ik}(\delta) + O(\delta) \quad (4.80)$$

for $\delta > 0$, where

$$S_{ik}(\delta) = \sum_{j=0}^{N_c-1} (-1)^{k \lceil \alpha_{ij}/\delta \rceil} |j\rangle\langle j| \quad (4.81)$$

and $[P]$ is the Boolean function for proposition P , with $[\text{True}] = 1$ and $[\text{False}] = 0$. Thus, we obtain an LCU decomposition of $C(H)$ as

$$C(H) = \delta \sum_{i=1}^L \sum_{k=1}^{\Lambda_i(\delta)} U_i \otimes S_{ik}(\delta) + O(L\delta). \quad (4.82)$$

The prepare circuit PREP is simple enough because the linear combination is uniform. Therefore, it can be accomplished using a Hadamard gate on each of

$$n_{C(H)} \in O \left(\log \sum_{i=0}^L \max_j |\alpha_{ij}|/\delta \right) \quad (4.83)$$

auxiliary qubits needed for a binary encoding. The unitaries $U_i \otimes S_{ik}(\delta)$ can be selected using two different SEL circuits: one for the original U_i (presumed available to us) and one for the signature

matrices $S_{ik}(\delta)$. These unitaries can be constructed using classical comparator circuits provided that each α_{ij} is computable.

We turn our attention now to Δ . Although already in LCU form, the coefficient has size $2/\delta t$ and is too large to be desirable. As discussed above, the problem stems from unnecessary high-frequency modes, which we wish to truncate. We start by converting Δ to Fourier space, i.e., diagonalizing via the Quantum Fourier Transform. The result may be computed by diagonalizing U_+ , and is found to be

$$\begin{aligned}\Delta &= \text{QFT} \sum_{j=0}^{N_c-1} \frac{N_c}{T} \sin\left(2\pi \frac{j}{N_c}\right) |j\rangle\langle j| \text{QFT}^\dagger \\ &= \text{QFT} \sum_{j=-N_c/2}^{N_c/2-1} \frac{N_c}{T} \sin\left(2\pi \frac{j}{N_c}\right) |j\rangle\langle j| \text{QFT}^\dagger\end{aligned}\tag{4.84}$$

where, in the second line, we define indices $-j = N_c - j$ for $j > 0$ and write the diagonalized Δ symmetrically about $j = 0$. The benefit of this parametrization is that small $|j|$ correspond to low-frequency modes, as we shall see. Let Δ_J be Δ truncated at frequencies above those determined by index $J \in [0, N_c/2] \cap \mathbb{Z}$.

$$\Delta_J := \text{QFT} \sum_{j=-J}^J \frac{N_c}{T} \sin\left(2\pi \frac{j}{N_c}\right) |j\rangle\langle j| \text{QFT}^\dagger\tag{4.85}$$

The error in a clock space evolution using Δ_J rather than Δ is upper bounded by $T\|\Delta|\phi_0\rangle - \Delta_J|\phi_0\rangle\|$, which can be evaluated and upper bounded as

$$\begin{aligned}T\|\Delta|\phi_0\rangle - \Delta_J|\phi_0\rangle\| &= \left\| \sum_{|j|>J} N_c \sin\left(2\pi \frac{j}{N_c}\right) |j\rangle\langle j| \text{QFT}^\dagger |\phi_0\rangle \right\| \\ &\leq N_c \sqrt{\sum_{|j|>J} |\langle j| \text{QFT}^\dagger |\phi_0\rangle|^2}.\end{aligned}\tag{4.86}$$

We thus desire a characterization of $\text{QFT}^\dagger |\phi_0\rangle$, which we naturally expect to be another Gaussian up to errors arising from the difference between discrete and continuous Fourier Transforms. This analysis was performed in Appendix C of [113], and we adapt that work to our present situation. As the reference shows, the error in each component j arises from three sources:

1. Truncation of the time variable to $O(T)$, which we denote ϵ_{trunc} .

2. Truncation of the frequency variable to $O(N_c/T)$ ("aliasing"), which we denote ϵ_{alias} .
3. Differences in normalizing in the continuum vs the discrete setting, which we denote ϵ_{norm} .

In our notation and setting, Rendon et al. [113] show that these errors satisfy the following asymptotic bounds.

$$\begin{aligned}
\epsilon_{\text{trunc}} &\in O\left(\sqrt{\frac{\sigma}{T}}e^{-\Omega(T^2/\sigma^2)}\right) \\
\epsilon_{\text{alias}} &\in O\left(\sqrt{\frac{\sigma}{T}}e^{-\Omega(N_c^2\sigma^2/T^2)}\right) \\
\epsilon_{\text{norm}} &\in O\left(e^{-\Omega(N_c)}\right)
\end{aligned} \tag{4.87}$$

Let's take these errors to all be $O(\epsilon_{\text{QFT}})$, with the required ϵ_{QFT} to be determined. The results from Theorem 16 and Appendix C of [113] imply that

$$\text{QFT}^\dagger |\phi_0\rangle = \sum_{j=-N_c/2}^{N_c/2-1} \left(\sqrt{\frac{\pi N_c \sigma}{N T}} e^{-\pi j \sigma / T} + O(\epsilon_{\text{QFT}}) \right) |j\rangle. \tag{4.88}$$

With this in hand, we return to (4.86). First,

$$|\langle j | \text{QFT}^\dagger |\phi_0\rangle|^2 = \frac{\pi N_c \sigma^2}{N T^2} e^{-2(\pi j \sigma / T)^2} + O\left(\sqrt{\frac{\sigma}{T}} e^{-\pi j \sigma / T} \epsilon_{\text{QFT}}\right) \tag{4.89}$$

where we assume the error ϵ_{QFT} is smaller asymptotically than the amplitude itself, to be justified.

Taking the sum over high frequencies,

$$\begin{aligned}
\sqrt{\sum_{|j|>J} |\langle j | \text{QFT}^\dagger |\phi_0\rangle|^2} &\in O\left(\sqrt{\frac{N_c \sigma}{N T}} e^{-\Omega(J^2 \sigma^2 / T^2)} + \sqrt{\frac{\sigma}{T}} e^{-\Omega(J^2 \sigma^2 / T^2)} \epsilon_{\text{QFT}}\right) \\
&\subseteq O\left(\sqrt{\frac{\sigma}{T}} e^{-\Omega(J^2 \sigma^2 / T^2)} (1 + \epsilon_{\text{QFT}})\right).
\end{aligned} \tag{4.90}$$

We next observe that $\epsilon_{\text{QFT}} \in O(1)$ by previous assumptions, and can now be removed. From (4.86), we get the full simulation error by multiplying by N_c

$$\epsilon_J \in O\left(N_c \sqrt{\frac{\sigma}{T}} e^{-\Omega(J^2 \sigma^2 / T^2)}\right). \tag{4.91}$$

In order for $\epsilon_J \in O(\epsilon)$, we want the cutoff J to satisfy

$$e^{-J^2 \sigma^2 / T^2} \in O\left(\sqrt{\frac{T}{\sigma}} \frac{\epsilon}{N_c}\right) \tag{4.92}$$

which can be satisfied provided J scales as

$$J \in \Theta\left(\frac{T}{\sigma} (\log \sigma/T + \log N_c + \log 1/\epsilon)\right) \subseteq \tilde{\Theta}(T/\sigma). \quad (4.93)$$

Letting $\tilde{\Delta} \equiv \Delta_J$ for this choice of J , we now switch to considering the simulation of $\tilde{\Delta}$. Let $\delta' > 0$, and let $\Gamma(\delta') := \lceil (N_c/T\delta') \sin(2\pi J/N_c) \rceil$. We have

$$\sum_{j=-J}^J \frac{N_c}{T} \sin\left(2\pi \frac{j}{N_c}\right) |j\rangle\langle j| = \delta' \sum_{\ell=1}^{\Gamma(\delta')} S_k^{(\Delta)}(\delta') + O(\delta') \quad (4.94)$$

where

$$S_k^{(\Delta)}(\delta') := \sum_{j=-J}^J \text{sgn}(j) (-1)^{k \lceil (N_c/T\delta') \sin(2\pi j/N_c) \rceil}. \quad (4.95)$$

Defining the unitary $V_\ell(\delta') := \text{QFT } S_\ell^{(\Delta)}(\delta') \text{QFT}^\dagger$, we have obtained an LCU decomposition of Δ .

The PREP circuit is, as with $C(H)$, only a column of Hadamards on

$$n_\Delta \in O(\log((N_c/T\delta') \sin(2\pi J/N_c))) \subseteq \tilde{O}\left(\log \frac{1}{\sigma\delta'}\right) \quad (4.96)$$

auxiliary qubits. Meanwhile the SEL circuit may be constructed as $\text{QFT SEL}' \text{QFT}^\dagger$, where SEL' is a select circuit using the $S_\ell^{(\Delta)}$ signature matrices that can, as before, be implemented with comparator circuits that compute sine.

Combining with (4.82), we obtain an approximate LCU decomposition of the approximate clock Hamiltonian \tilde{H}_c .

$$\tilde{H}_c = \delta \sum_{i=1}^L \sum_{k=1}^{\Lambda_i(\delta)} U_i \otimes S_{ik}(\delta) + \delta' \sum_{\ell=1}^{\Gamma(\delta')} I \otimes V_\ell(\delta') + O(\epsilon/T + L\delta + \delta') \quad (4.97)$$

To achieve an ϵ -accurate simulation, we will require $\delta \in O(\epsilon/LT)$ and $\delta' \in O(\epsilon/T)$. The 1-norm

$\|c\|_1$ of all of the coefficients is given by

$$\begin{aligned}
\|c\|_1 &= \delta \sum_{i=0}^{L-1} \Lambda_i(\delta) + \delta' \Gamma(\delta') \\
&\in O \left(\sum_{i=0}^{L-1} \max_j |\alpha_{ij}| + \frac{N_c}{T} \sin(2\pi J/N_c) \right) \\
&\subseteq O \left(\|\alpha\|_{\infty,1}^{\text{rev}} + J/T \right) \\
&\subseteq \tilde{O} \left(\|\alpha\|_{\infty,1}^{\text{rev}} + \sigma^{-1} \right) \\
&\subseteq \tilde{O} \left(\|\alpha\|_{\infty,1}^{\text{rev}} + \frac{\max_t \|\dot{H}\|T}{\epsilon} \right)
\end{aligned} \tag{4.98}$$

where $\|\alpha\|_{\infty,1} \equiv \sum_{i=0}^{L-1} \max_t |\alpha_i(t)|$ and \tilde{O} suppresses multiplicative logarithmic factors. Thus, the number of queries to SEL and PREP circuits in an LCU encoding scales as

$$Q \in \tilde{O} \left(\|\alpha\|_{\infty,1}^{\text{rev}} T + \frac{\max_t \|\dot{H}\|T^2}{\epsilon} + \frac{\log 1/\epsilon}{\log \log 1/\epsilon} \right). \tag{4.99}$$

The number of auxiliary qubits needed for the clock register is

$$n_c = \log N_p + \log N_q \in O \left(\log(\max_t \|\dot{H}\|T^2) + \log 1/\epsilon \right) \tag{4.100}$$

while the number of auxiliary qubits needed for the LCU block encoding is given by

$$\begin{aligned}
n_{\text{LCU}} &= n_{C(H)} + n_{\Delta} \\
&\in O \left(\log \frac{\|\alpha\|_{\infty,1}^{\text{rev}}}{\delta} + \log \frac{1}{\sigma \delta'} \right) \\
&\subseteq O \left(\log \frac{L \|\alpha\|_{\infty,1}^{\text{rev}} T}{\epsilon} + \log \frac{\max_t \|\dot{H}\|T^2}{\epsilon^2} \right) \\
&\subseteq O \left(\log L + \log(\|\alpha\|_{\infty,1}^{\text{rev}} T) + \log(\max_t \|\dot{H}\|T^2) + \log 1/\epsilon \right)
\end{aligned} \tag{4.101}$$

for a total number of auxiliary qubits $n \in O(n_{\text{LCU}})$.

Improvements over 1st-order Trotter in query complexity, with our bounds, only appear with very small time variations. We may be able to prove this through better characterizations in the error stemming from H_c . We use Trotter bounds in those calculations, but we may need to be smarter to avoid the limits seen here.

4.5 Discussion

In the circuit-based Hamiltonian simulation community, time dependent Hamiltonians are often treated on separate footing from time independent ones. The main contribution of this project is a new way of thinking about time dependent dynamics that allows us to replace time ordered operator exponentials with ordinary operator exponentials acting on a higher-dimensional finite space. We apply the discretized (t, t') trick and show that it encodes the time ordered exponential for sufficiently large clock sizes. The clock space framework can also be used directly in quantum simulation methods to extend the capabilities of certain quantum algorithms. Specifically, it can be used to extend qubitization to time dependent systems. While in many circumstances it will be more convenient to use a truncated Dyson series simulation method in preference to this approach, our work shows how the use of discrete clock spaces used to construct new quantum simulation algorithms that would otherwise be challenging.

Besides an LCU encoding, natural block encodings of H_c may be possible. For example, a very general input model for $H(t)$ is to take it as a d -sparse matrix with query access to the nonzero entries. This seems quite promising an avenue to take, because then $H_c = C(H) + \Delta$ is $d + 2$ sparse, and there is a natural way to query the entries of H_c . Hence, such a Hamiltonian should immediately simulatable by qubitization (or other quantum walk methods). The trouble is that the largest entry in absolute value $\|H_c\|_{\max}$ of H_c comes from Δ , which is of size $N_c/2T$. This is too large to yield an effective simulation algorithm. Of course, there is something odd about the need to care for the operator norm $\|\Delta\|$, since the typical state being acted on is a Gaussian $|\phi_j\rangle$. Thinking of Δ in frequency space, modes of frequency $\Omega(\sigma^{-1})$ should not be relevant for Gaussian states of width $O(\sigma)$ on the clock register. This suggests that a high-frequency truncation of Δ , say $\tilde{\Delta}$ would act approximately the same on the Gaussians while decreasing the norm. However, there is no guarantee that the modified operator, $\tilde{\Delta}$, is sparse in the basis of clock times. Perhaps considering a reduced clock Hamiltonian $\tilde{H}_{ij} = \langle \phi_i | H_c | \phi_j \rangle$, with all small elements set to zero, would have the sparseness conditions required, along with a subspace norm of $\|\Delta\|_{\phi} \in O(\sigma^{-1})$. Investigating such sparse encodings would make an interesting avenue for future work.

CHAPTER 5

MULTIPRODUCT FORMULAS FOR TIME DEPENDENT SIMULATION

One of the key developments of Chapter 4 is the construction of a discrete clock space which reduces the computation of time dependent Hamiltonians to time independent ones. This reduction provides a useful way to translate techniques and concepts between the time dependent and time independent settings. In this chapter, we investigate one of these connections: a generalization of Multiproduct Formulas (MPFs) for the time dependent setting based on product formula simulations of the clock space. After arguing that such "time dependent MPFs" should form good approximations to the time evolution operator U for sufficiently smooth $H(t)$, we propose an algorithm based on these MPFs. We then provide a rigorous characterization of error in these formulas, and from this derive a query complexity in a natural Hamiltonian input model. Numerical demonstrations are used to validate the effectiveness of time dependent MPFs at achieving high-accuracy simulations.

What we find for the properties of the MPF algorithm (and the qubitization algorithm) is summarized in Table 5.1, where we also display other leading algorithms for time dependent Hamiltonian simulation. Overall, the MPF algorithm has comparable performance to the Dyson method, with strengths and weaknesses on both sides. For example, unlike the Dyson method, the MPF simulation exhibits commutator scaling, meaning that the simulation is perfect for commuting Hamiltonian terms and no time dependence. It also scales as the more favorable L_1 norm of the Hamiltonian rather than the maximum value at a given time, as the Dyson series does. On the other hand, the Dyson series is not concerned with large derivatives, but only the size of H . Overall, time dependent MPF simulation enlarges the collection of available tools for the future practitioner who is looking for the right algorithm for their problem of interest.

This chapter is the subject of ongoing research, particularly with respect to the proof (or disproof) of Conjecture 1. An early preprint has been posted [133] which will be updated as the project reaches completion.

Method	Query Complexity	Auxiliary Qubits	CS?
Trotter [137]	$O(L(\ \Lambda\ _1)^{1+o(1)}/\epsilon^{o(1)})$	0	Yes
QDrift [16]	$O(\ \alpha\ _{1,1}^2/\epsilon)$	0	No
Dyson [89, 75]	$O(\ \alpha\ _{1,\infty}T \log(1/\epsilon))$	$\tilde{O}(\log(\ \dot{\alpha}\ _{1,1}/\epsilon) + \log(\ \alpha\ _{1,\infty}T/\epsilon))$	No
Qubitization*	$\tilde{O}(\ \alpha\ _{\infty,1}^{\text{rev}}T + \log 1/\epsilon)$ + $\max_t \ \dot{H}\ T^2/\epsilon$	$O(\log(L\ \alpha\ _{\infty,1}^{\text{rev}}T/\epsilon))$ + $\log(L \max_t \ \dot{H}\ T^2/\epsilon)$	No
MPF	$\tilde{O}(L\ \Lambda\ _1 \log^2(1/\epsilon))$	$\tilde{O}(\log(L\ \Lambda\ _1\ \dot{\alpha}\ _{\infty,\infty}T^2/\epsilon))$	Yes

Table 5.1 Summary of our results (green) and comparison to leading quantum simulation methods for time dependent Hamiltonians. We assume that $H = \sum_{j=1}^L \alpha_j(t)U_j$ for Hermitian unitaries U_j and real-valued $\alpha_j(t)$. Λ is a positive, time dependent function with dimensions of H , and quantifies the size of α_j and its derivatives (see Definition 3). $\|\alpha\|_{p,q}$ refers to a nested vector- p and functional- q norm for the coefficients $\alpha = (\alpha_j)_{j=1}^L$, and $\|\alpha\|_{p,q}^{\text{rev}}$ indicates these are taken in the reverse order. Commutator scaling (CS) here means the simulation error vanishes in the limit where H is time independent and $[U_j, U_k] = 0$ for all $j, k \in [L]$.

5.1 Introduction and Background

Multiproduct formulas (MPFs) are a generalization of the celebrated product formulas and span two of the pillars of quantum simulation: product formula and LCU methods. The aim of the MPF is to approximate the time evolution operator U as a linear combination of lower-order Trotter formulas, in such a way that higher order errors are cancelled [28, 24, 90]. They are, fundamentally, nothing more than a Richardson extrapolation of a product formula \mathcal{P} to Trotter step size $s \rightarrow 0$. This summation is done to address the primary deficiency of product formulas: the cost of constructing a high order product formula is exponentially large. This is not only true of the well-known Suzuki-Trotter formulas, but any similar construction, due to the need to cancel error terms that grow exponentially in the number of products considered. In contrast, since the MPF is a sum of product formula approximations, the number of error terms of a given order does not grow exponentially. This allows us to approximate the quantum dynamics using polynomially many, rather than exponentially many, operator exponentials.

The current central result concerning the use of MPFs for time independent simulation is

provided by the following theorem of Low, Kliuchnikov, and Wiebe [90].

Theorem 5.1.1 (Time independent MPFs (Theorem 1 of [90])). *Let H be a bounded, time independent Hamiltonian, and let $U_2(t)$ be the 2nd-order Suzuki-Trotter formula for the time evolution operator $U(t) = e^{-iHt}$. Let $a = (a_1, a_2, \dots, a_m) \in \mathbb{R}^m$ and $\vec{k} = (k_1, k_2, \dots, k_m) \in \mathbb{Z}_+^m$. There exist choices of a and \vec{k} such that the multiproduct formula*

$$U_{2,m}(t) := \sum_{j=1}^m a_j U_2^{k_j}(t/k_j)$$

is order $2m$ and satisfies

$$\max_j k_j \in O(m^2), \quad \|a\|_1 \in O(\text{polylog}(m)).$$

As a caution, we remark that, despite notation, the MPF $U_{2,m}$ is not generally unitary for $m > 1$, though when suitably constructed it will approximate the unitary U (hence be approximately unitary). The proof of Theorem 5.1.1 may be found in [90], but at a high level, the MPF $U_{2,m}$ is a Richardson extrapolation of U_2 with respect to the Trotter step size parameter $1/k$. Such an extrapolation is possible for arbitrary m because there exists an error series [18]

$$U_2^k(t/k) - U(t) = \sum_{j=1}^{\infty} E_{2j+1} \frac{t^{2j+1}}{k^{2j}} \quad (5.1)$$

with E_{2j+1} independent of k (but not t generically). The existence of this series suffices for a $1/k \rightarrow 0$ Richardson extrapolation [119]. In particular, cancellation occurs for coefficients a_j satisfying the following Vandermonde linear system.

$$\begin{pmatrix} 1 & \cdots & 1 \\ k_1^{-2} & \cdots & k_m^{-2} \\ \vdots & \ddots & \vdots \\ k_1^{-2m+2} & \cdots & k_m^{-2m+2} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (5.2)$$

Though the matrix is ill-conditioned, this is irrelevant to numerical stability, as the inverse Vandermonde matrix admits an analytic solution that may be reasoned from the theory of polynomial interpolation. What matters for our application is the one-norm $\|a\|_1$ of the coefficients, which

serves as our "condition number" because of how it amplifies small errors in the base formula U_2 [90]. The content of Theorem 5.1.1 is that Trotter steps \vec{k} may be chosen such that $\|a\|_1$ is not too large. For time-ordered U , the analysis of [18] does not carry over, although reasonable "time dependent" MPFs can be defined heuristically. One of our motivations in constructing a clock space is to be able to eliminate time ordering and, in this setting, show these formulas work.

As discussed in [90], specific choices of k_j can be found numerically to minimize $\|a\|_1$, and this may be the best approach in practice. However, for our analytical results it will be most appropriate to utilize the specific k_j chosen in their constructive proof of well-conditioned MPFs. Thus, for all results we will take the powers k_j as follows.

$$k_j = \left[\frac{\sqrt{8}m}{\pi} \left| \sin \left(\frac{\pi(2j-1)}{8m} \right) \right|^{-1} \right], \quad j = 1, \dots, m \quad (5.3)$$

We will use these same coefficients even in the time dependent MPFs to be introduced in the subsequent section. For error analysis, it will be useful to have simple, concrete bounds on k_j . We can achieve this by noting that $\sin(x) \leq x$ and $\sin(x) \geq 4x/5$ for $x \in [0, 1]$. This gives the lower bound

$$k_j \geq \left[\frac{8^{3/2}m^2}{\pi^2(2j-1)} \right] \geq \left[\frac{8^{3/2}m^2}{\pi^2(2m-1)} \right] > \frac{\sqrt{128}m}{\pi^2} > m \quad (5.4)$$

and the upper bound

$$k_j \leq \left[\frac{5 \times 8\sqrt{8}m^2}{4(2j-1)\pi^2} \right] \leq \left[\frac{5 \times 8\sqrt{8}m^2}{4\pi^2} \right] < 3m^2. \quad (5.5)$$

Note the consistency of (5.5) with the big- O scaling of Theorem 5.1.1.

5.2 Definition and Effectiveness

Multiproduct formulas have already been considered extensively in the Hamiltonian simulation community [24, 45, 141], however, they have yet to be seriously considered for use in time dependent Hamiltonian simulations. Because U generally has time-ordering, the techniques used in [18] involving Baker-Campbell-Hausdorff-type expansions do not carry over directly. An approach based instead on the Magnus expansion might be expected to work in its place, but no subset of terms in the expansion represents the exact evolution separated from error terms. Without this

generalization, MPFs cannot be applied to interaction picture algorithms as well as simulations of physical systems that have intrinsic time dependence.

It is rather easy to propose a reasonable generalization of the MPFs of Theorem 5.1.1 that would be expected to work well in the time dependent case. Simply replace the k_j th power with a sequence of k_j unitaries at each time slice. We will present this definition shortly. First, as a preliminary step, we want to clearly define a notion of approximation order of two-parameter operator functions (such as the propagator $U(t, t_0)$) that will suit our purposes.

Definition 1. For finite-dimensional \mathcal{H} , let $L : [0, T]^2 \rightarrow L(\mathcal{H})$. We say that $L_p : [0, T]^2 \rightarrow L(\mathcal{H})$ is a p th-order approximation to L if, for all $t \in [0, T)$,

$$\|L(t + \tau, t) - L_p(t + \tau, t)\| \in O(\tau^{p+1})$$

where τ is taken asymptotically to 0.

Observe that this aligns with the regular notion of p th order formulas when considered as a function of a single variable τ and fixed t . With p th order approximants defined, we now propose a generalization of MPFs for two-parameter operators such as the general propagator U .

Definition 2 (Time Dependent Multiproduct Formulas). For finite dimensional \mathcal{H} and $L : [0, T]^2 \rightarrow L(\mathcal{H})$, let $L_p : [0, T]^2 \rightarrow L(\mathcal{H})$ be a p th-order formula for L . Given $m \in \mathbb{Z}_+$, $\vec{k} \in \mathbb{Z}_+^m$, and $a \in \mathbb{R}^m$, define the time dependent Multiproduct Formula $L_{m,p} : [0, T]^2 \rightarrow L(\mathcal{H})$ to be

$$L_{m,p}(t, t_0) := \sum_{j=1}^m a_j L_p^{(k_j)}(t, t_0)$$

where

$$L_p^{(k)}(t, t_0) := \prod_{\ell=0}^{k-1} L_p(t_{\ell+1}, t_\ell)$$

and $t_\ell = t_0 + (t - t_0)\ell/k$.

The choice to take the t_ℓ as equally spaced is not entirely coincidental, for the same reason that, in the time independent setting, we take $U_2^k(t/k)$ instead of, say,

$$\prod_{j=1}^k U_2(s_j t) \tag{5.6}$$

where $s = (s_1, \dots, s_k)$ is a probability vector. Taking a simple power of k makes reasoning with the BCH expansion possible.

While these definitions are likely applicable in more general contexts (such as classical time dependent symplectic dynamics), our interest in simulation means we will consider $L = U$ to be a time evolution operator, satisfying all the corresponding properties. Moreover, we will assume H is in the so-called linear combination of Hamiltonians (LCH) form

$$H = \sum_{i=1}^L H_i(t) \quad (5.7)$$

which is suitable for product formula simulations. Later we will make the assumption that the exponentials of each $H_j(t)$ can be efficiently computed, a standard assumption. Because we utilize the well-conditioning results of [90], we want the base formula to be 2nd order and symmetric. Thus, we will take $L_p = U_2$ to be the *2nd order symmetric midpoint formula*

$$U_2(t + \tau, t) := \prod_{i=L}^1 \exp \left\{ -iH_i \left(t + \frac{\tau}{2} \right) \tau \right\} \prod_{i=1}^L \exp \left\{ -iH_i \left(t + \frac{\tau}{2} \right) \tau \right\}, \quad (5.8)$$

where H is the Hamiltonian generating U . That U_2 is second-order can be seen from Taylor expanding the Dyson series of U about $\tau = 0$ (H must be at least twice differentiable). Moreover, U_2 is time-reversal symmetric in the same sense as U : $U_2(t, t_0) = U_2(t_0, t)^\dagger$. This gives the nice property that the error series for $U(t + \tau, t) - U^{(k)}(t + \tau, t)$ has only even terms, such that higher order formulas can be reached with approximately half the number of summands. Thus, from now on we will be interested in the MPF

$$U_{2,m}(t, 0) = \sum_{j=1}^m a_j U_2^{(k_j)}(t, 0) \quad (5.9)$$

for the rest of this chapter.

We finally turn to the question of whether the time dependent MPFs of Definition 2 may be constructed for improved approximants. At the beginning of this section, we mentioned the difficulty presented by time-ordering in adopting the techniques from [18]. The reader of the previous chapter may recognize that clock spaces may be used to remove time ordering, circumventing the issue. However, when the clock variable t is continuous, the shift term $-E$ in the clock Hamiltonian is

an unbounded operator, complicating a BCH-type analysis. We conjecture, and provide a heuristic argument, that time dependent MPFs indeed boost the approximation order for sufficiently smooth Hamiltonians.

Conjecture 1. *Let $H = \sum_{i=1}^L H_i(t)$, and let*

$$U_2(t + \tau, t) = \prod_{i=L}^1 e^{-iH_i(t+\tau/2)\tau} \prod_{i=1}^L e^{-iH_i(t+\tau/2)\tau}$$

be the symmetric, 2nd order Trotterized midpoint formula. Suppose each H_i is $2m + 1$ time differentiable. Then the time dependent multiproduct formula $U_{2,m}(t + \tau, t)$ with base formula U_2 approximates $U(t + \tau, t)$ to order $2m$ in t .

We now discuss a potential path to proof of this conjecture. Without loss of generality, we take $t = 0$. Let $k \in \mathbb{Z}_+$, and consider a sequence of discrete clock constructions on interval $[0, \tau]$, with parameters $(N_p(\ell), N_q(\ell), \sigma(\ell))$, such that k always divides $N_c = N_p N_q$, and such that the limit reproduces the dynamics of $H(t)$ on the main register, as per Theorem 4.3.6. Consider one of the elements of this sequence. Using the form of H given in the conjecture statement, we may write

$$C(H) = \sum_{i=1}^L C(H_i). \quad (5.10)$$

Thus, the clock Hamiltonian H_c admits the following 2nd order symmetric Trotterization.

$$V_2(\tau) = e^{i\Delta\tau/2} \left(\prod_{i=L}^1 e^{-iC(H_i)\tau/2} \prod_{i=1}^L e^{-iC(H_i)\tau/2} \right) e^{i\Delta\tau/2} \quad (5.11)$$

From [18], we have that

$$V(\tau) - V_2^k(\tau/k) = \sum_{j=1}^{m-1} \mathcal{E}_{2j+1}(\tau) \frac{\tau^{2j+1}}{k^{2j}} + \mathcal{E}(\tau, k) \quad (5.12)$$

where $\mathcal{E} \in O(\tau^{2m+1})$ is analytic in τ . Thus the standard, well-conditioned multiproduct formula $V_{2,m}$ of Theorem 5.1.1 with base formula V_2 satisfies

$$V(\tau) - V_{2,m}(\tau) = \sum_{j=1}^m a_j \mathcal{E}(\tau, k_j). \quad (5.13)$$

We now wish to look at the action on the main register. Applying equation (5.13) to the state $|\psi\rangle|\phi_0\rangle$ of the full register, where $|\psi\rangle$ is arbitrary, and then taking the trace Tr_c over the clock register, one obtains

$$\text{Tr}_c(V(\tau)|\psi\rangle|\phi_0\rangle) - \text{Tr}_c(V_{2,m}(\tau)|\psi\rangle|\phi_0\rangle) = \sum_{j=1}^m a_j E(\tau, k_j)(|\psi\rangle) \quad (5.14)$$

where $E(\tau, k)$ is a linear map on the main register defined by

$$E(\tau, k)(|\psi\rangle) := \text{Tr}_c(\mathcal{E}(\tau, k)|\psi\rangle|\phi_0\rangle). \quad (5.15)$$

The above holds for every clock space in the sequence defined by $(N_p(\ell), N_q(\ell), \sigma(\ell))$. Taking the limit as $\ell \rightarrow \infty$ of equation (5.14) we may pass the limits through the finite sums and scalar multiplications

$$\lim_{\ell \rightarrow \infty} \text{Tr}_c(V(\tau)|\psi\rangle|\phi_0\rangle) - \lim_{\ell \rightarrow \infty} \text{Tr}_c(V_{2,m}(\tau)|\psi\rangle|\phi_0\rangle) = \sum_{j=1}^m a_j \lim_{\ell \rightarrow \infty} E(\tau, k_j)|\psi\rangle \quad (5.16)$$

provided that these limits exist. Indeed, by Theorem 4.3.6,

$$\text{Tr}_c(V(\tau)|\psi\rangle|\phi_0\rangle) = U(\tau, 0)|\psi\rangle. \quad (5.17)$$

As for the MPF, taking k steps of the Trotterization, we *should* find that

$$\lim_{\ell \rightarrow \infty} \text{Tr}_c(V_2(\tau/k)^k \psi |\phi_0\rangle_c) = U_2^{(k)}(\tau, 0)|\psi\rangle \quad (5.18)$$

though this must be shown. This shouldn't be too hard, as the idea is clear: perform a sequence of clock shifts followed by 2nd order Trotter on the main register. By passing the limit through the multiproduct sum,

$$\lim_{\ell \rightarrow \infty} \text{Tr}_c(V_{2,m}(\tau, 0)|\psi\rangle|\phi_0\rangle_c) = U_{2,m}(\tau, 0)|\psi\rangle. \quad (5.19)$$

It remains to show that the limit $\lim_{\ell} E(\tau, k)$ exists, and moreover is in $O(\tau^{2m+1})$. This is where the main challenge lies. To show that the limit of a sequence with terms of order $O(\tau^{2m+1})$ is also $O(\tau^{2m+1})$, we can show that the $2m + 1$ derivative is bounded at $\tau = 0$. Unfortunately, our current clock constructions have the width σ of the clock state shrinking to infinity, which means

the derivatives grow as well. If a different clock construction can be provided where the clock state can have width $\sigma \in O(1)$, a bound can be placed and thus the limit will be $O(\tau^{2m+1})$.

Current ongoing work is being undertaken to fill in the gaps of the previous argument. However, the numerics of Section 5.7 strongly suggest that the time dependent MPFs indeed work as expected. Moreover, the form of the time-dependent MPF of Definition 2 can be obtained by a naive Trotterization of the continuous clock space, which is very suggestive that, beyond formal issues, the approach is reasonable. Thus, we proceed with the assumption Conjecture 1 is true.

5.3 Time Dependent Multiproduct Simulation

Having argued that good time dependent MPFs exist, we now propose an algorithm for Hamiltonian simulation using these formulas. We will provide some accompanying discussion to explain our choices, and at the end we will more directly state the approach.

In order to present a concrete computational model for our Hamiltonian, we further specify that our LCH Hamiltonian $H(t)$ is of the form

$$H(t) = \sum_{i=1}^L \alpha_i(t) H_i \tag{5.20}$$

where each $\alpha_i(t) \in \mathbb{R}$ is assumed $2m+1$ differentiable for an m -term MPF. Without loss of generality we take $\|H_i\| \leq 1$.

From the onset, there are a couple of choices to make. The MPFs, in principle, could approximate the entire interval $[0, T]$ provided that the Trotter steps k_i are sufficiently large. However, this has several disadvantages. First, there is no flexibility to treat some subintervals of $[0, T]$ as more difficult than others and allocate resources appropriately. Second, the well-conditioned scheme of [90] would have to be abandoned or modified to accommodate larger \vec{k} . Instead, we divide $[0, T]$ into a mesh of r subintervals, not necessarily uniform, but rather constructed to account for more difficult parts of the simulation. We provide a greedy algorithm for constructing such a mesh in Section 5.9. The algorithm requires a computable Λ_{2m+1} -bound to work (see Definition 3), however, a practitioner might prefer a more heuristic approach to constructing the time mesh. For the moment, we will simply say that, given t_i , the next time point t_{i+1} is incremented roughly as

$1/\Lambda_{2m+1}(t)$ for t in a neighborhood of t_i , where Λ_{2m+1} is a positive real-valued function of H and its derivatives that grows for larger or faster fluctuating H .

Once the mesh points t_0, t_1, \dots, t_r are determined, a time dependent MPF is performed over each subinterval $[t_i, t_{i+1}]$ in sequence. We assume the MPF is implemented using the LCU technique. The base midpoint formula U_2 must be implemented by some scheme which depends on the structure of $H(t)$, though the approximating unitary W_2 should be at least 2nd-order and preserve the time-reversal symmetry of U_2 (and U). It is known that product formulas exhibit commutator scaling, meaning that, in the limit where all H_j commute pairwise and all α_j are constant functions, the simulation error goes to zero. Hence, the MPF will also inherit this desirable property. It is for precisely this reason that, in Table 5.1, we claim our MPFs exhibit commutator scaling.

Let us now supply our procedure for the multiproduct simulation. Given fundamental parameters, $[0, T]$, ϵ , and a description of $H(t)$:

1. Compute a Λ_{2m+1} bound (Definition 3) for some m larger than the expected number of MPF terms. This is more or less a bound on the generalized "size" of $H(t)$.
2. Construct a time mesh of r steps using the algorithm of Section 5.9.
3. Perform a sequence of MPFs over each time slice, with 2nd order base formula W_2 approximating the midpoint formula.

Not much more about the parameter choices, such as m or r , can be said without an error analysis. This will be supplied in the following section. We then return to the question of algorithmic cost via a query model.

5.4 Error Analysis

In this section, we analyze the errors arising between the exact unitary U and the MPF approximation \tilde{U} given by

$$\tilde{U}(T, 0) = \prod_{i=1}^r U_{2,m}(t_i, t_{i-1}). \quad (5.21)$$

This analysis will ignore hardware imperfections and decoherence, assume that U_2 is implemented perfectly, and assume exact coefficients a_j . In the query complexity analysis of Section 5.6 we will

consider additional algorithmic errors arising from a more precise specification of the Hamiltonian input model.

We introduce a useful definition to quantify errors succinctly. It is well understood that MPFs, like regular product formulas, have smoothness requirements to ensure convergence. To quantify errors and costs of MPFs, we provide a metric which captures the "size" of H and its derivatives at each point in time, in order to characterize the difficulty of simulation.

Definition 3. Let $H(t) = \sum_{i=1}^L \alpha_i(t)H_i$ be a time dependent, finite-dimensional Hamiltonian with H_i Hermitian and $\alpha_i(t) \in \mathbb{R}$ having $n \in \mathbb{N} \cup \{\infty\}$ continuous derivatives. For each i define a $\Lambda_{i,n}$ -bound ("Lambda i n bound") as any continuous function $\Lambda_{i,n} : [0, T] \rightarrow \mathbb{R}_+$ satisfying the following bounds with respect to H and its derivatives

$$\Lambda_{i,n}(t) \geq \sup_{j \in [n]} \sqrt[j+1]{\|\alpha_i^{(j)}(t)\|} \quad \forall t \in [0, T]$$

where $f^{(n)}$ represents an n th derivative of f , and $[n] := \{j \in \mathbb{N} \mid j \leq n\}$. Assuming such bounds exist for all $i = 1, \dots, L$, we say that $H(t)$ is Λ_n -bounded. We further say that $H(t)$ is Λ_n -boundable if it admits some Λ_n -bound. For convenience, we define $\Lambda_i \equiv \Lambda_{i,\infty}$. We also define a Λ_n bound as any continuous on $[0, T]$ satisfying

$$\Lambda_n(t) \geq \max_{i \in [L]} \Lambda_{i,n}(t).$$

For near-constant $\alpha_i(t)$, $\Lambda_{i,n}$ is simply an upper bound on $|\alpha_i|$, while for rapid oscillations the derivative terms will dominate. Observe that for finite n , our assumptions imply that $\Lambda_{i,n}(t)$ exists (H is $\Lambda_{i,n}$ -boundable), since $|\alpha_i^{(j)}|$ is continuous on a compact interval and hence a bounded function. Also in the finite case, the supremum may be replaced with a simple max, and $\Lambda_{i,n}(t)$ may be taken as equal to the right hand side because it is the maximum of a finite set of continuous functions, which is continuous. For this "minimal choice," $\Lambda_{i,n}(t)$ is a nondecreasing sequence in n . For each n , there also exists a $\Lambda_{i,n}$ that is constant in t . Allowing $\Lambda_{i,n}$ to vary in time, however, takes into consideration the possibility that the expense of simulating H will vary with time. We note that $\Lambda_{i,n}$ -bounds are additive in the sense that, for $H(t)$ and $G(t)$ admitting $\Lambda_{i,n}^H$ and $\Lambda_{i,n}^G$ -bounds, respectively, $\Lambda_{i,n}^H + \Lambda_{i,n}^G$ is a $\Lambda_{i,n}$ -bound on $H + G$.

In contrast to finite n , the existence of a $\Lambda_{i,\infty}$ -bound is not guaranteed, and amounts to the assumption that the derivatives of H grow at most exponentially for asymptotically large j and fixed t . There are smooth, even analytic functions which do not satisfy this, many of which are physically interesting. A simple example is a Gaussian pulse

$$\alpha(t) = e^{-t^2} \quad (5.22)$$

whose derivatives, generating the Hermite polynomials, grow factorially with n at $t = 0$. Other interesting cases, such as harmonic oscillations or exponential growth and decay, do admit a Λ -bound. Despite these restrictions, we adopt this approach for simplicity and in order to facilitate comparison with prior work on general-order Suzuki-Trotter formulas [137]. Admittedly, a modification of Definition 3 to be an upper bound on

$$\max_j j^{-1} \sqrt{\|\alpha_i^{(j)}(t)\|} \quad (5.23)$$

would expand the class of functions admitting Λ -bounds to analytic functions (though not generic smooth functions).

We now begin the error analysis of (5.21) in earnest. From a triangle inequality the error can be bounded as the error in each step.

$$\|U(T, 0) - \tilde{U}(T, 0)\| \leq \sum_{i=1}^r \|U(t_i, t_{i-1}) - U_{2,m}(t_i, t_{i-1})\| \quad (5.24)$$

Therefore, to ensure an error at most ϵ , it suffices that each subinterval has error at most ϵ/r . We thus focus a single subinterval. An upper bound on this error is supplied by the following theorem, which the main technical result of this section.

Theorem 5.4.1. *Let $H : [t_0, t_1] \rightarrow \text{Herm}(\mathcal{H})$ be a time dependent Hamiltonian on finite-dimensional \mathcal{H} with $2m + 1$ continuous derivatives on $[t_0, t_1]$ and Λ_{2m+1} -bound. Suppose further that*

$$eL \max_{\tau \in [t_0, t_1]} \Lambda_{2m+1}(\tau)(t_1 - t_0) < 1.$$

Then for any $m \in \mathbb{Z}_+$ there exists $\vec{k} \in \mathbb{Z}_+^m$ and $a \in \mathbb{R}^m$ such that

$$\|U(t_1, t_0) - U_{2,m}(t_1, t_0)\| < \frac{\|a\|_1}{\sqrt{\pi m}} \left(5L \max_{\tau \in [t_0, t_1]} \Lambda_{2m+1}(\tau)(t_1 - t_0) \right)^{2m+1}$$

and $\|a\|_1 \in O(\log(m))$.

Observe that convergence of the above error bound to zero as $m \rightarrow \infty$ is conditioned on sufficiently small $t_1 - t_0$. This is potentially unsurprising, as the Suzuki-Trotter formulas also do not provide an unconditionally converging sequence of approximations to the time evolution operator. Note as well the parallel roles between m and the Suzuki-Trotter order k in reducing the error. In our case, however, we shall see that the simulation cost increases only polynomially in m , whereas for product formulas the cost is necessarily exponential in k .

The term $\|a\|_1/\sqrt{\pi m}$ is $o(1)$ for large m and can be more or less ignored. Unfortunately, the Λ_{2m+1} scales as the worst α_i times the number of terms L , which seems too cynical. However, improving on this may greatly complicate the proof of the error bound. Theorem 5.4.1 will be the important result that informs the algorithmic choices and complexity analysis of subsequent sections.

Having characterized the error on a single subinterval of $[0, T]$, the full error over r subintervals may be found simply using (5.24).

We prove Theorem 5.4.1 using a similar strategy to that used to provide error estimates for the Suzuki-Trotter formulas [14, 137, 26]. As H is continuously differentiable at least $2m + 1$ times, $U_{2,m}$ is a valid extrapolant under our , and cancels the first m terms in the error series. We can thus express the difference $U_{2,m} - U$ using the integral Taylor remainder formulas

$$U_{2,m}(t, t_0) - U(t, t_0) = \mathcal{R}_{2m} - \mathcal{R}_{2m} \tag{5.25}$$

with

$$\mathcal{R}_{2m} := \frac{1}{2m!} \int_{t_0}^t (t - \tau)^{2m} U^{(2m+1)}(\tau, t_0) d\tau \tag{5.26}$$

$$\mathcal{R}_{2m} := \frac{1}{2m!} \int_{t_0}^t (t - \tau)^{2m} U_{2,m}^{(2m+1)}(\tau, t_0) d\tau, \tag{5.27}$$

where $U^{(n)}$ refers to derivatives in the first argument. By the triangle inequality,

$$\|U_{2,m}(t, t_0) - U(t, t_0)\| \leq \|\mathcal{R}_{2m}\| + \|\mathcal{R}_{2m}\| \tag{5.28}$$

and we upper bound each remainder in separate lemmas.

The easier bound is \mathcal{R}_{2m} , so we begin with the corresponding lemma.

Lemma 5.4.2. *The remainder term \mathcal{R}_{2m} in equation (5.27) satisfies*

$$\|\mathcal{R}_{2m}\| < \frac{1}{2\sqrt{\pi m}} \left(2L \max_{\tau \in [t_0, t]} \Lambda_{2m+1}(\tau)(t - t_0) \right)^{2m+1}.$$

Proof. Recall that U , as the exact propagator, satisfies the Schrödinger equation (2.16). Higher derivatives can easily be found through repeated application of the product rule. The result will be a polynomial in the derivatives of H times U itself. Under the spectral norm, using the triangle and submultiplicative properties, the ordering of terms doesn't matter, and therefore equivalent to the expression one gets taking derivatives of a scalar exponential. Noting that $\|U\| = 1$, the resulting polynomial is the complete exponential Bell polynomial from Faà di Bruno's formula (see Section 2.8). Letting $n = 2m + 1$, we have

$$\|\partial_t^n U(t, t_0)\| \leq Y_n \left(\|H(t)\|, \|\dot{H}(t)\|, \dots, \|H^{(n-1)}(t)\| \right). \quad (5.29)$$

From the definition of $\Lambda_{i,n}$, we have

$$\begin{aligned} \|H^{(j)}(t)\| &\leq \sum_{i=1}^L |\alpha_i^{(j)}(t)| \\ &\leq \sum_i \Lambda_{i,n}(t)^{j+1} \\ &\leq (L\Lambda_n(t))^{j+1} \end{aligned} \quad (5.30)$$

and since the Bell polynomials Y_n are monotonic in each argument,

$$\begin{aligned} Y_n \left(\|H\|, \|\dot{H}\|, \dots, \|H^{(n-1)}\| \right) &\leq Y_n(L\Lambda_n(t), (L\Lambda_n(t))^2, \dots, (L\Lambda_n(t))^n) \\ &= (L\Lambda_n(t))^n b_n \end{aligned} \quad (5.31)$$

where b_n are the Bell numbers (Section 2.8). Thus,

$$\|\partial_t^n U(t, t_0)\| \leq (L\Lambda_n(t))^n b_n. \quad (5.32)$$

Finally, returning to the bound on \mathcal{R}_{2m} , we have from the integral triangle inequality that

$$\begin{aligned}\|\mathcal{R}_{2m}\| &\leq \frac{1}{(2m)!} \int_{t_0}^t (t-\tau)^{2m} \|\partial_\tau^{2m+1} U(\tau, t_0)\| d\tau \\ &\leq \frac{1}{(2m)!} \int_{t_0}^t (t-\tau)^{2m} (L\Lambda_{2m+1}(\tau))^{2m+1} b_{2m+1} d\tau\end{aligned}\tag{5.33}$$

where we made use of equation (5.32). This, in turn, can be bounded by maximizing Λ_{2m+1} over $[t_0, t]$.

$$\begin{aligned}\|\mathcal{R}_{2m}\| &\leq \frac{b_{2m+1}}{(2m)!} (L \max_{\tau \in [t_0, t]} \Lambda_{2m+1}(\tau))^{2m+1} \int_{t_0}^t d\tau (t-\tau)^{2m} \\ &\leq \frac{b_{2m+1}}{(2m+1)!} \left(L \max_{\tau \in [t_0, t]} \Lambda_{2m+1}(\tau) (t-t_0) \right)^{2m+1}\end{aligned}\tag{5.34}$$

Finally, we upper bound the prefactor using a Stirling bound and bounds from [13] on the Bell numbers. For all $m \in \mathbb{Z}_+$,

$$\begin{aligned}\frac{b_{2m+1}}{(2m+1)!} &< \frac{\left(\frac{0.792(2m+1)}{\log(2m+2)}\right)^{2m+1}}{\sqrt{2\pi(2m+1)} \left(\frac{2m+1}{e}\right)^{2m+1}} \\ &= \frac{1}{\sqrt{2\pi(2m+1)}} \left(\frac{.792e}{\log(2m+2)}\right)^{2m+1}.\end{aligned}\tag{5.35}$$

Plugging this into equation (5.34),

$$\begin{aligned}\|\mathcal{R}_{2m}\| &< \frac{1}{\sqrt{2\pi(2m+1)}} \left(\frac{0.792e}{\log(2m+2)} L \max_{\tau \in [t_0, t]} \Lambda_{2m+1}(\tau) (t-t_0)\right)^{2m+1} \\ &< \frac{1}{2\sqrt{\pi m}} \left(2L \max_{\tau \in [t_0, t]} \Lambda_{2m+1}(\tau) (t-t_0)\right)^{2m+1}.\end{aligned}\tag{5.36}$$

The last line is the result of the lemma. \square

We now state the bound on the Taylor R_{2m} for the time dependent MPF.

Lemma 5.4.3. *In the notation above, suppose that*

$$eL \max_{\tau \in [t_0, t_1]} \Lambda_{2m+1}(\tau) (t_1 - t_0) < 1.$$

Then the remainder term R_{2m} in equation (5.27) satisfies

$$\|\mathcal{R}_{2m}\| < \frac{\|a\|_1}{2\sqrt{\pi m}} \left(5L \max_{\tau \in [t_0, t]} \Lambda_{2m+1}(\tau) (t-t_0)\right)^{2m+1}.$$

The proof is more technical than the previous bound, and is given at the end of this section. First, we quickly prove Theorem 5.4.1 assuming the truth of the above Taylor remainder lemmas.

Proof of Theorem 5.4.1. First, we note that $\|a\|_1 \geq 1$, since a necessarily satisfies $\sum_j a_j = 1$ from the Vandermonde constraints (5.2). From equation (5.28), the error $\|U(t, t_0) - U_{2,m}(t, t_0)\|$ is bounded by the sum of the remainder upper bounds derived in Lemmas 5.4.3 and 5.4.2. Comparing the two, we see that R_{2m} dominates \mathcal{R}_{2m} for all $m \geq 1$. We thus take twice the larger as an upper bound

$$\begin{aligned} \|U(t, t_0) - U_{2,m}(t, t_0)\| &< 2\|R_{2m}\| \\ &< \frac{\|a\|_1}{\sqrt{\pi m}} \left(5L \max_{\tau \in [t_0, t]} \Lambda_{2m+1}(\tau)(t - t_0) \right)^{2m+1}. \end{aligned} \quad (5.37)$$

This completes the proof. \square

To prove Lemma 5.4.3, we will first need a technical lemma that bounds the size of ordinary exponentials of time dependent matrices.

Lemma 5.4.4. *Let $A(t)$ be an anti-Hermitian valued function of $t \in \mathbb{R}$ with n bounded derivatives. Then*

$$\|d_t^n e^{A(t)}\| \leq Y_n \left(\|d_t A(t)\|, \|d_t^2 A(t)\|, \dots, \|d_t^n A(t)\| \right)$$

where Y_n is the complete exponential Bell polynomial.

In the scalar case, Faà di Bruno's bound is an exact expression (see Section 2.8), so the content of our result is that a corresponding bound holds even in the non-scalar case. The exponential disappears because $e^{A(t)}$ is unitary.

Proof of Lemma 5.4.4. From the Trotter product theorem, we have

$$\partial_t^n \exp(A(t)) = \partial_t^n \lim_{r \rightarrow \infty} (\exp(A(t)/r))^r. \quad (5.38)$$

Using the fact that the series converges uniformly, we may interchange the order of differentiation and the limit. This leads to

$$\|\partial_t^n \exp(A(t))\| \leq \lim_{r \rightarrow \infty} \sum_S \binom{n}{s_1, \dots, s_r} \prod_{q=1}^r \|\partial_t^{s_q} \exp(A(t)/r)\|. \quad (5.39)$$

Here the sum over S is constrained such that $s_j \geq 0$ and $s_1 + \dots + s_r = n$. Then using Taylor's theorem we have

$$\|\partial_t^{s_q} \exp(A(t)/r)\| \leq \frac{\|A^{(s_q)}(t)\|}{r} + O(1/r^2). \quad (5.40)$$

for $s_q > 0$, where the $O(1/r^2)$ terms will vanish as $r \rightarrow \infty$. The $s_q = 0$ case has upper bound 1 by unitarity. Hence, put together,

$$\|\partial_t^n \exp(A(t))\| \leq \lim_{r \rightarrow \infty} \sum_S \binom{n}{s_1, \dots, s_r} \prod_{q=1}^r \left(\frac{\|A^{(s_q)}(t)\| (1 - \delta_{s_q,0})}{r} + \delta_{s_q,0} \right). \quad (5.41)$$

Now let us define a scalar function $a(x)$ defined for x in a neighborhood of t such that, for any k such that $0 \leq k \leq n$,

$$a^{(k)}(t) = \|A^{(k)}(t)\| (1 - \delta_{k,0}) \quad (5.42)$$

for a particular $x = t$. Such a function can be seen to exist by considering the n th degree Taylor polynomial. We may apply the standard Faà di Bruno formula (2.49) to a , so that

$$\partial_x^n e^{a(x)} \Big|_{x=t} = e^{a(t)} Y_n(\|A^{(1)}(t)\|, \dots, \|A^{(n)}(t)\|) = Y_n(\|A^{(1)}(t)\|, \dots, \|A^{(n)}(t)\|). \quad (5.43)$$

On the other hand we can split $a(t)$ into r steps and compute the n th derivative, just as for the Trotter product theorem.

$$\partial_x^n e^{a(x)} \Big|_{x=t} = \lim_{r \rightarrow \infty} \sum_S \binom{n}{s_1, \dots, s_r} \prod_{q=1}^r \left(\frac{\|A^{(s_q)}(\Delta t)\| (1 - \delta_{s_q,0})}{r} + \delta_{s_q,0} \right) \quad (5.44)$$

By comparing expressions (5.41) and (5.44), we see that

$$\|\partial_t^n \exp A(t)\| \leq \partial_x^n e^{a(x)} \Big|_{x=t} \quad (5.45)$$

and applying (5.43), we reach our desired bound Faà di Bruno bound.

$$\|\partial_t^n \exp(A(t))\| \leq Y_n(\|A^{(1)}(t)\|, \dots, \|A^{(n)}(t)\|) \quad (5.46)$$

We evaluate the derivatives of $A(t)$, and express them in terms of the derivatives of the Hamiltonian, $H^{(j)}$ (for simplicity, we leave off the evaluation point. The derivative is with respect to the Hamiltonian's single argument). The result is

$$\partial_t^j A(t) = \frac{-i}{k} \left[\left(\frac{q-1/2}{k} \right)^j (t-t_0) H^{(j)} + j \left(\frac{q-1/2}{k} \right)^{j-1} H^{(j-1)} \right]. \quad (5.47)$$

Employing the Λ_n -bound from Definition 3, we have that

$$\begin{aligned} \|\partial_t^j A(t)\| &\leq \frac{1}{k} \left[\left(\frac{q-1/2}{k} \right)^j (t-t_0) \Lambda_{n,q}^{j+1} + j \left(\frac{q-1/2}{k} \right)^{j-1} \Lambda_{n,q} q^j \right] \\ &= \left(\frac{q-1/2}{k} \right)^j \Lambda_{n,q}^j \left[\frac{j}{q-1/2} + \frac{1}{k} (t-t_0) \Lambda_{n,q} \right]. \end{aligned} \quad (5.48)$$

Here,

$$\Lambda_{n,q} := \max_{\tau \in I_q} \Lambda_n(\tau) \quad (5.49)$$

and $I_q = [t_0 + (q-1)(t-t_0)/k, t_0 + q(t-t_0)/k]$ is the q th interval in the mesh from t_0 to t with k even spaces. Since $\Lambda_{n,q} \leq \max_{\tau \in [t_0, t]} \Lambda_n(\tau)$, from the assumptions of the lemma, $\Lambda_{n,q}(t-t_0) < 1$.

Hence,

$$\|\partial_t^j A(t)\| \leq \tilde{\Lambda}_{n,q}^j \left[\frac{j}{q-1/2} + \frac{1}{k} \right] \quad (5.50)$$

where $\tilde{\Lambda}_{n,q} \equiv \Lambda_{n,q}(q-1/2)/k$.

Plugging this into the formula into (5.46) and using the definition of Y_n given by (2.50), our bound becomes

$$\|\partial_t^n U_2(t)\| \leq \sum_C \frac{n!}{c_1! \dots c_n!} \prod_{j=1}^n \left(\frac{(\frac{j}{q-1/2} + \frac{1}{k}) \tilde{\Lambda}_{n,q}^j}{j!} \right)^{c_j}. \quad (5.51)$$

Using the sum property of the coefficients c_j , we can move the $\tilde{\Lambda}_{n,q}^j$ out of the sum.

$$\begin{aligned} \|\partial_t^n U_2(t)\| &\leq \left(\Lambda_{n,q} \frac{q-1/2}{k} \right)^n \sum_C \frac{n!}{c_1! \dots c_n!} \prod_{j=1}^n \left(\frac{\frac{j}{q-1/2} + \frac{1}{k}}{j!} \right)^{c_j} \\ &= \left(\Lambda_{n,q} \frac{q-1/2}{k} \right)^n Y_n(\vec{x}_{q,k}^{(n)}) \end{aligned} \quad (5.52)$$

In the last line, we reapplied the definition of Y_n and of the vectors $\vec{x}_{q,k}^{(n)}$. This completes our bound for the U_2 formula for the q th segment of mesh defined by k_j . \square

We conclude this section with a proof of the bound on R_{2m} .

Proof of Lemma 5.4.3. Without loss of generality, we take $t_0 = 0$. The relevant expressions are

$$U_{2,m}(t, 0) = \sum_{j=1}^m a_j U_2^{(k_j)}(t, 0) \quad (5.53)$$

and

$$U_2^{(k)}(t, 0) := \prod_{\ell=1}^k U_2(t_\ell, t_{\ell-1}) \quad (5.54)$$

with $t_\ell := t\ell/k$. The Taylor remainder in integral form is given by

$$\begin{aligned} R_{2m} &= \frac{1}{(2m)!} \int_0^t (t-\tau)^{2m} \frac{d^{2m+1}}{d\tau^{2m+1}} U_{2,m}(\tau, 0) d\tau \\ &= \frac{1}{(2m)!} \sum_{j=1}^m a_j \int_0^t (t-\tau)^{2m} \frac{d^{2m+1}}{d\tau^{2m+1}} U_2^{(k_j)}(\tau, 0) d\tau. \end{aligned} \quad (5.55)$$

With a couple triangle inequalities, this is upper bounded as

$$\begin{aligned} \|R_{2m}\| &\leq \frac{1}{(2m)!} \sum_{j=1}^m |a_j| \frac{t^{2m+1}}{2m+1} \max_{\tau \in [0, t]} \|d_\tau^{2m+1} U_2^{(k_j)}(\tau, 0)\| \\ &\leq \frac{\|a\|_1}{(2m+1)!} t^{2m+1} \max_{j, \tau} \|d_\tau^{2m+1} U_2^{(k_j)}(\tau, 0)\| \end{aligned} \quad (5.56)$$

where in the last line we employed a Hölder inequality. Our focus is now on bounding the derivative, which we unravel layer by layer using frequent multinomial expansions. First,

$$d_\tau^n U_2^{(k)}(\tau, 0) = \sum_N \binom{n}{n_1, \dots, n_k} \prod_{\ell=1}^k d_\tau^{n_\ell} U_2(\tau_\ell, \tau_{\ell-1}). \quad (5.57)$$

Next, we write

$$\begin{aligned} U_2(\tau_\ell, \tau_{\ell-1}) &= \prod_{i=L}^1 e^{-iH_i \alpha_i (\tau_{\ell-1/2}) \tau / k} \prod_{i=1}^L e^{-iH_i \alpha_i (\tau_{\ell-1/2}) \tau / k} \\ &= \prod_{i=1}^{2L} e^{A_{i,\ell}} \end{aligned} \quad (5.58)$$

where

$$A_{i,\ell} := -iH_i \alpha_i (\tau_{\ell-1/2}) \tau / k \quad (5.59)$$

and i is defined by reflection for $i > L$. Once again performing a multinomial expansion,

$$d_\tau^n U_2(\tau_\ell, \tau_{\ell-1}) = \sum_N \binom{n}{n_1, \dots, n_{2L}} \prod_{i=1}^{2L} d_\tau^{n_i} e^{A_{i,\ell}}. \quad (5.60)$$

We now bound the individual ordinary operator exponentials. Invoking Lemma 5.4.4,

$$\|d_\tau^n e^{A_{i,\ell}}\| \leq Y_n (\|d_\tau A_{i,\ell}\|, \dots, \|d_\tau^n A_{i,\ell}\|). \quad (5.61)$$

In turn, we have

$$\begin{aligned} d_\tau^n A_{i,\ell} &= -i \frac{H_i}{k} d_\tau^n (\alpha_i(\tau_{\ell-1/2})\tau) \\ &= -i \frac{H_i}{k} \left[\left(\frac{\ell-1/2}{k} \right)^n \tau \alpha_i^{(n)}(\tau_{\ell-1/2}) + n \left(\frac{\ell-1/2}{k} \right)^{n-1} \alpha_i^{(n-1)}(\tau_{\ell-1/2}) \right] \end{aligned} \quad (5.62)$$

where $\alpha^{(n)}(x)$ refers to the n th derivative of α with respect to its argument, then evaluated at x (i.e., not a τ derivative). Since $\|H_i\| \leq 1$ we have

$$\|d_\tau^n A_{i,\ell}\| < \frac{1}{k} (\ell/k)^{n-1} \left((\ell/k)\tau |\alpha_i^{(n)}(\tau_{\ell-1/2})| + n |\alpha_i^{(n-1)}(\tau_{\ell-1/2})| \right). \quad (5.63)$$

From Definition 3, $\alpha_i^{(j)}(t) \leq \Lambda_{i,n}(t)^{j+1}$. Dropping the n and t dependence for the moment,

$$\begin{aligned} \|d_\tau^n A_{i,\ell}\| &< (\ell/k)^n \left((\tau/k)\Lambda_i^{n+1} + (n/\ell)\Lambda_i^n \right) \\ &= (\Lambda_i \ell/k)^n (\Lambda_i \tau/k + n/\ell). \end{aligned} \quad (5.64)$$

We've reached the bottom, and now proceed to work our way back up to the Taylor remainder R_{2m} , starting with (5.61). Using Lemma 5.4.4,

$$\begin{aligned} \|d_\tau^n e^{A_{i,\ell}}\| &\leq \sum_C \frac{n!}{c_1! c_2! \dots c_n!} \prod_{j=1}^n \left(\frac{\|d_\tau^j A_{i,\ell}\|}{j!} \right)^{c_j} \\ &< \sum_C \frac{n!}{c_1! c_2! \dots c_n!} \prod_{j=1}^n \left(\frac{(\Lambda_i \ell/k)^j (\Lambda_i \tau/k + j/\ell)}{j!} \right)^{c_j}. \end{aligned} \quad (5.65)$$

Using the sum rule for C we can pull out a factor of $(\Lambda_i \ell/k)$. With the upper bound $j \leq n$ and the monotonicity of Y_n , we obtain the bound

$$\|d_\tau^n e^{A_{i,\ell}}\| < (\Lambda_i \ell/k)^n B_n(\Lambda_i \tau/k + n/\ell) \quad (5.66)$$

where B_n is the Bell polynomial (see Section 2.8). For simplicity, define

$$x_{i,\ell,n} = \Lambda_i \tau/k + n/\ell \quad (5.67)$$

as the argument to B_n . Employing the bound (2.56),

$$\|d_\tau^n e^{A_{i,\ell}}\| < (\Lambda_i \ell/k)^n \left(\frac{n}{\log(1 + n/x_{i,\ell,n})} \right)^n \quad (5.68)$$

which is valid for all $n > 0$, and for $n = 0$ when defined by the 0^+ limit. We can simplify the reciprocal log with the bound

$$\begin{aligned} \frac{1}{\log(1 + n/x_{i,\ell,n})} &< \left(\frac{1}{2} + \frac{x_{i,\ell,n}}{n} \right)^n \\ &= \frac{1}{2^n} \left(1 + \frac{2x_{i,\ell,n}}{n} \right)^n. \end{aligned} \quad (5.69)$$

This gives us the simplified exponential derivative

$$\|d_\tau^n e^{A_{i,\ell}}\| < (\Lambda_i \ell / 2k)^n (n + 2x_{i,\ell,n})^n. \quad (5.70)$$

We now move up a level to reconsider (5.60). Employing a triangle inequality,

$$\begin{aligned} \|d_\tau^n U_2(\tau_\ell, \tau_{\ell-1})\| &\leq \sum_N \binom{n}{n_1, \dots, n_{2L}} \prod_{i=1}^{2L} \|d_\tau^{n_i} e^{A_{i,\ell}}\| \\ &< \sum_N \binom{n}{n_1, \dots, n_{2L}} \prod_{i=1}^{2L} (\Lambda_i \ell / 2k)^{n_i} (n_i + 2x_{i,\ell,n_i})^{n_i}. \end{aligned} \quad (5.71)$$

Maximize Λ_i over all $i = 1, \dots, L$ and call it Λ . We can factor out the corresponding term, and with some rewriting obtain

$$(\Lambda \ell / 2k)^n \sum_N \binom{n}{n_1, \dots, n_{2L}} \prod_{i=1}^{2L} (n_i + 2x_{\ell,n_i})^{n_i} \quad (5.72)$$

where we've also let x_{ℓ,n_i} be x_{i,ℓ,n_i} with the subscript dropped on Λ_i . Focusing on the rightmost product over i , one can show using a Lagrange multiplier that the maximum is given by $n_i = n/2L$ for all i (we maximize over $n_i \in \mathbb{R}_+$, which is an upper bound). This is intuitive from symmetry of the product as well. Taking this as an upper bound, we have

$$\begin{aligned} \|d_\tau^n U_2(\tau_\ell, \tau_{\ell-1})\| &< (\Lambda \ell / 2k)^n \left(\frac{n}{2L} + \frac{2\Lambda\tau}{k} + \frac{n}{L\ell} \right)^n \sum_N \binom{n}{n_1, \dots, n_{2L}} \\ &= (\Lambda \ell / 2k)^n \left(n + \frac{4\Lambda\tau L}{k} + \frac{2n}{\ell} \right)^n \\ &= (\Lambda/k)^n \left(n + n\ell/2 + \frac{2\Lambda\tau L\ell}{k} \right)^n. \end{aligned} \quad (5.73)$$

where in going to the second line we evaluated the multinomial sum as $(2L)^n$ and simplified.

With this in hand, we return to (5.57) and bound it as

$$\begin{aligned} \|d_\tau^n U_2^{(k)}(\tau, 0)\| &\leq \sum_N \binom{n}{n_1, \dots, n_k} \prod_{\ell=1}^k \|d_\tau^{n_\ell} U_2(\tau_\ell, \tau_{\ell-1})\| \\ &< \sum_N \binom{n}{n_1, \dots, n_k} \prod_{\ell=1}^k (\Lambda/k)^{n_\ell} \left(n_\ell + n_\ell \ell / 2 + \frac{2\Lambda\tau L \ell}{k} \right)^{n_\ell}. \end{aligned} \quad (5.74)$$

Using the upper bound $\ell \leq k$ and factoring out the $(\Lambda/k)^{n_\ell}$ using the sum rule,

$$\|d_\tau^n U_2^{(k)}(\tau, 0)\| < (\Lambda/k)^n \sum_N \binom{n}{n_1, \dots, n_k} \prod_{\ell=1}^k (n_\ell + n_\ell k / 2 + 2\Lambda\tau L)^{n_\ell}. \quad (5.75)$$

Similar to, we upper bound the product using $n_\ell = n/k$ for all ℓ , which can be justified through a maximization using Lagrange multipliers. The corresponding bound is

$$\begin{aligned} \|d_\tau^n U_2^{(k)}(\tau, 0)\| &< (\Lambda/k)^n (n/k + n/2 + 2\Lambda\tau L)^n \sum_N \binom{n}{n_1, \dots, n_k} \\ &= (n\Lambda)^n \left(\frac{1}{k} + \frac{1}{2} + \frac{2\Lambda\tau L}{n} \right)^n. \end{aligned} \quad (5.76)$$

We are finally ready to return to equation (5.56) and bound R_{2m} . We recall that Λ has τ dependence, and let $\Lambda_{\max} := \max_{\tau \in [0, t]} \Lambda(\tau)$. We also upper bound any appearance of τ otherwise by t because these are always in the numerator. So far, using $n = 2m + 1$, these reductions give

$$\|R_{2m}\| < \frac{\|a\|_1}{(2m+1)!} ((2m+1)\Lambda_{\max} t)^{2m+1} \max_j \left(\frac{1}{k_j} + \frac{1}{2} + \frac{2\Lambda t L}{2m+1} \right)^{2m+1}. \quad (5.77)$$

Employing a Stirling bound on the factorial, and factoring out an additional L from the rightmost term,

$$\|R_{2m}\| < \frac{\|a\|_1}{\sqrt{2\pi(2m+1)}} (eL\Lambda_{\max} t)^{2m+1} \max_j \left(\frac{1}{Lk_j} + \frac{1}{2L} + \frac{2\Lambda t}{2m+1} \right)^{2m+1}. \quad (5.78)$$

We now apply the assumption that $eL\Lambda_{\max} t < 1$ to upper bound the \max_j term, along with $k_j, L \geq 1$.

$$\max_j \left(\frac{1}{Lk_j} + \frac{1}{2L} + \frac{2\Lambda t}{2m+1} \right)^{2m+1} < \left(\frac{3}{2} + \frac{2}{3e} \right)^{2m+1} \quad (5.79)$$

Thus,

$$\begin{aligned} \|R_{2m}\| &< \frac{\|a\|_1}{2\sqrt{\pi m}} \left(\left(\frac{3e}{2} + \frac{2}{3} \right) L \max_{\tau \in [0, t]} \Lambda_{2m+1}(\tau) t \right)^{2m+1} \\ &< \frac{\|a\|_1}{2\sqrt{\pi m}} \left(5L \max_{\tau \in [0, t]} \Lambda_{2m+1}(\tau) t \right)^{2m+1}. \end{aligned} \quad (5.80)$$

In these last lines, we remind ourselves that Λ has the subscript $2m + 1$ as per Definition 3. \square

5.5 Time Step Analysis

The next ingredient we need for a complexity analysis is asymptotic bounds on the number of subintervals r needed in the time mesh. This will be the concern of this section. Unfortunately, in pursuing best-case bounds on r , we eschew a practical procedure for generating the time points t_i . Section 5.9 provides a concrete procedure which is based on the analysis of this section.

For time dependent Hamiltonians, because the cost per unit time can vary with t in general, one should adaptively choose the step size depending on the cost. For our purposes, this means choosing a step size inversely proportional to the energy measure $\Lambda_{2m+1}(t)$. We will explore this adaptive time stepping and show L^1 -norm scaling with $\Lambda_{2m+1}(t)$ here.

To derive bounds on r , we will need to assume something about size of the derivative $\dot{\Lambda}_{2m+1}$ compared to Λ_{2m+1} itself. Given a Λ_n -bound, a differentiable (smooth, even) Λ_n -bound exists. From now on, we consider Λ_n -bounds for which there exists a $K \in \mathbb{R}_+$ be such that $|\dot{\Lambda}_n(t)| \leq K\Lambda_n(t)^2$ for all $t \in [0, T]$. Given H that is Λ_n boundable, there is always, in fact, a Λ_n bound such that K exists and is arbitrarily close to zero. For example, we may take a constant bound $\Lambda'_n := \max_t \Lambda_n(t)$, noting that Λ_n is continuous on a compact interval. Of course, Λ' does not capture the changing behavior of $H(t)$, and is therefore suboptimal. Nevertheless, we've demonstrated that our additional assumptions are not much more restrictive than those we've already made. Note that (in natural units) K is dimensionless.

With these preliminaries in place, the following result provides an upper bound on the number of time steps needed for our MPF algorithm.

Lemma 5.5.1. *Let H satisfy the assumptions of Theorem 5.4.1, and let Λ_{2m+1} be a Λ_{2m+1} -bound for H such that, for some $K \in \mathbb{R}_+$, $|\dot{\Lambda}_{2m+1}(t)| \leq K\Lambda_{2m+1}(t)^2$ for all $t \in [0, T]$. For every $\epsilon > 0$, there exists a list (t_0, t_1, \dots, t_r) of monotonically increasing times $t_j \in [0, T]$, with $t_0 = 0$ and $t_r = T$, such that*

$$\|U(T, 0) - \prod_{i=1}^r U_{2,m}(t_i, t_{i-1})\| \leq \epsilon$$

with the number of time steps r bounded above as

$$r \leq \left[\left(5 \left(1 + \frac{3}{2}K \right) L \|\Lambda\|_1 \right)^{1 + \frac{1}{2m}} \left(\frac{\|a\|_1}{\epsilon \sqrt{\pi m}} \right)^{\frac{1}{2m}} \right].$$

Here, $\|\Lambda_{2m+1}\|_1$ is the L^1 norm.

$$\|\Lambda_{2m+1}\|_1 := \int_0^T \Lambda_{2m+1}(t) dt$$

Proof. As discussed in Section 5.4 in order to satisfy the ϵ -error constraint of Lemma 5.5.1, it suffices that the error on each subinterval is less than ϵ/r . Using Theorem 5.4.1, the sum is bounded as

$$\sum_{i=1}^r \|U(t_i, t_{i-1}) - U_{2,m}(t_i, t_{i-1})\| \leq \frac{\|a\|_1}{\sqrt{\pi m}} \sum_{i=1}^r \left(5L \max_{\tau \in [t_{i-1}, t_i]} \Lambda_{2m+1}(\tau) (t_i - t_{i-1}) \right)^{2m+1}. \quad (5.81)$$

To ensure an overall error ϵ , it therefore suffices to produce a mesh such that for each i ,

$$\frac{\|a\|_1}{\sqrt{\pi m}} \left(5L \max_{\tau \in [t_{i-1}, t_i]} \Lambda_{2m+1}(\tau) (t_i - t_{i-1}) \right)^{2m+1} \leq \epsilon/r. \quad (5.82)$$

Rearranging, this corresponds to choosing t_i , given all other parameters, that satisfy

$$L \max_{\tau \in [t_{i-1}, t_i]} \Lambda_{2m+1}(\tau) (t_i - t_{i-1}) \leq \frac{1}{5} \left(\frac{\epsilon \sqrt{\pi m}}{\|a\|_1 r} \right)^{1/(2m+1)}. \quad (5.83)$$

We now digress in order to relate $\max_{\tau} \Lambda_{2m+1}(\tau)$ and its average. Here is where we will make use of the K -bounds on the derivative $\dot{\Lambda}$, we closely follow arguments found in [137]. From the inequality in the lemma statement, we have

$$\begin{aligned} \left| \frac{\dot{\Lambda}_{2m+1}(t)}{\Lambda_{2m+1}(t)^2} \right| &\leq K \\ \left| \frac{d}{dt} \frac{1}{\Lambda_{2m+1}(t)} \right| &\leq K. \end{aligned} \quad (5.84)$$

Suppose the time t_{i-1} has been chosen by the previous iteration (if $i = 1$, $t_0 = 0$). Let $t > t_{i-1}$ and integrate the above inequality from t_{i-1} to t .

$$\begin{aligned} \int_{t_{i-1}}^t \left| \frac{d}{d\tau} \frac{1}{\Lambda_{2m+1}(\tau)} \right| d\tau &\leq K(t - t_{i-1}) \\ \left| \int_{t_{i-1}}^t \frac{d}{d\tau} \frac{1}{\Lambda_{2m+1}(\tau)} d\tau \right| &\leq K(t - t_{i-1}) \\ \left| \frac{1}{\Lambda_{2m+1}(t)} - \frac{1}{\Lambda_{2m+1}(t_{i-1})} \right| &\leq K(t - t_{i-1}) \end{aligned} \quad (5.85)$$

Let us rearrange this in terms of $\Lambda_{2m+1}(t)$ alone.

$$\begin{aligned}
-K(t - t_{i-1}) &\leq \frac{1}{\Lambda_{2m+1}(t)} - \frac{1}{\Lambda_{2m+1}(t_{i-1})} \leq K(t - t_{i-1}) \\
\frac{1}{\Lambda_{2m+1}(t_{i-1})} - K(t - t_{i-1}) &\leq \frac{1}{\Lambda_{2m+1}(t)} \leq \frac{1}{\Lambda_{2m+1}(t_{i-1})} + K(t - t_{i-1}) \\
\frac{\Lambda_{2m+1}(t_{i-1})}{1 + K(t - t_{i-1})\Lambda_{2m+1}(t_{i-1})} &\leq \Lambda_{2m+1}(t) \leq \frac{\Lambda_{2m+1}(t_{i-1})}{1 - K(t - t_{i-1})\Lambda_{2m+1}(t_{i-1})}
\end{aligned} \tag{5.86}$$

The lowerbound inequality holds for all $t > t_{i-1}$, while the upper bound only holds when

$$(t - t_{i-1})\Lambda_{2m+1}(t_{i-1})K < 1. \tag{5.87}$$

We restrict our attention to t for which both bounds hold. Consider, for the moment, only the leftmost inequality. The lower bound on the left is monotonically decreasing with t . This means that it is also a uniform lower bound on $\Lambda_{2m+1}(t')$ for any $t' \in [t_{i-1}, t]$. Therefore, it is a lower bound for the average $\bar{\Lambda}_{2m+1}(t)$ on the interval $[t_{i-1}, t]$.

$$\bar{\Lambda}_{2m+1}(t, t_{i-1}) := \frac{1}{t - t_{i-1}} \int_{t_{i-1}}^t \Lambda_{2m+1}(\tau) d\tau \tag{5.88}$$

That is,

$$\frac{\Lambda_{2m+1}(t_{i-1})}{1 + K(t - t_{i-1})\Lambda_{2m+1}(t_{i-1})} \leq \bar{\Lambda}_{2m+1}(t, t_{i-1}), \tag{5.89}$$

or, after isolating for $\Lambda_{2m+1}(t_{i-1})$

$$\Lambda_{2m+1}(t_{i-1}) \leq \frac{\bar{\Lambda}_{2m+1}(t, t_{i-1})}{1 - K(t - t_{i-1})\bar{\Lambda}_{2m+1}(t, t_{i-1})}. \tag{5.90}$$

At this point, let's now consider the upper bound in equation (5.86). This bound is monotonically *increasing* in t , and therefore also upper bounds $\Lambda_{2m+1}(\tau)$ for any τ in $[t_{i-1}, t]$. Therefore, it is also a bound for the maximum.

$$\max_{\tau \in [t_{i-1}, t]} \Lambda_{2m+1}(\tau) \leq \frac{\Lambda_{2m+1}(t_{i-1})}{1 - K(t - t_{i-1})\Lambda_{2m+1}(t_{i-1})} \tag{5.91}$$

Substituting bounds for $\Lambda_{2m+1}(t_{i-1})$ from equation (5.90) gives us a bound on the maximum value in terms of the average.

$$\max_{\tau \in [t_{i-1}, t]} \Lambda_{2m+1}(\tau) \leq \frac{\bar{\Lambda}_{2m+1}(t, t_{i-1})}{1 - \frac{3}{2}K\bar{\Lambda}_{2m+1}(t, t_{i-1})(t - t_{i-1})} \tag{5.92}$$

Solving for the average value of Λ_{2m+1} , and multiplying by $t - t_{i-1}$ on both sides,

$$(t - t_{i-1})\bar{\Lambda}_{2m+1}(t, t_{i-1}) \geq \frac{(t - t_{i-1}) \max_{\tau \in [t_{i-1}, t]} \Lambda_{2m+1}(\tau)}{1 + \frac{3}{2}K(t - t_{i-1}) \max_{\tau \in [t_{i-1}, t]} \Lambda_{2m+1}(\tau)}. \quad (5.93)$$

Let us finally choose a $t = t_i$ which will serve as the next time step in the adaptive scheme. We would like come as close as possible to saturating equation (5.83) while staying within the constraint imposed by the maximum bound of equation (5.86). Thus, we choose t_i such that

$$\max_{\tau \in [t_{i-1}, t_i]} \Lambda_{2m+1}(\tau)(t_i - t_{i-1}) = \min \left\{ \frac{1}{K}, \frac{1}{5L} \left(\frac{\epsilon \sqrt{\pi m}}{\|a\|_1 r} \right)^{1/(2m+1)} \right\}. \quad (5.94)$$

Since K is a constant, for asymptotic purposes we will assume sufficiently small ϵ such that the right term is smaller. Plugging in to (5.93) yields

$$\bar{\Lambda}_{2m+1}(t_i, t_{i-1})(t_i - t_{i-1}) \geq \frac{\frac{1}{5L} \left(\frac{\epsilon \sqrt{\pi m}}{\|a\|_1 r} \right)^{1/(2m+1)}}{1 + \frac{3}{2} \frac{1}{5L} \left(\frac{\epsilon \sqrt{\pi m}}{\|a\|_1 r} \right)^{1/(2m+1)} K}. \quad (5.95)$$

We then find, by using the fact that $\frac{1}{5L} \left(\frac{\epsilon \sqrt{\pi m}}{\|a\|_1 r} \right)^{1/(2m+1)} < 1$ and by summing over $i = 1, \dots, r$ in (5.95), that

$$\|\Lambda\|_1 \geq r^{\frac{2m}{2m+1}} \frac{1}{5L} \left(\frac{\epsilon \sqrt{\pi m}}{\|a\|_1} \right)^{1/(2m+1)} \left(1 + \frac{3}{2}K \right)^{-1}. \quad (5.96)$$

Finally, rearranging the above, this implies that the number of steps required for the MPF algorithm is upper bounded as

$$r \leq \left(5 \left(1 + \frac{3}{2}K \right) L \|\Lambda\|_1 \right)^{1 + \frac{1}{2m}} \left(\frac{\|a\|_1}{\epsilon \sqrt{\pi m}} \right)^{\frac{1}{2m}}. \quad (5.97)$$

The result then directly follows from the requirement that r is an integer. \square

To summarize, we've provided an upper bound on the number of steps r needed given assumptions on the derivative of Λ_{2m+1} . What is perhaps objectionable is that, in determining our subsequent time stepping, we seemed to need information about the total number of steps r that we would end up with. While this does not detract from the correctness of our result, it does indicate possible difficulty in constructing a suitable set of t_j for which the Lemma holds. One approach is to guess the final number r_{try} of steps needed, construct the mesh according to the proof, then see if r_{try} can be made correct. This approach is considered in Section 5.9.

5.6 Query Complexity

With the results of the previous two sections, we proceed to bound the query complexity needed to perform a time dependent MPF simulation. First, we define a set of oracles that are appropriate for this simulation problem. We reemphasize that the most natural input model in our setting is the linear combinations of Hamiltonians model

$$H = \sum_{j=1}^L \alpha_j(t) H_j, \quad (5.98)$$

where $\alpha_j : [0, T] \rightarrow \mathbb{R}$ has $2m + 1$ continuous derivatives and $H_j \in \text{Herm}(\mathbb{C}^{2^n})$ satisfies $\|H_j\| \leq 1$. We discretize $[0, T]$ into 2^{n_t} uniform grid points $t_k = kT/2^{n_t}$ for $k \in [0, 2^{n_t}) \cap \mathbb{Z}$, and define $\alpha_{jk} := \alpha_j(t_k)$. Let $\delta t := T/2^{n_t}$. Let U_α and U_H be unitary oracles which provide the input Hamiltonian as follows.

$$\begin{aligned} U_\alpha |j\rangle |k\rangle |\tau\rangle |0\rangle &:= |j\rangle |k\rangle |\tau\rangle |\alpha_{jk}\tau\rangle \\ U_H |j\rangle |\alpha_{jk}\tau\rangle |\psi\rangle &:= |j\rangle |\alpha_{jk}\tau\rangle \exp\{-iH_j\alpha_{jk}\tau\} |\psi\rangle \end{aligned} \quad (5.99)$$

The oracle U_α encodes a reversible classical computation and may be taken as self-inverse. Here $|\tau\rangle$ encodes a step of size $\tau \in \mathbb{R}$ in binary using n_c qubits. Such step sizes are always nonnegative for the low-order formulas we consider, and therefore we take $\tau \in [0, T]$. Hence, $\delta t = T/2^{n_c}$ is the rounding error for the step sizes. We neglect rounding effects due to the values $\alpha_{jk}\tau$.

Our first result concerns the approximate implementation of U_2 using the two oracles.

Lemma 5.6.1. *Let $U_2(\tau + t, t)$ be the 2nd-order Suzuki-Trotter formula for the midpoint formula of equation (5.8). Then an approximation W_2 can be constructed using at most $6L - 3$ queries to U_H and U_α , such that*

$$\|U_2(t + \tau, t) - W_2(t + \tau, t)\| \leq L \max_{j, t \in [0, T]} |\dot{\alpha}_j(t + \tau/2)| \frac{T^2}{2^{n_c}}.$$

Proof. Define W_2 as U_2 but with each α_j evaluated at the nearest discrete times in $\{t_k\}$. Using the techniques of [137], two queries to U_α and one query to U_H are needed to exactly simulate each of the $2L - 1$ exponentials present in W_2 . Thus $3 \times (2L - 1)$ queries are needed total. To evaluate the

discretization error, by Box 4.1 of [101] we have that

$$\|W_2 - U_2\| \leq 2 \sum_{j=1}^L \|e^{-iH_j \alpha_j (\text{rnd}[t+\tau/2])\tau/2} - e^{-iH_j \alpha_j (t+\tau/2)\tau/2}\| \quad (5.100)$$

which in turn is upper bounded, through an application of the fundamental theorem of calculus, by

$$2 \sum_{j=1}^L \|H_j \alpha_j (\text{rnd}[t + \frac{\tau}{2}]) \frac{\tau}{2} - H_j \alpha_j (t + \frac{\tau}{2}) \frac{\tau}{2}\| \quad (5.101)$$

where rnd rounds to the nearest n_c -bit value. Since $\|H_j\| \leq 1$ this is merely upper bounded as

$$\tau \sum_{j=1}^L |\alpha_j (\text{rnd}[t + \frac{\tau}{2}]) - \alpha_j (t + \frac{\tau}{2})|. \quad (5.102)$$

By the fundamental theorem of calculus, with an integral upper bound, each term is upper bounded as $\delta t \max_{\delta t \in t \pm \delta t} |\partial_t \alpha_j (t + \tau/2)|$. Maximizing over $[0, T]$ instead, and making other simplifying choices, we get a crude upper bound

$$\begin{aligned} \|W_2 - U_2\| &\leq \tau L \delta t \max_{j, [0, T]} |\dot{\alpha}_j(t)| \\ &\leq L \frac{T^2}{2^{n_c}} \max_{j, [0, T]} |\dot{\alpha}_j(t)|. \end{aligned} \quad (5.103)$$

Rearranging this gives the inequality of the lemma statement. \square

Having supplied an approximate base formula W_2 with our queries, we next need to implement an approximate MPF $W_{2,m}$ over a subinterval $[t_0, t_1]$. This is conventionally done through the use of "select" SEL and "prepare" PREP circuits.

$$\text{PREP } |0\rangle := \sum_{j=1}^m \sqrt{\frac{|a_j|}{\|a\|_1}} |j\rangle \quad (5.104)$$

$$\text{SEL } |j\rangle |\psi\rangle := \text{sgn}(a_j) |j\rangle W_2^{(k_j)}(t_1, t_0) |\psi\rangle$$

The circuit PREP can be implemented without any queries to U_α or U_H whereas SEL requires $O(L\|\vec{k}\|_\infty)$ queries. Following the well-conditioned MPF scheme of [90] we have that $k_j \leq 3m^2$. This implies that a query to SEL requires $O(Lm^2)$ queries to U_H and U_α .

We can use the SEL and PREP for a standard LCU block encoding in order to construct a time dependent MPF with base formula W_2 .

Lemma 5.6.2. *Under the assumptions of Theorem 5.4.1 and the query model above, for any $[t_0, t_1] \subseteq [0, T]$ the time dependent MPF $W_{2,m}$ with base formula W_2 satisfies*

$$\|W_{2,m}(t_1, t_0) - U(t_1, t_0)\| \in O\left(\|a\|_1 \left(\max_{t \in [t_0, t_1]} \Lambda_{2m+1}(t)T\right)^{2m+1}\right),$$

provided that

$$n_c \geq \log\left(\frac{3\sqrt{\pi}m^{5/2}L \max_{t,j} |\partial_t \alpha_j(t)|(t_1 - t_0)^2}{(5L \max_{t \in [t_0, t_1]} \Lambda_{2m+1}(t)(t_1 - t_0))^{2m+1}}\right),$$

and can be constructed with a number of queries to U_H and U_α scaling as $O(m^2L)$.

Proof. From Lemma 4 of [15], we have

$$\begin{aligned} (\langle 0| \otimes I)(\text{PREP}^\dagger)\text{SEL}(\text{PREP})(|0\rangle \otimes I) &= \frac{1}{\|a\|_1} \sum_{j=1}^m a_j W_2^{(k_j)}(t_1, t_0) \\ &= W_{2,m}(t_1, t_0) / \|a\|_1. \end{aligned} \tag{5.105}$$

Let $\delta' > 0$ be such that, for all j and $\ell \in \{1, \dots, k_j\}$,

$$\left\| W_2\left(\Delta t \frac{\ell}{k_j} + t_0, \Delta t \frac{\ell-1}{k_j} + t_0\right) - U_2\left(\Delta t \frac{\ell}{k_j} + t_0, \Delta t \frac{\ell-1}{k_j} + t_0\right) \right\| \leq \delta' \tag{5.106}$$

where $\Delta t = t_1 - t_0$. Then, by invoking Box 4.1 from [101],

$$\left\| U_2^{(k_j)}(t_1, t_0) - W_2^{(k_j)}(t_1, t_0) \right\| \leq k_j \delta' \tag{5.107}$$

which, since $k_j \leq 3m^2$, implies that

$$\|V_{2,m}(t_1, t_0) - W_{2,m}(t_1, t_0)\| \leq 3m^2 \delta' \|a\|_1. \tag{5.108}$$

We supply δ' using Lemma 5.6.1, obtaining

$$3m^2 \delta' \|a\|_1 \leq 3m^2 \|a\|_1 L \max_{j,t \in [0,T]} |\dot{\alpha}_j(t + \tau/2)| \frac{T^2}{2^{n_c}}, \tag{5.109}$$

giving us a bound on the discretized MPF $W_{2,m}$ relative to the undiscretized $V_{2,m}$.

It then follows from the triangle inequality and Theorem 5.4.1 that

$$\begin{aligned} \|W_{2,m}(t_1, t_0) - U(t_1, t_0)\| &\leq \|U_{2,m}(t_1, t_0) - U(t_1, t_0)\| + \|W_{2,m}(t_1, t_0) - U_{2,m}(t_1, t_0)\| \\ &\leq \frac{\|a\|_1}{\sqrt{\pi m}} \left(5L \max_{t \in [0,T]} \Lambda_{2m+1}(t)T\right)^{2m+1} + 3m^2 L \|a\|_1 \max_{j,t} |\dot{\alpha}_j(t)| \frac{T^2}{2^{n_c}}. \end{aligned} \tag{5.110}$$

Under the assumption that

$$n_c \geq \log \left(\frac{3\sqrt{\pi}m^{5/2}L \max_{j,t} |\dot{\alpha}_j(t)| T^2}{(5L \max_{t \in [0,T]} \Lambda_{2m+1}(\tau) T)^{2m+1}} \right) \quad (5.111)$$

the second term is bounded by the first (5.110), so we have an upper bound

$$\left\| \sum_{j=1}^M a_j \prod_{q=1}^{k_j} W_2(Tq/k_j, T(q-1)/k_j) - U(T, 0) \right\| \leq \frac{2\|a\|_1}{\sqrt{\pi m}} \left(5L \max_{t \in [0,T]} \Lambda_{2m+1}(t) T \right)^{2m+1}. \quad (5.112)$$

Since $U(T, 0)$ is unitary, we know that the MPF implemented by our algorithm is close to a unitary.

This means that we satisfy the preconditions of robust oblivious amplitude amplification given by Lemma 5 of [15]. This result implies that using $O(\|a\|_1)$ applications of the unitary given by (5.105), we can implement an operator $\tilde{W}(T, 0)$ such that (for constant m)

$$\|W_{2,m}(t_1, t_0) - U(t_1, t_0)\| \in O \left(\|a\|_1 \left(\max_{t \in [0,T]} \Lambda_{2m+1}(t) (t_1 - t_0) \right)^{2m+1} \right). \quad (5.113)$$

The number of queries scales as

$$Q_{\text{step}} \in O(\|a\|_1 m^2 L) \subseteq \tilde{O}(m^2 L). \quad (5.114)$$

□

With the short-time simulation costs in place we are now ready to state our main theorem, which bounds the number of queries needed to perform the full multiproduct simulation of a time dependent Hamiltonian.

Theorem 5.6.3. *In the query setting above, and under the assumptions of Theorem 5.4.1, and Lemma 5.5.1 (Λ_{2m+1} -bounded H with K bound on $\dot{\Lambda}_{2m+1}$), we have that the number of queries Q_{tot} needed to U_α and U_H to construct an operator $W_{\text{tot}}(T, 0)$ simulate a time dependent Hamiltonian of the form $\sum_{j=1}^L \alpha_j(t) H_j$ such that $\|(\langle 0| \otimes I) W_{\text{tot}}(T, 0) (|0\rangle \otimes I) - U(T, 0)\| \leq \epsilon$ satisfies*

$$Q_{\text{tot}} \in \tilde{O} \left(L(1+K) \|\Lambda_{2m+1}\|_1 \log^2(1/\epsilon) \right),$$

and the total number of auxiliary qubits is in

$$\tilde{O} \left(\log \left(\frac{L(1+K) \|\Lambda_{2m+1}\|_1 \max_{j,t} |\dot{\alpha}_j(t)| T^2}{\epsilon} \right) \right).$$

Proof. From Lemma 5.5.1 we have that the number of segments needed to perform the simulation within error ϵ obeys

$$r \in \tilde{O}\left(\frac{((1+K)\|\Lambda_{2m+1}\|_1)^{1+1/(2m)}}{\epsilon^{1/(2m)}}\right). \quad (5.115)$$

Therefore, using Lemma 5.6.2,

$$Q_{\text{tot}} \in \tilde{O}(m^2 L r) \subseteq \tilde{O}\left(\frac{m^2 L ((1+K)\|\Lambda_{2m+1}\|_1)^{1+1/(2m)}}{\epsilon^{1/(2m)}}\right). \quad (5.116)$$

An approximation to the optimal value of $m \in \mathbb{Z}_+$ can be obtained by equating the exponentially shrinking component of the cost to the polynomially increasing value of m . We choose m to satisfy

$$m^2 = \left(\frac{(1+K)\|\Lambda_{2m+1}\|_1}{\epsilon}\right)^{1/2m}. \quad (5.117)$$

Solving for m yields

$$m = \frac{\log((1+K)\|\Lambda_{2m+1}\|_1/\epsilon)}{4 \text{LambertW}\left(\log((1+K)\|\Lambda_{2m+1}\|_1/\epsilon)/4\right)} \in \tilde{O}\left(\log\left(\frac{(1+K)\|\Lambda_{2m+1}\|_1}{\epsilon}\right)\right). \quad (5.118)$$

This implies that the query complexity Q_{tot} is in

$$\tilde{O}\left(L(1+K)\|\Lambda_{2m+1}\|_1 \log^2(1/\epsilon)\right). \quad (5.119)$$

The number of auxiliary qubits needed in the construction is in $O(\log(m))$ to implement the MPF and $(\lceil \log L \rceil + n_c)$ to implement the U_α oracle. From the result of Lemma 5.6.2 we see that n_c dominates this cost. We thus have a number of auxiliary qubits scaling as

$$\begin{aligned} n_{\text{aux}} &\in O\left(\log\left(\frac{m^2 L \max |\partial_t \alpha_j(t)| T^2}{(\max_{t \in [0, T]} \Lambda_{2m+1}(t) T)^{2m+1}}\right)\right) \\ &\in O\left(\log\left(\frac{m^2 L r \|a\|_1 \max |\partial_t \alpha_j(t)| T^2}{\epsilon}\right)\right) \\ &\in \tilde{O}\left(\log\left(\frac{L(1+K)\|\Lambda_{2m+1}\|_1 \max |\partial_t \alpha_j(t)| T^2}{\epsilon}\right)\right) \end{aligned} \quad (5.120)$$

where used Eq. (5.116) and Eq. (5.118) above. \square

This shows that the cost of quantum simulation using MPFs broadly conforms to the cost scalings that one would expect of previous methods. In particular, similar to the truncated Dyson series simulation method [89, 75] we obtain that the cost of simulating a time dependent Hamiltonian scales near-linearly with time T and poly-logarithmically with $1/\epsilon$.

5.7 Numerical Demonstrations

In the above sections, we developed and characterized MPFs for time dependent simulations by showing their existence and proving error bounds. However, these bounds are unlikely to be the final word on the performance of the algorithm. For example, we already mentioned that, for time independent H , the MPF of Definition 2 is exact in cases where the Hamiltonian consists of only commuting terms. Yet this behavior is not captured in the bound of Theorem 5.4.1 because Λ_{2m} is at least as large as $\|H\|$. This discrepancy is unrelated to the fact that, in practice, the 2nd-order formula U_2 can only be computed approximately.

To begin bridging the gap between algorithm's actual performance and our bounds, we investigate time dependent MPFs empirically through two numerical examples. We compute $U_{2,m}$ for these systems on a classical computer (using matrix computations) and compare the result with the exact propagator (computed within machine ϵ). The vector $\vec{k} \in \mathbb{Z}_+^m$ we will use comes from the bottom half of Table I from [90], which minimizes $\|\vec{k}\|$ for $\|a\|_1 \leq 2$.

In general, deriving an analytical solution for the propagator given a time dependent Hamiltonian is challenging or impossible. To bypass this problem, we will consider a time independent Hamiltonian which is viewed from a "non-inertial" frame, thereby rendering the dynamics time dependent in the new frame. More specifically, suppose H is a time independent Hamiltonian with propagator $U(t) = e^{-iHt}$ (henceforth the initial time is set to zero). Let $|\psi_t\rangle$ be the solution to the Schrödinger equation $i\partial_t |\psi_t\rangle = H |\psi_t\rangle$. Under a frame transformation $T(t)$, which transforms vectors as $|\tilde{\psi}_t\rangle = T(t) |\psi_t\rangle$, the Hamiltonian and propagator transform as

$$\begin{aligned}\tilde{U}(t) &= T(t)U(t) \\ \tilde{H}(t) &= i\frac{\partial T(t)}{\partial t}T(t)^\dagger + T(t)H(t)T(t)^\dagger.\end{aligned}\tag{5.121}$$

Thus, in order to benchmark the error of the MPF, we compute \tilde{U}_k for Hamiltonian \tilde{H} , then compare with the exact propagator (accurate to machine precision).

$$\epsilon_c = \|\tilde{U}_{2,m}(t) - T(t)U(t)\|\tag{5.122}$$

5.7.1 Example 1: Electron in Magnetic field, Rotating Frame

As a very simple first demonstration, consider a spin-1/2 particle (say, electron) in a homogeneous external magnetic field B . Choose a coordinate system such that B makes an angle θ with respect to the z -axis, and lies within the xz plane. This system can be described by the Hamiltonian

$$H = \mu B(\cos \theta Z/2 + \sin \theta X/2) \quad (5.123)$$

where Z and X (and later Y) are Pauli operators, and μ is a coupling parameter that will henceforth be set to one. The propagator $U(t) = e^{-iHt}$ is easy to compute, and corresponds to precession about the magnetic field axis with frequency B .

To obtain a time dependent problem, let's shift to a reference frame that rotates with angular frequency ω about the z -axis. The transformation is given by $R_z(\omega t)$, where R_a is the usual $SU(2)$ rotation operator about axis a . The Hamiltonian in the rotating frame is

$$\tilde{H}(t) = (\omega + B \cos \theta)Z/2 + B \sin \theta(\cos \omega t X/2 + \sin \omega t Y/2). \quad (5.124)$$

Because we know that this Hamiltonian is just a transformed time independent system, it is easy to compute the exact propagator $\tilde{U}(t)$.

$$\tilde{U}(t) = R_z(\omega t)U(t) \quad (5.125)$$

Though it is not strictly necessary to run the algorithm, let's compute an appropriate $\Lambda(t)$ upper bound. The spectral norm of \tilde{H} may be upper bounded as

$$\|\tilde{H}\| \leq \frac{|\omega + B \cos \theta|}{2} + |B \sin \theta| \quad (5.126)$$

while the derivatives $\tilde{H}^{(n)}(t)$ have the bound

$$\|\tilde{H}^{(n)}(t)\| \leq |B \sin \theta \omega^n| \quad (5.127)$$

$$\sqrt[n+1]{\|\tilde{H}^{(n)}(t)\|} \leq \omega \left| \frac{B \sin \theta}{\omega} \right|^{1/n+1}.$$

For ω not too much larger than B , we see then that $\Lambda(t) = \omega$ is an appropriate choice.

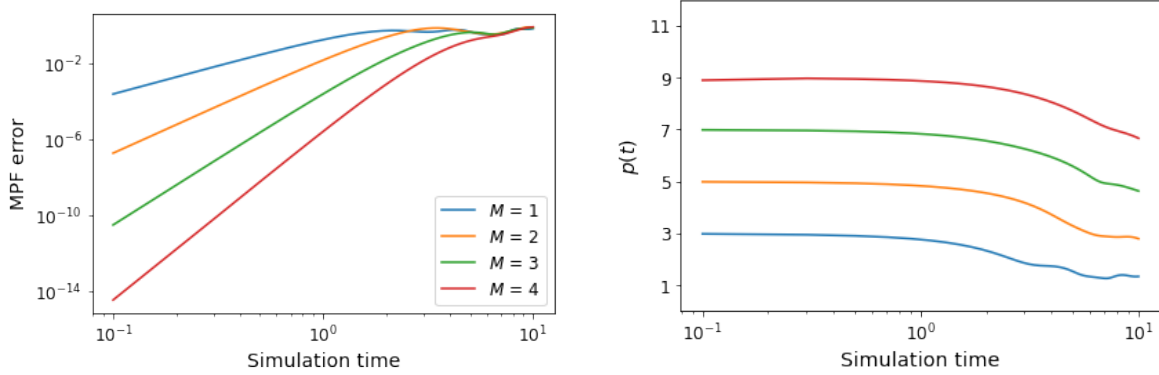


Figure 5.1 (left) Multiproduct errors plotted against simulation time, for several low-order MPFs, on a log-log plot. Notice the power law scaling for small values of t . The parameters used here are $B = 1$, $\omega = 4$, $\theta = \pi/6$. For larger m , one quickly runs into machine precision becoming the dominant error source. (right) The running power $p(t, t')$ defined in equation (5.128), with $t' = .3$. Note the plateau corresponds with the anticipated value of $2m + 1$.

The first thing to check will be that the error has the appropriate power law scaling. Namely, for M -term formulas, the error ϵ_c for small t should scale as $O(t^{2m+1})$ or better. We can check this by computing the "running power" $p(t, t')$.

$$p(t, t') := \frac{\log \epsilon_t / \epsilon_{t'}}{\log t / t'} \quad (5.128)$$

For different but small values of t, t' , the value of p should approach the expected order of the error: $2m + 1$. Indeed, this is precisely the behavior observed in Figure 5.1. For sufficiently small simulation times, a power-law dependence on the simulation error is observed, and the corresponding power is as anticipated. Additionally, we see that the error decreases by orders of magnitude with each additional term once the power-law regime is reached. Choosing $m > 4$ in this example quickly leads to machine precision being the dominant error source.

Next, we vary the MPF order m for fixed simulation time t . Since $\Lambda = \omega$, our bounds predict an exponential decay in the error, but only provided $t < 1/\omega$. Otherwise, the bounds grow exponentially and say nothing useful about performance. In Figure 5.7.1, we fix t at several different times and plot the error dependence on the multiproduct order m . Past a certain threshold value for m (which increases with t) an exponential decay in error is observed, possibly superexponential. It is promising that, even for $t = 10$, the exponential decay is eventually achieved at $m \gtrsim 6$.

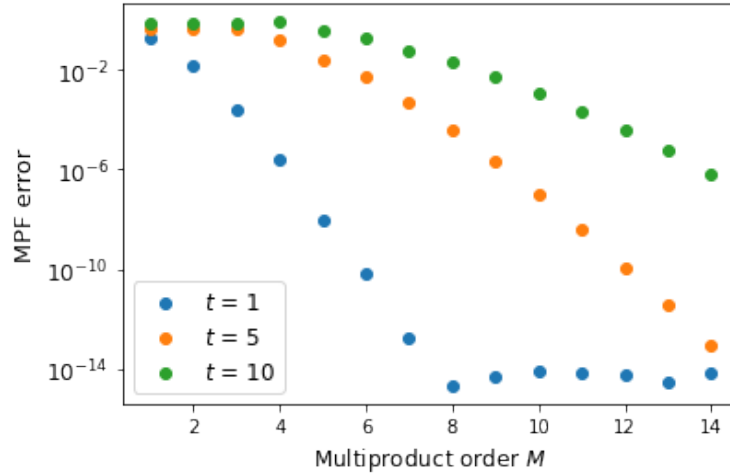


Figure 5.2 Multiproduct error shows an (super)exponential decrease in error for sufficiently large order m . The threshold for this regime is seen to increase as the simulation time increases. This behavior surpasses the expectation of our proven bounds, since there are no guarantees if the time step is too large. Note that, in practice, one should typically split a longer simulation time into smaller steps. The plateau for $t = 1, m > 8$ is a result of machine precision limitations. Parameter values: $B = 1, \omega = 4, \theta = \pi/6$.

This suggests our error bounds may be too conservative, and in particular MPFs could *absolutely* converge to U as $m \rightarrow \infty$ in certain circumstances. This would be a notable improvement to product formulas alone, which tend to lead to errors that diverge as $m \rightarrow \infty$ if the time step t remains fixed [14, 137, 26]. In contrast, Theorem 5.4.1 shows that if the time step is sufficiently small, then the MPF converges to the exact result. However, such convergence is not anticipated from the bounds for a large value such as $t = 10$.

Indeed, there are good reasons to believe the absolute convergence property holds more generically than this example. No matter how large the order m , we are still using a low order formula (such as the midpoint formula U_2) as a base. Moreover, recall that the MPF is essentially a sum of product formulas with different numbers of time steps (for the same time interval). As the order m increases, higher weight is given to terms in the multiproduct sum with finer meshes. Correspondingly, terms which have larger time steps, and therefore may not converge properly, become suppressed at large m . Such behavior is not reflected in our derived error bounds, so there is likely room for improvement.

Practitioners in quantum simulation will likely want to know how MPFs fare against the more-familiar and simpler Trotter techniques. To facilitate this, numerical studies across a broad range of physically interesting systems would be desirable. Such a comprehensive analysis must be left to future work; here we will be satisfied with comparing MPFs with Trotterization for our spin-1/2 example. Our Trotterization is just an MPF with $m = 1$, corresponding to a midpoint-formula approximation. To facilitate as fair a comparison as possible, we will keep the number of midpoint-formula queries between the two methods the same. That is, we will enforce the requirement

$$r_{\text{trot}} = r_{\text{mpf}} \max_j |k_j| \quad (5.129)$$

where r_{trot} and r_{mpf} are the number of time steps for Trotter and MPF, respectively. Note that the number of midpoint queries per time step for Trotter and MPFs are 1 and $\max_j |k_j|$ respectively.

Figure 5.7.1 shows the results of these head-to-head comparisons for the several values of the magnetic field B and rotation frequency ω . The number of MPF steps r_{mpf} is fixed at 10, a reasonable value since it makes $\Lambda\Delta t \sim 1$ on each subinterval. As the MPF order increases, so does the number of Trotter steps r_{trot} by the condition (5.129). These results show that, for m not too large, MPFs outperform Trotterization, at a value of the error ϵ which is large enough to be of practical significance for scientific or industrial applications.

Admittedly, the spin-1/2 system considered above is rather simplistic. However, we anticipate most of the inferences drawn above to hold even as we increase the dimensionality of the Hilbert space. For example, though the complexity of simulating U_2 generally increases as $\dim(H)$ grows, it does so both for MPFs and Trotterization. Nevertheless, benchmarking of MPFs on more complex systems would be a welcomed proof (or disproof) of concept.

5.7.2 Example 2: Spin Chain in Interaction Picture

As a first step towards more complicated many-body quantum systems, we investigate the use of MPFs for a particular one-dimensional chain of spins with nearest-neighbor interactions. As before, we will take advantage of a change of reference frame, allowing us to compare the multiproduct simulations with a machine-precise simulation in an equivalent, time independent

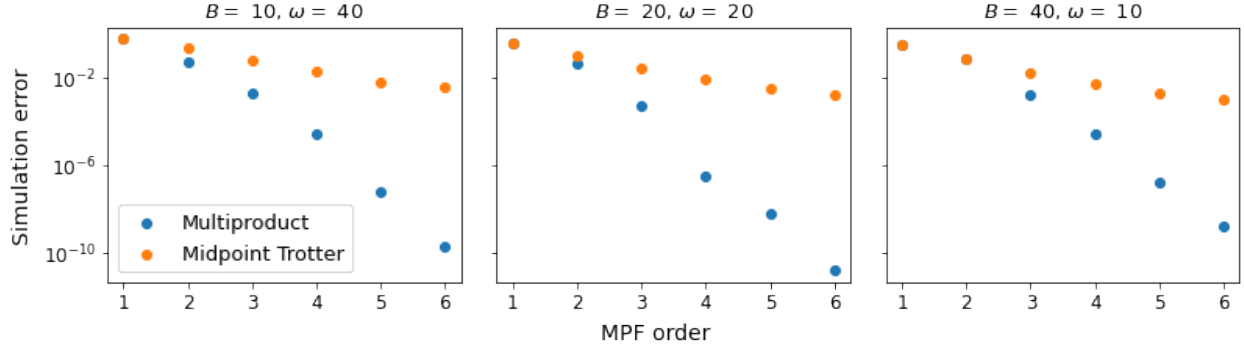


Figure 5.3 Simulation error (spectral norm) of MPFs and midpoint-formula Trotterization, for the spin-1/2 system, with number of midpoint-formula queries kept fixed between the two. Each plot corresponds to different values for the parameters B and ω , always with $\theta = \pi/6$. The number of MPF steps r_{mpf} is fixed at 10. The crossover point tends to occur for error $\epsilon > 10^{-3}$, which is large enough for practical significance. Such error tolerances can be orders of magnitude larger than those required in many quantum simulation proposals [112, 84].

frame. In pursuit of a good case study, we seek a (time independent) Hamiltonian $H = H_0 + H_1$ which produces nontrivial time-dependence in the so-called "interaction picture." We also ask that it satisfies a simple conservation law. A special instance of the 1D XX model will suffice to meet these conditions. Consider a circular chain of N qubits with nearest-neighbor hopping interactions, with Hamiltonian $H = H_0 + H_1$ of the form

$$\begin{aligned}
 H_0 &= \sum_{k=1}^N \frac{\omega_k}{2} Z_k \\
 H_1 &= \sum_{k=1}^N \frac{J_k}{2} (X_k X_{k+1} + Y_k Y_{k+1}).
 \end{aligned}
 \tag{5.130}$$

Here, ω_k, J_k are real, site-dependent parameters, and any index increments are done modulo N . For any value of the parameters, the Hamiltonian conserves the total magnetization $\mu := \sum_k Z_k$.

$$[\mu, H] = 0
 \tag{5.131}$$

Conceptually will think of H_0 as a "base" Hamiltonian, with perturbation H_1 generating interactions, though we make no assumptions as to the smallness of H_1 . We will switch to an interaction picture which is comoving with the simple dynamics of H_0 . In this frame, the Hamiltonian $\tilde{H}(t)$ is

given by

$$\begin{aligned}\tilde{H}(t) &= e^{iH_0 t} H_1 e^{-iH_0 t} \\ &= \sum_{k=1}^N \frac{J_k}{2} (X_k(t)X_{k+1}(t) + Y_k(t)Y_{k+1}(t))\end{aligned}\quad (5.132)$$

where

$$\begin{aligned}X_k(t) &:= e^{iH_0 t} X_k e^{-iH_0 t} = \cos(\omega_k t)X_k - \sin(\omega_k t)Y_k \\ Y_k(t) &:= e^{iH_0 t} Y_k e^{-iH_0 t} = \cos(\omega_k t)Y_k + \sin(\omega_k t)X_k\end{aligned}\quad (5.133)$$

correspond to rotating the pauli vectors about the z -axis with frequency ω_k . We can express equation (5.132) in terms of the time independent X_k and Y_k of the original frame,

$$\tilde{H}(t) = \sum_{k=1}^N \frac{J_k}{2} \{ \cos(\Delta\omega_k t)(X_k X_{k+1} + Y_k Y_{k+1}) + \sin(\Delta\omega_k t)(X_k Y_{k+1} - Y_k X_{k+1}) \}, \quad (5.134)$$

where $\Delta\omega_k = \omega_{k+1} - \omega_k$. We see that having different qubit frequencies ω_k on neighboring sites should give rise to a nontrivial time-dependence in \tilde{H} . Another indication is gleaned from the commutator of H_0 and H_1 .

$$[H_0, H_1] = -i \sum_k \frac{J_k}{2} (X_k Y_{k+1} - Y_k X_{k+1})(\Delta\omega_k) \quad (5.135)$$

The time dependence in H_I will be nontrivial when the commutator does not vanish, as occurs when $\Delta\omega_k \neq 0$. A simple choice is to set

$$J_k = J, \quad \omega_k = (-1)^k \omega. \quad (5.136)$$

That is, the qubit frequency alternates sign at each site, and the coupling is translation invariant. For simplicity, we consider only even numbers of qubits to avoid frequency-matching at $k = N$. Plugging (5.136) into the expression for \tilde{H} in (5.134),

$$\tilde{H}(t) = \frac{J}{2} (\cos(2\omega t)G_1 + \sin(2\omega t)G_2) \quad (5.137)$$

where

$$\begin{aligned}G_1 &= \sum_{k=1}^N X_k X_{k+1} + Y_k Y_{k+1} \\ G_2 &= \sum_{k=1}^N (-1)^k (X_k Y_{k+1} - Y_k X_{k+1}).\end{aligned}\quad (5.138)$$

As a final check, one can see that G_1 and G_2 do not commute with each other. Yet they both commute with μ . Thus, $\tilde{H}(t)$ given in (5.137) is our model system to investigate.

Assuming \tilde{H} commutes with an observable μ , to what degree does the MPF $U_{2,m}$ conserve μ ? Since $U_{2,m}$ is an algebraic combination of exponentials of \tilde{H} , $U_{2,m}$ also commutes with μ . If $U_{2,m}$ were truly unitary, then the operator μ would evolve in the Heisenberg picture as

$$\mu_{2,m}(t) := U_{2,m}^\dagger(t)\mu U_{2,m}(t) = \mu \quad (5.139)$$

as it would under the exact propagator U . However, $U_{2,m}$ is not necessarily unitary.

$$U_{2,m}^\dagger(t)U_{2,m}(t) \neq I \quad (5.140)$$

This implies that conservation laws are only approximately conserved.

$$\mu_{2,m}(t) - \mu = \left(U_{2,m}^\dagger(t)U_{2,m}(t) - I \right) \mu \neq 0 \quad (5.141)$$

Because $U_{2,m}(t) - U(t) \in O(t^{2m+1})$, so is $\left(U_{2,m}^\dagger(t)U_{2,m}(t) - I \right)$.

Figure 5.7.2 plots the deviations in the conserved μ , $\|\mu - \mu_{2,m}(t)\|$, with respect to the simulation time. As the simulation time tends to zero, we see the expected power-law scaling, as evidence by the linear relationship on a log-log plot. For larger m , the slope and hence power p increases, corresponding to improved performance. We can extract the power as the slope of the line, and this is plotted in the right frame. Notice there are sudden dips in the error at specific simulation times, which tend to occur before reaching the power law scaling regime. This could be due to cancellation between two terms in an error series of comparable magnitude. Similar phenomenon occurs in several other contexts, such as the error from adiabatic evolution [135]. Conclusive identification of these phenomenon will require further study.

Naively, we would expect $p = 2m + 1$, but here we actually get slightly better: $p = 2m + 2$. In fact, this scaling can be justified. The following argument, a variant of which can be found in [28], shows that the integrator is nearly unitary.

Theorem 5.7.1. *The deviation of $U_{2,m}$ from being unitary obeys*

$$\|U_{2,m}^\dagger(t)U_{2,m}(t) - I\| \in O(t^{2m+2}).$$

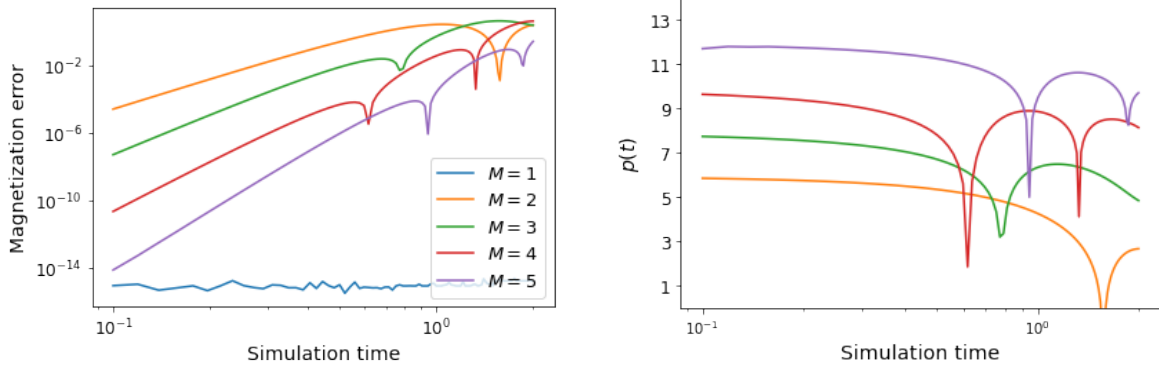


Figure 5.4 (left) Deviations from the conservation of magnetization μ under time-evolution by MPFs. Note that the order $m = 1$ is simply a product formula evolution, which conserves μ exactly. For small simulation times, the expected power-law scaling is observed, with larger powers as m increases. (right) The running power $p(t, t')$ as defined in (5.128), with $t' = .3$. Note the plateau at $2m + 2$, which indicates slightly better convergence than naively expected ($p = 2m + 1$). This phenomenon generalizes to other systems and is formalized by Theorem 5.7.1. Parameter values: $N = 4, J = 1, \omega = 4$.

Proof. We suppress all function evaluations at t when convenient. Let $E := U_{2,m} - U$, so that $U_{2,m} = U + E$. Then, using the unitarity of U and the fact that $E \in O(t^{2m+1})$,

$$U_{2,m}^\dagger U_{2,m} = I + N + O(t^{4m+2}) \quad (5.142)$$

where

$$N := U^\dagger E + E^\dagger U. \quad (5.143)$$

Since $N \in O(t^{2m+1})$, all of its derivatives up to degree $2m$ vanish when evaluated at $t = 0$. Hence, it suffices to show that

$$N^{(2m+1)}(0) = 0. \quad (5.144)$$

We can expand this derivative in terms of E and U using the binomial theorem. When we evaluate at $t = 0$, those terms with derivative less than degree $2m + 1$ in E vanish. We are left with

$$N^{(2m+1)}(0) = E^{\dagger(2m+1)}(0)U(0) + U^\dagger(0)E^{(2m+1)}(0). \quad (5.145)$$

We have $U(0) = U^\dagger(0) = I$. Moreover, by the time-symmetric property of U and $U_{2,m}$, $E(t)$ is also symmetric. Therefore

$$E^{\dagger(2m+1)}(0) = E^{(2m+1)}(-t)\Big|_{t=0} = -E^{(2m+1)}(0). \quad (5.146)$$

Hence, the two terms in (5.145) cancel, yielding $N^{(2m+1)}(0) = 0$. This completes the proof. \square

In summary, though MPFs do not inherently preserve commutations laws, the error is due to nonunitarity in $U_{2,m}$. This can be bounded and reduced in a systematic way, either by decreasing the time step or increasing the MPF order.

5.8 Discussion

In this chapter, we presented an algorithm for time dependent Hamiltonian simulation that uses multiproduct formulas to boost the accuracy compared to product formula simulation. Our algorithm inherits the commutator scaling of product formulas, giving a benefit over comparable methods such as the Dyson series approach. We provide a rigorous characterization of the simulation error as well as query computational complexity. Numerical demonstrations validate the effectiveness of time dependent MPFs in achieving high-accuracy simulations.

Several avenues for future research are immediately apparent from this chapter. First, a proof of Conjecture 1 is highly desirable. Currently, we are investigating modifications of the clock space construction that keep the clock state width σ fixed, allowing for completion of the argument. Numerical demonstrations beyond the simple examples here would be desirable for showing scaling to larger systems, and of course, eventually the method should be tested on actual quantum hardware. To summarize, time dependent MPF simulation is a new algorithm which complements many existing approaches and likely will perform well on systems with a large degree of locality.

5.9 Algorithm for Time Mesh

For completeness, I include the greedy algorithm for generating the time mesh used in the MPF simulation. I thank Alessandro Roggero for devising the approach presented below.

The mesh construction of Section 5.5, although theoretically sound, is not directly implementable since it requires knowing the total number of steps while constructing each new point based on local data. To avoid this issue, as well as the restriction $|\dot{\Lambda}(\tau)| \leq K\Lambda^2(\tau)$ we seek a simple-to-use greedy algorithm.

One possibility is to use a direct approach which first selects a candidate number of steps r_{try} . Starting from $r_{\text{try}} = 1$, we then build recursively a sequence of times using the condition (see

Eq. (5.83) in the main text)

$$\max_{t \in [t_{i-1}, t_i]} \Lambda_{2m}(t) (t_i - t_{i-1}) \leq \frac{1}{41} \left(\frac{\epsilon}{0.32 \|a\|_1 r} \right)^{1/(2m+1)}, \quad (5.147)$$

with $r = r_{\text{try}}$. Starting from $t_0 = 0$ and looking for the largest t_i that satisfies the condition, we finally check whether the generated number of intervals is greater than r_{try} in which case we increase r_{try} by one and repeat. When the algorithm stops at the optimal value r_{opt} , we have performed a total of $r_{\text{opt}}(r_{\text{opt}} + 1)/2$ non-linear optimization steps, each one requiring multiple evaluations of the left hand side of Eq. (5.147). This can be very demanding when the left hand side of Eq. (5.147) is expensive to evaluate and the optimal number of intervals is around half the upperbound

$$r_{\text{max}} = \left(41(t - t_0) \max_{\tau \in [t_0, t]} \Lambda_{2m}(\tau) \right)^{\frac{2m+1}{2m}} \left(\frac{0.32 \|a\|_1}{\epsilon} \right)^{\frac{1}{2m}} \quad (5.148)$$

obtained considering identical intervals and bounding $\Lambda_{2m}(t)$ with its maximum value over the whole simulation interval $[0, T]$. In this case, finding an approximation to the optimal decomposition requires $O(r_{\text{max}}^2)$ optimization steps, each one requiring multiple evaluations of the left hand side of Eq. (5.147).

We now describe an alternative approach which determines r_{opt} within a factor of 2 and uses only r_{max} evaluations of $\max_{t \in [t_{i-1}, t_i]} \Lambda_{2m}(t)$ and additional $O(\log(r_{\text{max}})r_{\text{max}})$ simple arithmetic operations. This procedure can be used to find a viable, and approximately optimal, decomposition of the time interval or as a good starting point to find the optimal one using a procedure as the one described above. The idea is to start by decomposing the interval $[0, T]$ into r_{max} segments with equal length and storing the maximum of $\Lambda_{2m}(t)$ in each segment in an array A of size r_{max} . We then introduce an additional array of the same size

$$L_m = \left[\max_{k \leq m} A_k \right] m \frac{T}{r_{\text{max}}}, \quad (5.149)$$

together with an additional set of vectors of the same size

$$R_m^{(n)} = \left[\max_{n \geq k > m} A_k \right] (n - m) \frac{T}{r_{\text{max}}}, \quad (5.150)$$

with n an additional index between 1 and r_{max} . The first vector stores the left hand side of Eq. (5.147) for the interval up to the m -th time while the second vector stores the same information

for the interval starting at the m -th time and ending at the n -th one. The algorithm proceeds by splitting the time interval recursively into two parts so that the left hand side of Eq. (5.147) takes (approximately) the same value on both halves (ie. we are splitting the error equally on both sides). At every iteration the number of intervals doubles and the right hand side of Eq. (5.147) shrinks accordingly. We stop the procedure once Eq. (5.147) is satisfied on one interval (since we are guaranteed it will in all others). The procedure will stop at some r_K at which point we know the optimal value r_{opt} is in $[[r_K/2], r_K]$. The algorithm can then be described as follows

1. Compute L_m for all $m = 1, \dots, r_{\text{max}}$
2. Set $n = r_{\text{max}}$ and $r = 2$
3. Compute the elements of $R_m^{(n)}$ for all $m = 1, \dots, n - 1$
4. Initialize an auxiliary array D_m as $D_m = L_m - R_m^{(n)}$
5. Find the least index k for which $D_k > 0$
6. If L_k is less than the right hand side of Eq. (5.147) with the current value of r , set $r_K = r$ and exit
7. If $2r \geq r_{\text{max}}$ set $r_K = r_{\text{max}}$ and exit
8. set $r = 2r$, $n = k$ and repeat from step 3

Step 1 requires r_{max} operations while Steps 3 and 4 cost n operations each. Since the number of iterations is bounded by $\log_2(r_{\text{max}})$, their combined cost is bounded by $2 \log_2(r_{\text{max}})r_{\text{max}}$. If we use binary search, Step 5 costs $\log_2(n)$ operations so its total cost is at most $\log_2(r_{\text{max}})^2$ operations. From this analysis we see that Steps 3 and 4 are the most expensive ones and they dominate the cost of the scheme. On exit we have $r_K \approx r_{\text{opt}}$ together with the first interval $[t_0, t_1]$. The rest of the intervals can then be found keeping $r = r_K$ fixed with additional $O(r_{\text{max}})$ operations.

CHAPTER 6

A SIMULATION MIXED BAG

Each of the prior chapters of this dissertation, starting with Chapter 3, focuses on a (mostly) self-contained theoretical project. However, I also contributed to several other projects during my PhD, especially during earlier years. These projects are more concerned with applications on noisy, small-scale devices than with methods suited for relatively noiseless processors. While my contributions were often of a mathematical and theoretical flavor, the ultimate goal was to apply our algorithmic gadgets to existing quantum devices, emphasizing demonstration over rigorous proof.

Each section of the present chapter corresponds to one of these projects, presented in the order they were completed, with corresponding publication referenced within the first few paragraphs. I will begin with a topic that technically falls outside of quantum computing proper, in which we propose investigating discrete scale invariance and anomalous symmetry breaking with trapped-ion quantum simulators. Moving back to quantum computing, I will present a heuristic quantum algorithm for preparing low-lying energy states known as the Projected Cooling algorithm. Finally, I will discuss the aptly but oddly named Rodeo Algorithm, which is a randomized iterative phase estimation algorithm for determining eigenvalues and preparing eigenstates of an observable, provided that observable can be efficiently time evolved.

6.1 Discrete Scale Invariance on Trapped-Ion Systems

Closely related to the idea of simulation by quantum computation is the emulation of a desired Hamiltonian on a controllable and accessible quantum system, namely, *analog* Hamiltonian simulation. See Subsection 2.7.4 for a short background and references to more detailed introductions. In one of my first projects as a PhD student, I proposed, along with collaborators [82], a scheme for simulating scale invariant Hamiltonians on trapped-ion quantum simulators. As of writing, trapped ions are one of the most mature platforms for exquisite manipulation of quantum systems. We begin this topic by introducing some of the deep physics ideas motivating our study, such as anomalous symmetry breaking. Then we will discuss our proposal for investigating these concepts on certain Hamiltonians accessible to trapped-ion simulators. We will conclude with some of my

findings on the self-similar dynamics which these systems exhibit in for certain nearly-unbound states.

6.1.1 Scale Invariance and Quantum Anomalies

Understanding how a system changes with scale is a fundamental question in physics. For example: How do aspects of a microscopic system affect behavior at larger scales? Which of these aspects are relevant, and which get washed out by averaging or some other mechanism? In modern parlance, such questions are formalized in the theory of renormalization [91], which touches on essentially all aspects of modern physics, from the Standard Model to phase transitions. Closely related is the concept of emergent phenomena, which deals with how complex many-body systems achieve their properties, not through microscopic properties themselves, but through the complex interactions of these properties. It is remarkable that the new descriptions produced by these microscopic interactions are even intelligible. Renormalization provides a partial answer, showing how the effect of changing scales can often be incorporated, to good approximation, by changes to couplings in the physical theory.

Another important concept in physics is symmetry. Symmetries provide such a powerful framework for understanding a physical system that little progress in physics could be made without them. Noether's theorem relates certain continuous symmetries to conservation laws, and in practice symmetries allow for the reduction of computation and analysis needed to solve a problem. While symmetries are by nature elegant and simplifying, the *breaking* of symmetry has proven an essential idea to modern physics. In this project, we explored how quantum anomalies—the breaking of symmetry in a classical Hamiltonian by quantization—can be studied using a class of Hamiltonians that are implementable on existing trapped-ion simulators.

Symmetry is merely a change that, in fact, leads to no change at all. When a scale transformation is performed, yet the system looks the same at this new scale, we have *scale invariance*. This is an important idea in field theory and renormalization (fixed points and conformal field theory). Even more "mundane" standard quantum mechanical systems can exhibit scale invariance. Take,

for example, the $1/r^2$ potential

$$H = \alpha \frac{p^2}{2} + \frac{\beta}{r^2} \quad (6.1)$$

for $\alpha \in \mathbb{R}_+$ and $\beta \in \mathbb{R}$. Performing a canonical scale transformation $r \mapsto \lambda r$ and $p \mapsto \lambda^{-1} p$ for any $\lambda \in \mathbb{R}_+$, we find that $H(\lambda r, p/\lambda) = \lambda^{-2} H(r, p)$. Classically, this leads to a trivial change in the phase space dynamics whereby the trajectories are the same, but time rescales as $\lambda^2 t$. Quantum mechanically, the time evolution operator is rescaled by the same factor. Remarkably, even this simple example leads to a quantum anomaly for sufficiently attractive potentials [20, 62]. The full scale invariance is broken to a discrete one, giving rise to a tower of bound states approaching $E \rightarrow 0$ related by a geometric sequence. Thus, this simple Hamiltonian, which may arise from the interaction of an electromagnetic dipole and point charge, already exhibits a surprising richness. What's more, the potential turns out to be intimately related to the Efimov effect [41, 100], a curious phenomenon in which bosons, with short range interactions, cannot form a two-body bound state, but three bosons together have an infinite number of bound states. The binding of three particles, but not two, recalls the intriguing Borromean rings [34], which are linked such that cutting any single ring unlinks all the rings. Since the original discovery, Efimov physics has been linked to a broad class of phenomena, and theoretical interest has been recently growing [100].

6.1.2 Trapped-Ion Systems

One of the most developed hardware platforms for quantum information processing is the ion trap [19], first proposed as a platform for universal quantum computation shortly after Shor's groundbreaking factoring algorithm [30]. Atomic ions, typically Ytterbium or one of several alkaline earth metals, are trapped using lasers in a 1D chain. External control and readout of the system is mediated by laser pulses, and interactions between the ions are mediated, interestingly, through coupling with phononic modes [101] in the 1D chain. Trapped ions possess a number of internal states that can be used as the informational degree of freedom (e.g., qubit) including hyperfine, Zeeman, optical, and fine transitions. The impressive degree of control, all-to-all connectivity, high fidelity of operations, and long coherence times make trapped ions a promising platform for quantum hardware experiments and computations.

Although the technology is being pursued for general computation, these systems can also be operated as analog simulators. Our study is motivated by a particular long range interaction that can be engineered between the ions. Using two hyperfine "clock" states, the following effective spin Hamiltonian can be approximately generated [104].

$$H = \sum_{i,j} \sum_{k=x,y,z} J_{ij}^k \sigma_k^{(i)} \sigma_k^{(j)} + \sum_i \sum_{k=x,y,z} B_{ik} \sigma_k^{(i)} \quad (6.2)$$

Here $\sigma_k^{(i)}$ is the k th Pauli matrix on site for ion i . The parameters J are related to the spatial coordinates of the ions and can be tuned separately for each k using laser pulses. At long ranges, the couplings fall off as

$$J_{ij}^k \approx \frac{J_0^k}{|i-j|^\alpha} \quad (6.3)$$

for chosen $\alpha \in (0, 3)$ [104, 72] and couplings J_0^k .

It is the long-range character (6.3) that we find particularly interesting. By factorizing (6.2) into an xy piece and a z piece, and choosing appropriate coupling parameters, we can create effective kinetic-potential Hamiltonians with scale invariant properties. First, making the choice

$$J_{ij}^x = J_{ij}^y \equiv J_{ij}, \quad J_{ij}^z = V_{ij}, \quad B_{ik} = \delta_{kz} U_i \quad (6.4)$$

we find that the Hamiltonian factorizes as $H = T + V + U$, where T is a kinetic term, V is a potential interaction term, and U is a on-site potential. With this parametrization, the full Hamiltonian becomes

$$H = \sum_{i \neq j} J_{ij} (X_i X_j + Y_i Y_j) + \sum_{i \neq j} V_{ij} Z_i Z_j + \sum_i U_i Z_i. \quad (6.5)$$

The "particles" of the system are given by the spin state: we say that there is a particle at site j if a Z_j measurement yields +1, i.e., we equate particle occupancy with the binary value of the Z_j observable. Notice that $\sum_i Z_i$ is conserved by H , hence particle number is conserved. We can interpret these particles as hard-core bosons, with raising and lowering operators given by σ_\pm , and strong repellant interactions at short distances preventing multiple occupancy of a site. Hard-core bosons might serve as a reasonable model for a collection of nuclei with even nucleon number, since nuclei are repulsive at low energies.

We want to consider the Hamiltonian (6.5) as a discretization of some continuous-variable system with scale invariance. As such, we will need to consider low energies where wavelengths are large compared to the lattice. Considering only the kinetic term T alone, with asymptotic form (6.3) of the hopping coefficient, and neglecting boundary effects by assuming a large chain, the eigenenergies are given by

$$E(p) = 4J_0 \sum_{m=1}^{\infty} \frac{1}{m^\alpha} \cos(pm). \quad (6.6)$$

Performing a low momentum expansion, we find that for $\alpha < 3$

$$E(p) = 2J_0 \sin(\alpha\pi/2)\Gamma(1 - \alpha)|p|^{\alpha-1} + O(p^2). \quad (6.7)$$

Here is the first part of our scale invariant Hamiltonian. Now we consider the potential part. We fix one of our bosons ("spin up") on one of the sites, say site 0, with a very deep on-site potential U_0 . We then add a second boson and tune the potential interaction V_{ij} such that, at large distances,

$$V_{ij} \sim \frac{V_0}{|i - j|^{\alpha-1}}. \quad (6.8)$$

Then, we see that the total Hamiltonian

$$H = 2J_0 \sin(\alpha\pi/2)\Gamma(1 - \alpha)|p|^{\alpha-1} + \frac{V_0}{|r|^{\alpha-1}} + O(p^2) \quad (6.9)$$

is scale invariant at low momenta. With choice of parameters $J_0 < 0, V_0 < 0$, this Hamiltonian has an attractive potential. In our paper's Supplemental Material [82], we provide the limit cycle boundary $E = 0$ state as well as the discrete scaling factor relating the bound state energies near this threshold.

6.1.3 Self-Similar Dynamics

In the previous sections, we have discussed discrete scale invariance in the bound state spectra of certain Hamiltonians, and how these could be constructed on trapped-ion systems. Discrete scale invariance implies self-similarity as one changes their "field of view" by certain discrete amounts.

Besides providing equilibrium information via the Gibbs state, the bound state spectrum impacts the closed-system dynamics. Can the self-similarity of the bound state spectrum manifest in the

dynamical evolution of a chosen initial state? We will show that the answer is yes, though the initial state must be chosen carefully and may be difficult to prepare.

The idea is to prepare a weakly bound state that overlaps many of the eigenstates just below the $E = 0$ threshold. Take as initial state

$$|\psi(0)\rangle = \sum_{n=0}^{\infty} c_n |n\rangle \quad (6.10)$$

where n indexes the bound states of energy $E_n < 0$, with some chosen E_0 as the lowest energy state. These energies are related in geometric sequence as $E_n = E_0/\lambda^n$ for some $\lambda > 1$. The time evolved state is given by

$$|\psi(t)\rangle = \sum_{n=0}^{\infty} c_n e^{-iE_n t} |n\rangle = \sum_{n=0}^{\infty} c_n e^{-iE_0 t/\lambda^n} |n\rangle. \quad (6.11)$$

Suppose now we rescale t by λ . Because $E_{n-1} = E_n \lambda$, this is simply $\sum_n c_n e^{-iE_{n-1} t} |n\rangle$. Without relating the coefficients c_n , there isn't much more that can be said. Imposing the condition $c_n = \gamma c_{n-1}$, however, we find

$$\begin{aligned} |\psi(\lambda t)\rangle &= \gamma \sum_{n=1}^{\infty} c_{n-1} e^{-iE_{n-1} t} |n\rangle + c_0 e^{-iE_0 \lambda t} |0\rangle \\ &= \gamma U_+ |\psi(t)\rangle + c_0 e^{-iE_0 \lambda t} |0\rangle \end{aligned} \quad (6.12)$$

where $U_+ |n\rangle := |n+1\rangle$. Besides the piece proportional to $|0\rangle$, we have a representation of the initial state $|\psi(0)\rangle$ on the subspace $\text{span}\{|n\rangle\}_{n=1}^{\infty}$. The scaling factor $\gamma \in \mathbb{C}$ must satisfy $|\gamma| < 1$ for normalization purposes, and in fact $|c_0|^2 + |\gamma|^2 = 1$. Already, we see some manifestation of self-similarity.

Some feasible measurement must be performed to extract the information about the state (6.12). Let's consider the overlap observable $W(t) := \langle \psi(0) | \psi(t) \rangle$, which might be obtained with two copies of the initial state, one of which time-evolved, then performing some version of the SWAP test. We find

$$W(t) = (1 - |\gamma|^2) \sum_{n=0}^{\infty} |\gamma|^{2n} e^{-iE_0 t/\lambda^n}. \quad (6.13)$$

Up to normalization and nomenclature, the real part of this is nothing more than a Weierstrass function, defined in the original paper as

$$\sum_{n=0}^{\infty} a^n \cos(b^n \pi x). \quad (6.14)$$

In particular, we make the identifications

$$a = |\gamma|^2, \quad b = \lambda^{-1}, \quad x = \frac{E_0 t}{\pi}. \quad (6.15)$$

For certain values of the parameters a, b the function (6.14) is a fractal, being continuous everywhere but differentiable nowhere. This is not the case we are currently in; our evolution is mathematically smooth. As n increases in the series, the frequencies b^n decrease exponentially, leading to an utterly smooth behavior. This is not surprising, since all of the bound eigenstates present in our initial state have frequencies within $[E_0, 0]$.

Despite this, we can recover an approximate self similarity at long time scales. Consider $\gamma = 1 - \delta$, with $\delta > 0$ taken very small. Then, for values $N \in \mathbb{Z}_+$ such that $n\delta \ll 1$ for all $n < N$, the first N levels have approximately equal weight in the superposition. Shifting $t \mapsto \lambda t$ then leaves the function approximately unchanged, up to a small high-frequency component. For these N levels, then, we get

$$\sum_{n=0}^N \cos(\lambda^{-n} E_0 t) \quad (6.16)$$

exhibiting self similar behavior. Moreover, if we rescale to $t \mapsto \lambda^n t$ and reindex $n \mapsto N - n$, we get something which looks like a truncation of a Weierstrass function

$$\sum_{n=0}^N \cos(\lambda^n E_0 t). \quad (6.17)$$

In short, by zooming out to larger time scales, the short time scales appear fractal. The Weierstrass function is famous as being an example of a continuous function that is differentiable nowhere. It is also a fractal, with fractal dimension

$$D = 2 + \frac{\log a}{\log b} = 2 \left(1 - \frac{\log |\gamma|}{\log \lambda} \right). \quad (6.18)$$

In our case $D = 2$, so our curve is space filling. Different choices of coefficient c_j , in which lower frequencies have higher amplitude, could produce different values of a and therefore different fractal dimensions.

6.1.4 Discussion

In this Section, we show how trapped-ion quantum simulators can be used to investigate Hamiltonians with anomalous symmetry breaking due to quantization, with applications to Efimov physics. We characterized the nature of the discrete scale invariance of the bound state spectrum for a family of scale-invariant Hamiltonians. Finally, we indicate how self-similar dynamics, reminiscent of the Weierstrass function, can be obtained through particular state preparation and measurement.

Unfortunately, technological challenges remain to implementing the low energy Hamiltonian (6.9), particularly due to the need for long wavelengths relative to the ion spacing. As of writing, trapped ion quantum computers are of size at most 60 [50, 140], which appears small enough to introduce unwanted boundary effects. While there are no fundamental limitations to the size of the ion trap, implementing interactions between the ions becomes increasingly challenging and expensive as the size scales up [98]. We hope these challenges may be resolved by future improvements in technology or clever implementation of our approach.

6.2 Projected Cooling Algorithm

Having considered analog quantum simulators, we return back to digital quantum computing, but still with an analog flavor. Absent of error correction and logical qubits, current quantum computers mainly reside in the regime of Noisy Intermediate Scale Quantum (NISQ) devices. Such devices implement imperfect operations and are subject to decoherence, severely restricting maximum computation time. For algorithm developers looking to find near-term application of quantum computers, it is important that the proposed algorithms are relatively noise insensitive.

In this section, we will discuss the Projected Cooling Algorithm [83], which prepares ground states, or generally low-lying states, of a kinetic-potential Hamiltonian with localized interactions and translation-invariant kinetic terms. Such systems are common in nuclear physics, as nuclear

interactions are spatially local and often only a few deeply bound states exist. I will discuss the main idea of the algorithm, followed by an application to the Dirac delta potential.

6.2.1 Background

Preparing ground states of a Hamiltonian is valuable for both scientific computing and mathematical optimization. This is known to be a hard problem in general [78] and even determining basic properties, such as the existence of a spectral gap, is undecidable [35]. However, the importance of determining ground state properties is so great that generic hardness is not a deterrent to trying. Moreover, generic hardness of the ground state problem does not necessarily imply hardness among the instances of "physical interest."

Several heuristic or partial algorithms exist for preparing ground states, or low-lying eigenstates. One of these is the adiabatic algorithm, in which a ground state of a simple Hamiltonian H_0 is prepared, then evolved according to a slowly varying time dependent $H(t)$ which takes H_0 to the Hamiltonian H of interest. The premise of the algorithm, as the name suggests, rests on adiabaticity: provided that the evolution is "sufficiently slow," the state $|\psi(t)\rangle$ will remain in the ground state of $H(t)$ for any time t . The slowness required is related to the inverse gap between the ground state and the next excited state at any given time. Unfortunately, it can be challenging to ensure that the gap does not decrease exponentially with system size, despite clever choice of initial Hamiltonian and trajectory $H(t)$ in model space. As another example, we briefly mention Variational Quantum Eigensolver (VQE) [22], which is an optimization approach to finding the ground state whereby circuit parameters are optimized to reduce the energy of the state. Unfortunately, variational methods are known to suffer from barren plateaus [95], which provide a poor optimization landscape for gradient-based methods to succeed.

A distinct class of approaches, in which we include projected cooling, may be characterized as measurement-based. Simple projective measurements, as the name suggests, project onto a state, or subspace, determined by the measurement outcome. By performing a measurement compatible with the energy basis, it is possible to post-select on measuring a result compatible with the ground state. The main limitation arises when the probability of successful measurement is vanishingly

small. Indeed, for a randomly prepared initial state, the expected overlap with the ground state should decrease exponentially with system size. A natural objection to this grim outlook is the existence of clever ansatzes, such as Hartree-Fock states, that have been developed in the domain sciences long before quantum computing, which can be expected to have much higher overlap than a randomly chosen state.

The canonical and highly general measurement-based scheme for state preparation is the suite of phase estimation algorithms, to be discussed in greater detail in the next Section. At a high level, phase estimation is nothing more than a projective measurement in the eigenbasis of a chosen unitary U . When U is a time-evolution operator for time independent H , this corresponds to a measurement of energy states. Given initial state $|\psi_0\rangle$ with fidelity p_{gs} with respect to the ground space of H , a phase estimation protocol will produce the (approximate) ground state with (approximate) probability p_{gs} . Thus, repeating $O(1/p_{\text{gs}})$ times is sufficient to produce the ground state.

The generality of phase estimation suggests a tradeoff in the form of computational difficulty. Phase estimation algorithms require at least one auxiliary qubit to use as a control register for a controlled time evolution. Moreover these operations do not necessarily respect hardware connectivity, as the control register must talk to all qubits in the main system. For present hardware, these demands can be prohibitive.

6.2.2 Projected Cooling

To avoid the demands imposed by phase estimation, focusing on a more specific class of Hamiltonians is desirable. We take inspiration from nuclear physics. Nucleons exhibit strong, short-range interactions, and thus their bound states are also spatially localized. Many nuclei have only a few deeply bound states, and possibly many more shallow bound states. For example, the simple deuteron is known to have only one bound state. When nucleons are freed from the short-range potential, they enter scattering, or continuum states that evolve according to a translationally-invariant kinetic Hamiltonian. These scattering states tend to disperse away from the interaction region, while the bound states naturally remain localized in the interaction region.

The above discussion suggests a simple criterion for distinguishing bound and unbound states for such systems: long-term localization. Evolved under the nuclear Hamiltonian, bound states will stay localized in the interaction region, while unbound states will disperse away. A measurement of the nuclear configuration (i.e., many-particle position measurement) should distinguish the two cases. To retain coherence, a binary measurement should be made which only records whether particles are found outside the range of interaction. Assuming none are, we can expect the system to have lower energy content, since the particle distance from the interaction region is correlated with energy.

Such reasoning is the basis of the Projected Cooling Algorithm. We begin by preparing an initial state localized with respect to the interaction. We then perform a time-evolution according to the natural Hamiltonian of the system for some time T . Finally, a binary measurement is performed which asks whether or not particles are found outside the region in which the potential V has greatest support.

It is reasonable to expect the initial state to have higher overlap with the ground state, barring special symmetry, than a generic state far from the interaction region. This suggests simple yet effective ansatzes are available for these localized systems. The simulation time T required depends, in principle, on the nature of the potential. Resonances will tend to weaken the effectiveness of the algorithm by keeping continuum states around longer. At the same time, T cannot be chosen longer than the velocity of the dispersive component of the wavefunction, or else there will be reflections due to the finite box size. The final measurement M is in the configuration basis, which can be made simple by choosing it as our computational basis. Altogether, no auxiliary registers and additional controlled operations are necessary. The time evolution itself can be expected, generally, to be the most difficult subroutine of the algorithm.

6.2.3 Application to Approximate 1D Dirac Delta Potential Well

We demonstrate the Projected Cooling procedure with a 1D particle in an attractive, localized, deep potential square well. The Hamiltonian is

$$H = \frac{p^2}{2m} + V \tag{6.19}$$

where

$$V(x) = \begin{cases} -V_0 & |x| < L \\ 0 & |x| \geq 0 \end{cases} \quad (6.20)$$

for $V_0, L > 0$. We take a discretization of this into $2N$ sites labeled $n = -N, \dots, N$. We aren't concerned so much with the interior details of the potential as much as its localization, for this simple example. As such, we will assume only the $n = 0$ point is located within the potential; that is, the lattice spacing a is larger than L . The discretized potential then becomes simply

$$\hat{V} = -V_0|0\rangle\langle 0|. \quad (6.21)$$

For the kinetic term, we take a symmetric finite difference approximation. That is,

$$\hat{K} = \frac{1}{2ma^2}(2I - U_+ - U_-) \quad (6.22)$$

where $U_+ = U_-^\dagger$ is the right-shifting unitary operation $U_+|n\rangle = U_+|n+1\rangle$. The discretized Hamiltonian $\hat{H} = \hat{K} + \hat{V}$ has a bound state spectrum that can be analyzed using an ansatz borrowed from the continuum Dirac delta potential. Neglecting finite boundary, we take an ansatz bound state of the form

$$|\kappa\rangle = \sum_{n=-\infty}^{\infty} e^{-\kappa a|n|} |n\rangle. \quad (6.23)$$

Applying \hat{H} to $|\kappa\rangle$, we see that $|\kappa\rangle$ is an eigenstate with energy E provided the following two conditions are satisfied.

$$\begin{aligned} Ema^2 &= 1 - \cosh(\kappa a) \\ 1 - e^{-\kappa a} &= ma^2(V_0 + E) \end{aligned} \quad (6.24)$$

This transcendental equation admits a unique solution for $\kappa > 0$ and $E < 0$. Thus, provided our cutoff N is sufficiently large, we might expect to find such a state following Projected Cooling.

For our numerical example, we work in lattice units where $a = 1$, and we will also take $m = V_0 = 1$. The full discretized Hamiltonian is then

$$\hat{H} = \left(I - \frac{1}{2}U_+ - \frac{1}{2}U_-\right) - |0\rangle\langle 0|. \quad (6.25)$$

We now need to discuss a mapping of this Hamiltonian onto a set of qubits. One natural choice is to encode position into the computational basis. This has the advantage of requiring $O(\log N)$ qubits representing the system, which is valuable because Projected Cooling requires a large enough region for the unbound states to disperse into. In contrast, we could directly represent each lattice site with a qubit. This leads to a less-favorable $O(N)$ scaling, but the advantage is that multiple particles could be allowed, with hard-core repulsion preventing multiple occupancy. Moreover, the operations U_+, U_- are much simpler to implement in this "unary" encoding.

The case of unary encoding is discussed in the paper (Model 1A) [83]. Here, we supplement this work with a discussion of the binary encoding approach. The number of qubits needed to represent the system scales as $n \in O(\log N)$. Preparing Gaussian wavepackets on a quantum register is a well-studied problem [110, 80, 79], and we assume this can be done efficiently.

For simulating \hat{H} , several methods could be employed. Trotterizing along the two terms \hat{K} and \hat{V} , one could simulate \hat{V} using a $C^{n-1}(R_z)$ gate, and \hat{K} by diagonalizing via the Quantum Fourier Transform. Alternatively, we observe that \hat{H} can be expressed as a linear combination of unitaries (LCU), hence is amenable to simulation by qubitization. The number of required queries to the block-encoding "select" SEL and "prepare" PREP circuits scales as $O(T + \log 1/\epsilon)$ for simulation time T and accuracy ϵ . The PREP circuit is only on two qubits, because once the identity terms are removed, there are only 3 unitary pieces. The SEL cost is dominated by the controlled incrementer, which requires $O(n^2) = O(\log^2 N)$ CNOT gates [85]. The potential part of SEL requires the reflection operator $I - 2|0^{\otimes n}\rangle\langle 0^{\otimes n}|$, controlled on the "prepare register." This requires only $O(n)$ gates. The number of auxiliary qubits for the LCU is 2, and does not change with system size.

Once the time evolution is performed, a measurement must be performed to determine if the particle is found significantly outside the region of interaction. A simple computational basis (i.e., position) measurement won't do, since this will destroy the state of interest. Instead, a binary measurement must be done which asks whether the position of the particle is outside of a range R determined by the locality of the state. This can be done with a comparator circuit [115] which requires $O(n)$ gates and a couple of auxiliary qubits to be measured.

6.2.4 Discussion

Here we considered the Projected Cooling algorithm for preparing ground states using the dispersion of unbound states away from the interaction region. While we motivated our approach using nuclear systems, the method can be applied to any system with attractive, localized potential interactions and a translation-invariant kinetic energy. Following our initial work, the method was used to investigate the transverse Ising model [58], with success for models which exhibited dispersion rather than localization. Like other measurement-based state preparation algorithms, success is contingent upon having sufficient overlap with the bound state of interest. For localized interactions, an effective ansatz may correspond to, for example, a Gaussian packet of width on the order of the interaction length.

Determining the required T , ϵ and N analytically to ensure good fidelity with the ground state is beyond our scope. In our paper, we mainly employ classical numerical simulations that suggest good convergence to the desired state, and refer the reader to these results [83]. However, a more careful theoretical treatment is left to be desired. This might be done using a scattering theory treatment. This would likely elucidate the role of resonances, which should propagate slowly and thus hinder the method's effectiveness.

6.3 Rodeo Algorithm

This final section of the chapter concerns a new addition to the suite of iterative Quantum Phase Estimation (IQPE) protocols known as the Rodeo Algorithm. The original algorithm was introduced in [29], where it was tested on the Heisenberg model using classical simulations. Subsequently, an actual quantum computation on IBM quantum computer Casablanca was performed for a simple single-qubit Hamiltonian [106]. In more recent work [11], the Rodeo Algorithm was tested on two-qubit Hamiltonians and used to benchmark a protocol for efficiently compiling complex sequences of controlled operations. Rather than cover all of these works in detail, here I will describe the principles of the algorithm, its relation to iterative phase estimation, and my contributions to the theoretical characterization of the method.

6.3.1 Phase Kickback and Phase Estimation

Needless to say, unitary operations play a vital role in quantum mechanics, particularly quantum computing. A fundamental computational task one might be interested in is estimating the eigenvalues of a unitary U . Given a quantum state $|\psi\rangle$ on the Hilbert space of U , one way to learn the eigenvalues of U is through a projective measurement in the eigenbasis. This has the added benefit of approximately preparing a corresponding eigenstate.

Phase estimation algorithms accomplish precisely this goal. For readers familiar with basic optics, a satisfying analogy exists between phase estimation and Mach-Zehnder interferometers [67]. In fact, the analogy is so close that it is more accurate to say they share the same working principle: measuring a phase shift via interference.

Phase estimation algorithms have been around for as long as quantum computing has garnered significant attention. Shor's famous algorithm for factoring [118], and the more general problem of finding discrete logarithms, rests on phase estimation techniques, as does the HHL algorithm for solving linear systems [64]. Standard QPE, based on the Quantum Fourier Transform, is described in detail in Nielsen and Chuang's well-known text [101]. Kitaev supplied perhaps the first iterative QPE protocol [77], and several improvements to the method have been made since [125, 67, 102]. Adaptive protocols allow for improved phase measurement schemes based on prior measurements [136]. Often iterative QPE refers simply to a "iteratization" of the standard QFT by pushing all controls past the measurements [59].

6.3.2 Basic Circuit

Figure 6.1 exhibits the fundamental iteration of the Rodeo Algorithm, which we term a "cycle." The upper wire represents a single auxiliary qubit, on which single qubit gates such as the Hadamard H and the parametrized phase gate

$$S(\alpha) := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \quad (6.26)$$

act. The lower register is the main register of interest, where $U(t) = e^{-iOt}$ is the time evolution operator for some observable O of interest. The parameter E is called the *target energy*, and set

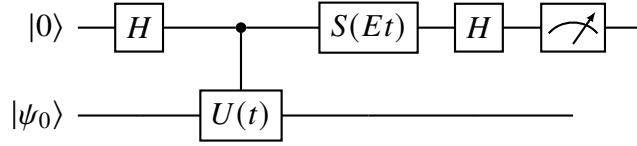


Figure 6.1 Elementary iteration ("cycle") of the Rodeo Algorithm. Times t are randomly sampled from a normal distribution of center 0 and width Γ^{-1} . Performing M cycles acts as a band-pass filter, only allowing eigenvalues within a range centered around E with width Γ . Eigenvalues outside this interval are exponentially suppressed in the number of cycles M . "Success" is conditioned on all M measurement outcomes being 0.

by the user. Meanwhile, t is a random variable sampled from some distribution ρ centered about zero. We take t to be normally distributed with variance $1/\Gamma^2$.

$$\rho(t) = \frac{\Gamma}{\sqrt{2\pi}} e^{-(\Gamma t)^2/2} \quad (6.27)$$

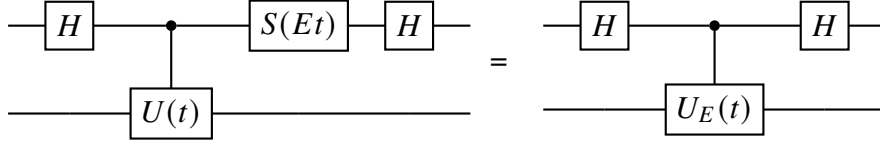
Other reasonable choices exist, but the normal distribution is simple to analyze and performs well enough. Some choices, such as the uniform distribution over $[-\Gamma^{-1}, \Gamma^{-1}]$, are less favorable as they have poorer filtering properties resulting from the distribution having long tails in Fourier (energy) space.

Roughly speaking, the parameters E, Γ define an interval $[E - \Gamma, E + \Gamma]$ for which the Rodeo measurement protocol asks the question: Is there an eigenvalue of O located in $[E - \Gamma, E + \Gamma]$? Like all quantum measurements, this will depend on the state $|\psi_0\rangle$ prepared, and a successful detection will occur with frequency given by the Born rule. As Γ shrinks, longer time evolutions $U(t)$ will occur, and this is expensive. Taking the cost to increase linearly in t , which saturates lower bounds of the no-fast-forwarding theorem [14], the cost scales inversely with the accuracy. This is in accord with the Heisenberg limit for quantum parameter estimation.

The role of the phase gate $S(Et)$ is to shift the spectrum of O by $-E$. This is because the phase acts equivalently to a controlled phase multiplication on the system.

$$\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} S(\alpha) \text{---} = \begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array} e^{i\alpha} \text{---}$$

Applying this identity to the circuit of Figure 6.1 and combining the control gates,



where

$$U_E(t) = e^{iEt}U(t) = e^{-i(O-E)t}. \quad (6.28)$$

For eigenvalues λ of O within $O(\Gamma)$ of E , $e^{-i(\lambda-E)t}$ is relatively unaffected by variations in t of order $O(\Gamma^{-1})$. Meanwhile, other eigenvalues are shifted dramatically as t varies randomly. This we analogize as the "bucking" in the Rodeo Algorithm, where far-away eigenvalues are likely to be kicked off. This will be elucidated more concretely in subsequent analysis.

As far as the author is aware, our algorithm is the first of the phase estimation family to employ random parameters and shift the Hamiltonian in this fashion. Even if our initial state $|\psi_0\rangle$ has small overlap with the eigenstates we are interested in, successive iterations of this circuit, with E and Γ trained on the energy range of interest, allow us to amplify these states and determine whether our operator O has some eigenvalue in the range set by these parameters. We do require $|\psi_0\rangle$ to have some overlap with these eigenstates. However, the threshold for detection can be made increasingly small with repeated cycles.

6.3.3 A Single Buck of the Bull

Let $|\psi_0\rangle$ be the initial state of the main register, as in Figure 6.1. We decompose $|\psi_0\rangle$ into its spectral components in the following way.

$$|\psi_0\rangle = \sum_{\alpha \in \sigma(O)} c_\alpha |\alpha\rangle \quad (6.29)$$

Here, $\sigma(O)$ is the spectrum of O , i.e., the set of (real) eigenvalues, $c_\alpha \in \mathbb{C}$ is the component of $|\psi_0\rangle$ along the eigenspace of α , and $|\alpha\rangle$ is the projection onto this subspace.

$$|\alpha\rangle := \frac{P_\alpha |\psi\rangle}{\sqrt{\langle \psi | P_\alpha | \psi \rangle}}, \quad c_\alpha := \langle \alpha | P_\alpha | \psi \rangle = \sqrt{\langle \psi | P_\alpha | \psi \rangle} \quad (6.30)$$

We will suppress the notation $\sigma(O)$ in the sum (6.29) from now on. The output of the rodeo cycle, before measurement, is given by the well-known result for the Hadamard test.

$$|0\rangle |\psi_0\rangle \rightarrow |0\rangle \left(\frac{I + U_E(t)}{2} |\psi_0\rangle \right) + |1\rangle \left(\frac{I - U_E(t)}{2} |\psi_0\rangle \right) \quad (6.31)$$

Expressing this in terms of the eigenbasis $|\alpha\rangle$ gives

$$\begin{aligned}
|0\rangle |\psi_0\rangle &\rightarrow |0\rangle \otimes \sum_{\alpha} c_{\alpha} e^{-i\Delta\alpha t/2} \cos\left(\frac{\Delta\alpha t}{2}\right) |\alpha\rangle \\
&+ |1\rangle \otimes \sum_{\alpha} c_{\alpha} i e^{-i\Delta\alpha t/2} \sin\left(\frac{\Delta\alpha t}{2}\right) |\alpha\rangle
\end{aligned} \tag{6.32}$$

where we have defined $\Delta\alpha = \alpha - E$. If the measurement succeeds (0), we continue on with the iteration of the circuit and preserve the state of the main register, while for failure we either halt or discard the result at the end of the computation. Assuming success, the new state vector $|\psi'\rangle$ is given by

$$\begin{aligned}
|\psi'\rangle &= \sum_{\alpha} c'_{\alpha} |\alpha\rangle \\
c'_{\alpha} &= c_{\alpha} e^{-i\Delta\alpha t/2} \cos\left(\frac{\Delta\alpha t}{2}\right)
\end{aligned} \tag{6.33}$$

up to normalization. The probability $p'_{\alpha} = |c'_{\alpha}|^2$ is slightly more illuminating.

$$p'_{\alpha} = p_{\alpha} \cos^2\left(\frac{\Delta\alpha t}{2}\right) \tag{6.34}$$

Observe that a relative enhancement of the probability amplitudes p_{α} occurs for $\cos^2(\Delta\alpha t/2) \approx 1$. This certainly occurs for $\Delta\alpha t \approx 0$, i.e., for α near the target energy, but also for $\Delta\alpha t = 2\pi k$ with $k \in \mathbb{Z}$. These peak locations for $k \neq 0$ fluctuate with t . With many cycles, it is unlikely that any eigenvalue far from E will stay on a maximum across multiple trials, as we will see shortly.

The probability of success P_0 and failure P_1 is computed from the squared norm of each term in equation (6.32).

$$\begin{aligned}
P_0 &= \sum_{\alpha} p_{\alpha} \cos^2\left(\frac{\Delta\alpha t}{2}\right) \\
P_1 &= \sum_{\alpha} p_{\alpha} \sin^2\left(\frac{\Delta\alpha t}{2}\right) = 1 - P_0
\end{aligned} \tag{6.35}$$

6.3.4 Multiple Bucks: the Full Rodeo

The extension of the previous analysis from a single run to M runs through the basic circuit of Figure 6.1 is relatively straightforward. Let $(t_i)_{i=1}^M$ be the time samples for each cycle. Then,

with repeated application of equation (6.34), the probability amplitude $p_\alpha^{(M)}$ of α after M runs, conditioned on the circuit succeeding, is given by

$$p_\alpha^{(M)} = p_\alpha \prod_{i=1}^M \cos^2 \left(\frac{\Delta\alpha t_i}{2} \right). \quad (6.36)$$

The success probability $P_0^{(M)}$ may be factorized as

$$P_0^{(M)} = \prod_{k=1}^M Z_k \quad (6.37)$$

where Z_k is the probability of measuring zeros on the k th measurement conditioned on measuring zeros in every prior measurement. Let $|\Psi_{k-1}\rangle = |0\rangle \otimes \sum_\alpha c_\alpha^{(k-1)} |\alpha\rangle$ be the state of the entire register after $k-1$ successful measurements and directly before the k th measurement. From Born's rule,

$$Z_k = \frac{\| \langle 0 | \Psi_{k-1} \rangle \|^2}{\langle \Psi_{k-1} | \Psi_{k-1} \rangle}, \quad (6.38)$$

where $\langle 0 | \equiv \langle 0 | \otimes I$. Adapting equation (6.32) to the present situation,

$$\begin{aligned} Z_k &= \frac{\sum_\alpha p_\alpha^{(k-1)} \cos^2(\Delta\alpha t_k/2)}{\sum_\beta p_\beta^{(k-1)}} \\ &= \frac{\sum_\alpha p_\alpha^{(k)}}{\sum_\beta p_\beta^{(k-1)}}. \end{aligned} \quad (6.39)$$

Returning to expression (6.37), we see that each Z_k in the product telescopes, giving a simple formula.

$$P_0^{(M)} = \sum_\alpha p_\alpha^{(M)} = \sum_\alpha p_\alpha \prod_{k=1}^M \cos^2 \left(\frac{\Delta\alpha t_k}{2} \right) \quad (6.40)$$

That is, the success probability is simply the sum of the unnormalized probability amplitudes. In hindsight, we could have anticipated that, for each α , the measurement probabilities over each iteration behave independently, as exhibited in equation (6.40). Performing the analysis for an eigenvector input state, one finds that the state is unaffected by the measurement outcomes, thus the measurements are independent. The full result follows by linearity.

6.3.5 Statistics

Now that we've determined the behavior for a particular choice of times $(t_i)_{i=1}^M$, we must recall that these times were randomly chosen according to the normal distribution (6.27). We hope,

though have yet to fully justify, that the measurement statistics correlate strongly, and predictably, with the presence of an eigenvalue within $E \pm \Gamma$ that significantly overlaps the initial state. To do this, it makes sense to compute some basic statistics about the measurement results, particularly the expected behavior over the distribution of t_i . Thus, we compute the expectation values of $p_\alpha^{(M)}(t)$ and $\mathcal{P}_0^{(M)}(t)$. Using equation (6.36),

$$\begin{aligned}
\langle p_\alpha^{(M)} \rangle &= \int_{\mathbb{R}^M} \rho(t) p_\alpha^{(M)} dt^M \\
&= \frac{\Gamma}{(2\pi)^{M/2}} \int_{\mathbb{R}^M} e^{-(\Gamma t)^2/2} p_\alpha \prod_{j=1}^M \cos^2\left(\frac{\Delta\alpha t_j}{2}\right) dt^M \\
&= \frac{p_\alpha \Gamma}{(2\pi)^{M/2}} \left(\int_{\mathbb{R}} e^{-(\Gamma t)^2/2} \cos^2\left(\frac{\Delta\alpha t}{2}\right) dt \right)^M.
\end{aligned} \tag{6.41}$$

In the last step, we used the fact that the M -dimensional integral factorizes. This expression is easy to evaluate.

$$\langle p_\alpha^{(M)} \rangle = p_\alpha \left(\frac{1 + e^{-\Delta\alpha^2/2\Gamma^2}}{2} \right)^M \tag{6.42}$$

We see that, for $\alpha - E$ on the order of Γ or greater, the probability decays in the number of cycles as 2^{-M} , whereas for $\sqrt{M}\Delta\alpha/\Gamma \ll 1$ the amplitudes are approximately preserved. The expected success probability $\langle P_0^{(M)} \rangle$ can be easily obtained from equations (6.40) and (6.42), using the linearity of the expectation value.

$$\boxed{\langle P_0^{(M)} \rangle = \sum_{\alpha} p_\alpha \left(\frac{1 + e^{-\Delta\alpha^2/2\Gamma^2}}{2} \right)^M} \tag{6.43}$$

Observe how this serves as an indicator function for the existence of eigenvalues. If all α are outside of $O(\Gamma)$ from E , the amplitudes decay exponentially in M . For any α within this range, however, there will be a success probability which goes roughly as the initial overlap with those states. As expected from measurement-based procedures, we cannot overcome the inherent $O(1/\text{overlap}^2)$ cost scaling. A quadratic improvement to a Heisenberg limit may be feasible with amplitude amplification, but only with additional quantum overhead and operations.

Our indicator $\langle P_0^{(M)} \rangle$ might not be of practical use if the behavior of a typical run deviates

wildly from the average. We thus investigate the variance. From equation (6.40),

$$\begin{aligned}
\text{Var}(P_0^{(M)}) &= \langle (P_0^{(M)})^2 \rangle - \langle P_0^{(M)} \rangle^2 \\
&= \langle \sum_{\alpha\beta} p_\alpha^{(M)} p_\beta^{(M)} \rangle - \left(\sum_\alpha \langle p_\alpha^{(M)} \rangle \right)^2 \\
&= \sum_{\alpha\beta} \text{Cov} \left(p_\alpha^{(M)}, p_\beta^{(M)} \right),
\end{aligned} \tag{6.44}$$

where

$$\text{Cov}(X, Y) := \langle XY \rangle - \langle X \rangle \langle Y \rangle \tag{6.45}$$

is the covariance. To clean up the math below, define the dimensionless parameters $a := \Delta\alpha/\Gamma$ and $b := \Delta\beta/\Gamma$. Then,

$$\begin{aligned}
\langle p_\alpha^{(M)} p_\beta^{(M)} \rangle &= p_\alpha p_\beta \left(\frac{2 + e^{-(a+b)^2/2} + e^{-(a-b)^2/2} + 2e^{-a^2/2} + 2e^{-b^2/2}}{8} \right)^M \\
\langle p_\alpha^{(M)} \rangle \langle p_\beta^{(M)} \rangle &= p_\alpha p_\beta \left(\frac{1 + e^{-(a^2+b^2)/2} + e^{-a^2/2} + e^{-b^2/2}}{4} \right)^M
\end{aligned} \tag{6.46}$$

so that

$$\text{Cov} \left(p_\alpha^{(M)}, p_\beta^{(M)} \right) = p_\alpha C_{\alpha\beta}^{(M)} p_\beta \tag{6.47}$$

where

$$\begin{aligned}
C_{\alpha\beta}^{(M)} &= \left(\frac{1 + e^{-a^2/2} + e^{-b^2/2} + e^{-(a^2+b^2)/2} \cosh(ab)}{4} \right)^M \\
&\quad - \left(\frac{1 + e^{-a^2/2} + e^{-b^2/2} + e^{-(a^2+b^2)/2}}{4} \right)^M
\end{aligned} \tag{6.48}$$

is positive definite. Hence, the variance $\text{Var}(\mathcal{P}_0^{(M)})$ is a contraction of the matrix $C^{(M)}$ with the vector of initial probability amplitudes p for each eigenvalue.

It is fruitful to consider $C_{\alpha\beta}^{(M)}$ as a function of two real parameters $a, b \in \mathbb{R}$ for each $M \in \mathbb{Z}_+$. First, we observe that C is an even function in both a and b , so that only the positive quadrant need be considered. We also observe that C is symmetric under $a \leftrightarrow b$. It is more or less clear that C should approach 0 for a, b large and for a, b near zero, but showing this analytically is rather awkward (though straightforward in principle). Figure 6.2 provides plots of $C_{\alpha\beta}^{(M)}$ for various values

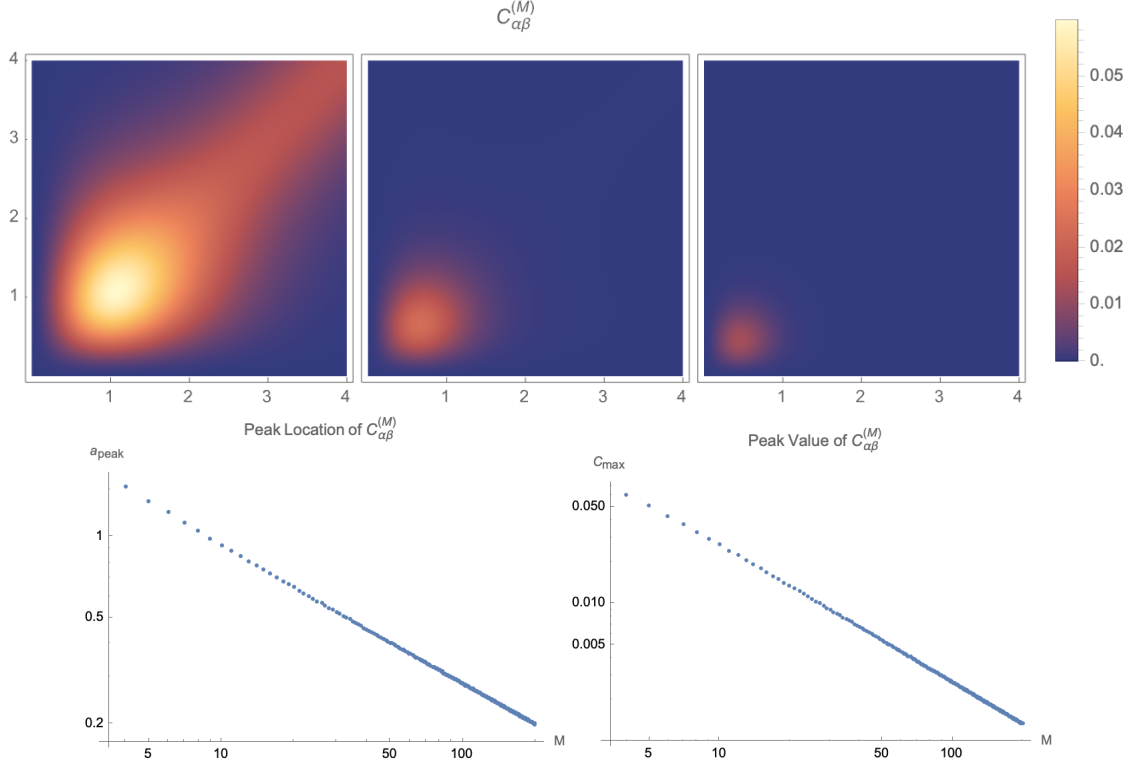


Figure 6.2 (Top) Density plot of covariance function $C_{\alpha\beta}^{(M)}$ with respect to $a = \Delta\alpha/\Gamma$ and $b = \Delta\beta/\Gamma$ for $M = 4, 10,$ and 20 going left to right. We observe the function diminishing with M , with maximum along the $a = b$ line which moves slightly inward with M . (Bottom) Location and values of maximum $C_{\alpha\beta}^{(M)}$ for M from 4 to 200. Line of best fit indicates inverse square root power law for location, whereas value c_{\max} appears to follow inverse power law with M .

of M . We observe the correlations are peaked for $a = b = a_{\text{peak}}$, where a_{peak} appears to follow an inverse square root power law. Even for $M = 4$, the correlations are never larger than 0.06 and only decrease with M .

The peak location corresponds to eigenvalues being on the order of $\Theta(\Gamma)$ away from the target energy. Larger values of $C_{\alpha\beta}^{(M)}$ contribute to a larger variance via (6.44). We interpret this as follows: the Rodeo Algorithm struggles to properly classify eigenvalues close to, but not entirely within, the rough interval $[E - \Gamma, E + \Gamma]$. When eigenvalues are clearly within Γ , the success probability is 1 for those eigenstates, while for far eigenvalues the success probability is a coin toss. This "resonant peak", despite being a nuisance, can be handled by varying the target energy, and in any case the resonance decreases in amplitude and width with M . We will make use of some properties of $C_{\alpha\beta}^{(M)}$ in our subsequent analysis of the Rodeo Algorithm's performance.

6.3.6 Performance on Eigenvalue Detection

Having characterized the statistical properties of our randomized circuit, we now turn to the question of performance. To make progress analytically, we make some simplifying assumptions. Suppose we choose some target energy E and width Γ , and wish to determine whether there is a nearby eigenvalue of α_* , by which we mean an eigenvalue such that $|\alpha_* - E|/\Gamma < d$ for d on the order of 1. We are promised that, if such an eigenstate exists, the input state $|\psi\rangle$ has overlap $p_{\alpha_*} = |c_{\alpha_*}|^2 \geq \delta$, where $\delta > 0$ is some given threshold. Any other populated eigenstates are assumed to have eigenvalues at least $g\Gamma$ away from E for some $g > d$.

We analyze the question of eigenvalue existence in the language of hypothesis testing, with null hypothesis H_{null} of no eigenvalue present. The alternative hypothesis H_{alt} is the presence of α_* within Γd of E that has overlap at least δ with the initial state. Practically speaking, without the promises given above, our algorithm will simply fail to detect eigenstates whose overlap is too low (without the δ promise), or fail to resolve multiple eigenvalues within the detection range (without the g promise).

Under H_0 , the expected success probability P_{null} of the M -cycle Rodeo Algorithm is upper bounded as

$$P_{\text{null}} \leq \left(\frac{1 + e^{-g^2/2}}{2} \right)^M. \quad (6.49)$$

On the other hand, for alternative hypothesis H_1 , the expected success probability P_{alt} is lower bounded as

$$\begin{aligned} P_{\text{alt}} &\geq \delta \left(\frac{1 + e^{-d^2/2}}{2} \right)^M + (1 - \delta) \frac{1}{2^M} \\ &> \delta \left(\frac{1 + e^{-d^2/2}}{2} \right)^M. \end{aligned} \quad (6.50)$$

To distinguish the two cases, we estimate the success probability through a normalized count N samples of the M cycle Rodeo Algorithm. A detection corresponds to determining $P_{\text{alt}} > P_{\text{null}}$ with confidence determined by uncertainty in the method. We denote the estimated probability by \bar{P} . Ignoring other reasonable sources of error, such as imperfect gates, decoherence, and imperfect implementation of $U(t)$, the error in the estimate is $O(\sigma_{\text{sample}})$, where σ_{sample} is the standard

deviation in the estimate of expected success probability. To analyze this uncertainty, we use a heuristic error propagation approach. Assuming an exact, nonrandom success probability P_0 , our uncertainty comes from the binomial variance

$$\sigma_{\text{binom}} = \sqrt{\frac{P_0(1-P_0)}{N}} < \sqrt{\frac{P_0}{N}}. \quad (6.51)$$

To characterize deviations from the binomial distribution due to random fluctuations σ_{fluc} of P_0 caused by the times t_j , we use an error propagation formula.

$$\sigma_{\text{fluc}} \approx \left| \frac{\partial}{\partial P_0} \sqrt{\frac{P_0(1-P_0)}{N}} \right| \sigma_{P_0} < \frac{1}{2} \frac{1}{\sqrt{P_0 N}} \sigma_{P_0} \quad (6.52)$$

Let's now consider the two cases separately, in the case of H_{null} , there is no eigenvalue within $g\Gamma$ and the upper bound

$$\sigma_{\text{fluc}} < \frac{1}{\sqrt{N}} 2^{M/2} \sigma_{P_0} \quad (6.53)$$

is valid. In this case, σ_{P_0} will shrink exponentially, and for some $g \in O(1)$ this will be enough to bound σ_{fluc} by $O(\sqrt{P_0/N})$. Consider now the case H_{alt} . From now on, we make the assumption

$$d^2 M \ll 1 \quad (6.54)$$

so that $P_{\text{alt}} > \delta + O(\delta d^2 M)$. In this case, we require $\sigma_{P_0} < \sqrt{\delta + O(\delta d^2 M)}$. A careful analysis of $C_{\alpha\beta}^{(M)}$, particularly the $\alpha = \beta = d\Gamma + E$ term, reveals that $C_{\alpha\beta}^{(M)} \in O(d^4 M)$. This is sufficient to guarantee an $O(\sqrt{P_0/N})$ scaling as desired.

Overall, we've found, through semi-heuristic derivation, that the error in our Rodeo estimation protocol goes as $O(1/\sqrt{N})$. To ensure the two hypotheses can be distinguished by the Rodeo Algorithm, we thus require

$$\delta \left(\frac{1 + e^{-d^2/2}}{2} \right)^M - \left(\frac{1 + e^{-g^2/2}}{2} \right)^M \in \Omega(\sqrt{\delta/N}). \quad (6.55)$$

Taking as the worst case scenario $g = 1$ in the above, we have the requirement

$$\left(\frac{1 + e^{-d^2/2}}{2} \right)^M \in \Omega \left(\frac{1}{\sqrt{\delta N}} + \frac{1}{\delta} e^{-\Theta(M)} \right). \quad (6.56)$$

We utilize our requirement that $d^2M \ll 1$, so that the right-hand side is $\Omega(1)$. This is implicitly an upper bound on M , meaning that not all d and δ will allow for solutions using our approach. However, we will see presently that the size of M need not be too large. Looking at the 2nd term on the right-hand side of (6.56), we find that

$$M \in \Omega(\log 1/\delta) \tag{6.57}$$

suffices to ensure $O(1)$ error. Finally, we choose $N \in \Omega(1/\delta)$.

We see that there are choices of parameters for which the algorithm can succeed in the setting we've constructed. The only real requirement is that

$$\log 1/\delta \ll 1/d^2 \tag{6.58}$$

which is not a great restriction in practice, provided the overlap is not exceedingly small. This restriction comes about, more or less, from the fact that, as M increases, the effective width of the search window shrinks as $O(1/\sqrt{M})$.

A rough cost estimate C_{ost} can be assigned as

$$C_{\text{ost}} := \frac{NM}{\Gamma} \in O\left(\frac{(1/\delta) \log 1/\delta}{\Gamma}\right) \tag{6.59}$$

which attributes a cost of Γ^{-1} to the time evolution, assuming it dominates the full rodeo cycle. The cost $c_{\text{ost}} = C_{\text{ost}}/N$ per cycle is also the maximum circuit depth, and has the same Γ scaling but favorable δ dependence. Interpreting Γ as our accuracy parameter, we see that the Rodeo Algorithm achieves Heisenberg scaling.

In practice, given a range $[E_{\text{min}}, E_{\text{max}}]$ of possible eigenvalues for the operator O of interest, Γ can be varied using a procedure akin to binary search. This has little effect on the total cost, which will still be dominated by the smallest Γ used assuming it is reduced exponentially each refinement.

6.3.7 Noise

Noise and decoherence are significant sources of error for near-term quantum computers without error correction. Significant advances in error mitigation techniques have increased what can be achieved from limited, noisy hardware [76], though there may be fundamental limitations to these

approaches for large systems [108]. In search of applications for near-term quantum devices, there is a desire for methods which are inherently robust to noise.

We investigated the noise robustness of the Rodeo Algorithm under the simplest possible model of decoherence: depolarizing noise. At each gate, we assume there is some probability p_{depol} of the current state being replaced with the maximally mixed state, thus ruining the computation. Computations were performed using Qiskit Runtime with the "ibmq_qasm_simulator." We considered the simple 3-qubit Hamiltonian $Z_1 + Z_2 + Z_3$, whose eigenvalues are easily seen to be the odd integers from -3 to 3 . We took as initial state $(H|0\rangle)^{\otimes 3}$ where H is the Hadamard. We varied p_{depol} from 0 to 0.05 and investigated the effect on performance. We take $\Gamma = 1$ and $M = 6$, and perform a scan from $[-5, 5]$ in increments of 0.1 . Figure 6.3 shows the results of the numerical simulation. We observe that, for increased noise, the peaks decrease in height but remain in the same location. By $p_{\text{depol}} = 0.05$, the peaks become hard to distinguish from one another, but not hard to distinguish from background. Importantly, the location of the peaks is relatively unaffected, which is the most important part. We expect that, for many reasonable models of noise, not just symmetric depolarization, the peaks will not shift greatly as a result. Imperfect gates will change the effective operator \tilde{O} , however, thus leading to changes in eigenvalues. Understanding how decoherence affects the effective Hamiltonian being evolved would greatly advance our understanding of the impact, not only on the Rodeo Algorithm, but other phase-estimation protocols.

6.3.8 Discussion

Here we presented the Rodeo Algorithm, a new addition to the collection of phase estimation protocols which is simple and allows for targeted search of eigenvalues and preparation of eigenstates. To complement previous work on the method [29, 106, 11], here we focused on theoretical aspects not previously covered. We provide a quasi-rigorous cost analysis of the method under a binary hypothesis model, and conclude with some numerics showing behavior of the method under symmetric depolarizing noise.

Following the initial publication of the method, subsequent research removed the need for randomness in the algorithm while providing a more rigorous characterization of performance [96].

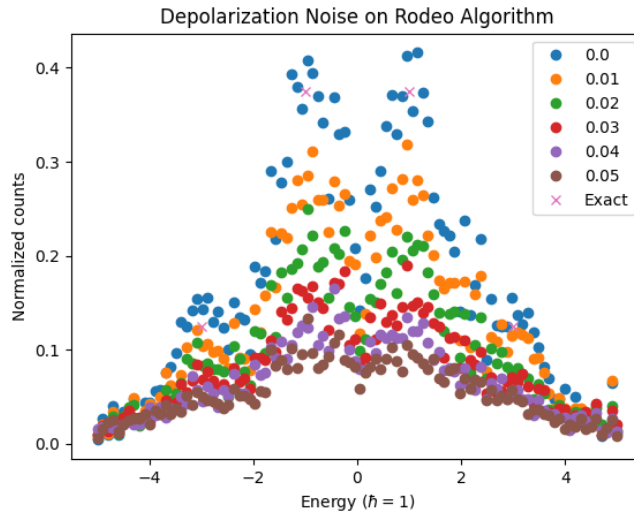


Figure 6.3 Simulation of Rodeo Algorithm for three non-interacting qubits for various amounts of depolarizing noise. Eigenvalues are located at -3 , -1 , 1 , and 3 , and the initial state has overlap $1/8$, $3/8$, $3/8$, $1/8$ with each eigenspace. We see that the algorithm reproduces the correct qualitative behavior in spite of noise reductions in signal. The algorithm benefits from the symmetry of depolarization, whereas other noise models may affect the phase rotation being applied.

These authors were interested particularly in state preparation, and their findings of performance are consistent with our randomized approach.

For current noisy devices, the Rodeo Algorithm can serve as a simple and practical alternative for standard QPE for eigenvalue estimation and state preparation. Thus, it may eventually serve as a useful subroutine for algorithms aimed at achieving quantum advantage, that is, practical advantage of quantum computers over classical computers for interesting tasks. For example, recently an end-to-end simulation algorithm for nuclear effective field theories was proposed, which used standard Quantum Phase Estimation (QPE) in analysis for estimating total resource costs [134]. Replacing this final measurement step with the Rodeo Algorithm in actual applications should increase the feasibility of the full algorithm for near-term hardware.

CHAPTER 7

CONCLUSION AND OUTLOOK

Quantum computing today shares similarities with thermodynamics of the 19th century. From its birth, thermodynamics was intimately linked to technology. Practitioners sought to create more efficient engines, refrigerators, and other cyclical systems to perform work using the resource of heat gradients. These pragmatic goals led to foundational scientific progress, such as discovering laws about optimal efficiencies and allowed physical transformations. Gradually, the powerful and subtle notion of entropy developed, so fruitful as to remain at the forefront of modern physics and information theory.¹

Quantum computers, unlike engines and refrigerators, have yet to yield any practical benefit. Interest in the field is driven by the expectation that they eventually will, and that effective, error-corrected quantum computers can be made that will simulate quantum dynamics, solve optimization problems, and perform hitherto unrecognized yet valuable computational tasks. In the background, deeper questions are very much present. How different are the quantum and classical worlds from a computational viewpoint? What makes quantum probabilities special compared to standard probability? Broadly, what is quantum mechanics even about? Because the building blocks of quantum information, qubits, are so simple, they provide an excellent playground to tackle such questions with clarity. It is imaginable that large-scale manipulation of entanglement will aid our "gut" understanding of the theory and elucidate connections between the microscopic world and our macroscopic reality.

For what purposes will we manipulate this enormous entanglement? Besides Shor's groundbreaking factoring algorithm, Grover's unstructured search algorithm, and a handful of others, not many compelling use cases are known. The truth is, if a scalable, fault-tolerant quantum computer existed today, we wouldn't know many uses for it. The idea of quantum-based quantum simulation has been around about as long as quantum computing itself, and remains one of the few concrete and useful tasks we know of. Thankfully, its importance to scientific computing is enough to command

¹While the "entropy" from physics and information theory are technically distinct, Landauer's principle [81] and other ideas suggest a deep connection that, to my knowledge, is still not fully understood.

attention, and moreover, Hamiltonian simulation finds application in at least a handful general tasks such as solving linear systems of equations. Even if not part of an algorithm, Hamiltonian simulation concepts can assist in the design of quantum algorithms, especially given the absence of many other conceptual guideposts.

This dissertation offers additional tools for Hamiltonian simulation and, importantly, rigorous characterizations of each of their performance. We demonstrate increased effectiveness of product formulas for accurately estimating observable dynamics using polynomial interpolation. We provide several new approaches to time dependent Hamiltonian simulation, as well as a computational reduction of the time dependent to time independent dynamics. Finally, we survey novel tools for near-term quantum simulation, including trapped-ion simulation of anomalous symmetry breaking, state preparation via Projected Cooling, and resource-effective eigenvalue estimation and eigenvector preparation using the Rodeo Algorithm.

Despite enormous progress, fresh ideas are needed in quantum algorithms research. We offer tools which may eventually be implemented on the quantum computers of the future, and hope that that future is not so far away.

BIBLIOGRAPHY

- [1] Scott Aaronson. Quantum computing and hidden variables. *Physical Review A*, 71(3):032325, 2005.
- [2] Scott Aaronson. *How big are quantum states?*, page 200–216. Cambridge University Press, 2013.
- [3] Andrew Adamatzky. A brief history of liquid computers. *Philosophical Transactions of the Royal Society B*, 374(1774):20180372, 2019.
- [4] Dorit Aharonov, Wim Van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Review*, 50(4):755–787, 2008.
- [5] Thomas D Ahle. Sharp and simple bounds for the raw moments of the binomial and poisson distributions. *Statistics & Probability Letters*, 182:109306, 2022.
- [6] Rizwanul Alam, George Siopsis, Rebekah Herrman, James Ostrowski, Phillip C Lotshaw, and Travis S Humble. Solving maxcut with quantum imaginary time evolution. *Quantum Information Processing*, 22(7):281, 2023.
- [7] Alain Aspect, Jean Dalibard, and Gérard Roger. Experimental test of bell’s inequalities using time-varying analyzers. *Physical Review Letters*, 49(25):1804, 1982.
- [8] Brian M Austin, Dmitry Yu Zubarev, and William A Lester Jr. Quantum monte carlo and related approaches. *Chemical Reviews*, 112(1):263–288, 2012.
- [9] Mohsen Bagherimehrab, Yuval R Sanders, Dominic W Berry, Gavin K Brennen, and Barry C Sanders. Nearly optimal quantum algorithm for generating the ground state of a free quantum field theory. *PRX Quantum*, 3(2):020364, 2022.
- [10] Christian W. Bauer, Plato Deliyannis, Marat Freytsis, and Benjamin Nachman. Practical considerations for the preparation of multivariate gaussian states on quantum computers, 2021.
- [11] Max Bee-Lindgren, Zhengrong Qian, Matthew DeCross, Natalie C. Brown, Christopher N. Gilbreth, Jacob Watkins, Xilin Zhang, and Dean Lee. Controlled gate networks applied to eigenvalue estimation, 2024.
- [12] John S Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1(3):195, 1964.
- [13] Daniel Berend and Tamir Tassa. Improved bounds on bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 30(2):185–205, 2010.

- [14] Dominic W Berry, Graeme Ahokas, Richard Cleve, and Barry C Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. *Communications in Mathematical Physics*, 270:359–371, 2007.
- [15] Dominic W Berry, Andrew M Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th annual symposium on foundations of computer science*, pages 792–809. IEEE, 2015.
- [16] Dominic W Berry, Andrew M Childs, Yuan Su, Xin Wang, and Nathan Wiebe. Time-dependent hamiltonian simulation with 11-norm scaling. *Quantum*, 4:254, 2020.
- [17] Alexandre Blais, Arne L Grimsmo, Steven M Girvin, and Andreas Wallraff. Circuit quantum electrodynamics. *Reviews of Modern Physics*, 93(2):025005, 2021.
- [18] S Blanes, F Casas, and J Ros. Extrapolation of symplectic integrators. *Celestial Mechanics and Dynamical Astronomy*, 75:149–161, 1999.
- [19] Colin D Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2), 2019.
- [20] Horacio E Camblong, Luis N Epele, Huner Fanchiotti, and Carlos A Garcia Canal. Quantum anomaly in molecular physics. *Physical Review Letters*, 87(22):220402, 2001.
- [21] Giulio Casati and Luca Molinari. “quantum chaos” with time-periodic hamiltonians. *Progress of Theoretical Physics Supplement*, 98:287–322, 1989.
- [22] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.
- [23] Chi-Fang Chen and Fernando G. S. L. Brandão. Average-case speedup for product formulas. *Communications in Mathematical Physics*, 405(2):32, 2024. doi: 10.1007/s00220-023-04912-5.
- [24] Andrew M Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Information & Computation*, 12(11-12):901–924, 2012.
- [25] Andrew M Childs, Dmitri Maslov, Yunseong Nam, Neil J Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018.
- [26] Andrew M Childs, Aaron Ostrander, and Yuan Su. Faster quantum simulation by randomization. *Quantum*, 3:182, 2019.
- [27] Andrew M Childs, Yuan Su, Minh C Tran, Nathan Wiebe, and Shuchen Zhu. Theory of

- trotter error with commutator scaling. *Physical Review X*, 11(1):011020, 2021.
- [28] Siu A Chin. Multi-product splitting and runge-kutta-nyström integrators. *Celestial Mechanics and Dynamical Astronomy*, 106(4):391–406, 2010.
- [29] Kenneth Choi, Dean Lee, Joey Bonitati, Zhengrong Qian, and Jacob Watkins. Rodeo algorithm for quantum computing. *Physical Review Letters*, 127(4):040505, 2021.
- [30] Juan I Cirac and Peter Zoller. Quantum computations with cold trapped ions. *Physical Review Letters*, 74(20):4091, 1995.
- [31] Louis Comtet. *Advanced Combinatorics: The Art of Finite and Infinite Expansions*. Springer Science & Business Media, 2012.
- [32] B. Jack Copeland. The Church-Turing Thesis. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2020 edition, 2020.
- [33] Christopher J Cramer. *Essentials of Computational Chemistry: Theories and Models*. John Wiley & Sons, 2013.
- [34] Peter Cromwell, Elisabetta Beltrami, and Marta Rampichini. The borromean rings. *Mathematical Intelligencer*, 20(1):53–62, 1998.
- [35] Toby S Cubitt, David Perez-Garcia, and Michael M Wolf. Undecidability of the spectral gap. *Nature*, 528(7581):207–211, 2015.
- [36] André Pierro de Camargo. On the numerical stability of newton’s formula for lagrange interpolation. *Journal of Computational and Applied Mathematics*, 365:112369, 2020. ISSN 0377-0427. doi: <https://doi.org/10.1016/j.cam.2019.112369>.
- [37] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.
- [38] Bryce Seligman Dewitt and Neill Graham. *The Many-Worlds Interpretation of Quantum Mechanics*, volume 63. Princeton University Press, 2015.
- [39] Paul Adrien Maurice Dirac. Quantum mechanics of many-electron systems. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 123(792):714–733, 1929.
- [40] 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(4):040305, 2022.

- [41] Vitaly Efimov. Energy levels arising from resonant two-body forces in a three-body system. *Physics Letters B*, 33(8):563–564, 1970.
- [42] Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 47(10):777, 1935.
- [43] Jens Eisert, Marcus Cramer, and Martin B Plenio. Colloquium: Area laws for the entanglement entropy. *Reviews of Modern Physics*, 82(1):277, 2010.
- [44] Suguru Endo, Qi Zhao, Ying Li, Simon Benjamin, and Xiao Yuan. Mitigating algorithmic errors in a hamiltonian simulation. *Physical Review A*, 99:012334, 1 2019. doi: 10.1103/PhysRevA.99.012334.
- [45] Paul K Faehrmann, Mark Steudtner, Richard Kueng, Maria Kieferova, and Jens Eisert. Randomizing multi-product formulas for hamiltonian simulation. *Quantum*, 6:806, 2022.
- [46] Richard P Feynman. Simulating physics with computers. In *Feynman and Computation*, pages 133–153. CRC Press, 2018.
- [47] Fulvio Flamini, Nicolo Spagnolo, and Fabio Sciarrino. Photonic quantum information processing: a review. *Reports on Progress in Physics*, 82(1):016001, 2018.
- [48] Edward Fredkin and Tommaso Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3):219–253, 1982. doi: 10.1007/BF01857727.
- [49] Stuart J Freedman and John F Clauser. Experimental test of local hidden-variable theories. *Physical Review Letters*, 28(14):938, 1972.
- [50] Nicolai Friis, Oliver Marty, Christine Maier, Cornelius Hempel, Milan Holzäpfel, Petar Jurcevic, Martin B Plenio, Marcus Huber, Christian Roos, Rainer Blatt, et al. Observation of entangled states of a fully controlled 20-qubit system. *Physical Review X*, 8(2):021012, 2018.
- [51] W Gautschi. How (un)stable are vandermonde systems? asymptotic and computational analysis. In *Lecture Notes in Pure and Applied Mathematics*, pages 193–210. Marcel Dekker, Inc, 1990.
- [52] Iulia M Georgescu, Sahel Ashhab, and Franco Nori. Quantum simulation. *Reviews of Modern Physics*, 86(1):153, 2014.
- [53] Elizabeth Gibney. The quantum gold rush. *Nature*, 574(7776):22–24, 2019.
- [54] 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*,

- page 193–204, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367059.
- [55] Sheldon Goldstein. Bohmian Mechanics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2021 edition, 2021.
- [56] Dmitry Grinko, Julien Gacon, Christa Zoufal, and Stefan Woerner. Iterative quantum amplitude estimation. *npj Quantum Information*, 7:52, 2021. doi: 10.1038/s41534-021-00379-1.
- [57] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions, 2002.
- [58] Erik J Gustafson. Projective cooling for the transverse ising model. *Physical Review D*, 101(7):071504, 2020.
- [59] Dipanjali Halder, V Srinivasa Prasanna, Valay Agarawal, and Rahul Maitra. Iterative quantum phase estimation with variationally prepared reference state. *International Journal of Quantum Chemistry*, 123(3):e27021, 2023.
- [60] Jack K. Hale and Hüseyin Koçak. *Scalar Nonautonomous Equations*, pages 107–132. Springer New York, New York, NY, 1991. ISBN 978-1-4612-4426-4. doi: 10.1007/978-1-4612-4426-4_4.
- [61] Brian Hall. *Quantum Theory for Mathematicians*, volume 267 of *Graduate Texts in Mathematics*. Springer-Verlag New York, 2013.
- [62] H-W Hammer and Brian G Swingle. On the limit cycle for the $1/r^2$ potential in momentum space. *Annals of Physics*, 321(2):306–317, 2006.
- [63] Dylan Harley, Ishaun Datta, Frederik Ravn Klausen, Andreas Bluhm, Daniel Stilck França, Albert Werner, and Matthias Christandl. Going beyond gadgets: The importance of scalability for analogue quantum simulators. *arXiv:2306.13739*, 2023.
- [64] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009.
- [65] Loïc Henriët, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. Quantum computing with neutral atoms. *Quantum*, 4:327, 2020.
- [66] Markus Heyl, Philipp Hauke, and Peter Zoller. Quantum localization bounds trotter errors in digital quantum simulation. *Science Advances*, 5(4):eaau8342, 2019. doi: 10.1126/sciadv.aau8342.

- [67] Brendon L Higgins, Dominic W Berry, Stephen D Bartlett, Howard M Wiseman, and Geoff J Pryde. Entanglement-free heisenberg-limited phase estimation. *Nature*, 450(7168):393–396, 2007.
- [68] Nicholas J Higham. The numerical stability of barycentric lagrange interpolation. *IMA Journal of Numerical Analysis*, 24(4):547–556, 2004.
- [69] Mark Howard, Joel Wallman, Victor Veitch, and Joseph Emerson. Contextuality supplies the ‘magic’ for quantum computation. *Nature*, 510(7505):351–355, 2014.
- [70] Jacky Huyghebaert and Hans De Raedt. Product formula methods for time-dependent schrodinger problems. *Journal of Physics A: Mathematical and General*, 23(24):5777, 1990.
- [71] Jason Iaconis, Sonika Johri, and Elton Yechao Zhu. Quantum state preparation of normal distributions using matrix product states. *npj Quantum Information*, 10(1):15, 2024.
- [72] R Islam, Crystal Senko, Wes C Campbell, S Korenblit, J Smith, A Lee, EE Edwards, C-CJ Wang, JK Freericks, and C Monroe. Emergence and frustration of magnetism with variable-range interactions in a quantum simulator. *Science*, 340(6132):583–587, 2013.
- [73] Dollard John Day and Friedman Charles N. *Product Integration with Application to Differential Equations*. Number v. 10. Section, Analysis in Encyclopedia of Mathematics and Its Applications. Cambridge University Press, 1984. ISBN 9780521302302.
- [74] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.
- [75] Mária Kieferová, Artur Scherer, and Dominic W Berry. Simulating the dynamics of time-dependent hamiltonians with a truncated dyson series. *Physical Review A*, 99(4):042314, 2019.
- [76] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout Van Den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, et al. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965):500–505, 2023.
- [77] A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995.
- [78] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, USA, 2002. ISBN 0821832298.
- [79] Alexei Kitaev and William A. Webb. Wavefunction preparation and resampling using a

- quantum computer, 2009.
- [80] Natalie Klco and Martin J. Savage. Minimally entangled state preparation of localized wave functions on quantum computers. *Physical Review A*, 102:012612, 2020. doi: 10.1103/PhysRevA.102.012612.
 - [81] Rolf Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961.
 - [82] Dean Lee, Jacob Watkins, Dillon Frame, Gabriel Given, Rongzheng He, Ning Li, Bing-Nan Lu, and Avik Sarkar. Time fractals and discrete scale invariance with trapped ions. *Physical Review A*, 100(1):011403, 2019.
 - [83] Dean Lee, Joey Bonitati, Gabriel Given, Caleb Hicks, Ning Li, Bing-Nan Lu, Abudit Rai, Avik Sarkar, and Jacob Watkins. Projected cooling algorithm for quantum computation. *Physics Letters B*, 807:135536, 2020.
 - [84] Joonho Lee, Dominic W Berry, Craig Gidney, William J Huggins, Jarrod R McClean, Nathan Wiebe, and Ryan Babbush. Even more efficient quantum computations of chemistry through tensor hypercontraction. *PRX Quantum*, 2(3):030305, 2021.
 - [85] Xiaoyu Li, Guowu Yang, Carlos Manuel Torres Jr, Desheng Zheng, and Kang L Wang. A class of efficient quantum incrementer gates for quantum circuit synthesis. *International Journal of Modern Physics B*, 28(01):1350191, 2014.
 - [86] HQ Lin. Exact diagonalization of quantum-spin models. *Physical Review B*, 42(10):6561, 1990.
 - [87] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.
 - [88] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019.
 - [89] Guang Hao Low and Nathan Wiebe. Hamiltonian simulation in the interaction picture. *arXiv:1805.00675*, 2018.
 - [90] Guang Hao Low, V. Kliuchnikov, and N. Wiebe. Well-conditioned multiproduct hamiltonian simulation. *arXiv: Quantum Physics*, 2019.
 - [91] Shang-keng Ma. Introduction to the renormalization group. *Reviews of Modern Physics*, 45(4):589, 1973.
 - [92] Pesi R Masani. *Norbert Wiener 1894–1964*, volume 5. Birkhäuser, 2012.
 - [93] John C Mason and David C Handscomb. *Chebyshev Polynomials*. CRC press, 2002.

- [94] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C Benjamin, and Xiao Yuan. Quantum computational chemistry. *Reviews of Modern Physics*, 92(1):015003, 2020.
- [95] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812, 2018.
- [96] Richard Meister and Simon C Benjamin. Resource-frugal hamiltonian eigenstate preparation via repeated quantum phase estimation measurements. *arXiv:2212.00846*, 2022.
- [97] Barry Simon Michael Reed. *Methods of Modern Mathematical Physics*, volume 1. Academic Press, 1980.
- [98] Christopher Monroe and Jungsang Kim. Scaling the ion trap quantum processor. *Science*, 339(6124):1164–1169, 2013.
- [99] Chandra Sekhar Mukherjee, Subhamoy Maitra, Vineet Gaurav, and Dibyendu Roy. Preparing dicke states on a quantum computer. *IEEE Transactions on Quantum Engineering*, 1:1–17, 2020.
- [100] Pascal Naidon and Shimpei Endo. Efimov physics: a review. *Reports on Progress in Physics*, 80(5):056001, 2017.
- [101] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.
- [102] Caleb J O’Loan. Iterative phase estimation. *Journal of Physics A: Mathematical and Theoretical*, 43(1):015301, 2009.
- [103] Uri Peskin and Nimrod Moiseyev. The solution of the time-dependent schrödinger equation by the (t, t') method: Theory, computational algorithm and applications. *The Journal of Chemical Physics*, 99(6):4590–4596, 1993.
- [104] Diego Porras and J Ignacio Cirac. Effective quantum spin systems with trapped ions. *Physical Review Letters*, 92(20):207901, 2004.
- [105] David Poulin, Angie Qarry, Rolando Somma, and Frank Verstraete. Quantum simulation of time-dependent hamiltonians and the convenient illusion of hilbert space. *Physical Review Letters*, 106(17):170501, 2011.
- [106] Zhengrong Qian, Jacob Watkins, Gabriel Givern, Joey Bonitati, Kenneth Choi, and Dean Lee. Demonstration of the rodeo algorithm on a quantum computer. *arXiv:2110.07747*, 2021.
- [107] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics*, volume 37. Springer Science & Business Media, 2010.

- [108] Yihui Quek, Daniel Stilck França, Sumeet Khatri, Johannes Jakob Meyer, and Jens Eisert. Exponentially tighter bounds on limitations of quantum error mitigation. *arXiv:2210.11505*, 2022.
- [109] Arthur G. Rattew and Bálint Koczor. Preparing arbitrary continuous functions in quantum registers with logarithmic complexity, 2022.
- [110] Arthur G Rattew, Yue Sun, Pierre Minssen, and Marco Pistoia. The efficient preparation of normal distributions in quantum registers. *Quantum*, 5:609, 2021.
- [111] Robert Raussendorf, Daniel E Browne, and Hans J Briegel. Measurement-based quantum computation on cluster states. *Physical Review A*, 68(2):022312, 2003.
- [112] Markus Reiher, Nathan Wiebe, Krysta M Svore, Dave Wecker, and Matthias Troyer. Elucidating reaction mechanisms on quantum computers. *Proceedings of the National Academy of Sciences*, 114(29):7555–7560, 2017.
- [113] Gumaro Rendon, Jacob Watkins, and Nathan Wiebe. Improved accuracy for trotter simulations using chebyshev interpolation. *Quantum*, 8:1266, 2024.
- [114] T.J. Rivlin. *Chebyshev Polynomials*. Dover Books on Mathematics. Dover Publications, 2020. ISBN 9780486842332. URL <https://books.google.com/books?id=3s0mygEACAAJ>.
- [115] Ankur Sarker, M Shamiul Amin, Avishek Bose, and Nafisah Islam. An optimized design of binary comparator circuit in quantum computing. In *2014 International Conference on Informatics, Electronics & Vision (ICIEV)*, pages 1–5. IEEE, 2014.
- [116] Nicolas PD Sawaya, Tim Menke, Thi Ha Kyaw, Sonika Johri, Alán Aspuru-Guzik, and Gian Giacomo Guerreschi. Resource-efficient digital quantum simulation of d-level systems for photonic, vibrational, and spin-s hamiltonians. *npj Quantum Information*, 6(1):49, 2020.
- [117] M.L. Shetterly and L. Freeman. *Hidden Figures*. HarperCollins, 2018. ISBN 9780062881885.
- [118] 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.
- [119] Avram Sidi. *The Richardson Extrapolation Process*, page 21–41. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003. doi: 10.1017/CBO9780511546815.003.
- [120] Daniel R Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997.

- [121] Ionel Stetcu, Alessandro Baroni, and Joseph Carlson. Projection algorithm for state preparation on quantum computers. *Physical Review C*, 108(3):L031306, 2023.
- [122] Masuo Suzuki. General theory of fractal path integrals with applications to many-body theories and statistical physics. *Journal of Mathematical Physics*, 32(2):400–407, 1991.
- [123] Masuo Suzuki. General theory of higher-order decomposition of exponential operators and symplectic integrators. *Physics Letters A*, 165(5-6):387–395, 1992.
- [124] Masuo Suzuki. Methodology of analytic and computational studies on quantum systems. *Journal of Statistical Physics*, 110(3):945–956, 2003.
- [125] Krysta M. Svore, Matthew B. Hastings, and Michael H. Freedman. Faster phase estimation. *Quantum Information & Computation*, 14(3-4):306–328, 2014. doi: 10.26421/QIC14.3-4-7.
- [126] Lloyd N. Trefethen. *Appendix A. Six Myths of Polynomial Interpolation and Quadrature*, pages 263–271. SIAM, 2019. doi: 10.1137/1.9781611975949.appa. URL <https://epubs.siam.org/doi/abs/10.1137/1.9781611975949.appa>.
- [127] Jean-Philippe Uzan. The fundamental constants and their variation: observational and theoretical status. *Reviews of Modern Physics*, 75(2):403, 2003.
- [128] Dyon van Vreumingen and Kareljan Schoutens. Adiabatic ground-state preparation of fermionic many-body systems from a two-body perspective. *Physical Review A*, 108(6):062603, 2023.
- [129] Farrokh Vatan and Colin Williams. Optimal quantum circuits for general two-qubit gates. *Physical Review A*, 69:032315, 2004. doi: 10.1103/PhysRevA.69.032315.
- [130] Almudena Carrera Vazquez, Daniel J Egger, David Ochsner, and Stefan Woerner. Well-conditioned multi-product formulas for hardware-friendly hamiltonian simulation. *Quantum*, 7:1067, 2023.
- [131] Libor Veis and Jiří Pittner. Adiabatic state preparation study of methylene. *The Journal of Chemical Physics*, 140(21), 2014.
- [132] Marco Vianello and Federico Piazzon. Stability inequalities for lebesgue constants via markov-like inequalities. *Dolomites Research Notes on Approximation*, 11(DRNA Volume 11.1):1–9, 2018.
- [133] Jacob Watkins, Nathan Wiebe, Alessandro Roggero, and Dean Lee. Time-dependent hamiltonian simulation using discrete clock constructions. *arXiv:2203.11353*, 2022.
- [134] James D Watson, Jacob Bringewatt, Alexander F Shaw, Andrew M Childs, Alexey V Gorshkov, and Zohreh Davoudi. Quantum algorithms for simulating nuclear effective field

- theories. *arXiv:2312.05344*, 2023.
- [135] Nathan Wiebe and Nathan S Babcock. Improved error-scaling for adiabatic quantum evolutions. *New Journal of Physics*, 14(1):013024, 2012.
- [136] Nathan Wiebe and Chris Granade. Efficient bayesian phase estimation. *Physical Review Letters*, 117(1):010503, 2016.
- [137] Nathan Wiebe, Dominic Berry, Peter Høyer, and Barry C Sanders. Higher order decompositions of ordered operator exponentials. *Journal of Physics A: Mathematical and Theoretical*, 43(6):065203, 2010. doi: 10.1088/1751-8113/43/6/065203.
- [138] Nathan Wiebe, Dominic W Berry, Peter Høyer, and Barry C Sanders. Simulating quantum dynamics on a quantum computer. *Journal of Physics A: Mathematical and Theoretical*, 44(44):445308, 2011.
- [139] Changhao Yi and Elizabeth Crosson. Spectral analysis of product formulas for quantum simulation. *npj Quantum Information*, 8(1):37, 2022. ISSN 2056-6387. doi: 10.1038/s41534-022-00548-w.
- [140] Jiehang Zhang, Guido Pagano, Paul W Hess, Antonis Kyprianidis, Patrick Becker, Harvey Kaplan, Alexey V Gorshkov, Z-X Gong, and Christopher Monroe. Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator. *Nature*, 551(7682):601–604, 2017.
- [141] Sergiy Zhuk, Niall Robertson, and Sergey Bravyi. Trotter error bounds and dynamic multi-product formulas for hamiltonian simulation, 2023.
- [142] Christa Zoufal, Ryan V Mishmash, Nitin Sharma, Niraj Kumar, Aashish Sheshadri, Amol Deshmukh, Noelle Ibrahim, Julien Gacon, and Stefan Woerner. Variational quantum algorithm for unconstrained black box binary optimization: Application to feature selection. *Quantum*, 7:909, 2023.
- [143] Wojciech Hubert Zurek. Decoherence, einselection, and the quantum origins of the classical. *Reviews of Modern Physics*, 75(3):715, 2003.
- [144] Karol Życzkowski, Paweł Horodecki, Anna Sanpera, and Maciej Lewenstein. Volume of the set of separable states. *Physical Review A*, 58(2):883, 1998.