ABSTRACT

Title of Dissertation: QUANTUM ALGORITHMS FOR NONCONVEX OPTIMIZATION: THEORY AND IMPLEMENTATION

Jiaqi Leng
Doctor of Philosophy, 2024

Dissertation Directed by: Professor Xiaodi Wu
Department of Computer Science

Continuous optimization problems arise in virtually all disciplines of quantitative research. While convex optimization has been well-studied in recent decades, large-scale nonconvex optimization problems remain intractable in both theory and practice. Quantum computers are expected to outperform classical computers in certain challenging computational problems. Some quantum algorithms for convex optimization have shown asymptotic speedups, while the quantum advantage for nonconvex optimization is yet to be fully understood. This thesis focuses on Quantum Hamiltonian Descent (QHD), a quantum algorithm for continuous optimization problems. A systematic study of Quantum Hamiltonian Descent is presented, including theoretical results concerning nonconvex optimization and efficient implementation techniques for quantum computers.

Quantum Hamiltonian Descent is derived as the path integral of classical gradient descent algorithms. Due to the quantum interference of classical descent trajectories, Quantum Hamilto-

nian Descent exhibits drastically different behavior from classical gradient descent, especially for nonconvex problems. Under mild assumptions, we prove that Quantum Hamiltonian Descent can always find the global minimum of an unconstrained optimization problem given a sufficiently long runtime. Moreover, we demonstrate that Quantum Hamiltonian Descent can efficiently solve a family of nonconvex optimization problems with exponentially many local minima, which most commonly used classical optimizers require super-polynomial time to solve. Additionally, by using Quantum Hamiltonian Descent as an algorithmic primitive, we show a quadratic oracular separation between quantum and classical computing.

We consider the implementation of Quantum Hamiltonian Descent for two important paradigms of quantum computing, namely digital (fault-tolerant) and analog quantum computers. Exploiting the product formula for quantum Hamiltonian simulation, we demonstrate that a digital quantum computer can implement Quantum Hamiltonian Descent with gate complexity nearly linear in problem dimension and evolution time. With a hardware-efficient sparse Hamiltonian simulation technique known as Hamiltonian embedding, we develop an analog implementation recipe for Quantum Hamiltonian Descent that addresses a broad class of nonlinear optimization problems, including nonconvex quadratic programming. This analog implementation approach is deployed on large-scale quantum spin-glass simulators, and the empirical results strongly suggest that Quantum Hamiltonian Descent has great potential for highly nonconvex and nonlinear optimization tasks.

QUANTUM ALGORITHMS FOR NONCONVEX OPTIMIZATION: THEORY AND IMPLEMENTATION

by

Jiaqi Leng

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2024

Advisory Committee:
Professor Xiaodi Wu, Chair/Advisor
Professor Andrew M. Childs
Professor Ming Lin
Professor Yi-Kai Liu
Professor Haizhao Yang, Dean's representative

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost, I'd like to thank my advisor, Professor Xiaodi Wu, for giving me an invaluable opportunity to work on challenging and exciting projects over the past years. He has supported me throughout my entire doctoral journey, even during the most challenging moments. It has been a pleasure to work with and learn from such an extraordinary individual.

I would also like to thank Professor Andrew M. Childs, Professor Ming Lin, Professor Yi-Kai Liu, and Professor Haizhao Yang for agreeing to serve on my thesis committee and sparing their invaluable time reviewing the manuscript. In particular, I would like to extend my special thanks to Professor Ming Lin, who graciously agreed to join my dissertation defense as a voting committee member at the last minute due to an unexpected situation.

At the University of Maryland, I had the privilege of learning and discussing with many fantastic faculty members, including Gorjan Alagic, Abhinav Bhatele, Maria Cameron, Sandra Cerrai, Mohammad Hafezi, Furong Huang, Lise-Marie Imbert-Gerard, Dio Margetis, Carl Miller, and Rodrigo Trevino. I would also like to thank Jessica Sadler, the AMSC program manager. She was always there to offer help when I needed assistance.

My colleagues at the Joint Center for Quantum Information and Computer Science (QuICS) have enriched my graduate life in many ways and deserve a special mention. Tongyang Li played

a crucial role in initiating and completing my first academic paper on quantum computing. I had numerous engaging discussions, both academic and non-academic, with Yuxiang Peng. Ethan Hickman and Joseph Li have been outstanding collaborators from the early stages of the Quantum Hamiltonian Descent project. My interaction with Yingkang Cao, Shouvanik Chakrabarti, Connor Clayton, Haowei Deng, Yi Lee, Jin-Peng Liu, Suying Liu, Haohai Shi, Daochen Wang, Yuxin Wang, Xuchen You, Jacob Young, Yufan Zheng, and Shaopeng Zhu, has been very fruitful. I would also like to thank Andrea Svejda, the QuICS coordinator. Whenever I was looking for assistance, she was always there to offer help.

I appreciate the discussion with Dong An, Brandon Augustino, Lei Fan, Aram Harrow, Wuchen Li, Lin Lin, Yizhou Liu, Mohammadhossein Mohammadisiahroudi, Giacomo Nannicini, Stanley Osher, Tamás Terlaky, Lexing Ying, Yu Tong, Chunhao Wang, Chenyi Zhang, and Chaoyue Zhao.

I am profoundly grateful to my family—my parents and siblings—who have consistently supported and guided me throughout my career and have helped me overcome seemingly insurmountable challenges at times. Words fall short of expressing the immense gratitude I feel towards them.

My housemates at my place of residence have been a crucial factor in my finishing smoothly. I'd like to express my gratitude to Yiling Qiao and Hsien-Yu Meng for their friendship and support.

Finally, I must acknowledge that it is impossible to remember every individual who has contributed to my journey. I sincerely apologize to those I may have inadvertently omitted from this acknowledgment. Your support and contributions have not gone unnoticed, and I am grateful for your part in this journey.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Continuous optimization

Continuous optimization concerns finding the minimum of a real-valued function with one or several continuous variables, subject to constraints. It can be formulated as the following mathematical problem,

$$\min_{x \in S} f(x), \quad S \subset \mathbb{R}^n,$$

where the function $f(x)$ is known as the *objective function*, and the subset S is called the *feasible set* of the problem. This problem lies at the core of modern computational sciences, ranging from training large machine learning models to managing investment portfolios.

When the objective function and the feasible set are both convex, the problem is referred to as a **convex optimization** problem. The simplest case is to optimize a convex differentiable function over the entire Euclidean space, a task known as unconstrained optimization. This problem can be solved by the standard gradient descent algorithm in polynomial time [1]. Another notable example is linear programming (LP, or linear optimization), in which the objective function is linear, and the feasible set is defined by a set of linear inequalities. In 1984, Karmarkar [2] proposed the first provably efficient algorithm that solves linear programming in polynomial time.

In the past decades, the theoretical foundation of convex optimization has been well established and many effective computer software programs (i.e., solvers) have been created to facilitate the solution of convex problems in practice.

When either the objective function or the feasible set is not convex, the optimization problem falls into the category of **nonconvex optimization**. Nonconvex optimization problems usually admit more than one local solution or even disconnected feasible sets, which pose significant challenges for known optimization algorithms and computer solvers. In general, nonconvex optimization problems are known to be NP-hard.

There are usually two strategies for solving nonconvex optimization problems. The first strategy is *branch and bound*, which decomposes the original problem into several sub-problems, and each sub-problem is solved to an optimal level; sub-optimal solutions can be eliminated by comparing the solutions obtained from the sub-problems. The second strategy is *local search*, which leverages first- and/or second-order information (i.e., gradient and/or Hessian) of the objective function to produce an iterative procedure that converges to a local stationary solution. While in theory, a branch-and-bound strategy can find a global optimization solution through brutal-force search, the runtime could be exponentially long and therefore not ideal for large-scale applications. Local search algorithms enjoy faster convergence for large and regular problems; however, the global optimality of the obtained solution is not guaranteed. In practice, finding approximately global optimal solutions for highly nonconvex problems is considered intractable when the problem dimension exceeds several hundred.

Between the realms of convexity and extreme nonconvexity (NP-hardness) lies a complex territory, where mathematical understanding and computational methods are still in their infancy and largely uncharted. While certain nonconvex problems are genuinely difficult to solve for

classical computers, emerging computing paradigms based on non-traditional computer architecture could pave an unprecedented path toward nonconvex optimization. Quantum computing, a technology that exploits quantum physics for information processing, is regarded as one of the most competitive candidates to address the intractable computational challenges faced by humanity. In this regard, it is more than natural to extend the classical study of nonconvex optimization to the quantum context.

1.2 Quantum algorithms for continuous optimization

Quantum algorithms run on quantum computers. The concept of a quantum computer was initially proposed by Manin [3] and Feynman [4] over four decades ago. They believed that to understand quantum physics, we need a machine that mimics the quantum-mechanical systems found in nature. In 1999, Peter Shor [5] discovered a quantum algorithm (now known as the *Shor's algorithm*) that can factorize a large integer into prime factors exponentially faster than any known classical algorithms. This finding was a historical milestone and has significantly motivated the investigation of quantum speedups for other classes of computational tasks that do not stem from quantum physics.

As a fundamental problem class in mathematical sciences, optimization has long been a target for quantum speedups. While quantum algorithms for combinatorial optimization problems have been extensively studied since 2000 [6, 7, 8, 9, 10], the exploration of quantum algorithms for continuous optimization problems is a more recent development.

Convex optimization. Semidefinite programming (SDP) is arguably one of the most important problems in convex optimization, which minimizes a linear objective function over the inter-

section of the positive semidefinite cone and a hyperplane. Existing quantum algorithms for SDP can be roughly categorized by their underlying solution methodology. The first category is the so-called first-order methods, which are based on the multiplicative weights update framework [11, 12, 13, 14]. The second category is based on second-order methods (more precisely, interior point methods), which utilize quantum linear system solvers (QLSS) to improve the overall time complexity of classical interior point methods [15, 16, 17]. Many of the quantum solutions to SDP can be readily applied to the simpler class of linear programming (LP), and often lead to better run times [18, 19, 20, 21].

Note that, nevertheless, the time complexity of the aforementioned quantum algorithms heavily depends on the input model and problem structure. Also, most of these quantum algorithms assume access to a quantum data structure named Quantum Random Access Memory (QRAM), for which the scalable fault-tolerant implementation is yet to be understood [22, 23, 24, 25]. One exception is a recent work based on a quantum central path method framework [26], which only assumes a classical binary description of the problem data but still achieves asymptotic speedup in certain parameter regimes.

Nonconvex optimization. It is unlikely that a polynomial-time quantum algorithm exists for general nonconvex optimization problems, otherwise, it would imply that NP is contained in BQP (which is the quantum analog of the complexity class P). Zhang and Li proved a quantum query lower bound for general nonconvex optimization problems [27]. In practice, gradient descent is the workhorse for differentiable nonconvex optimization problems. Rebentrost et al. [28] formulated a quantum gradient descent algorithm that achieves $\mathrm{poly}\log(n)$ run time for an n-dimensional optimization problem; however, except for some special cases [15, 29], the run time

of the quantum gradient descent algorithm scales exponentially in terms of the number of iterations, rendering this method less interesting in practice. There seem to be some fundamental difficulties in directly generalizing classical optimization algorithms to quantum. Many existing classical first- and second-order algorithms (e.g., gradient descent, Newton's method) can be interpreted as highly nonlinear dynamical systems [30, 31, 32]. Quantum computers, nevertheless, may not be the most ideal computing device to simulate genuinely nonlinear dynamics [33, 34].

1.3 Quantum Hamiltonian Descent: solving continuous optimization via quantum dynamics

As we have seen, many classical optimization algorithms are formulated as iterative processes. To accelerate these algorithms using a quantum computer, prior attempts usually rely on replacing a key component (e.g., gradient update, solving Newton systems, etc.) in an iteration step with a faster quantum subroutine [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 28]. By carefully balancing the potential speedup and possible overheads, it is possible to achieve asymptotic quantum speedups. However, two major drawbacks significantly limit the practical advantage of this approach:

1. Replacing a classical subroutine with a quantum one does not change the solution quality, as the overall solution path in an iterative algorithm is not affected. Therefore, it is unlikely to observe any quantum speedup if the overhead introduced by the quantum subroutine, possibly due to error correction code and advanced input models, outweighs its potential benefit.

2. Commonly used quantum subroutines for optimization problems, such as quantum phase

estimation [35], quantum linear system solver [36, 37], and quantum singular value transformation [38], require large fault-tolerant quantum computers. This means quantum optimization algorithms based on these subroutines can not be deployed and evaluated with the progress of near-term quantum technology.

In this thesis, we consider an alternative approach to speed up classical gradient descent. Instead of using quantum subroutines to accelerate individual iteration steps, we *quantize* gradient descent as a whole. Many gradient methods have well-defined continuous-time limits as differential equations. For example, the standard gradient descent can be regarded as (time-discretized) gradient flow, and the so-called accelerated gradient descent can be modeled by continuous-time evolution generated by (classical) Lagrangian mechanics [31]. We then *quantize* these classical dynamical systems and lift them to quantum dynamical systems, which are referred to as **Quantum Hamiltonian Descent**. By simulating these quantum dynamics, we effectively implement a quantum algorithm for optimization problems.

Compared with the conventional approach to designing quantum optimization algorithms, our new approach is conceptually novel and has some unique advantages for practical implementation. First, Quantum Hamiltonian Descent is described by quantum Hamiltonians, which can be readily simulated using quantum computers. We develop a technique known as Hamiltonian embedding, which even allows us to simulate the dynamics in a near-term analog quantum computer. Second, Quantum Hamiltonian Descent exhibits the quantum tunneling effect, which means the dynamics can escape from sub-optimal local minima and locate the global minimum of the optimization problem. In other words, the solution quality is improved by the quantum computer, a goal that is not achieved by prior art. In summary, Quantum Hamiltonian Descent produces

solutions with higher quality, while not necessarily requiring sophisticated gate-based implementation or even fault-tolerant quantum computers. These desired features make it a promising candidate for solving real-world nonconvex optimization problems.

It is worth mentioning that Quantum Hamiltonian Descent itself can be used as a subroutine to accelerate or improve many well-known algorithms in the literature of classical optimization theory. For example, we can integrate Quantum Hamiltonian Descent into a branch-and-bound strategy to reduce the total runtime. Alternatively, we can use Quantum Hamiltonian Descent as a more effective solver to find better solutions in an augmented Lagrangian framework. Quantum Hamiltonian Descent has the potential to be an important algorithm that expands the traditional optimization toolbox and influences the future design of optimization infrastructure, such as software toolchains and benchmarking.

1.4 Overview

Theory of Quantum Hamiltonian Descent. Accelerated gradient descent algorithms, a prominent variant of the standard gradient descent, exploit the momentum information to accelerate the convergence to (locally) optimal solutions. It has been shown that many accelerated gradient descent algorithms can be interpreted as certain Lagrangian-mechanical flows, in which the dynamics are generated by some time-dependent Lagrangian functions [30, 31]. However, these Lagrangian dynamics are described by nonlinear ordinary differential equations that can not be sped up by applying existing quantum subroutines [34]. In Chapter 2, we employ Feynman's well-known path integral formulation of quantum mechanics [39] to *lift* the classical Lagrangian-mechanical systems describing accelerated gradient descent to quantum dynamics. The resulting

"quantum accelerated gradient descent" is governed by a linear Schrödinger equation with a time-dependent Hamiltonian, which we named Quantum Hamiltonian Descent (QHD). Through a Lyapunov function approach, we prove that Quantum Hamiltonian Descent has a similar convergence rate as its classical counterpart, while numerical results suggest Quantum Hamiltonian Descent is more robust in terms of time discretization. Moreover, compared with classical accelerated gradient descent, Quantum Hamiltonian Descent exhibits the quantum tunneling effect in the presence of high-energy barriers on the optimization landscape, making it advantageous for general nonconvex optimization problems. A comprehensive numerical study shows that Quantum Hamiltonian Descent outperforms other classical and quantum optimization methods (including quantum adiabatic algorithm).

In Chapter 3, we further investigate the mathematics of Quantum Hamiltonian Descent. In particular, we are concerned with the asymptotic and non-asymptotic behavior of Quantum Hamiltonian Descent when applied to nonconvex optimization problems. First, we show that, under mild conditions, Quantum Hamiltonian Descent can also find the global solution to a nonconvex problem given a sufficiently long evolution time (Section 3.2). This asymptotic convergence result is based on the semi-classical theory of Schrödinger operators and a standard adiabatic approximation technique from quantum physics. Second, we identify a family of structured search problems, each containing N items but only one of them is marked. This problem has a classical query lower bound $\Omega(N)$. We find these problems admit a reformulation as nonconvex optimization problems, for which Quantum Hamiltonian Descent can find the global solution using $\tilde{O}(\sqrt{N})$ queries to the corresponding quantum oracle (Section 3.3). This result reproduces a quadratic (i.e., Grover-type) oracular separation between quantum and classical computation, indicating that Quantum Hamiltonian Descent is a genuinely quantum algorithm that can not be

trivially de-quantized. Third, in Section 3.4, we prove that Quantum Hamiltonian Descent can efficiently solve a family of nonconvex optimization instances, each characterized by 2^d local minima, in $\tilde{O}(d^3)$ quantum queries to the function value and $\tilde{O}(d^4)$ elementary quantum gates. On the other side, we test a comprehensive collection of classical optimization algorithms and solvers with the same family of problem instances. The run time required by these classical optimizers turns out to scale super-polynomially in the parameter d. This result constitutes new evidence of a significant performance separation between quantum and classical computation in nonconvex optimization.

Implementation of Quantum Hamiltonian Descent. Quantum computers can efficiently simulate quantum evolution. Since Quantum Hamiltonian Descent is formulated as real-space quantum dynamics described by a time-dependent Schrödinger equation, it can be efficiently implemented on quantum computers. In this thesis, we consider the implementation of Quantum Hamiltonian Descent for two types of quantum computers, namely, gated-based digital (fault-tolerant) quantum computers and analog quantum computers. In Chapter 4, we introduce a digital implementation of Quantum Hamiltonian Descent to solve unconstrained continuous optimization problems, based on the product formula (i.e., operator splitting method). Both the rigorous gate (time) complexity and the numerical performance of this implementation technique are presented.

Analog quantum computing is another prominent paradigm of quantum computation, which tackles computational tasks by emulating physical quantum systems (i.e., analog quantum simulation). Unlike gate-based (digital) quantum computers, analog quantum computers are usually made for specific-purpose computational tasks (e.g., quantum simulation, quantum annealing,

etc.) but are easier to control and scale. In Chapter 5, we develop a technique named *Hamiltonian embedding*, which allows hardware-efficient simulation of sparse quantum Hamiltonian. Using this technique, we can simulate some high-dimensional quantum evolution without involving advanced quantum input models such as sparse-input oracles or block-encodings. Based on this technique, we explore an implementation of Quantum Hamiltonian Descent by leveraging Ising-type analog quantum computers in Chapter 6. We show that every box-constrained quadratic programming problem can be mapped to an Ising Hamiltonian, and the quantum evolution generated by this Ising Hamiltonian is equivalent to a Quantum Hamiltonian Descent. With this analog implementation, we deploy Quantum Hamiltonian Descent to solve up to 75-dimensional nonconvex quadratic programming on a quantum spin glass simulator with over 5000 qubits. In our experiment, Quantum Hamiltonian Descent outperforms a selection of state-of-the-art classical solvers and the quantum adiabatic algorithm.

Software design based on Quantum Hamiltonian Descent. In Chapter 7, we build an open-source, end-to-end software library named `QHDOPT` to facilitate the implementation of Quantum Hamiltonian Descent on real machines, including gate-based quantum computers and analog quantum computers. `QHDOPT` is developed following the Hamiltonian-Oriented Programming (HOP) paradigm, in which the quantum Hamiltonian serves as the central object in the quantum stack. `QHDOPT` eliminates the technical barrier to implementing quantum algorithms by offering an accessible interface and automatically maps tasks to various supported quantum backends. These features enable users, even those without prior knowledge or experience in quantum computing, to utilize the power of existing quantum devices for large-scale nonlinear and nonconvex optimization tasks. This software is available at `https://github.com/jiaqileng/`

`QHDOPT`.

Acknowledgment. This thesis is based on the following work:

[1] [40]: Jiaqi Leng, Ethan Hickman, Joseph Li, Xiaodi Wu. *Quantum Hamiltonian Descent*. 2023.

[2] [41]: Jiaqi Leng, Yufan Zheng, Xiaodi Wu. *A quantum-classical performance separation in nonconvex optimization*. 2023.

[3] [42]: Jiaqi Leng, Joseph Li, Yuxiang Peng, Xiaodi Wu. *Expanding hardware-efficiently manipulable Hilbert space via Hamiltonian embedding*. 2024.

[4] [43]: Samuel Kushnir, Jiaqi Leng, Yuxiang Peng, Lei Fan, Xiaodi Wu. *QHDOPT: A software for nonlinear optimization with Quantum Hamiltonian Decent*. 2024. Under review.

Chapter 2 is based on [1]. Chapter 3 is based on [1] and [2]. Chapter 4 is based on [1]. Chapter 5 is based on [3]. Chapter 6 is based on [1], [3], and [4]. Chapter 7 is based on [4].

1.5 Preliminaries

Set-theoretic notations. $\mathbb{N}$ denotes the set of positive integers, $\mathbb{Z}$ denotes the set of integers, $\mathbb{R}$ denotes the set of real numbers, and $\mathbb{C}$ denotes the set of complex numbers. $\mathbb{R}_+$ denotes the set of all non-negative real numbers. For an integer $n \in \mathbb{N}$, $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. Given two sets A and B, $A \times B$ denotes the Cartesian product of the two sets,

i.e., $A \times B \coloneqq \{(a, b) \colon a \in A, b \in B\}$; $A \backslash B$ denotes the difference between the two sets, i.e., $A \backslash B \coloneqq \{x \colon x \in A, x \notin B\}$.

Asymptotic notations. Let $f, g \colon \mathbb{R}_+ \to \mathbb{R}_+$. We write $f = O(g)$ if there exists a constant $c > 0$ and an $x_0 \in \mathbb{R}_+$ such that $f(x) \leq cg(x)$ for all $x \geq x_0$. We write $f = \tilde{O}(g)$ if there exists a polynomial $p \colon \mathbb{R} \to \mathbb{R}$ and an $x_0 \in \mathbb{R}_+$ such that $f(x) \leq g(x) \cdot p\left(\log(g(x))\right)$ for all $x \geq x_0$. We write $f = \Omega(g)$ if there exists a constant $c > 0$ and an $x_0 \in \mathbb{R}_+$ such that $f(x) \geq cg(x)$ for all $x \geq x_0$. Note that $f = O(g)$ is equivalent to $g = \Omega(f)$. We write $f = \Theta(g)$ if $f = O(g)$ and $f = \Omega(g)$.

Norms. For a vector $v \in \mathbb{R}^n$, we denote the vector ℓ_p norm as $\|a\|_p \coloneqq (\sum_{k=1}^n |a_k|^p)^{1/k}$ for $1 \leq p < \infty$. The vector ℓ_∞ norm is defined by $\|a\|_\infty \coloneqq \max_k |a_k|$. For a matrix $A \in \mathbb{R}^{n \times n}$, we denote the operator p-norm induced by the vector ℓ_p norm, i.e.,

$$\|A\|_p \coloneqq \sum_{v \neq 0} \frac{\|Av\|_p}{\|v\|_p}.$$

For a Lesbesgue integrable real function $f(x) \colon \mathbb{R}^n \to \mathbb{R}$, we denote the L^p norm ($1 \leq p < \infty$) of f as

$$\|f\|_p \coloneqq \left(\int_{\mathbb{R}^n} |f(x)|^p \right)^{1/p} \mathrm{d}x.$$

The L^∞ norm of f is denoted as

$$\|f\|_\infty \coloneqq \sup_{x \in \mathbb{R}^n} |f(x)|.$$

In this thesis, when no subscript is used in the vector/matrix/function norm notation, we mean $\|\cdot\| = \|\cdot\|_2$ by default.

Quantum information. An N-dimension quantum state is a vector $v \in \mathbb{C}^N$ with $\|v\|_2 = 1$. The vector v is written in Dirac notation as $|v\rangle$ and called a "ket". The complex conjugate transpose of $|v\rangle$ is written as $\langle v|$ and called a "bra", i.e., $\langle v| \coloneqq v^\dagger$. The computational basis of $\mathbb{C}^N$ is the set of vectors $\{b^{(1)}, \ldots, b^{(N)}\}$, where $b^{(i)} = (0, \ldots, 1, \ldots, 0)$ is the all-zero vector except with a 1 in the i-th coordinate. By convention, the i-th computational basis $b^{(i)}$ is written as $|i\rangle$ without further definition. A quantum state in $\mathbb{C}^N$ can be written as $|v\rangle = \sum_{i=1}^N v_i |i\rangle$ and $\langle v| = \sum_{i=1}^N \bar{v}_i \langle i|$. The inner product between two vectors v and w can be written as $\langle v|w\rangle \coloneqq v^\dagger w \in \mathbb{C}$. The outer product between two vectors w and v can be written as $|w\rangle \langle v| = wv^\dagger \in \mathbb{C}^{N \times N}$. The tensor product of quantum states refers to their Kronecker product: if $|v\rangle \in \mathbb{C}^{N_1}$ and $\langle w| \in \mathbb{C}^{N_2}$, then $|v\rangle |w\rangle \coloneqq |v\rangle \otimes |w\rangle \in \mathbb{C}^{N_1} \otimes \mathbb{C}^{N_2}$.

A *qubit* (or quantum-bit) is a two-dimensional complex-valued vector with the unit norm, i.e., $|v\rangle = (v_1, v_2)^\top \in \mathbb{C}^2$ and $\|v\|_2 = 1$. We can write $|v\rangle = v_1 |0\rangle + v_2 |1\rangle$. Pauli operators are three linear (and unitary) operators acting on a single qubit, defined by

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

Chapter 2: Quantum Hamiltonian Descent

Chapter summary. Gradient descent is a fundamental algorithm in both theory and practice for continuous optimization. Identifying its quantum counterpart would be appealing to both theoretical and practical quantum applications. A conventional approach to quantum speedups in optimization relies on the quantum acceleration of intermediate steps of classical algorithms while keeping the overall algorithmic trajectory and solution quality unchanged. We propose Quantum Hamiltonian Descent (QHD), which is derived from the path integral of dynamical systems referring to the continuous-time limit of classical gradient descent algorithms, as a truly quantum counterpart of classical gradient methods.

Leveraging a Lyapunov function approach, we prove that Quantum Hamiltonian Descent achieves an $\mathcal{O}(t^{-2})$ convergence rate for convex optimization, the same as its classical counterparts. However, for nonconvex optimization problems, Quantum Hamiltonian Descent exhibits completely different behaviors compared to classical gradient descent. This is because Quantum Hamiltonian Descent can be interpreted as the path integral of classical trajectories in the search space, where the contribution from classically prohibited trajectories can significantly boost QHD's performance for non-convex optimization. In this chapter, we present numerical evidence of Quantum Hamiltonian Descent's potent performance in nonconvex optimization, deferring the theoretical analysis to the next chapter.

2.1 Introduction

Continuous optimization, stemming from the mathematical modeling of real-world systems, is ubiquitous in applied mathematics, operations research, and computer science [1, 44, 45, 46, 47]. In this chapter, we consider the unconstrained continuous optimization problem,

$$\min_{x \in \mathbb{R}^d} f(x), \tag{2.1}$$

where $f(x)$ is assumed to be continuously differentiable but not necessarily convex. Such problems often come with high dimensionality and non-convexity, posing great challenges for the design and implementation of optimization algorithms [48, 49]. Continuous optimization has been an active research field in past decades, and it is a natural question to investigate potential quantum speedups for continuous optimization problems.

Gradient descent (and its variant) is arguably the most fundamental optimization algorithm in continuous optimization, both in theory and in practice, due to its simplicity and efficiency in converging to critical points. However, many real-world problems have spurious local optima [50], for which gradient descent is subject to slow convergence since it only leverages first-order information [51]. On the other hand, quantum algorithms have the potential to escape from local minima and find near-optimal solutions by leveraging the *quantum tunneling* effect [52, 53]. Therefore, it is desirable to identify a quantum counterpart of gradient descent that is simple and efficient on quantum computers while leveraging the quantum tunneling effect to escape from spurious local optima.

A conventional approach to creating new quantum algorithms involves replacing compo-

nents in a classical algorithm with corresponding quantum subroutines, while carefully balancing the potential speedup and possible overheads. Several proposals for quantum optimization along this line have been made, including [11, 14, 13, 54, 55, 56, 57]. Rebentrost et al. [28] attempted to accelerate gradient descent by computing gradient update steps using quantum circuits. However, these proposals often achieve only moderate quantum speedups, if any at all. More importantly, they rarely improve the quality of solutions because they essentially follow the same solution trajectories in the original classical algorithms. It appears that we need a completely new idea to make a genuine and effective quantum algorithm for continuous optimization.

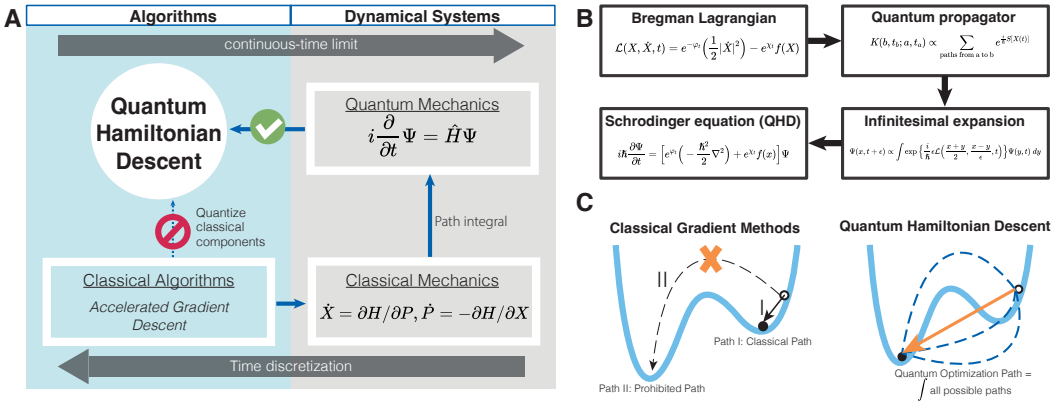


Figure 2.1: **Schematic of Quantum Hamiltonian Descent (QHD). A.** Comparing our approach to quantizing classical gradient descent (GD) with the conventional one. **B.** Four major technical steps in the derivation of QHD. **C.** An illustrative example where classical GD with bad initialization will be trapped in a local minimum, while QHD can easily escape and find near-optimal solutions by taking the path integral of trajectories prohibited by classical mechanics.

In this chapter, we propose a novel approach to quantizing gradient descent that is radically different from any existing proposals. Our main observation is a perhaps unintuitive connection between gradient descent and dynamical systems satisfying classical physical laws. Precisely, it is known that the continuous-time limit of many gradient-based algorithms can be understood as classical physical dynamical systems, e.g., Wibisono et al. [31] models accelerated gradient descent algorithms using a Lagrangian-mechanical framework. Conversely, variants of

gradient-based algorithms could be emerged through the time discretization of these continuous-time dynamical systems [58]. This two-way correspondence inspired us a second approach to quantization: instead of quantizing a specific subroutine in gradient descent, we can quantize the continuous-time limit of gradient descent as a whole, and the resulting quantum dynamical systems give rise to quantum optimization algorithms, as illustrated in Figure 2.1A. In Section 2.3, we derive the quantized gradient descent using the path integral formulation of quantum mechanics. The resulting dynamical system is a quantum evolution described by a Schrödinger equation. This Schrödinger dynamics generates a family of quantum gradient descent algorithms that we will refer to as **Quantum Hamiltonian Descent**, or simply **QHD**.

As desired, Quantum Hamiltonian Descent inherits simplicity and efficiency from classical gradient descent algorithms. Since Quantum Hamiltonian Descent is formulated as the evolution of a quantum system, it can be either implemented with a digital quantum computer using standard Hamiltonian simulation techniques, or with analog quantum simulators. For convex problems, Quantum Hamiltonian Descent is proven to find the unique global solution at a convergence rate similar to its classical counterpart, as detailed in Section 2.3. The true power of Quantum Hamiltonian Descent is revealed in nonconvex optimization problems. The dynamics described by Quantum Hamiltonian Descent can be regarded as the *path integral* of all solution trajectories, including those prohibited in classical gradient descent. Interference among all these trajectories gives rise to a unique quantum phenomenon called *quantum tunneling*, which assists Quantum Hamiltonian Descent in overcoming high-energy barriers and locating the global minimum within complex optimization landscapes. In Section 2.5, we demonstrate the behavior of Quantum Hamiltonian Descent for nonconvex optimization through numerical simulations. A rigorous theoretical analysis is deferred to the next chapter.

2.2 Preliminaries: the Bregman-Lagrangian framework for acceleration

The Bregman-Lagrangian framework is a variational formulation for accelerated gradient methods proposed by Wibisono, Wilson and Jordan [31]. This framework is formulated using Lagrangian mechanics. By defining the Lagrangian function

$$\mathcal{L}(X, \dot{X}, t) = e^{\alpha_t + \gamma_t} \left(\underbrace{\frac{1}{2} |e^{-\alpha_t} \dot{X}|^2}_{\text{kinetic energy}} - \underbrace{e^{\beta_t} f(X)}_{\text{potential energy}} \right), \tag{2.2}$$

where $t \geq 0$ is the time, $X \in \mathbb{R}^d$ is the position, $\dot{X} \in \mathbb{R}^d$ is the velocity, and $\alpha_t, \beta_t, \gamma_t$ are arbitrary smooth functions that control the damping of energy in the system.

Lagrangian mechanics is formulated by variational principle. For a trajectory of motion $X(t) \colon [0, T] \to \mathbb{R}^d$, we define a functional S as the *action* of this trajectory

$$S[X(t)] = \int_0^T \mathcal{L}(X, \dot{X}, t) \, \mathrm{d}t, \tag{2.3}$$

where $\mathcal{L}$ is the Lagrangian of the system. A *physical* path $X(t)$ from a to b is the curve $X(t)$ such that $X(0) = a$, $X(T) = b$, and it minimizes the action functional $S[X(t)]$. By calculus of variations, a least-action curve necessarily solves the Euler-Lagrange equation

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial \mathcal{L}}{\partial \dot{X}} \right) - \frac{\partial \mathcal{L}}{\partial X} = 0. \tag{2.4}$$

With the Lagrangian function (2.2), the resulting Euler-Lagrange equation is a second-order differential eqaution,

$$\ddot{X} + (\dot{\gamma}_t - \dot{\alpha}_t)\dot{X} + e^{2\alpha_t + \beta_t}\nabla f(X) = 0. \tag{2.5}$$

The convergence of the Bregman-Lagrangian framework for continuously differentiable convex functions is established by constructing a Lyapunov function. Suppose the global minimizer of f is x^*, we define the following function $\mathcal{E}_t$:

$$\mathcal{E}_t = \frac{1}{2}\left|e^{-\gamma_t}\dot{X}_t + X_t - x^*\right|^2 + e^{\beta_t}\left(f(X_t) - f(x^*)\right). \tag{2.6}$$

When f is convex, it is shown that $\mathcal{E}_t$ is a Lyapunov function of (2.5), i.e., non-increasing along the solution trajectory X_t. The monotonicity of $\mathcal{E}_t$ leads to the convergence result [31, Theorem 2.1],

$$f(X(t)) - f(x^*) \leq O(e^{-\beta_t}). \tag{2.7}$$

At first glance, (2.7) says that any desired convergence rate can be achieved by choosing different β_t. This is true for continuous-time flows, as different time-dependent parameters $\alpha_t, \beta_t, \gamma_t$ result in the same path in spacetime while converging at different speeds However, it does not imply that gradient-based methods generated by this dynamics can achieve any arbitrary convergence rate, which contradicts the fact that gradient-based methods can not converge faster than $O(t^{-2})$ in the worst case [59]. The devil is in time discretization: when translating the ODE model to practical gradient-based algorithms, we have to discretize the continuous time into dis-

crete steps. The authors devise a family of gradient-based algorithms via an *ad-hoc* discretization of (2.5), and these algorithms fail to convergence when $e^{-\beta_t}$ decays too fast. This matches our intuition that gradient-based optimization is bounded in convergence rate.

It is worth noting that the variational framework in [31] can be reformulated via Hamiltonian mechanics. The Lagrangian system is equivalently described by its Hamiltonian, the Legendre conjugate of the Lagrangian. In particular, the Hamiltonian corresponding to (2.2) takes the formal

$$\mathcal{H}(X, P, t) = e^{\alpha_t + \gamma_t} \left(\underbrace{\frac{1}{2}|e^{-\gamma_t} P|^2}_{\text{kinetic energy}} + \underbrace{e^{\beta_t} f(X)}_{\text{potential energy}} \right), \tag{2.8}$$

where X is the position, and P is the momentum. This Hamiltonian function has the form of the sum of the kinetic and potential energy. The dynamics of the Hamiltonian system is then given by the Hamilton's equations,

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \frac{\partial \mathcal{H}}{\partial P} = e^{\alpha_t - \gamma_t} P, \tag{2.9a}$$

$$\frac{\mathrm{d}P}{\mathrm{d}t} = -\frac{\partial \mathcal{H}}{\partial X} = -e^{\alpha_t + \beta_t + \gamma_t} \nabla f(X). \tag{2.9b}$$

One can show this system of ODEs is identical to the Euler-Lagrange equation (2.5).

Besides the variational formulation of accelerated gradient descent in [31], Maddison et. al. [60] propose a family of gradient-based methods via discretizations of conformal Hamiltonian dynamics, known as *Hamiltonian descent methods*. This framework assumes an additional

access to a kinetic energy k that incorporates information about f. The resulting continuous-time trajectories achieve linear convergence on convex functions: $f(x_t) - f(x^*) \leq O(e^{-ct})$, where c depends on k and f. They consider one implicit and two explicit discretization schemes in order to obtain gradient-based optimization algorithms. These algorithms exhibit similar convergence rates as the continuous-time flows.

2.3 Derivation: path integrals of gradient descent

Now, we quantize the Lagrangian formulation of accelerated gradient methods by introducing the path integral formulation of quantum mechanics. For simplicity, the derivation works with a one-dimensional objective function $f\colon \mathbb{R} \to \mathbb{R}$, while the generalization to higher dimensions is trivial.

In classical mechanics, only the curves that solve the Euler-Lagrange equation are of interests because they are predicted by the variational principle. All other curves are considered "unphysical" because they are not stationary points of the action function $S[X(t)]$.

For quantum mechanics, however, Feynman postulates that not only the "physical" trajectory but all trajectories contribute to quantum evolution. These trajectories contribute equal magnitudes but different phases to the total amplitude. More precisely, the probability to go from a point a at time t_a to the point b at time t_b is $P(b, a) = |K(b, t_b; a, t_a)|^2$, where the amplitude function $K(b, t_b; a, t_a)$ is the sum of contributions of all paths from a to b:

$$K(b, t_b; a, t_a) \propto \sum_{\text{paths from } a \text{ to } b} e^{(i/\hbar)S[X(t)]}. \tag{2.10}$$

Here, i is the imaginary unit and $\hbar$ is the Planck constant. The amplitude function $K(b, t_b; a, t_a)$ is also known as the *propagator* of the quantum dynamics because it can be used to compute the evolution of the wave function from t_a to t_b:

$$\Psi(x, t_b) = \int_{\mathbb{R}} K(x, t_b; y, t_a)\Psi(y, t_a)\, \mathrm{d}y. \tag{2.11}$$

For a detailed discussion on the propagator $K(b, t_b; a, t_a)$, we refer the readers to [39, Chapter 2].

To quantize accelerated gradient methods, we begin with the Lagrangian formulation,

$$\mathcal{L}(X, \dot{X}, t) = e^{-\varphi_t}\left(\frac{1}{2}|\dot{X}|^2\right) - e^{\chi_t} f(X), \tag{2.12}$$

where φ_t and χ_t are time-dependent functions that control the energy dissipation in the system.[1]

Suppose the quantum particle at time t is described by the wave function $\Psi(x, t)$. To get a differential equation of $\Psi(x, t)$, we consider an infinitesimal time interval $[t, t + \epsilon]$. In this short time, the action $S[X(t)]$ is approximately ϵ times the Lagrangian,

$$S[X(t)] = \epsilon\mathcal{L}\left(\frac{a+b}{2}, \frac{b-a}{\epsilon}, t\right), \tag{2.13}$$

which is correct to first order in ϵ [39, Section 2.5]. The propagator can be evaluated by

[1] In [31], the authors introduced three time-dependent functions $\alpha_t, \beta_t, \gamma_t$ in the Bregman Lagrangian framework. Here, we use a simplified description with only two time-dependent parameters. The original formulation can be recovered by setting $\varphi_t = \alpha_t - \gamma_t$, $\chi_t = \alpha_t + \beta_t + \gamma_t$.

$$K(x, t+\epsilon; y, t) = \frac{1}{A} \exp\left\{ \frac{i}{\hbar} \epsilon \mathcal{L}\left(\frac{x+y}{2}, \frac{x-y}{\epsilon}, t \right) \right\}, \tag{2.14}$$

where A is a normalization factor that will be specified later. Plugging (2.14) into the (2.11), we obtain the following equation:

$$\Psi(x, t+\epsilon) = \frac{1}{A} \int_{-\infty}^{\infty} \exp\left\{ \frac{i}{\hbar} \epsilon \mathcal{L}\left(\frac{x+y}{2}, \frac{x-y}{\epsilon}, t \right) \right\} \Psi(y, t)\, \mathrm{d}y. \tag{2.15}$$

In the infinitesimal time interval, the smooth time-dependent functions φ_t, χ_t in (2.12) can be treated as constant functions. We plug the optimization Lagrangian (2.12) into (2.15) and introduce the change of variable $y = x + \eta$. It follows that,

$$\Psi(x, t+\epsilon) = \frac{1}{A} \int_{-\infty}^{\infty} \exp\left\{ \frac{i}{\hbar} \frac{e^{-\varphi_t} \eta^2}{2\epsilon} \right\} \exp\left\{ -\frac{i}{\hbar} \epsilon f\left(x + \frac{\eta}{2}, t \right) \right\} \Psi(x+\eta, t)\, \mathrm{d}\eta. \tag{2.16}$$

Note that we absorb the e^{χ_t} coefficient into potential field $f(x)$ and write them simply as $f(x, t)$. Then, we expand the wave function Ψ to first order in ϵ and second order in η. Note that the term $\epsilon f(x + \eta/2, t)$ is replaced by $\epsilon f(x, t)$ since the error term is of higher order than ϵ. It turns out

that

$$\Psi(x,t)+\epsilon\frac{\partial\Psi}{\partial t}=\frac{1}{A}\int_{-\infty}^{\infty}\exp\left\{\frac{i}{\hbar}\frac{e^{-\varphi_t}\eta^2}{2\epsilon}\right\}\left[1-\frac{i}{\hbar}\epsilon f(x,t)\right]\left[\Psi(x,t)+\eta\frac{\partial\Psi}{\partial x}+\frac{\eta^2}{2}\frac{\partial^2\Psi}{\partial x^2}\right]\,\mathrm{d}\eta. \tag{2.17}$$

On the left-hand side of (2.17), the $O(\epsilon^0)$[2] term is $\Psi(x,t)$; meanwhile, the $O(\epsilon^0)$ term on the right-hand side is $\Psi(x,t)$ times the coefficient:

$$\frac{1}{A}\int_{-\infty}^{\infty}\exp\left\{\frac{i}{\hbar}\frac{e^{-\varphi_t}\eta^2}{2\epsilon}\right\}\mathrm{d}\eta=\frac{1}{A}(2\pi i\hbar e^{\varphi_t}\epsilon)^{1/2}. \tag{2.18}$$

To match the $O(\epsilon^0)$ term on both sides of (2.17), we must have the coefficient (2.18) equal to 1, i.e., $A=(2\pi i\hbar e^{\varphi_t}\epsilon)^{1/2}$.

Next, we match the higher order terms in (2.17) (up to $O(\epsilon^1)$ and $O(\eta^2)$). With some algebraic manipulation, we end up with

$$\epsilon\frac{\partial\Psi}{\partial t}=C_1\frac{\partial\Psi}{\partial x}+C_2\frac{\partial^2\Psi}{\partial x^2}+C_3 f\Psi, \tag{2.19}$$

[2]We use $O(\epsilon^k)$ to indicate the correction terms in the infinitesimal expansion (2.17). For example, $O(\epsilon^0)$ means constant term, $O(\epsilon^1)$ means the first-order correction term, etc.

where the coefficients C_1, C_2, C_3 can be explicitly evaluated:

$$C_1 = \frac{1}{A}\int_{\mathbb{R}} \eta \exp\{ie^{-\varphi_t}\eta^2/2\hbar\epsilon\}\,\mathrm{d}\eta = 0, \tag{2.20}$$

$$C_2 = \frac{1}{A}\int_{\mathbb{R}} \frac{\eta^2}{2} \exp\{ie^{-\varphi_t}\eta^2/2\hbar\epsilon\}\,\mathrm{d}\eta = \frac{1}{2}i\hbar\epsilon e^{\varphi_t}, \tag{2.21}$$

$$C_3 = -\frac{i}{\hbar}\epsilon. \tag{2.22}$$

Substituting the coefficients C_1, C_2, C_3 to (2.19), we obtain the QHD dynamics described by the Schrödinger equation,

$$i\hbar\frac{\partial\Psi}{\partial t} = \left[e^{\varphi_t}\left(-\frac{\hbar^2}{2}\frac{\partial^2}{\partial x^2}\right) + e^{\chi_t}f(x)\right]\Psi, \tag{2.23}$$

which defines the QHD dynamics.

In higher dimensions, the kinetic operator will be replaced by $-\frac{\hbar^2}{2}\nabla^2$. We set $\hbar = 1$, and the general QHD Hamiltonian operator is given by

$$\hat{H}(t) = e^{\varphi_t}\left(-\frac{1}{2}\nabla^2\right) + e^{\chi_t}f(x). \tag{2.24}$$

In the rest of the paper, we sometimes also consider the QHD with three time-dependent parameters,

$$\hat{H}(t) = e^{\alpha_t-\gamma_t}\left(-\frac{1}{2}\nabla^2\right) + e^{\alpha_t+\beta_t+\gamma_t}f(x), \tag{2.25}$$

where the parameters α_t, β_t, and γ_t satisfy the *ideal scaling condition*:

$$\dot{\beta}_t \leq e^{\alpha_t}, \tag{2.26a}$$

$$\dot{\gamma}_t = e^{\alpha_t}. \tag{2.26b}$$

This setting is closely related to classical accelerated gradient descent because it has the same time-dependent parameters as in the variational formulation in [31]; see (2.8). As we will show in Section 2.5, quantum and classical gradient descent have radically different behavior even with the same time-dependent parameters.

The QHD Hamiltonian can be seen as a weighted sum of the kinetic and potential operator. The time-dependent functions φ_t and χ_t contribute to the limiting behavior of this quantum system. If these functions are constant in time, the system Hamiltonian $\hat{H}(t)$ is time-independent and the system energy is conserved. In this case, the wave function is indefinitely oscillatory. This is not what we want: as the word "descent" suggests, we want to gradually drain out the kinetic energy from the system so that the quantum particle will eventually land still on the rock bottom of the potential landscape $f(x)$. To achieve this goal, we will consider φ_t as a decreasing function and χ_t as an increasing function so that the potential energy will dominate in the long run.

2.4 Convergence analysis for convex optimization

Notations. We denote the position and momentum operators as (choosing $\hbar = 1$):

$$\hat{x} = x, \ \hat{p} = -i\nabla. \tag{2.27}$$

Given a quantum observable $\hat{O}$, its expectation value at time t with respect to the quantum wave function $|\Psi_t\rangle$ is computed by

$$\langle \hat{O} \rangle_t \coloneqq \left\langle \Psi_t \middle| \hat{O} \middle| \Psi_t \right\rangle = \int \overline{\Psi(t,x)} \hat{O} \Psi(t,x) \, \mathrm{d}x. \tag{2.28}$$

In particular, when $\hat{O} = f(x)$ is the objective function, we define

$$\mathbb{E}[f]_{\sim \Psi_t} \coloneqq \langle f \rangle_t \tag{2.29}$$

as the (average) loss function at time t.

To recover the convergence rate shown in [31] for QHD, we consider the QHD Hamiltonian with three parameters (2.25) and these parameters satisfy the ideal scaling condition (2.26).

Theorem 1. *Assume that f is a continuously differentiable convex function and the ideal scaling condition (2.26) holds. Let x^* be the unique local minimizer of f. Then, for any smooth initial wave function $|\Psi_0\rangle$, the solution $|\Psi_t\rangle$ to the Schrödinger equation $i\partial_t |\Psi_t\rangle = \hat{H}(t) |\Psi_t\rangle$ with*

Hamiltonian (2.25) satisfies

$$\mathbb{E}[f]_{\sim\Psi_t} - f(x^*) \leq O(e^{-\beta_t}). \tag{2.30}$$

Proof. Without loss of generality, we may assume $x^* = 0$ and $f(x^*) = 0$. It suffices to prove that

$$\mathbb{E}[f]_{\sim\Psi_t} \leq O(e^{-\beta_t}). \tag{2.31}$$

We take a Lyapunov function approach to prove Theorem 1. We construct the following quantum Lyapunov function:

$$\mathcal{W}(t) = \langle \hat{J}^2/2 \rangle_t + e^{\beta_t} \langle f \rangle_t, \tag{2.32}$$

in which we introduce the new operator $\hat{J} \coloneqq e^{-\gamma_t}\hat{p} + \hat{x}$. $\hat{J}$ is a legal quantum observable because both $\hat{p}$ and $\hat{x}$ are Hermitian operators. In Proposition 1, we show that $\mathcal{W}(t) \leq \mathcal{W}(0)$ for any $t \geq 0$. Meanwhile, notice that $\hat{J}^2/2$ is a positive-definite operator, so $\langle \hat{J}^2/2 \rangle_t \geq 0$, and we have:

$$e^{\beta_t}\mathbb{E}[f]_{\sim\Psi_t} \leq \mathcal{W}(t) \leq \mathcal{W}(0), \tag{2.33}$$

or equivalently,

$$\mathbb{E}[f]_{\sim\Psi_t} \leq \mathcal{W}(0)e^{-\beta_t} \leq O(e^{-\beta_t}). \tag{2.34}$$

□

Lemma 1. *Given a (time-dependent) quantum observable $\hat{O}(t)$ and let $|\Psi_t\rangle$ be the solution of the Schrödinger equation $i\partial_t |\Psi_t\rangle = \hat{H}(t) |\Psi_t\rangle$, we have*

$$\frac{\mathrm{d}}{\mathrm{d}t}\langle\hat{O}(t)\rangle_t = \langle\frac{\mathrm{d}}{\mathrm{d}t}\hat{O}(t)\rangle_t + \langle i[\hat{H}(t), \hat{O}(t)]\rangle_t. \tag{2.35}$$

Proof.

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\langle\hat{O}(t)\rangle_t &= \frac{\mathrm{d}}{\mathrm{d}t}\left\langle\Psi_t\middle|\hat{O}(t)\middle|\Psi_t\right\rangle && (2.36)\\
&= \left\langle\dot{\Psi}_t\middle|\hat{O}(t)\middle|\Psi_t\right\rangle + \left\langle\Psi_t\middle|\hat{O}(t)\middle|\dot{\Psi}_t\right\rangle + \left\langle\Psi_t\middle|\frac{\mathrm{d}}{\mathrm{d}t}\hat{O}(t)\middle|\Psi_t\right\rangle && (2.37)\\
&= i\left\langle\Psi_t\middle|\hat{H}(t)\hat{O}(t)\middle|\Psi_t\right\rangle - i\left\langle\Psi_t\middle|\hat{O}(t)\hat{H}(t)\middle|\Psi_t\right\rangle + \langle\frac{\mathrm{d}}{\mathrm{d}t}\hat{O}(t)\rangle_t && (2.38)\\
&= \left\langle\Psi_t\middle|i[\hat{H}(t), \hat{O}(t)]\middle|\Psi_t\right\rangle + \langle\frac{\mathrm{d}}{\mathrm{d}t}\hat{O}(t)\rangle_t. && (2.39)
\end{aligned}$$

□

Lemma 2 (Commutation relations)**.** *Let $\hat{x}$, $\hat{p}$ be the position and momentum operators, we have*

$$i[\hat{p}^2, \hat{x}^2] = 4\hat{x}\hat{p} - 2i, \tag{2.40a}$$

$$i[\hat{p}^2, \hat{x}\hat{p}] = 2\hat{p}^2, \tag{2.40b}$$

$$i[f, \hat{x}\hat{p}] = -\hat{x} \cdot \nabla f. \tag{2.40c}$$

Proof. We note that $[\hat{x}, \hat{p}] = i$, or equivalently, $\hat{p}\hat{x} = \hat{x}\hat{p} - i$. To show (2.40a), we compute:

$$\begin{aligned} i[\hat{p}^2, \hat{x}^2] &= i\left(\hat{p}^2\hat{x}^2 - \hat{x}^2\hat{p}^2\right) = i\left(\hat{p}(\hat{x}\hat{p} - i)\hat{x} - \hat{x}^2\hat{p}^2\right) \\ &= i\left(\hat{p}\hat{x}(\hat{x}\hat{p} - i) - i\hat{p}\hat{x} - \hat{x}^2\hat{p}^2\right) \\ &= i\left(\hat{p}\hat{x}^2\hat{p} - 2i\hat{p}\hat{x} - \hat{x}^2\hat{p}^2\right) \\ &= i\left((\hat{x}\hat{p} - i)\hat{x}\hat{p} - 2i\hat{p}\hat{x} - \hat{x}^2\hat{p}^2\right) \\ &= i\left(\hat{x}(\hat{x}\hat{p} - i)\hat{p} - i\hat{x}\hat{p} - 2i\hat{p}\hat{x} - \hat{x}^2\hat{p}^2\right) \\ &= 2\hat{x}\hat{p} + 2\hat{p}\hat{x} = 4\hat{x}\hat{p} - 2i. \end{aligned}$$

Similarly, one can exploit the commutation relation of $\hat{x}$ and $\hat{p}$ to show (2.40b). To show (2.40c), we introduce a smooth L^2 test function φ:

$$\begin{aligned} i[f, \hat{x}\hat{p}]\varphi &= i\left(f\hat{x}(-i\nabla)\varphi - \hat{x}(-i\nabla)(f\varphi)\right) \\ &= f\hat{x}\nabla\varphi - \hat{x}\left(\nabla f\varphi + f\nabla\varphi\right) \\ &= -\hat{x}\nabla f\varphi, \end{aligned}$$

so it follows that $i[f, \hat{x}\hat{p}] = -\hat{x}\nabla f$. □

Proposition 1. *The function $\mathcal{W}(t)$ is non-increasing in time, i.e., $\mathcal{W}(t) \leq \mathcal{W}(0)$ for any $t \geq 0$.*

Proof. Recall that the Lyapunov function $\mathcal{W}(t)$ is given by (2.32), in which the operator $\hat{J} = e^{-\gamma_t}\hat{p} + \hat{x}$. Invoking the commutation relation $[\hat{x}, \hat{p}] = i$, we have

$$\mathcal{W}(t) = \langle \hat{J}^2/2 \rangle_t + e^{\beta_t}\langle f \rangle_t = \langle e^{-2\gamma_t}\hat{p}^2/2 + \hat{x}^2/2 + e^{-\gamma_t}(\hat{x}\hat{p} - i/2) + e^{\beta_t} f \rangle_t. \tag{2.41}$$

By Lemma 1, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{W}(t) = \langle -\dot{\gamma}_t e^{-2\gamma_t}(\hat{p}^2) - \dot{\gamma}_t e^{-\gamma_t}(\hat{x}\hat{p} - i/2) + \dot{\beta}_t e^{\beta_t} f\rangle_t \tag{2.42}$$

$$+ \langle i[H(t), e^{-2\gamma_t}\hat{p}^2/2 + \hat{x}^2/2 + e^{-\gamma_t}(\hat{x}\hat{p} - i/2) + e^{\beta_t} f]\rangle_t, \tag{2.43}$$

where $H(t) = e^{\alpha_t - \gamma_t}\hat{p}^2/2 + e^{\alpha_t + \beta_t + \gamma_t} f$ is the QHD Hamiltonian as in (2.25). We expand the commutator (2.43) and simplify it using Lemma 2:

$$i[H(t), e^{-2\gamma_t}\hat{p}^2/2 + \hat{x}^2/2 + e^{-\gamma_t}(\hat{x}\hat{p} - i/2) + e^{\beta_t} f] \tag{2.44}$$

$$= i\Big(\cancel{\frac{1}{2}e^{\alpha_t + \beta_t - \gamma_t}[f, \hat{p}^2]} + \frac{1}{4}e^{\alpha_t - \gamma_t}[\hat{p}^2, \hat{x}^2] \tag{2.45}$$

$$+ \frac{1}{2}e^{\alpha_t - 2\gamma_t}[\hat{p}^2, \hat{x}\hat{p}] + e^{\alpha_t + \beta_t}[f, \hat{x}\hat{p}] + \cancel{\frac{1}{2}e^{\alpha_t + \beta_t - \gamma_t}[\hat{p}^2, f]}\Big) \tag{2.46}$$

$$= e^{\alpha_t - \gamma_t}(\hat{x}\hat{p} - i/2) + e^{\alpha_t - 2\gamma_t}\hat{p}^2 + e^{\alpha_t + \beta_t}\left(-\hat{x}\cdot\nabla f\right). \tag{2.47}$$

Inserting (2.47) into (2.43), we have:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{W}(t) = (e^{\alpha_t} - \dot{\gamma}_t)\left[e^{-2\gamma_t}\langle\hat{p}^2\rangle_t + e^{-\gamma_t}\langle\hat{x}\hat{p} - i/2\rangle_t\right] + \dot{\beta}_t e^{\beta_t}\langle f\rangle_t + e^{\alpha_t + \beta_t}\langle -\hat{x}\cdot\nabla f\rangle_t. \tag{2.48}$$

By the ideal scaling condition (2.26), we have $e^{\alpha_t} - \dot{\gamma}_t = 0$ and $\dot{\beta}_t \leq e^{\alpha_t}$, therefore, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{W}(t) \leq e^{\alpha_t + \beta_t}\langle f - x\cdot\nabla f\rangle_t. \tag{2.49}$$

Note that f is continuously differentiable and convex, and $x^* = 0$, we have $f(x) \leq x\cdot\nabla f(x)$ for

any $x \in \mathbb{R}^d$. We conclude that $\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{W}(t) \leq 0$, □

Remark 1. *In Theorem 1, we prove that the continuous-time QHD dynamics has the convergence rate $O(e^{-\beta_t})$ for convex differentiable problems. Like classical gradient methods, which can be derived by discretizing continuous-time ordinary differential equations, quantum optimization algorithms can be formulated by directly discretizing the QHD dynamics, as detailed in Chapter 4. Our numerical results suggest that the discrete-time quantum algorithm converges and it is robust in the sense that a comparatively large stepsize is allowed in practice. However, it remains an open question if the same convergence rate still holds for the discrete-time quantum algorithms. Also, the discrete-time version of QHD may exhibit pathological behaviors for certain non-smooth optimization problems, as in the case of classical non-smooth optimization.*

2.5 Behavior of QHD for nonconvex optimization

2.5.1 Numerical simulation

To showcase the difference between QHD and other classical and quantum algorithms, we test four algorithms to find the global minimum of the Levy function. The tested algorithms include: (1) Quantum Hamiltonian Descent (QHD), (2) Quantum Adiabatic Algorithm (QAA), (3) Nesterov's accelerated gradient descent (NAGD), and (4) stochastic gradient descent (SGD)). The Levy function is an objective function with many local minima, as shown in Figure 2.2A. Due to its sophisticated landscape, this function is often used as a test case for local and global optimization problems. Note that in this numerical experiment, the performance of QHD and QAA are simulated using a classical computer. The simulation of QHD follows Algorithm 2 in Chapter 4.

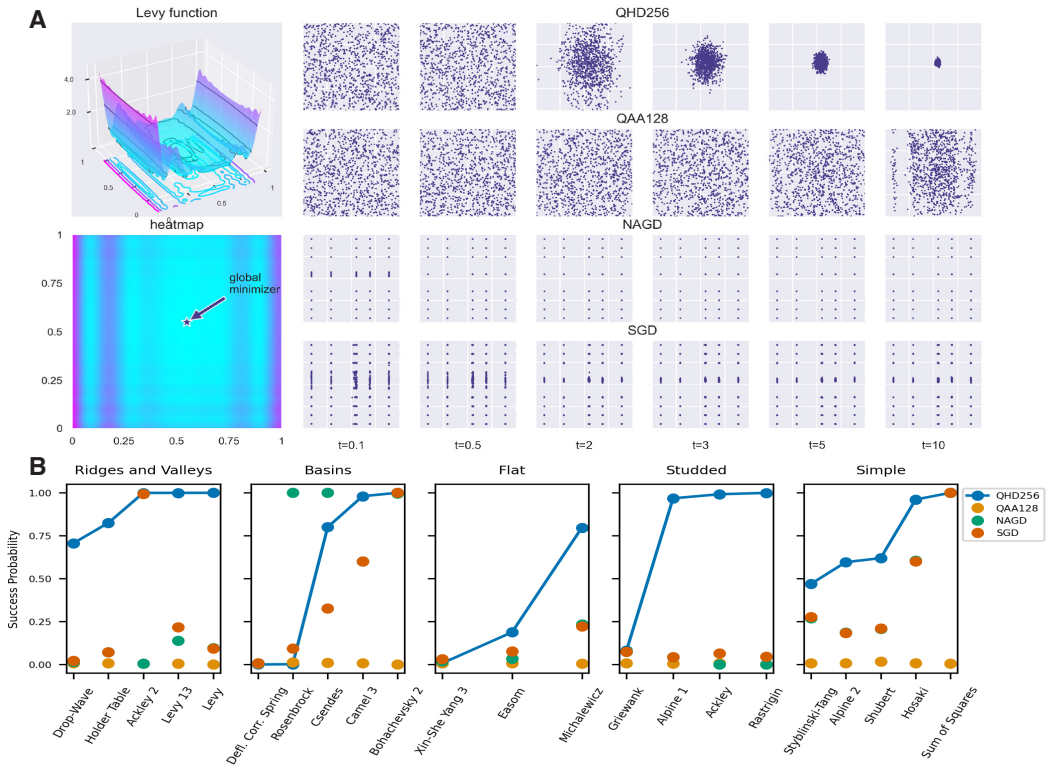


Figure 2.2: **Quantum and classical optimization methods for two-dimensional test problems. A.** Surface and heatmap plots of the Levy function. Samples from the distributions of QHD, QAA, NAGD, and SGD at different (effective) evolution times $t = 0.1, 0.5, 2, 3, 5, 10$ are shown as scatter plots. **B.** Final success probabilities of QHD, QAA, NAGD, and SGD for all 22 instances. A final solution x_k is considered "successful" if $|x_k - x^*| < 0.1$, where x^* is the (unique) global minimizer of f. Data are categorized into five groups by landscape features of the objective functions.

In Figure 2.2A, we also illustrate the solutions from the four algorithms are shown for different evolution times t. For QHD and QAA, t is the evolution time of the quantum dynamics; for the two classical algorithms, the effective evolution time t is computed by multiplying the step size and the number of iterations so that it is comparable to the one used in QHD and QAA. Compared with QHD, QAA converges at a much slower speed and little apparent convergence is observed within the time window. Although the two classical algorithms seem to converge faster than quantum algorithms, they have lower success probability because many solutions are trapped in spurious local minima. Our observation made with Levy function is consistent with

the results of other functions: as shown in Figure 2.2B, QHD has a higher success probability in most optimization instances at the same choice of total effective evolution time.

2.5.2 Three phases in QHD

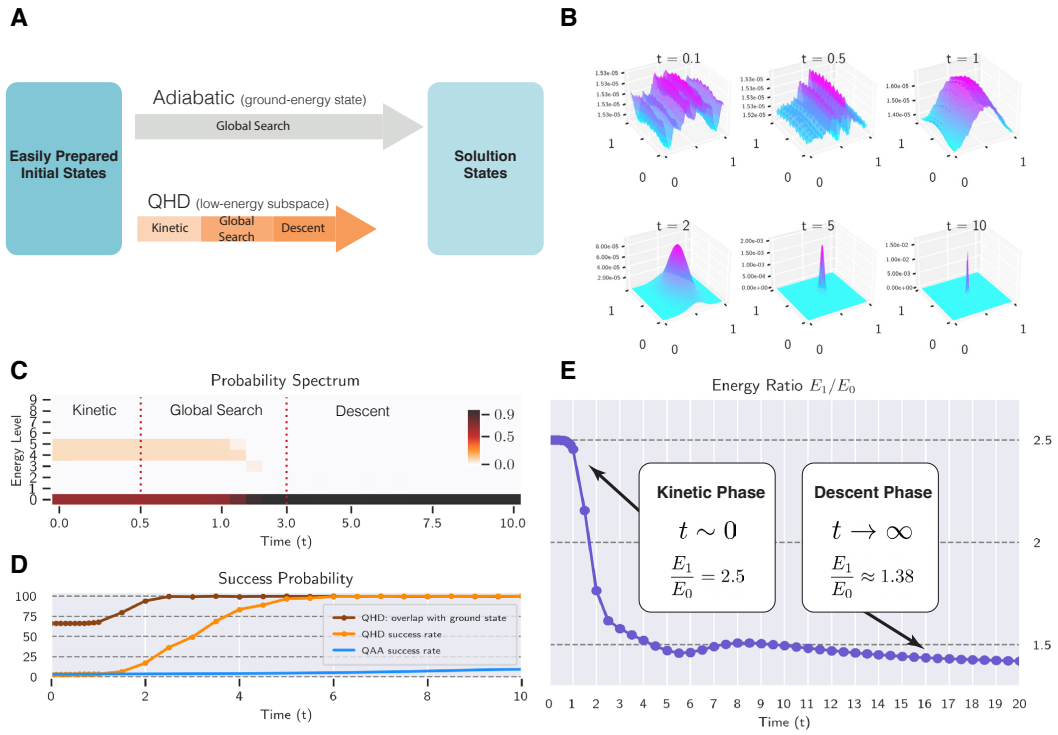


Figure 2.3: **The three-phase picture of QHD. A.** Schematic of the three-phase picture of QHD. **B.** Surface plots of the probability density in QHD for the Levy function. **C.** Probability spectrum of QHD. **D.** Success probabilities of QHD and QAA. **E.** The energy ratio E_1/E_0 in QHD shown as a function of time t.

Focusing on the QHD dynamics, we find rich dynamical properties at different stages of evolution. Figure 2.3B shows the quantum probability densities of QHD at different evolution times. First, the initial wave function becomes highly oscillatory and spreads to the full search space ($t = 0.1, 0.5$). Then, the wave function *sees* the landscape of f and starts moving towards the global minimum ($t = 1, 2$). Finally, the wave packet is clustered around the global minimum and converges like a classical gradient descent ($t = 5, 10$). This three-stage evolution is

not only seen for the Levy function but also observed in many other instances. We thus propose to divide QHD's evolution in solving optimization problems into three consecutive phases called the **kinetic phase**, the **global search phase**, and the **descent phase** according to the above observations.

The three-phase picture of QHD could be supported by several quantitative characterizations of the QHD evolution. One such characterization is the probability spectrum of QHD, which shows the decomposition of the wave function to different energy levels (Figure 2.3C). QHD begins with a major ground-energy component and a minor low-energy component. Note that no significant high-energy component with energy level ≥ 10 is found when applied to the Levy function. During the global search phase, the low-energy component is absorbed into the ground-energy component, indicating that QHD finds the global minimum (Figure 2.3D). The energy ratio E_1/E_0 is another characterization of the three phases in QHD (Figure 2.3E), where E_0 (or E_1) is the ground (or first excited) energy of the QHD Hamiltonian $\hat{H}(t)$. In the kinetic phase, the kinetic energy $-\frac{1}{2}\Delta$ dominates in the system Hamiltonian so we have $E_1/E_0 \approx 2.5$, which is the same as in a free-particle system. In the descent phase, the QHD Hamiltonian enters the "semi-classical regime" and the energy ratio can be theoretically computed based on the objective function. For the Levy function, the predicted semi-classical energy ratio reads $E_1/E_0 \approx 1.38$, which matches our numerical data.

Remark 2. *Note that the energy-absorbing behavior is highly initial-state-dependent and does not apply to arbitrary initial states; otherwise, it violates the unitarity of quantum evolution. In most cases, the initial state is a superposition of low-energy eigenstates of the initial Hamiltonian* $\hat{H}(0)$*, and it remains in the low-energy subspace of* $\hat{H}(t)$ *in the entire evolution. Since all the*

low-energy eigenstates of the final Hamiltonian are locally clustered near the global minimum of f (see Section 3.1 for a detailed discussion), the final state is also highly concentrated near the global minimizer of f.

The three-phase picture of QHD sheds light on why QAA has slower convergence. Compared to QHD, QAA has neither a kinetic phase nor a descent phase. In the kinetic phase, QHD averages the initial wave function over the whole search space to reduce the risk of poor initialization, while QAA remains in the ground state throughout its evolution, so it never gains as much kinetic energy. In the descent phase, QHD exhibits convergence similar to classical gradient descent and is insensitive to spatial resolution; such fast convergence is not seen in QAA.

In QAA, the use of the radix-2 representation scrambles the Euclidean topology so that the resulting discrete problem is even harder than the original problem (e.g., see Figure B.2). Failing to incorporate the continuity structure, QAA is hence sensitive to the resolution of spatial discretization; we observe that higher resolutions often cause worse QAA performance, see Figure 4.3.

It is worth noting that the radix-2 representation is not the only way to discretize a continuous problem. One can lift QAA to the continuous domain by choosing its Hamiltonian over a continuous space in a general way. From this perspective, QHD could be interpreted as a special version of the general QAA with a particular choice of the Hamiltonian. However, some existing results [61] suggest that QHD may have fast convergence properties that the general theory of QAA fails to explain.

Chapter 3: Quantum Hamiltonian Descent for nonconvex optimization

Chapter summary. Nonconvex optimization is a central object in machine learning and operations research. However, lots of nonconvex problems naturally arising from application domains possess sophisticated optimization landscapes with a huge number of saddle points and spurious local minima, posing great challenges for classical optimization algorithms. Quantum Hamiltonian Descent (QHD), a genuine quantum-upgraded version of classical gradient descent, could leverage the quantum tunneling effect to go through barriers on the optimization landscape, rendering itself a competitive candidate for complex nonconvex optimization. In this chapter, we present three theoretical results regarding QHD's speedup for nonconvex optimization problems, including the asymptotic convergence of QHD in the slow-varying time limit, a quadratic quantum speedup with QHD for spatial search on hypercube graphs, and a super-polynomial performance separation between QHD and several classical state-of-the-art optimizers. Throughout this chapter, we assume access to fault-tolerant digital quantum computers.

3.1 Preliminaries: the theory of Schrödinger operators

Convexity and spectral gap

Our theoretical analysis in this work heavily relies on the spectrum of the Schrödinger operator,

$$\hat{H} = -\nabla^2 + f, \tag{3.1}$$

where ∇^2 is the Laplacian operator in the real space $\mathbb{R}^d$, and $f\colon \mathbb{R}^d \to \mathbb{R}$ is a continuous objective function that we want to minimize. The physical meaning of f is a "potential field" that confines the motion of a quantum particle.

When f is non-negative and it diverges at infinity (i.e., $\lim_{\|x\|\to\infty} f(x) = \infty$), the standard theory of elliptic operators [62, Theorem 10.7] guarantees that the spectrum of $\hat{H}$ is discrete and all eigenvalues are positive. We can arrange the eigenvalues of $\hat{H}$ in ascending order: $E_0 < E_1 \leq \ldots$. The difference between the first two eigenvalues (i.e., $E_1 - E_0$) is often called the *spectral gap*. It is well-known that the spectral gap of the Schrödinger operator (3.1) has a deep connection to the convexity of the potential field f. Since the eighties, several authors (van den Berg [63], Ashbaugh and Benguria [64], and Yau [65]) have independently observed that the spectral gap of a Schrödinger operator with a convex potential field has a lower bound $3\pi^2/D^2$, where D is the diameter of the domain on which the Schrödinger operator is defined. This observation, known as the Fundamental Gap Conjecture, has been proven by Andrews and Clutterbuck [66] in 2011 using an involved analysis of the heat equation.

We now discuss a simple example known as the quantum harmonic oscillator. A quantum harmonic oscillator is described by a Schrödinger operator with a quadratic potential field,

$$H = \hbar^2 \left(-\frac{1}{2}\nabla^2 \right) + \left(\frac{1}{2}\omega^2 x^2 \right), \tag{3.2}$$

where ω is a fixed frequency and $\hbar$ is the so-called Planck constant. The eigenvalues of the quantum harmonic oscillator are given by $E_n = (2n+1)\frac{\hbar}{2}\omega$. We refer the readers to [67] for more discussions on quantum harmonic oscillators.

If we consider the rescaled operator,

$$\frac{1}{\hbar}H = \hbar \left(-\frac{1}{2}\nabla^2 \right) + \frac{1}{\hbar}\left(\frac{1}{2}\omega^2 x^2 \right),$$

we find the eigenvalues of $H/\hbar$ are $E'_n = (n+1/2)\omega$, which only depend on the curvature of the potential field regardless of the value of $\hbar$. The spectral gap of the rescaled quantum harmonic oscillator $H/\hbar$ seems to be an invariant quantity determined by the geometry of the quadratic potential field.

Inspired by the case of quantum harmonic oscillators, we define a one-parameter family of Schrödinger operators for a given objective function f,

$$\hat{H}(\lambda) = \frac{1}{\lambda}\left(-\frac{1}{2}\nabla^2 \right) + \lambda f, \tag{3.3}$$

where $\lambda > 0$ is an interpolating parameter and we assume f is non-negative and diverges at infinity. For any positive λ, the spectrum of $\hat{H}(\lambda)$ is discrete and we arrange the eigenvalues of

$\hat{H}(\lambda)$ in ascending order,

$$E_0(\lambda) < E_1(\lambda) \leq E_2(\lambda) \leq \dots.$$

The spectral gap of $\hat{H}(\lambda)$ is denoted by

$$\Delta(\lambda) \coloneqq E_1(\lambda) - E_0(\lambda). \tag{3.4}$$

If f is a convex quadratic function, $\hat{H}(\lambda)$ is precisely a quantum harmonic oscillator. According to our discussion above, all the eigenvalues of the operator $\hat{H}(\lambda)$ only depend on f, regardless of the value of λ. Nevertheless, when f is nonconvex and non-quadratic, the eigenvalues of $\hat{H}(\lambda)$ change along with λ. The spectral gap $\Delta(\lambda)$ depends on the nonconvexity of f [68, 66] and it also changes with λ. In particular, when f is a double-well function (described by a quartic polynomial), the spectral gap can be made arbitrarily small by tuning the coefficients in the quartic polynomial [69].

Ground states of the Schrödinger operator

In the Schrödinger operator (3.1), there are two major components: the kinetic operator $-\nabla^2$ and the potential operator f. When the kinetic operator dominates, the low-energy subspace of the Schrödinger operator tends to behave like that of a free particle. In another case, if the potential f plays a leading role, the ground state of the Schrödinger operator exhibits properties as

like a Gaussian distribution. For example, if we consider the following 1D Schrödinger operator,

$$\hat{H}(\lambda) = \frac{1}{\lambda}\left(-\frac{1}{2}\nabla^2\right) + \lambda f, \tag{3.5}$$

where $f\colon \mathbb{R} \to \mathbb{R}$ is a smooth one-dimensional potential function with a single, non-degenerate zero at $x^* = 0$: $f(0) = 0$, $\nabla f(0) = 0$, and $f''(0) > 0$. The function f admits a Taylor expansion near the global minimum:

$$f(x) = \frac{1}{2}f''(x^*)(x - x^*)^2 + O(x^3).$$

As λ becomes large, the potential landscape observed by the quantum particle is essentially the quadratic part of the potential f near x^*. Let $\gamma = \sqrt{f''(x^*)}$, we can define a comparison Schrödinger operator

$$\hat{K}(\lambda) = \frac{1}{\lambda}\left(-\frac{1}{2}\nabla^2\right) + \lambda\left(\frac{1}{2}\gamma^2(x - x^*)^2\right). \tag{3.6}$$

This new operator $\hat{K}(\lambda)$ is a quantum harmonic oscillator and it gives a nice approximation of the low-energy subspace of $\hat{H}(\lambda)$ for large λ.

Theorem 2 (Theorem 4.1, [70])**.** *Assume $f(x)$ is polynomially bounded (i.e., $|f(x)| \leq C(1 + |x|^m)$ for some constant C and a fixed integer m). Let n be a fixed integer such that $E_n(\lambda)$ is a simple*[1] *eigenvalue of $\hat{H}(\lambda)$. Let $\Phi_n(\lambda)$ be the corresponding eigenfunction. Denote e_n and $\phi_n(\lambda)$*

[1] A *simple* eigenvalue means the multiplicity is 1. In other words, the corresponding eigenspace is non-degenerate.

to be the n-th eigenvalue and eigenfunction of $\hat{K}(\lambda)$. Then, for sufficiently large λ, we have that

$$E_n(\lambda) - e_n = \mathcal{O}(\lambda^{-1}),\ \|\Phi_n(\lambda) - \phi_n(\lambda)\| = \mathcal{O}(\lambda^{-1/2}). \tag{3.7}$$

The ground state of the quantum harmonic oscillator $\hat{K}(\lambda)$ is a Gaussian state

$$\phi_0(\lambda) = \left(\frac{\lambda\gamma}{\pi}\right)^{1/4} e^{-\frac{\gamma\lambda(x-x^*)^2}{2}}, \tag{3.8}$$

and the eigenvalues of $\hat{K}$ are given by $e_n = \left(n + \frac{1}{2}\right)\gamma$.

As λ increases, the Gaussian state $\phi_0(\lambda)$ is localized in the vicinity of the global minimizer x^* and its tail (i.e., the probability of landing far away from x^*) is exponentially small. One would naturally guess that $\Phi_0(\lambda)$, the actual ground state of the Schrödinger operator (3.5), also has a very small tail for large λ. However, this is not immediately clear from Theorem 2. In Section 3.4, we will prove a stronger result to characterize the smallness of the tail of $\Phi_0(\lambda)$ (see Lemma 14). It turns out that, given a fast-growing potential field f, the ground state $\Phi_0(\lambda)$ corresponds to a *sub-Gaussian* distribution whose variance scales proportional to $1/\lambda$.

Quantum adiabatic theorem for unbounded Hamiltonian

Given a quantum Hamiltonian $H(t)$, $t \in [0, t_f]$, we consider the dynamics described by the Schrödinger equation,

$$i\epsilon\frac{\mathrm{d}}{\mathrm{d}t}\left|\psi^\epsilon(t)\right\rangle = H(t)\left|\psi^\epsilon(t)\right\rangle, \tag{3.9}$$

subject to an initial state $|\psi^\epsilon(0)\rangle = |\psi_0\rangle$. We suppose that $U^\epsilon(t)$ is the propagator of the dynamics described by (3.9), so the solution at time t is given by

$$|\psi^\epsilon(t)\rangle = U^\epsilon(t)\,|\psi_0\rangle\,. \tag{3.10}$$

The parameter $\epsilon > 0$ controls the time scale on which the quantum Hamiltonian $H(t)$ varies. For a small ϵ, the system evolves slowly and the dynamics are relatively simple: if the system begins with an eigenstate of $H(0)$, it remains close to an eigenstate of $H(t)$. This process is called *adiabatic quantum evolution*.

Formally, We define a new Hamiltonian operator,

$$H_a(t) = H(t) + i\epsilon[\dot{P}, P], \tag{3.11}$$

where $P(t)$ is a rank-1 projector onto the ground state of $H(t)$, $\dot{P}$ is the time derivative of P, and $[\cdot,\cdot]$ is the commutator. Let $U_a^\epsilon(t)$ be the propagator of the quantum dynamics generated by (3.11), i.e., $U_a^\epsilon(t)$ is given as the solution of

$$i\epsilon\frac{\mathrm{d}}{\mathrm{d}t}U_a^\epsilon(t) = H_a(t)U_a^\epsilon(t), \tag{3.12}$$

subject to $U_a^\epsilon(0) = \mathbf{1}$. The propagator $U_a^\epsilon(t)$ is called the *adiabatic intertwiner* because it preserves the ground-energy subspace of $H(t)$ [71]. Precisely, we have that

$$U_a^\epsilon(t)P(0) = P(t)U_a^\epsilon(t). \tag{3.13}$$

The quantum adiabatic theorem states that adiabatic intertwiner (i.e., the exact adiabatic propagator) $U_a^\epsilon(t)$ is a good approximation of the actual propagator $U^\epsilon(t)$ for sufficiently small ϵ. Various formulations of the quantum adiabatic theorem exist in the literature, while most of them assume the Hamiltonian is a bounded linear operator. In Quantum Hamiltonian Descent, however, the Hamiltonian is an unbounded operator defined in the real space $\mathbb{R}^d$. Here, we introduce a quantum adiabatic theorem for unbounded Hamiltonian [72, 73], which allows us to have a neat analysis without discretizing the unbounded operator in QHD.

For a self-adjoint operator $H(t)$ with discrete spectrum at any t, the spectral gap of $H(t)$, denoted by $\Delta(t)$, is the difference between the first two eigenvalues of $H(t)$ at time t.

Theorem 3 (Theorem 2.1 [73])**.** *Assume that for all $t \in [0, t_f]$, there exist positive numbers c_0 and c_1 such that*

$$\dot{H}^2 \leq c_0 + c_1 H^2. \tag{3.14}$$

Moreover, we assume $H > 0$ and the spectral gap $\Delta(t)$ has a uniform lower bound, i.e., $\Delta(t) \geq \Delta_{\min} > 0$. We denote ϕ_0 as the ground state of $H(0)$. Then, we have that

$$\left\| \left(U^\epsilon(t_f) - U_a^\epsilon(t_f)\right) \phi_0 \right\| \leq \epsilon \theta t_f, \tag{3.15}$$

where θ is a constant that only depends on c_0, c_1 and $\Delta_{\min}$.

Remark 3. *In Theorem 3, we rescale the time variable compared to the original version in [73, Theorem 2.1]. After the time rescaling, the adiabatic error from the original theorem reads $b = \frac{\theta t_f}{1/\epsilon} = \epsilon \theta t_f$, which is precisely what we have in (3.15).*

3.2 Global convergence of QHD with slow-varying time dependence

Non-convex optimization problems usually have many local minima, and the theoretical result for convex problems does not naturally generalize. Nevertheless, we can prove the global convergence of QHD for non-convex problems under certain realistic assumptions.

In this section, we consider the QHD Hamiltonian with two parameters (2.24) because the convergence rate is not guaranteed for non-convex problems. Also, for simplicity, we only consider the one-dimensional case, i.e., $f : \mathbb{R} \to \mathbb{R}$. However, our argument is readily generalized to arbitrary finite dimensions. We denote $\Pi_N(t)$ as the orthogonal projection onto the subspace spanned by the first N eigenstates of the operator $\hat{H}(t)$, given a fixed positive integer N.

Assumption 1. *The objective function f is continuously differentiable, unbounded at infinity, and has a unique global minimum at x^*. Without loss of generality, we assume $f(x^*) = 0$.*

Assumption 2. *The time-dependent parameters are slow-varying in time (i.e., $|\dot{\varphi}_t|, |\dot{\chi}_t| \ll 1$) and $\lim_{t\to\infty} e^{\varphi_t - \chi_t} = 0$. The Hamiltonian $\hat{H}(t)$ has no crossing eigenvalues for $t \geq 0$. Moreover, the semi-classical approximation holds. This means that in the regime $e^{\varphi_t - \chi_t} \ll 1$, the first N eigenstates of $\hat{H}(t)$ are approximated by the first N eigenstates of the quantum Harmonic oscillator:*

$$\hat{H}_{HO} \coloneqq e^{\varphi_t}\left(-\frac{1}{2}\nabla^2\right) + \frac{e^{\chi_t}}{2} f''(x^*)(x - x^*)^2. \tag{3.16}$$

Remark 4. *In what follows, we will call $f_q \coloneqq f(x^*) + \frac{1}{2} f''(x^*)(x - x^*)^2$ as the quadratic model of the function f.*

Assumption 3. *The initial wave function $|\Psi_0\rangle$ is in the low-energy subspace of the QHD Hamiltonian at $t = 0$, i.e.,*

$$\Pi_N(0)\,|\Psi_0\rangle = |\Psi_0\rangle\,. \tag{3.17}$$

Now, we justify why the assumptions we made are realistic for many non-convex optimization problems.

- Assumption 1 is a very standard assumption on the objective function in optimization theory. We assume $f(x)$ has a unique global minimum to avoid degenerate cases when we do semi-classical approximation. It is worth mentioning that the uniqueness of global minimum is just a technical assumption and it does not mean QHD can not solve optimization problems with multiple global minima!

- The semi-classical approximation in Assumption 2 is a well-known result in the spectral theory of Schrödinger operators. For a Schrödinger operator $H = -\hbar\Delta + \frac{1}{\hbar}f$, it is shown that when $\hbar \to 0$, the low-energy spectrum is well approximated by that of a quantum harmonic oscillator $H' = -\hbar\Delta + \frac{1}{\hbar}f_q$ (i.e., the potential field f is replaced by its quadratic model), see [62, Theorem 11.3, informal]. In our assumption, we take a stronger form so not only the low-energy spectrum but also the low-energy eigenstates of H are approximated by those of H' as well. In Figure 3.1, we show the low-energy subspace of the Schrödinger operator H and its semi-classical limit H'. The numerical evidence suggests that our assumption is valid in practical problems.

- Though it is difficult to prepare $|\Psi_0\rangle$ to satisfy Assumption 3 without enough knowledge

of f, we observe that $(I - \Pi_N(0))\,|\Psi_0\rangle$ is often very small for $N \approx 10$. For example, in our 2D experiments, we usually have $\|(I - \Pi_N(0))\,|\Psi_0\rangle\,\| \leq 10^{-4}$ for $|\Psi_0\rangle$ as a uniform superposition state. The reason is that the eigenstates of $\hat{H}(0)$ form a complete basis set, so it is always possible to find an integer N such that the projection of $|\Psi_0\rangle$ onto the low-energy subspace is close enough to $|\Psi_0\rangle$ itself.

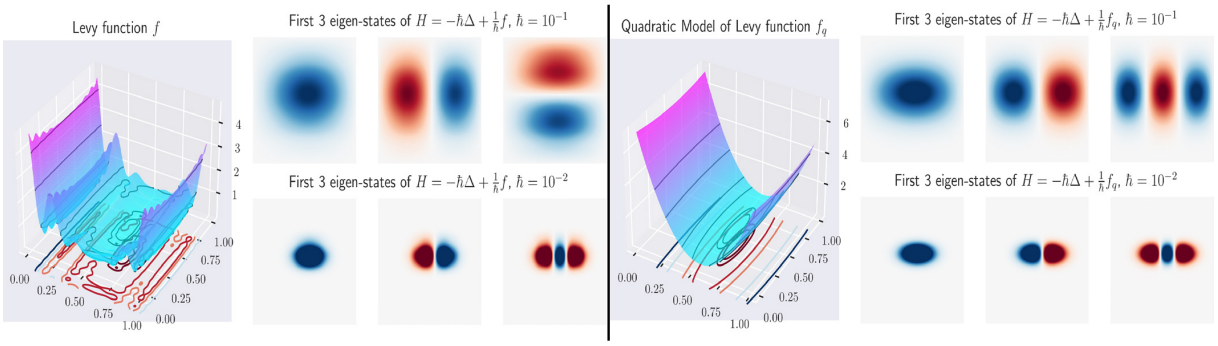


Figure 3.1: The low-energy subspace of the Schrödinger operator, illustrated with Levy function f and its quadratic model f_q. The ground state, the first excited state, and the second excited eigenstate of the Schrödinger operator are plotted for $\hbar = 0.1, 0.01$. When $\hbar$ is small, it is clear that the low-energy subspae of the Schrödinger operator converges to its semi-classical limit (i.e., with the quadratic model as potential field).

Theorem 4. *Suppose Assumption 1, Assumption 2, and Assumption 3 are satisfied. Then, QHD will converge to the global minimum of f when t is large enough, i.e.,*

$$\lim_{t\to\infty} \mathbb{E}[f]_{\sim\Psi_t} = f(x^*). \tag{3.18}$$

Remark 5. *In the proof of the theorem, we invoke the adiabatic approximation. Since the same technique is also used in the analysis of Quantum Adiabatic Algorithm (QAA)[2], we are no worse than QAA (in terms of convergence time) in the worst case. As [61] suggests, QHD is likely to converge much faster than the standard QAA because the QHD Hamiltonian is more regular.*

[2]For more details, see Section A.

Proof. Since f is continuously differentiable and unbounded at infinity, by [62, Theorem 10.7], the spectrum of $\hat{H}(t)$ is purely discrete for all $t \geq 0$. Denote the eigen-pairs of $H(t)$ by $\{E_n(t), |n;t\rangle\}_n$. We now represent the QHD wave function using this basis:

$$|\Psi_t\rangle = \sum_n c_n(t) e^{i\theta_n(t)} |n;t\rangle \,, \tag{3.19}$$

where $\theta_n(t)$ are real-valued functions given by $\theta_n(t) \coloneqq -\int_0^t E_n(s)\,\mathrm{d}s$. This representation is common in the adiabatic approximation of quantum evolution. For detailed background information and technical discussions, we refer the readers to [74, Section 5.6].

Inserting (3.19) into the Schrödinger equation $i\partial_t |\Psi_t\rangle = \hat{H}(t) |\Psi_t\rangle$, we obtain the following equations (which is precisely the same as [74, Equation (5.6.10)]):

$$\dot{c}_m(t) = \underbrace{-c_m(t) \langle m;t| \left[\frac{\partial}{\partial t} |m;t\rangle\right]}_{\text{Part I.}} \underbrace{- \sum_{n\neq m} c_n(t) e^{i(\theta_n - \theta_m)} \frac{\langle m;t| \dot{H} |n;t\rangle}{E_n(t) - E_m(t)}}_{\text{Part II.}} . \tag{3.20}$$

No crossing eigenvalues in $\hat{H}(t)$ implies that $E_n(t) - E_m(t) \neq 0$. By the slow-varying assumption on the time-dependent parameters, we have $\left\langle m;t \middle| \dot{H} \middle| n;t \right\rangle \ll 1$. Now, we invoke the adiabatic approximation [74, Section 5.6] and neglect Part II in (3.20).

As for Part I in (3.20), we note that $\langle m;t| \left[\frac{\partial}{\partial t} |m;t\rangle\right]$ is purely imaginary for all m. To see this, we differentiate the identity $\langle m;t|m;t\rangle = 1$ w.r.t the time t and it turns out that

$$0 = \frac{\partial}{\partial t} \langle m;t|m;t\rangle = 2\,\mathrm{Re}\left\{ \langle m;t| \left[\frac{\partial}{\partial t} |m;t\rangle\right] \right\} . \tag{3.21}$$

Therefore, (3.20) is simplified to the following first-order linear ODE:

$$\dot{c}_m(t) = -ic_m(t)\mathscr{A}_m(t), \tag{3.22}$$

in which $\mathscr{A}_m(t)$ are real-valued functions. This simplification leads to a closed-form solution for $c_m(t)$:

$$c_m(t) = c_m(0)e^{-i\int_0^t \mathscr{A}(s)\mathrm{d}s}, \tag{3.23}$$

which implies that $|c_m(t)| = |c_m(0)|$ for all $t \geq 0$. Due to Assumption 3, we conclude that $|\Psi_t\rangle$ remains in the low-energy subspace during the evolution:

$$|\Psi_t\rangle = \sum_{m<N} c_m(t)e^{i\theta_m(t)} |m;t\rangle \,. \tag{3.24}$$

With this finite expansion of $|\Psi_t\rangle$, we compute the expectation value:

$$\mathbb{E}[f]_{\sim\Psi_t} = \langle\Psi_t|f|\Psi_t\rangle = \sum_{m<N} |c_m|^2 \langle m;t|f|m;t\rangle + \sum_{\substack{m\neq n\\ m,n<N}} \overline{c_m}c_n e^{-i(\theta_m-\theta_n)} \langle m;t|f|n;t\rangle \,. \tag{3.25}$$

For large enough t, the parameters $e^{\varphi_t-\chi_t}$ is so small that we can invoke the semi-classical approximation. We denote the eigenstates of $\hat{H}_{HO}(t)$ as $|e_n;t\rangle$. By Assumption 2), we approximate the eigenstates of $\hat{H}(t)$ by those of $\hat{H}_{HO}(t)$. We write $V = \frac{e^{\chi_t}}{2}f''(x^*)(x-x^*)^2$, by Lemma 3

(using $\hbar^2 = e^{\varphi_t}$),

$$\langle m;t|V|m;t\rangle \approx \langle e_m;t|V|e_m;t\rangle = \frac{1}{2}\sqrt{e^{\varphi_t+\chi_t}f''(x^*)}\left(m+\frac{1}{2}\right). \tag{3.26}$$

Note that $f = e^{-\chi_t}V$ in the semi-classical limit, then it follows that

$$\langle m;t|f|m;t\rangle = \frac{1}{2}\sqrt{e^{\varphi_t-\chi_t}f''(x^*)}\left(m+\frac{1}{2}\right). \tag{3.27}$$

Next, we use the Cauchy-Schwarz inequality to bound the cross terms in (3.25):

$$\begin{aligned}
|\langle m;t|V|n;t\rangle| &= \left|\int_{\mathbb{R}} \overline{e_m(x)}Ve_n(x)\,\mathrm{d}x\right| && (3.28)\\
&\leq \left[\left(\int_{\mathbb{R}} V|e_m|^2\,\mathrm{d}x\right)\left(\int_{\mathbb{R}} V|e_n|^2\,\mathrm{d}x\right)\right]^{1/2} && (3.29)\\
&= \left(\langle e_m;t|V|e_m;t\rangle\langle e_n;t|V|e_n;t\rangle\right)^{1/2} && (3.30)\\
&= \frac{1}{2}\sqrt{e^{\varphi_t+\chi_t}f''(x^*)}\sqrt{(m+1/2)(n+1/2)}, && (3.31)
\end{aligned}$$

and similarly, we insert $f = e^{-\chi_t}V$ and end up with

$$|\langle m;t|f|n;t\rangle| \leq \frac{1}{2}\sqrt{e^{\varphi_t-\chi_t}f''(x^*)}\sqrt{(m+1/2)(n+1/2)}. \tag{3.32}$$

Substituting both (3.27) and (3.32) to (3.25), we get an upper bound of $\mathbb{E}[f]_{\sim\Psi_t}$:

$$\left|\mathbb{E}[f]_{\sim\Psi_t}\right| \leq \frac{1}{2}\sqrt{e^{\varphi_t-\chi_t}f''(x^*)}\left(\sum_{m<N}\left(m+\frac{1}{2}\right)+\sum_{\substack{m\neq n\\ m,n<N}}\sqrt{(m+1/2)(n+1/2)}\right) \tag{3.33}$$

$$\leq \frac{1}{2}\sqrt{e^{\varphi_t-\chi_t}f''(x^*)}\left(\frac{1}{2}N^2+\frac{1}{2}N^3-\frac{1}{2}N^2\right)=\frac{N^3}{4}\sqrt{e^{\varphi_t-\chi_t}f''(x^*)}, \tag{3.34}$$

which will converge to the global minimum $f(x^*)=0$ because N and $f''(x^*)$ are fixed positive numbers and $\lim_{t\to\infty}e^{\varphi_t-\chi_t}=0$. □

Lemma 3. *Consider the quantum harmonic oscillator*

$$\hat{H}=\hat{K}+\hat{V}, \tag{3.35}$$

where $\hat{K}=-\frac{\hbar^2}{2}\frac{\partial^2}{\partial x^2}$ *is the kinetic operator and* $\hat{V}=\frac{1}{2}\omega^2x^2$ *is the quadratic potential field. We denote the eigenstates of* $\hat{H}$ *as* $\{|e_n\rangle\}_{n=0}^{\infty}$*. Then, we have*

$$\left\langle e_n\middle|\hat{K}\middle|e_n\right\rangle=\left\langle e_n\middle|\hat{V}\middle|e_n\right\rangle=\frac{1}{2}\hbar\omega\left(n+\frac{1}{2}\right), \tag{3.36}$$

$$\left\langle e_n\middle|\hat{K}\middle|e_{n\pm1}\right\rangle=\left\langle e_n\middle|\hat{V}\middle|e_{n\pm1}\right\rangle=0. \tag{3.37}$$

Proof. We introduce the ladder operators:

$$\hat{a}_{\pm}:=\frac{1}{\sqrt{2\hbar\omega}}\left(\mp i\hat{p}+\omega x\right), \tag{3.38}$$

where $\hat{p} = -i\frac{\partial}{\partial x}$ is the momemtum operator. The $\hat{a}_+$ (or $\hat{a}_-$) is known as the *raising* (or *lowering*) operator because

$$\hat{a}_+ |e_n\rangle \propto |e_{n+1}\rangle \,,\ \hat{a}_- |e_n\rangle \propto |e_{n-1}\rangle \,.$$

And in particular, we have (c.f. [67, Section 2.3])

$$\hat{a}_+\hat{a}_- |e_n\rangle = n |e_n\rangle \,,\ \hat{a}_-\hat{a}_+ |e_n\rangle = (n+1) |e_n\rangle \,. \tag{3.39}$$

We use the definition of ladder operators (3.38) to express $\hat{p}$ and x,

$$\hat{p} = i\sqrt{\frac{\hbar\omega}{2}} (\hat{a}_+ - \hat{a}_-) \,,\ x = \sqrt{\frac{\hbar}{2\omega}} (\hat{a}_+ + \hat{a}_-) \,. \tag{3.40}$$

We are interested in $\hat{K}$ and $\hat{V}$:

$$\hat{K} = \frac{1}{2}\hat{p}^2 = -\frac{\hbar\omega}{4} \left[(\hat{a}_+)^2 - (\hat{a}_+\hat{a}_-) - (\hat{a}_-\hat{a}_+) + (\hat{a}_-)^2\right] , \tag{3.41}$$

$$\hat{V} = \frac{\omega^2}{2}x^2 = \frac{\hbar\omega}{4} \left[(\hat{a}_+)^2 + (\hat{a}_+\hat{a}_-) + (\hat{a}_-\hat{a}_+) + (\hat{a}_-)^2\right] . \tag{3.42}$$

Note that $(\hat{a}_+)^2$ in $\hat{K}$ or $\hat{V}$ will raise $|e_n\rangle$ to $|e_{n+2}\rangle$ and thus has no overlap with $|e_n\rangle$. Similarly, $\langle e_n| (\hat{a}_-)^2 |e_n\rangle = 0$. It turns out that,

$$\left\langle e_n \middle| \hat{K} \middle| e_n \right\rangle = \frac{\hbar\omega}{4} \langle e_n| \left[(\hat{a}_+\hat{a}_-) + (\hat{a}_-\hat{a}_+)\right] |e_n\rangle = \frac{\hbar\omega}{2} \left(n + \frac{1}{2}\right) . \tag{3.43}$$

Similarly, $\left\langle e_n \middle| \hat{V} \middle| e_n \right\rangle = \frac{\hbar\omega}{2} \left(n + \frac{1}{2}\right)$.

As for $\left\langle e_n \middle| \hat{K} \middle| e_{n+1} \right\rangle$, we note that $|K\rangle$ maps the state $|e_{n+1}\rangle$ to a linear combination of $|e_{n-1}\rangle$, $|e_{n+1}\rangle$, and $|e_{n+3}\rangle$. It has no overlap with the state $|e_n\rangle$, thus $\left\langle e_n \middle| \hat{K} \middle| e_{n+1} \right\rangle = 0$. The same argument applies to show that $\left\langle e_n \middle| \hat{K} \middle| e_{n-1} \right\rangle = 0$ and $\left\langle e_n \middle| \hat{V} \middle| e_{n\pm1} \right\rangle = 0$. □

3.3 A quadratic speedup by QHD

In this section, we show that Quantum Hamiltonian Descent can provide a quadratic speedup for the spatial search problem, a combinatorial optimization problem with classical query lower bound $\Omega(N)$. The key idea is to embed the spatial search problem into a nonconvex continuous optimization problem. Using an argument due to Childs and Goldstone [75], we show that QHD solves the problem with $\tilde{O}(\sqrt{N})$ quantum queries.

Problem formulation

The spatial search problem is to find a marked item in a structured database. The database structure is modeled as an undirected graph G. At each vertex x on the graph, there is a black-box function $f(x)$ that returns 1 if x is the marked item; otherwise, the black-box function $f(x)$ returns 0. We also assume access to the neighborhood of a given vertex x. In each search step, one can only move to a neighboring vertex on the graph and check if it is the marked item. Here, we consider the spatial search problem on the d-dimensional hypercube $\{0,1\}^d$, in which there are $N = 2^d$ vertices in total. Clearly, any classical algorithm that solves this problem necessarily has to query f at least $\Omega(N)$ times.

Reformulation as nonconvex optimization

The spatial search problem on the d-dimensional hypercube can be formulated as a continuous optimization problem. We consider the following function:

$$F(x) = \left(\sum_{k=1}^{d} a x_k^4 - c x_k^2 \right) + g(x), \tag{3.44}$$

where $a, c > 0$, and

$$g(x) = \begin{cases} -\frac{B}{e} \exp\left(-\frac{r^2}{|x-x_0|^2 - r^2} \right) & (|x - x_0| \leq r) \\ 0 & (|x - x_0| > r), \end{cases} \tag{3.45}$$

in other words, $g(x)$ is compacted supported in a radius-r ball centered at $x_0 \in \mathbb{R}^d$ and $g(x) \geq g(0) = -B$. Since both the quartic polynomial and $g(x)$ are smooth functions, $F(x)$ is also smooth.

Suppose the marked vertex is $w \in \{0,1\}^d$. When $a, c > 0$, the polynomial function $ax^4 - cx^2$ has two local minimizers at $x^{\pm} = \pm\sqrt{\frac{c}{2a}}$, respectively. It turns out that the quartic polynomial $\sum_{k=1}^{d} a x_k^4 - c x_k^2$ has 2^d local minima, and each local minimizer has the k-th ($1 \leq k \leq d$) coordinate to be either x^- or x^+. Now, if we choose $r \ll \sqrt{\frac{c}{2a}}$ and $x_0 \in \mathbb{R}^d$ such that

$$(x_0)_k = \begin{cases} x^- & (w_k = 0) \\ x^+ & (w_k = 1), \end{cases}$$

the function $F(x)$ has 2^d local minima but a unique global minimum at $x = x_0$. Therefore, the

spatial search problem discussed in Section 3.3 reduces to the global minimization problem of $F(x)$: each local minimum of $F(x)$ corresponds to a vertex on $\{0,1\}^d$, and the global minimizer x_0 corresponds to the marked vertex w.

Complexity analysis

In this section, we present a theoretical argument demonstrating that QHD can solve the continuous optimization problem F using $\tilde{O}(\sqrt{N})$ queries to the black-box function f. It is important to note that the argument provided below is a proof sketch; a rigorous mathematical proof for this quadratic speedup will be presented in future work.

To solve this continuous optimization problem with smooth objective F, we consider the following QHD Hamiltonian (for simplicity, we do not introduce time-dependent damping parameters):

$$\hat{H} = -h\nabla^2 + F(x) = \hat{H}_0 + g(x), \tag{3.46}$$

where $\hat{H}_0 = -h\nabla^2 + \left(\sum_{k=1}^{d} ax_k^4 - cx_k^2\right)$ and $h > 0$ will be specified soon.

To understand the spectrum of $\hat{H}_0$, we first consider the 1D case. When $d = 1$, $\hat{H}_0$ is the quantum Hamiltonian with a symmetric double-well potential function, and it is well known that the first two eigenstates of the operator $\hat{H}_0$ are:

$$|\Phi_0\rangle = \frac{1}{\sqrt{2}}\left(\left|\Phi^-\right\rangle + \left|\Phi^+\right\rangle\right), \tag{3.47}$$

$$|\Phi_1\rangle = \frac{1}{\sqrt{2}}\left(\left|\Phi^-\right\rangle + \left|\Phi^+\right\rangle\right), \tag{3.48}$$

where $|\Phi^{\pm}\rangle$ are two wave function functions clustered near the left or right well, respectively, as illustrated in Figure 3.2. Moreover, for sufficiently small $h > 0$, the wave function $|\Phi^{+}\rangle$ (or $|\Phi^{-}\rangle$) are approximately zero outside of a small area near x^{+} (or x^{-}), i.e., $\langle\Phi^{+}|\Phi^{-}\rangle = 0$ for small $h > 0$.

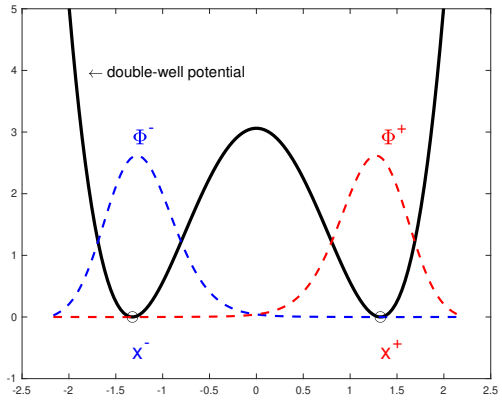


Figure 3.2: The first two eigenstates of 1D symmetric double well.

Suppose the first two energy levels of $\hat{H}_0$ are E_0 and E_1, i.e., $\hat{H}_0 |\Phi^{-}\rangle = E_0 |\Phi^{-}\rangle$, $\hat{H}_0 |\Phi^{-}\rangle = E_1 |\Phi^{-}\rangle$. By choosing appropriate a, c (and possibly adding a global phase $C \in \mathbb{R}$ to $\hat{H}_0$), we may assume $E_0 = -\gamma, E_1 = \gamma$ for some $\gamma > 0$. Denote $|0\rangle \coloneqq |\Phi^{-}\rangle$ and $|1\rangle \coloneqq |\Phi^{+}\rangle$, and define the two-dimensional subspace $\mathcal{S}$ spanned by $|\Phi^{-}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and $|\Phi^{+}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, we can express the low-energy part of $\hat{H}_0$ using the Pauli-X operator:

$$\hat{H}_0|_{\mathcal{S}} = -\gamma |+\rangle \langle +| + \gamma |-\rangle \langle -| = -\gamma\sigma_x.$$

Now, we consider the low-energy part of $\hat{H}_0$ in the general d-dimensional case. Note that

$\hat{H}_0$ is separable:

$$\hat{H}_0 = \sum_{k=1}^{d} \left(-h \frac{\partial^2}{\partial x_k^2} + a x_k^4 - c x_k^2 \right),$$

so the low-energy subspace of $\hat{H}_0$ is spanned by basis functions of the form:

$$\Phi_b = \Phi^{s_1}(x_1)\Phi^{s_2}(x_2)\ldots\Phi^{s_d}(x_d), \tag{3.49}$$

where a length-d bitstring b is used to enumerate the basis functions:

$$s_k = \begin{cases} - & (b_k = 0) \\ \\ + & (b_k = 1). \end{cases}$$

Clearly, for small $h > 0$, each basis function Φ_b can be regarded to be compactly supported within a small neighborhood of the point x_b, where the k-th coordinate of x_b is x^{s_k}. Each basis function $|\Phi_b\rangle$ can be identified with the computational basis $|b\rangle$ (and thus representing a vertex on the d-dimensional hypercube). The subspace spanned by all the 2^d basis is denoted as $\mathcal{H}$. Effectively, $\mathcal{H}$ can be regarded as the Hilbert space associated to d qubits, and

$$\hat{H}_0|_{\mathcal{H}} = -\gamma \sum_{j=1}^{d} \sigma_x^{(j)}. \tag{3.50}$$

Meanwhile, by choosing a small radius r and appropriate constant B, we have that (recall

that w is the marked vertex)

$$\langle b'|g|b\rangle = \begin{cases} -1 & (b = b' = w) \\ 0 & (\text{otherwise}) \end{cases}$$

It follows that $g|_{\mathcal{H}} = -|w\rangle\langle w|$, and the full QHD Hamiltonian has the low-energy part:

$$\hat{H}|_{\mathcal{H}} = -\gamma A - |w\rangle\langle w|, \tag{3.51}$$

where $A = \sum_{j=1}^{d} \sigma_x^{(j)}$ is exactly the adjacency matrix of the d-dimensional hypercube.

In the QHD Hamiltonian $\hat{H} = \hat{H}_0 + g$, the function g is small compared to $\hat{H}_0$. Therefore, g can be regarded as a perturbation term, and the spectrum of $\hat{H}$ is dominated by that of $\hat{H}_0$. The double-well potential function induces a huge energy gap between the subspace $\mathcal{H}$ and the higher-energy subspace, so if we start with an initial state $|\psi_0\rangle \in \mathcal{H}$, the QHD evolution will stay within the low-energy subspace $\mathcal{H}$. This heuristic argument based on perturbation theory can be made rigorous by (1) truncating the unbounded Hamiltonian $\hat{H}$ using a finite-dimensional Hilbert space spanned by the low-energy eigenstates of $\hat{H}_0$, and (2) utilizing the Schrieffer–Wolff transformation for the finite-dimensional truncated Hamiltonian to show that the effective quantum evolution is close to the evolution generated by $\hat{H}|_{\mathcal{H}}$, as defined in (3.51).

Now, according to [75, Section III.B], we may choose the parameter $\gamma = \frac{2}{d} + O(d^{-2})$ and the initial state $|\psi_0\rangle = |+\rangle^{\otimes d}$, and the probability of finding $|w\rangle$ will be of order 1 after an evolution time $T = O(\sqrt{N})$. Given this evolution time, the real-space quantum evolution given in (3.46) can be simulated using $\tilde{O}(\sqrt{N})$ quantum queries to the function F [76]. Note that the

quantum oracle of F can be constructed using $O(1)$ queries to the oracle f. It turns out that QHD can solve the spatial search problem on the hypercube with query complexity $\tilde{O}(\sqrt{N})$.

3.4 Super-polynomial quantum-classical performance separation

In this section, we identify a family of nonconvex continuous optimization instances, each d-dimensional instance with 2^d local minima, to demonstrate a quantum-classical performance separation. We prove that Quantum Hamiltonian Descent can solve any d-dimensional instance from this family using $\widetilde{\mathcal{O}}(d^3)$[3] quantum queries to the function value and $\widetilde{\mathcal{O}}(d^4)$ additional 1-qubit and 2-qubit elementary quantum gates. On the other side, a comprehensive empirical study suggests that representative state-of-the-art classical optimization algorithms/solvers (including Gurobi) would require a super-polynomial time to solve such optimization instances. This result constitutes new evidence of a significant performance separation between quantum and classical algorithms in nonconvex optimization.

3.4.1 Problem formulation

We construct a family of optimization instances representing nonconvex optimization problems with sophisticated landscapes. These hard problem instances are constructed as follows. First, we consider certain *well-formed* one-dimensional "double well" functions with two local minima (see Figure 3.3A).

Definition 1 (Gapped 1D potential)**.** *Given a 1-dimensional twice differentiable function* $w\colon \mathbb{R} \to \mathbb{R}$ *such that* w *is bounded from below and diverges to infinity as* $|x| \to \infty$*. For any* $\lambda > 0$*, let*

[3] $\widetilde{\mathcal{O}}(\cdot)$ suppresses poly-logarithmic factors in d and $1/\delta$ where δ is the precision.

$\Delta(\lambda)$ *denote the spectral gap of the Hamiltonian operator (C.1). We say this function* w *is* ***gapped*** *if the spectral gap is uniformly bounded by a positive constant* $\Delta_{\min} > 0$*, i.e.,*

$$\forall \lambda > 0 \colon \Delta(\lambda) \geq \Delta_{\min} > 0. \tag{3.52}$$

Definition 2. *We call a non-negative function* $w \colon \mathbb{R} \to \mathbb{R}$ *a* ***well-formed*** *asymmetric double well if* w *satisfies the following conditions:*

1. w *has at least 2 local minima and a unique non-degenerate global minimum at* $x = x^*$*, i.e.,* $w'(x^*) = 0$ *and* $w''(x^*) \geq \gamma^2 > 0$*;*

2. *There exist positive numbers* $C > c > 0$ *such that* $c|x - x^*|^4 \leq w(x) \leq C(1 + |x - x^*|^4)$ *for all* $x \in \mathbb{R}$*;*

3. $w \in C^\infty$ *and* w *is* ρ*-smooth, i.e.,* $\|w''\| \leq \rho$*;*

4. w *is a gapped function in the sense of Definition 1.*

These well-formed double well functions can be instantiated as degree-4 polynomials,[4] as detailed in Appendix C.1. Suppose that $w(x)$ is a well-formed 1D double well function with global minimizer at $x = x^*$, we then define a d-dimensional separable function $F(\boldsymbol{x}) \coloneqq \sum_{k=1}^{d} w(x_k)$. We note that the function $F(\boldsymbol{x})$ has 2^d local minima and a unique global minimizer at $\boldsymbol{x}^* = (x^*, \ldots, x^*)$. In Figure 3.3B, we show a two-dimensional separable function with 4 local minima.

[4]Precisely speaking, a degree-4 polynomial is not ρ-smooth, as its Hessian is a quadratic function that diverges at infinity. Thanks to the localization of wave function in the quantum evolution (because w grows as $|x|^4$), we may restrict the evolution to a compact domain in quantum simulation. In this case, the double-well potential function is effectively ρ-smooth.

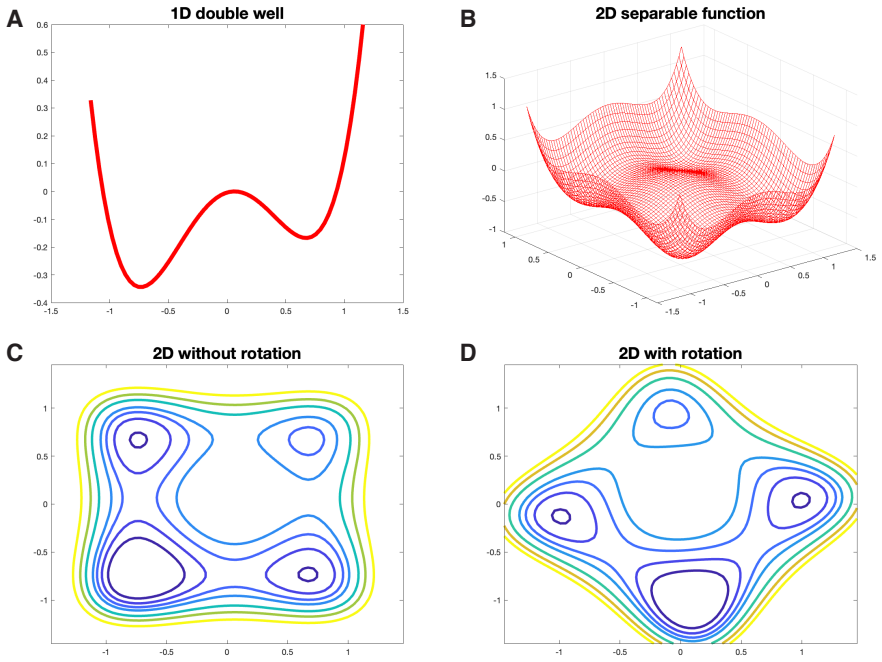


Figure 3.3: **Construction of optimization instances. A.** A well-formed double well function. **B.** A separable function built from two identical well-formed double well functions. **C.** Contour plot of the 2-dimensional separable function $F(\boldsymbol{x})$. **D.** Contour plot of the rotated 2-dimensional function $F_U(\boldsymbol{x})$. The local geometry of $F(\boldsymbol{x})$ is preserved, while the new function $F_U(\boldsymbol{x})$ is not separable.

Although the function $F(\boldsymbol{x})$ has exponentially many local minima, it is separable in each independent variable x_k. The separability structure of $F(\boldsymbol{x})$ can be easily detected from its closed-form expression. Even if the closed-form expression is not available and the function is given via a black-box query model, a simple classical algorithm such as coordinate descent can split the problem into d independent sub-problems, each can be solved in constant time.

To create hard optimization instances for classical algorithms, we apply a random rotation to *hide* the separability structure in $F(\boldsymbol{x})$. More precisely, let U be an orthogonal matrix and we define a new function,

$$F_U(\boldsymbol{x}) \coloneqq F(U\boldsymbol{x}). \tag{3.53}$$

The newly generated function $F_U(\boldsymbol{x})$ still has 2^d local minima as the rotation preserves the geometry of the optimization landscape. However, unlike the original function $F(\boldsymbol{x})$, the rotated function $F_U(\boldsymbol{x})$ is no longer separable (see Figure 3.3C, D).

More importantly, the random rotations on nonconvex functions are difficult to revert. For example, if the double-well function $w(x)$ is a degree-4 polynomial, so is the rotated function $F_U(\boldsymbol{x})$ because the rotation by U is an affine transformation. However, to learn the rotation U from the closed-form expression of $F_U(\boldsymbol{x})$ amounts to diagonalizing a 4-tensor. Diagonalizing general 4-tensors is known to be NP-hard [77]. Therefore, it does not appear to be a trivial task for classical algorithms to revert the rotation and decompose $F_U(\boldsymbol{x})$ into d separate sub-problems.

Definition 3 (Nonconvex optimization instances)**.** *Given an instance of a well-formed asymmetric double well $w(x)$, we define a class of optimization instances $\mathscr{C}(w, d)$ for dimension $d \geq 1$:*

$$\mathscr{C}(w, d) \coloneqq \{F_U(\boldsymbol{x}) : U \in \mathcal{Q}(d)\}, \tag{3.54}$$

where $F_U(\boldsymbol{x})$ is defined in (3.53) and $\mathcal{Q}(d)$ is the set of all d-by-d orthogonal matrix:

$$\mathcal{Q}(d) \coloneqq \{U \in \mathbb{R}^{d\times d} : UU^\top = I\}.$$

3.4.2 Main results

Due to our construction of the optimization instances, any instance $f \in \mathscr{C}(w, d)$ has 2^d local minima but a unique global minimum $f(\boldsymbol{x}^*)$. We say $\boldsymbol{x} \in \mathbb{R}^d$ is an δ-*approximate* solution to the optimization problem f if $\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \delta$. Our main theoretical contribution is to prove that QHD can find a δ-approximate solution to any $f \in \mathscr{C}(w, d)$ in polynomial time.

We assume our quantum algorithm only has access to the *quantum evaluation oracle* (i.e., zeroth-order oracle), which is defined as a unitary map O_f on $\mathbb{R}^d \otimes \mathbb{R}$ such that for any $|\boldsymbol{x}\rangle \in \mathbb{R}^d$,

$$O_f\left(|\boldsymbol{x}\rangle \otimes |0\rangle\right) = |\boldsymbol{x}\rangle \otimes |f(\boldsymbol{x})\rangle \tag{3.55}$$

Note that the quantum evaluation oracle can be coherently accessed. Namely, for any $m \in \mathbb{N}$, let $|\boldsymbol{x}_1\rangle, \ldots, |\boldsymbol{x}_d\rangle \in \mathbb{R}^d$ and $\mathbf{c} \in \mathbb{C}^m$ such that $\|\mathbf{c}\| = 1$, we have

$$O_f\left(\sum_{j=1}^{m} c_j |\boldsymbol{x}_j\rangle \otimes |0\rangle\right) = \sum_{j=1}^{m} c_j |\boldsymbol{x}_j\rangle \otimes |f(\boldsymbol{x}_j)\rangle . \tag{3.56}$$

Given an instance $f \in \mathscr{C}(w, d)$, we consider a Quantum Hamiltonian Descent algorithm with parameters δ, λ_f, η and η.

Algorithm 1: Quantum Hamiltonian Descent

1 Specify positive parameters δ, λ_f, ϵ, and η. Let $t_f = \log(\lambda_f)$ and $\lambda(t) = e^{2t - t_f}$. For $0 \le t \le t_f$, we define the QHD Hamiltonian over Ω,

$$H(t) = \frac{1}{\lambda(t)}\left(-\frac{1}{2}\nabla^2\right) + \lambda(t) f.$$

2 Prepare the ground state of $H(0)$, denoted by $|\Phi(0)\rangle$.

3 Use $|\Phi(0)\rangle$ as the initial state and simulate the quantum dynamics governed by the Schrödinger equation,

$$i\epsilon \frac{\mathrm{d}}{\mathrm{d}t} |\Psi^\epsilon(t)\rangle = H(t) |\Psi^\epsilon(t)\rangle , \tag{3.57}$$

up to an additive error η, i.e., a final state $|\Psi^\epsilon_{\mathrm{sim}}(t_f)\rangle$ is obtained such that $\|\Psi^\epsilon_{\mathrm{sim}}(t_f) - \Psi^\epsilon(t_f)\| \le \eta$.

4 Measure the final state $|\Psi^\epsilon_{\mathrm{sim}}(t_f)\rangle$ using the position quadrature $\hat{\boldsymbol{x}}$ and return a d-dimensional vector $\boldsymbol{x}$.

Our main theoretical result is summarized in the following theorem, where the $\widetilde{\mathcal{O}}$ notation suppresses poly-logarithmic factors in d and $1/\delta$.

Theorem 5. *Given a well-formed asymmetric double well w (see Definition 2) and a fixed integer d, let $f \in \mathscr{C}(w, d)$ be an optimization instance. For a sufficiently small $\delta > 0$, we specify the choices of parameters:*

$$\lambda_f \coloneqq \frac{\beta(2d + 6\log(3))}{\delta^2},\ \epsilon \coloneqq \frac{1}{18d\theta\log(\lambda_f)},\ \eta \coloneqq \frac{1}{18}, \tag{3.58}$$

where β and θ are constants that only depend on w. Then, Algorithm 1 produces a δ-approximate global solution to f using

$$\mathcal{O}\left(\frac{d^3}{\delta^2} \cdot \frac{\log^2(\frac{d^2}{\delta^2})}{\log\log(\frac{d^2}{\delta^2})}\right)$$

queries to O_f and additional

$$\mathcal{O}\left(\frac{d^3}{\delta^2}\left(d + \log^{2.5}\left(\frac{d}{\delta^2}\right)\right)\frac{\log^2(\frac{d^2}{\delta^2})}{\log\log(\frac{d^2}{\delta^2})}\right)$$

1- and 2-qubit gates.

In what follows, we give a sketch of the proof of Theorem 5, while some technical details are provided in Appendix C. The main idea is that the Laplacian operator in the QHD Hamiltonian is rotationally invariant. As a consequence, the spectral gap of the QHD Hamiltonian, given a rotated $F_U(\boldsymbol{x})$, remains the same as that of a 1-dimensional, well-formed asymmetric double well. In other words, Quantum Hamiltonian Descent effectively solves the rotated problem $f \in \mathscr{C}(w, d)$ as if solving d separable 1-dimensional problems.

We define $B_\delta \coloneqq \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \delta\}$, i.e., the δ-ball entered at $\boldsymbol{x}^*$. Let $1_{B_\delta}(\boldsymbol{x})$ be

the indicator function of the set B_δ, i.e.,

$$1_{B_\delta}(\boldsymbol{x}) = \begin{cases} 1 & (\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \delta) \\ 0 & (\text{otherwise}). \end{cases} \tag{3.59}$$

Then, for a quantum particle described by a quantum state $|\Psi\rangle$, its position is a random variable $X \in \mathbb{R}^d$ such that $X \sim |\Psi|^2$. The success probability can be evaluated through the expectation,

$$\Pr_{X \sim |\Psi|^2}[\|X - \boldsymbol{x}^*\| \leq \delta] = \langle \Psi | \, 1_{B_\delta}(\boldsymbol{x}) \, | \Psi \rangle = \int_{\mathbb{R}^d} 1_{B_\delta}(x) |\Psi(\boldsymbol{x})|^2 \, \mathrm{d}\boldsymbol{x}. \tag{3.60}$$

Proof. Let $|\Phi(t_f)\rangle$ be the ground state of the final Hamiltonian $H(t_f) = \frac{1}{\lambda_f}\left(-\frac{1}{2}\nabla^2\right) + \lambda_f f$. The modulus square of the quantum state, $|\Phi(t_f)|^2$, gives a probability density on $\mathbb{R}^d$. We denote $X \in \mathbb{R}^d$ as a random vector following the distribution $|\Phi(t_f)|^2$. Due to our construction of the optimization instances, it is clear that $Y \coloneqq U^\top(X - \boldsymbol{x}^*)$ is a d-dimensional random vector with independent coordinates. Moreover, by Lemma 14,

$$\mathbb{E}\left[e^{r^\top Y}\right] \leq e^{\|r\|^2 \sigma^2 / 2} \tag{3.61}$$

for all $r \in \mathbb{R}^d$, where $\sigma^2 = \beta/\lambda$ and β is a constant that only depends on w. Then, by Theorem 10, we have that

$$\Pr\left[\|Y\|^2 > d\sigma^2 + 2\sqrt{cd\sigma^4} + 2\sigma^2 c\right] \leq e^{-c}. \tag{3.62}$$

We choose $c = 2\log(3)$ (so $e^{-c} = 1/9$) so that

$$d\sigma^2 + 2\sqrt{cd\sigma^4} + 2\sigma^2 c \leq \frac{\beta(2d + 6\log(3))}{\lambda_f}.$$

If specify the parameter λ_f as in Theorem 5, it follows that

$$\Pr[\|X_1 - \boldsymbol{x}^*\| > \delta] = \Pr[\|Y\| > \delta] \leq \frac{1}{9}. \tag{3.63}$$

Let $|\Psi^\epsilon(t_f)\rangle$ be the final state of the dynamics (3.57). If we choose ϵ as in Theorem 5, by Proposition 4, we have that

$$\|\Psi^\epsilon(t_f) - \tilde{\Phi}(t_f)\| \leq \frac{1}{18}, \tag{3.64}$$

where $\tilde{\Phi}(t_f)$ only differs from the ground state $|\Phi(t_f)\rangle$ by a global phase so $|\Phi(t_f)|^2$ and $|\tilde{\Phi}(t_f)|^2$ give exactly the same probability distribution.

Now, if we simulate the quantum dynamics in Algorithm 1 up to an additive error $\eta = 1/18$, by the triangle inequality and (3.64), we have that

$$\|\Psi^\epsilon_{\text{sim}}(t_f) - \tilde{\Phi}(t_f)\| \leq \frac{1}{9}. \tag{3.65}$$

Then, we deduce from Lemma 18 that

$$\left|\langle\Psi^\epsilon_{\text{sim}}(t_f)|\, 1_{B_\delta}\, |\Psi^\epsilon_{\text{sim}}(t_f)\rangle - \left\langle\tilde{\Phi}(t_f)\right|\, 1_{B_\delta}\, \left|\tilde{\Phi}(t_f)\right\rangle\right| \leq \frac{2}{9}. \tag{3.66}$$

Since $\tilde{\Phi}(t_f)$ describes the same probability distribution as $\Phi(t_f)$, we combine (3.63) and (3.66) to yield the desired estimate,

$$\Pr_{X\sim|\Psi^{\epsilon}_{\mathrm{sim}}(t_f)|^2}\left[\|X-\boldsymbol{x}^*\|>\delta\right]\leq\frac{1}{3}. \tag{3.67}$$

It remains to figure out the overall complexity of the quantum simulation. Since the function f grows as $\Theta(|x|^4)$, we can choose a large enough constant M such that the quantum dynamics in $\mathbb{R}^d$ can be faithfully simulated over a compact domain $\Omega \coloneqq [-M, M]^d$ (with periodic boundary conditions). In what follows, we let $V(\boldsymbol{x})$ be the truncated objective function $f(\boldsymbol{x})$ on Ω.

The initial state of the quantum simulation can be prepared using the procedures in Appendix C.4. The query/gate complexity of the initial state preparation is at most $\mathcal{O}(d)$.

By Lemma 16, simulating the quantum simulation in Algorithm 1 is equivalent to simulating the effective quantum dynamics (C.37). We use the quantum algorithm in Theorem 11 to simulate (C.37). For $s \in J$, the time-dependent potential is $V(s, x) = \varphi(s)V(x)$, so we have

$$\|V\|_{\infty,1}=\left(\int_{1/\epsilon\lambda_f}^{\lambda_f/\epsilon}\varphi(s)\,\mathrm{d}x\right)\|V\|_\infty\leq\frac{\lambda_f}{\epsilon}\|V\|_\infty=27\theta\|V\|_\infty d\lambda_f\log(\lambda_f)\leq\mathcal{O}\left(\frac{d^3}{\delta^2}\right). \tag{3.68}$$

Note that $\|V\|_\infty \leq CdM^4$ for an absolute constant $C > 0$. Similarly, we can prove that the time-dependent potential function $V(x, s)$ is Lipschitz in s and the Lipschitz constant is

$$L=\max_{s\in J}\|\dot{V}\|_\infty\leq 2\frac{\lambda_f^3}{27d\theta\lambda(\lambda_f)}\|V\|_\infty\leq\mathcal{O}\left(\frac{d^3}{\delta^6}\right). \tag{3.69}$$

Therefore, by Theorem 11, the quantum simulation task can be implemented with

$$\mathcal{O}\left(\frac{d^3}{\delta^2}\cdot\frac{\log^2(\frac{d^3}{\delta^2})}{\log\log(\frac{d^3}{\delta^2})}\right) \tag{3.70}$$

queries to the quantum evaluation oracle O_f and additional

$$\mathcal{O}\left(\frac{d^3}{\delta^2}\left(d+\log^{2.5}\left(\frac{d^3}{\delta^4}\right)\right)\frac{\log^2(\frac{d^3}{\delta^2})}{\log\log(\frac{d^3}{\delta^2})}\right) \tag{3.71}$$

1- and 2-qubit gates. □

3.4.3 Empirical study of classical algorithms

In this section, we perform an empirical study to evaluate the performance of classical optimization algorithms for nonconvex optimization instances. We consider 6 state-of-the-art classical optimization algorithms (covering four major categories in the optimization literature):

1. Annealing-based and Monte-Carlo algorithms: dual annealing [78], basin-hopping [79];
2. Gradient method: stochastic (or perturbed) gradient descent;
3. Newton and quasi-Newton methods: interior-point method (Ipopt [80]), sequential quadratic programming [1];
4. Branch-and-bound algorithm: Gurobi [81].

Methodology

The problem instances are introduced in Section 3.4.1. For a given positive integer d and a d-by-d orthogonal matrix U, we define a problem instance: $\min F_U(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbb{R}^d$, $F_U(\boldsymbol{x}) \coloneqq F(U\boldsymbol{x})$ and $F(\boldsymbol{x}) = F(x_1, \ldots, x_d) \coloneqq \sum_{k=1}^{d} w(x_k)$. Here, we choose $w(x) = x^4 - \left(x - \frac{1}{32}\right)^2 - c$, with the constant c introduced to ensure that $\min_{\boldsymbol{x}} F_U(\boldsymbol{x}) = 0$.

We use *Time-to-solution (TTS)*, a standard performance metric for randomized optimization algorithms [82], to evaluate the performance of classical optimization algorithms/solvers. Suppose that a randomized algorithm solves a problem instance with success probability p and average runtime t (over its internal randomness), its TTS with success probability p_0 is defined as

$$T_{p_0} \coloneqq Rt, \quad \text{where } R \coloneqq \left\lceil \frac{\ln(1-p_0)}{\ln(1-p)} \right\rceil. \tag{3.72}$$

Intuitively, T_{p_0} is the expected time spent for R consecutive runs of the algorithm on a problem instance to guarantee that the overall success probability is at least p_0. In the literature, TTS is often evaluated for a fixed problem instance. In our case, however, the optimization instances are also randomly generated. To reflect the average performance of algorithms when applied to a class of randomly generated optimization instances, we introduce the notion of *average TTS*. Suppose that $\mathcal{A}(\theta)$ is an algorithm (parameterized by some adjustable parameter θ) and f are problem instances randomly drawn from a set $\mathscr{C}$. We define the success probability $p(\theta)$ and

average runtime $t(\theta)$ as follows:

$$p(\theta) \coloneqq \Pr[\mathcal{A}(\theta) \text{ solves a problem instance } f], \tag{3.73}$$

$$t(\theta) \coloneqq \mathbb{E}[\text{the time required by } \mathcal{A}(\theta) \text{ to halt}]. \tag{3.74}$$

Similar to the standard definition, we define the average TTS of $\mathcal{A}(\theta)$:

$$T_{p_0}(\theta) \coloneqq \left\lceil \frac{\ln(1-p_0)}{\ln(1-p(\theta))} \right\rceil \cdot t(\theta). \tag{3.75}$$

In practice, $p(\theta)$ can be estimated by counting the successful events out of a large number of trials, each with a different optimization instance. The average runtime t will be estimated similarly. In what follows, for simplicity, we will refer to "average TTS" as TTS if not stated otherwise.

For an optimization algorithm $\mathcal{A}(\theta)$, we can fine-tune the adjustable parameter θ (e.g., number of iterations, learning rate, etc.) to yield better performance. It is therefore challenging to justify which adjustable parameter θ we should choose in the estimation of TTS. To address this issue, we adopt the same approach proposed by Rønnow et al. [82] in their study comparing quantum and classical algorithms. We regard an algorithm parameterized by θ as an algorithm family and define its TTS (for a given dimensionality d) by the minimal TTS in the family over all possible θ. Admittedly, we can only try a finite number of θ in reality. Therefore, we pick the parameter range such that it is more likely to cover the optimal choice(s).

We conduct all of our experiments below on the Zaratan cluster [83] if not stated otherwise. For a fixed dimensionality d and parameter θ, we run an optimization algorithm multiple times on instances F_U with Haar-random U. We regard an algorithm successfully solves a problem

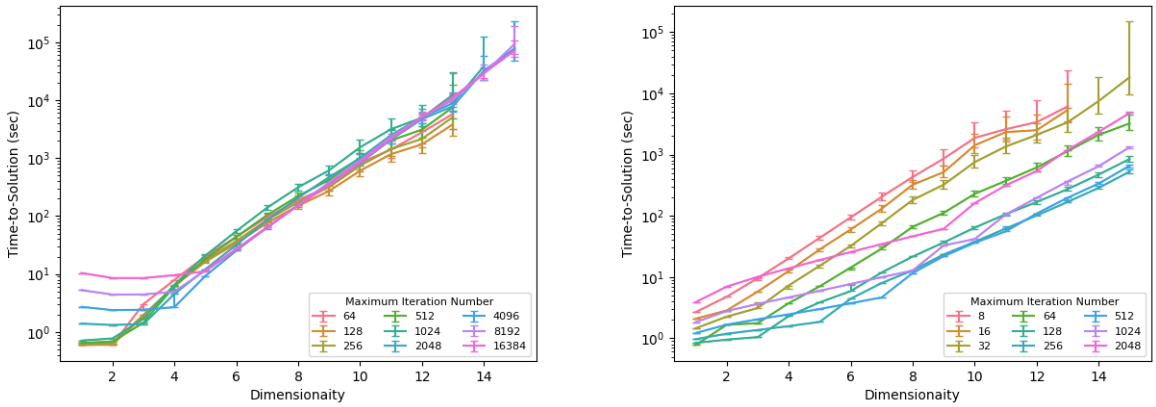
instance "minimize $F_U(\boldsymbol{x})$" if it returns a solution $\boldsymbol{y}$ such that $F_U(\boldsymbol{y}) \leq C/2$, where $C \approx 0.088$ is the (globally) second lowest local minima of $F_U(\boldsymbol{x})$. The success probability (i.e., p) of an algorithm $\mathcal{A}(\theta)$ is evaluated as the fraction of successful events out of a large number of tested instances drawn from $\mathcal{C}(w, d)$. We will not report corresponding TTS if less than 5 success runs are detected since in this case the corresponding estimation of p is likely unreliable. Each run is single-threaded and assigned one CPU core. Runtime for estimating t is measured by CPU core time used but not wall time elapsed. We choose the overall success probability threshold $p_0 = 0.99$.

Performance of classical solvers

Figure 3.4 shows the scaling of two stochastic algorithms, **dual annealing** [78] and **basin-hopping** [79], using their SciPy [84] implementation in Python. Each curve in a plot represents the TTS (parameterized by a specific θ) as a function of the dimension$d$. Note that the "true" TTS of an algorithm is the lower envelope of all the TTS curves, each parameterized by a different θ. It is clear that TTS for both algorithms scale exponentially in the problem dimension d.

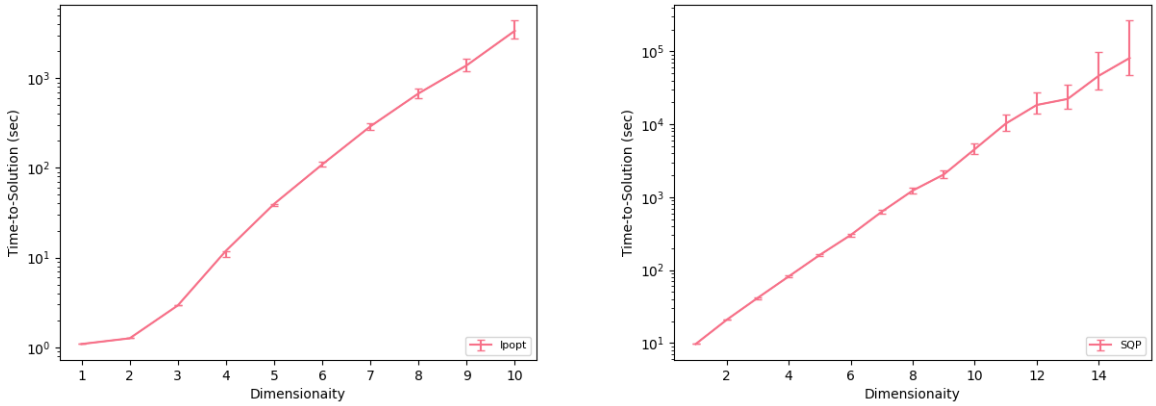
Figure 3.5 shows the scaling of the TTS of the two algorithms based on Newton and quasi-Newton methods, **Ipopt** [80] and **SQP (sequential quadratic programming)** [1] in its SciPy [84] implementation. We use the default parameter setup in these solvers, as we do not find material adjustable parameters in these algorithms. Both curves suggest super-polynomial scaling problem dimension d.

Figure 3.6 shows the scaling of **stochastic gradient descent (SGD)** for two types of learning rate schedules: (1) constant learning rate; and (2) the learning rate schedules that correspond



(a) Dual annealing (b) Basin-hopping

Figure 3.4: TTS scaling of dual annealing and basin-hopping. Confidence intervals are calculated at the 95% level. Results suggest that TTS for both algorithms (lower envelopes of curves) scale exponentially in dimensionality d.



(a) Ipopt (b) SQP

Figure 3.5: TTS scaling of Ipopt and SQP. Confidence intervals are calculated at the 95% level. Both scale exponentially in dimensionality d.

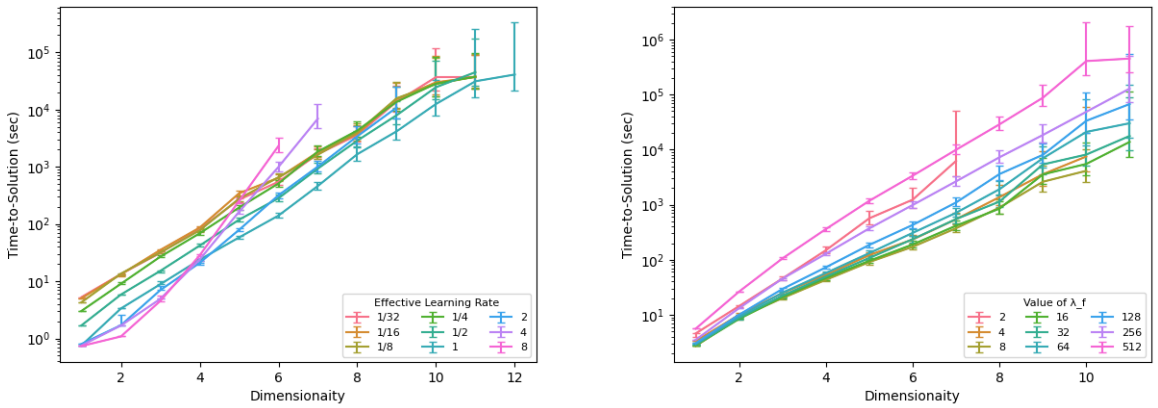
to the time-dependent functions in QHD. As suggested by Liu et al [85], the role of the learning rate s in SGD is similar to the *quantum learning rate* h in the Schrödinger equation,

$$i\frac{\mathrm{d}}{\mathrm{d}t}\Psi = \left(h^2\left(-\frac{1}{2}\nabla^2\right) + f(x)\right)\Psi. \tag{3.76}$$

If we allow the parameter h to be time-dependent, (3.76) turns out to be nothing but QHD in disguise: let $t^\star := 1/2\lambda(t)$ in (3.57) from Algorithm 1. (3.57) becomes

$$i\epsilon \frac{\mathrm{d}}{\mathrm{d}t^\star} \left|\Psi^\epsilon(t^\star)\right\rangle = H(t^\star) \left|\Psi^\epsilon(t^\star)\right\rangle, \quad \text{where } H(t^\star) = \frac{1}{(2t^\star)^2}\left(-\frac{1}{2}\nabla^2\right) + f(x). \tag{3.77}$$

And the range of $t^\star$ is $[1/2\lambda_f, \lambda_f/2]$. The quantum learning rate is then $s = 1/2t^\star$. Translating back to SGD we get the following learning rate schedule $s_{\lambda_f} : [0, \lambda_f/2 - 1/2\lambda_f] \to \mathbb{R}_+$, $s_{\lambda_f}(t) = \frac{1}{2\left(t+1/2\lambda_f\right)}$. Leveraging this connection between SGD and QHD, we set up the second learning rate schedule (i.e., "QHD-type learning rate") in our numerical experiment.



(a) Fixed learning rate

(b) QHD-type learning rate

Figure 3.6: TTS scaling of SGD under two types of learning rate schedules respectively. Confidence intervals are calculated at the 95% level. It is clear that TTS in both settings (lower envelopes of curves) scale exponentially in dimensionality d. In some curves, the data points are missing because we do not report the TTS if less than 5 success runs are detected.

We also experiment to test the performance of **Gurobi** [81], an industrial-level optimization software based on the branch-and-bound algorithm. Note that we need to reformulate our optimization instances (represented by degree-4 polynomials) as Quadratically Constrained Quadratic Program (QCQP) because Gurobi only accepts quadratic objectives. Figure 3.7 shows

the TTS scaling of Gurobi. In total, 10 TTS curves are depicted. The runtime is measured using the *work* metric, introduced by Gurobi, which is proportional to the CPU time spent.[5] The TTS of Gurobi essentially scales exponentially in dimensionality d.

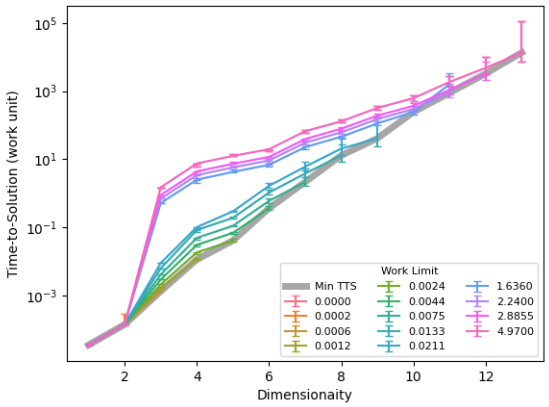


Figure 3.7: TTS scaling of Gurobi which is roughly exponential in dimensionality d. Confidence intervals are calculated at the 95% level. 1 work unit is proportional to 1 second of CPU time. In some curves, the data points are missing because we do not report the TTS if less than 5 success runs are detected.

3.4.4 Interpretation of the results

Our empirical study covers a wide spectrum of classical state-of-the-art optimization algorithms, where the numerical results suggest the run time of all tested classical algorithms scales super-polynomially in dimension d. This empirical evidence implies that QHD could potentially replace the role of these classical solvers and become one competitive *off-the-shelf* optimization solution in practice. On the contrary, our theoretical result implies that the time-to-solution of QHD for the same class of nonconvex optimization instances scales polynomially in dimension d.

[5] In our setting, a rough estimate suggests that 1 work unit corresponds to 10–100 seconds of CPU time.

However, our empirical study does not exclude the possibility of efficient classical algorithms specially designed for our tested problem instances. We expect any such efficient classical algorithm, if exists, would, nevertheless, require a special design to leverage the hidden separable structure in the instances.[6] Moreover, our empirical study is by no means complete as it is bound by limited computing resources and time. In light of that, we make all of our empirical study publicly available and invite the whole community to test it further.

We note further that our construction in the empirical test is unlikely to yield any super-polynomial oracle separation directly. This is because our optimization instances are d-dimensional degree-4 polynomials whose precise form can be fully determined with a polynomial, precisely $O(d^4)$, number of queries of the function value. Thus, any bigger than $\Omega(d^4)$ classical lower bound needs to be established in the plain model, which is likely beyond the reach of known techniques. We want to emphasize that our instances, however, could be more inspiring about concrete features of optimization problems that can be efficiently solved by QHD, compared with oracle separations, if possible, obtained via some technical routes. For example, any reduction from instances efficiently solvable by Shor's algorithm to optimization ones will likely not reveal any new quantum capability, as it is essentially running Shor's algorithm with an optimization disguise.

Finally, we want to emphasize that our specific construction of the hard optimization instances, however, should not be treated as a limitation of the applicability of the QHD algorithm, but rather a limitation of our method in analyzing QHD's behavior on general optimization instances. Indeed, as demonstrated in Chapter 4, QHD could perform well on optimization in-

[6]Note that Theorem 5 can be established with more general instances than those used in the empirical test. Any specially designed classical algorithm should aim to work with the general condition and its hidden structure.

stances much beyond the specific construction used in this paper. Analyzing the QHD's performance in these instances would require investigating the spectrum of Schrödinger's operator with general potential energy functions, which we believe is an interesting but challenging problem in pure math.

Chapter 4: Digital implementation of Quantum Hamiltonian Descent

Chapter summary. In this chapter, we introduce an implementation of Quantum Hamiltonian Descent with gate-based digital (fault-tolerant) quantum computers. This implementation utilizes the product formula (i.e., operator splitting method) to simulate the time-dependent Hamiltonian evolution given by Quantum Hamiltonian Descent. We give a rigorous resource analysis (in terms of gate complexity) of this implementation for solving unconstrained continuous optimization problems. The proposed implementation method is evaluated through numerical experiments with standard test problems in the literature of numerical optimization. Empirical results show that Quantum Hamiltonian Descent exhibits advantageous performance for nonconvex optimization problems.

4.1 Computational model and problem formulation

In this section, we discuss how to implement QHD on gate-based digital quantum computers (i.e., quantum circuits) to solve an unconstrained optimization problem f with black-box query access. Under this computational model, a quantum algorithm can be expressed as a sequence of quantum gates, and measurements, with appropriate initialization of qubits. For more details on quantum circuits and digital quantum computing, we refer the readers to the seminal textbook by Nielsen and Chuang [86].

The unconstrained optimization problem is assumed to be accessed through a quantum function value evaluation (i.e., zeroth order) order oracle, as described as follows.

Problem formulation.

We consider the following (unconstrained) optimization problem,

$$\min_{x\in\mathbb{R}^d} f(x),$$

where $f(x)$ is a smooth function. We assume access to a unitary $O_f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d \times \mathbb{R}$ such that for any $x \in \mathbb{R}^d$, and $z \in \mathbb{R}$,

$$O_f |x, z\rangle \to |x, f(x) + z\rangle .$$

4.2 Implementing QHD via product formulae

In the quantum simulation literature, several algorithms have been proposed for simulating real-space quantum dynamics, including Trotter-type algorithms (product formulae) [87], truncated Taylor series (with finite difference scheme [88] or Fourier spectral method [76]), and interaction picture simulations [76]. Although methods like truncated Taylor series or interaction picture simulations achieve better asymptotic scaling in terms of error dependence, we will consider Trotter-type algorithms for implementing QHD. Quantum simulation algorithms based on product formulae usually take a simpler form and are thus more resource-efficient in practice [89]. Besides, product formulae are closely related to a family of numerical schemes known

as *symplectic integrators* [90], which are proven to be the key in the time discretization of the Bregman-Lagrangian framework [58].

The product-formula-based implementation of QHD follows from [76, Section 2.3]. It takes two steps: *spatial discretization* and *time discretization*.

Spatial discretization

We introduce a standard mesh-grid discretization of the continuous space. Suppose the objective function f is unbounded at infinity, then there must exist a large enough region $\Omega = [a, b]^d$ such that the global minimizer of f is within Ω. We consider a regular mesh grid $\mathcal{M}$ in the domain $\Omega = [a, b]^d$ and divide each edge into N cells, i.e.,

$$\mathcal{M} = \{(x_1, ..., x_d) : x_j \in \{a, a + \Delta x, ..., b - \Delta x\}\}, \tag{4.1}$$

where $\Delta x = (b - a)/N$ is the stepsize of the mesh $\mathcal{M}$. Let $N = 2^q$ for an integer q, and we assign a q-qubit register to represent a single edge: the computational basis of the quanutm register enumerates mesh coordinates $x_j \in \{a, a + \Delta x, ..., b - \Delta x\}$. The full full mesh $\mathcal{M}$ is represented by concatenating d such registers: the mesh point $\boldsymbol{x} = (x_1, ..., x_d)$ is represented by the basis state $|x_1, ..., x_d\rangle$. With this mesh-grid discretization, a quantum wave function in the real space $\Psi(x) \colon \Omega \to \mathbb{C}$ can be represented as a quantum state in the register:

$$|\psi\rangle = \sum_{\boldsymbol{x} \in \mathcal{M}} c(x_1, ..., x_d) \, |x_1, ..., x_d\rangle \, , \ \|\psi\|_2^2 = \sum_{\boldsymbol{x} \in \mathcal{M}} |c(x_1, ..., x_d)|^2 = 1. \tag{4.2}$$

Time discretization

We utilize the product formula to decompose the quantum evolution into an alternating sequence of kinetic and potential evolution. Given a Hamiltonian of the form $H = A + B$, the first-order Trotter-Suzuki formula [91] is defined as $\mathscr{T}_1(t) = e^{-itA}e^{-itB}$. Recall the QHD Hamiltonian takes the form $H(t) = e^{\varphi_t}(-\nabla^2/2) + e^{\chi_t}f$, let $U(0,T)$ be the time-evolution operator in QHD, we can approximate $U(0,T)$ by a sequence of product formulae:

$$U(T,0) \approx \mathcal{T}\prod_{j=0}^{N-1}\Big[\exp\big(-i\Delta t a_j(-\nabla^2/2)\big)\exp(-i\Delta t b_j f)\Big], \tag{4.3}$$

where $\mathcal{T}$ is the time-ordering operator, $a_j = e^{\varphi_{t_j}}$, and $b_j = e^{\chi_{t_j}}$.

Assuming the domain Ω is of periodic boundary condition, one observes that the Laplacian operator ∇^2 is diagonalized by the Fourier basis. This observation leads to a neat implementation of the exponential of the kinetic operator $(-\nabla^2/2)$ [76, Section 3.2]:

$$-\frac{1}{2}\nabla^2 \to \mathbf{F}_s\mathbf{L}\,(\mathbf{F}_s)^{-1}\,,\ \exp\big(-i\Delta t a_j(-\nabla^2/2)\big) \to \mathbf{F}_s\exp(-i\Delta t a_j\mathbf{L})\,(\mathbf{F}_s)^{-1}\,, \tag{4.4}$$

where $\mathbf{F}_s$ is the *quantum shifted Fourier transform* (QSFT)[1], and $\mathbf{L}$ is a diagonal matrix with eigenvalues of $(-\nabla^2/2)$. These eigenvalues can be computed from the frequency number of a Fourier basis function [76, Eq.(26)]. On the other hand, the operator $\exp(-i\Delta t b_j f)$ can be

[1]QSFT is a close variant of the Quantum Fourier Transform (QFT) and can be efficiently implemented on quantum computers. According to [92, Lemma 5], QSFT can be performed to an k-dimensional vector with gate complexity $\log(k)\log\log(k)$.

readily computed because f is diagonal and can be directly queried from the oracle O_f at any point $\boldsymbol{x} \in \mathcal{M}$.

We now summarize the full procedures of the product formula implementation of QHD in Algorithm 2. We also analyze the gate complexity of this algorithm, see Theorem 6.

Algorithm 2: Implementation of QHD by product formulae

1 Specify a hypercubic domain $\Omega = [a, b]^d$ where the bounds a, b are prefixed or determined by the knowledge of f. Let q be a positive integer (a pre-fixed precision parameter). We perform the regular-mesh discretization

$$\mathcal{M} = \{(x_1, ..., x_d) : x_j \in \{a, a + \Delta x, ..., b - \Delta x\}, \text{ where } \Delta x = (b - a)/2^q\}.$$

2 Initialize a dq-qubit quantum register to $|0\rangle^{\otimes dq}$, where the parameter d is the dimension of the problem. Note that all grid points in $\mathcal{M}$ are enumerated by computational basis $|x_1, ..., x_d\rangle$ of the quantum register.

3 Prepare the *initial guess state* $|\psi_0\rangle$. For instance, it can be the uniform superposition state (i.e., uniform distribution) or a Gaussian state (i.e., Gaussian distribution).

4 Specify two time-dependent parameters φ_t, χ_t, an integer R (the number of iterations), and a positive scalar $s > 0$ (the step size). The (effective) end time is $T = Rs$.

5 Inductively apply the following update rule to the quantum state for $j = 0, 1, ..., R - 1$:

$$|\psi_{j+1}\rangle = \mathbf{F}_s \exp(-isa_j\mathbf{L})\mathbf{F}_s^{-1} \exp(-isb_j\mathbf{V}) |\psi_j\rangle ,$$

where $a_j = e^{\varphi(t_j)}$, $b_j = e^{\chi(t_j)}$, $t_j = js$, $\mathbf{L}$ is defined as above, and $\mathbf{V}$ is a diagonal operator such that $\mathbf{V} |\boldsymbol{x}\rangle = f(\boldsymbol{x}) |\boldsymbol{x}\rangle$ for any $|\boldsymbol{x}\rangle \in \mathcal{M}$. We end up with a quantum state $|\psi_R\rangle$.

6 Measure the quantum state $|\psi_R\rangle$ with computational basis (i.e., position observable), and the outcome is a grid point $|\boldsymbol{x}\rangle \in \mathcal{M}$.

4.3 Complexity analysis

We show that the gate complexity of Algorithm 2 is linear in the dimension d of the problem and the number of iterations R. Our result is exponentially better than the previous quantum gradient descent algorithm [28] in terms of the iteration number R.

Theorem 6. *Let d be the dimension of the problem, q be a pre-fixed precision parameter, and R be the number of iterations. The gate complexity of Algorithm 2 is $\tilde{O}(dqR)$.*

Proof. First, we study the cost of each iteration step. Let $n = dq$ be the total number of qubits. Note that the operator $\mathbf{V}$ is diagonal in the computational basis and is given by the oracle O_f, the evolution operator $\exp(-isb_j\mathbf{V})$ can be computed by standard techniques (see for example Rule 1.6 in [93, Section 1.2]) with gate complexity $\tilde{O}(n)$. The quantum shifted Fourier transform $\mathbf{F}_s$ is performed with gate complexity $O(n\log(n))$ [92, Lemma 5], so is its inverse $\mathbf{F}_s^{-1}$. The operator $\mathbf{L}$ is diagonal in the computational basis and the diagonal entries are evaluated from a closed-form formula [76, Section 2.1]. Similar as $\exp(-isb_j\mathbf{V})$, the evolution operator $\exp(-isa_j\mathbf{L})$ is computed with gate complexity $\tilde{O}(n)$. It turns out that the gate complexity of each iteration is $\tilde{O}(n)$.

As the algorithm iterations in R steps, and each step has exactly the same gate complexity, the overall gate complexity is $\tilde{O}(dqR)$. □

4.4 Numerical experiments

In Figure 2.2, we numerically simulate the execution of Algorithm 2 for the Levy function. In this section, we provide more numerical results for Quantum Hamiltonian Descent when applied to two-dimensional nonconvex optimization problems.

Test problems

We select 22 optimization instances in the literature [94, 95, 96]. The optimization problems are split into five categories by the landscape features, see Table 4.1.

Custom Classification (no repetition)		
Features	**Count**	**Function names**
Ridges or Valleys	5	Ackley 2, Dropwave, Holder Table, Levy, Levy 13
Basin	5	Bohachevsky 2, Camel 3, Csendes, Deflected Corrugated Spring, Rosenbrock
Flat	3	Bird, Easom, Michalewicz
Studded	4	Ackley, Alpine 1, Griewank, Rastrigin
Simple	5	Alpine 2, Hosaki, Shubert, Styblinski-Tang, Sum of Squares

Table 4.1: A classification of the functions, separated by features we believe to be interesting to investigate in the comparison between QHD and alternate optimization algorithms.

The category "Ridges or Valleys" is characterized by small deep sections and/or steep walls that divide the domain. "Basin" is characterized by flatness at function values near the global optimum. "Flat" is characterized by flatness at high function values. "Studded" is characterized by a base shape with higher frequency perturbation. Local search algorithms may be confused by many local minima and high gradient magnitudes. "Simple" functions are those on which standard gradient descent algorithms work efficiently. Although the instances in the "Simple" category are not difficult for classical optimization methods, we test these problems as a sanity check on the behavior of the gradient methods and QHD. In Figure 4.1, we plot two representative instance from each of the five categories.

To unify the simulation accuracy and comparability of the functions, we truncate the objective function $f(x, y)$ over a squared domain $[a, b]^2$, and then rescale the function so it is supported on the unit square $[0, 1]^2$. The rescaled objective function is given by:

$$\tilde{f}(x, y) = \frac{1}{L} f(a + Lx, b + Ly) - f_{\min}, \tag{4.5}$$

where $L = b - a$ and $f_{\min}$ is the global minimum value of f. The global minimum of the rescaled

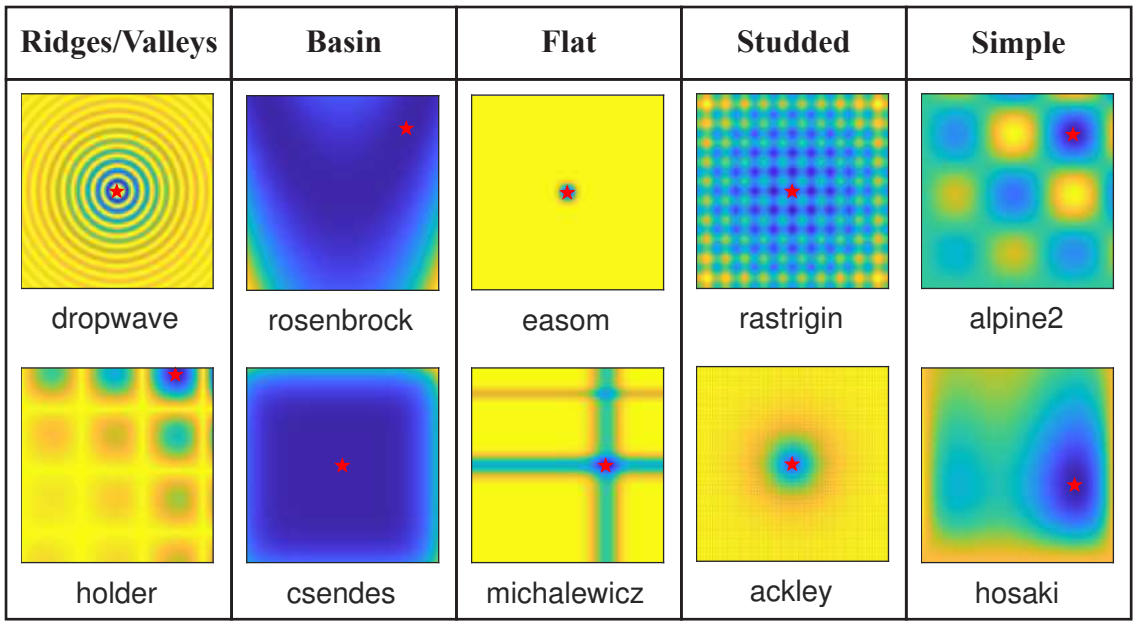


Figure 4.1: Representatives from the benchmark set of test functions. High function values shown in yellow, down to low values in blue. The unique global minimum is marked with a red star.

function $\tilde{f}$ is 0. The normalization factor $1/L$ is introduced to preserve the size of the gradient and Hessian of f. We do this to avoid affecting the performance of the (classical) gradient-based algorithms, preserving fair comparison.

Experiment setup

In our experiment, we implement SGD by adding a small Gaussian perturbation to the analytical gradient. Unlike deterministic algorithms like NAGD, this stochastic perturbation seems to help with non-convex optimization. QAA solves an optimization problem by simulating a quantum adiabatic evolution [52], and it has mostly been applied to discrete optimization in the literature. To solve continuous optimization with QAA, a common approach is to represent each continuous variable with a finite-length bitstring so the original problem is converted to combinatorial optimization defined on the hypercube $\{0,1\}^N$, where N is the total number of bits

(e.g., [97, 98]). In our experiment, we adopt the radix-2 representation (i.e., binary expansion) and assign 7 bits for each continuous variable. This allows QAA to handle the optimization instances as discrete problems over $\{0,1\}^{14}$. Effectively, this means we discretize the continuous domain $[0,1]^2$ into a 128×128 mesh grid.

Numerical results

As described above, we fix the same total evolution time $T = 10$ among all tested algorithms. The final success probability (measured at $t = 10$) for all 22 instances has been shown in panel B of Figure 2.2. We observe that QHD has a higher success probability in most optimization instances. The full success probability data of all 22 instances is shown in Figure 4.2A.

We also find that QHD is more stable than Nesterov and SGD. We use the same stepsize $s = 0.001$ for QHD and the two classical iterative methods. However, we observe that NAGD fails to converge for Ackley2 and Rastrigin functions, while QHD manages to converge for both instances. This observation suggests that our quantization approach not only improves the solution quality but also the stability of the resultant quantum dynamics in time discretization.

In Figure 4.3, we show that the performance of QAA highly depends on the spatial resolution. For many instances, we find that the high resolution (128) in QAA causes worse performance. This is because the Radix-2 representation used in QAA does not preserve the Euclidean topology in the original problem, therefore the discrete problem can become much harder for higher resolution. On the contrary, QHD does not suffer from this issue – the performance of QHD is consistent in both low and high-resolution cases.

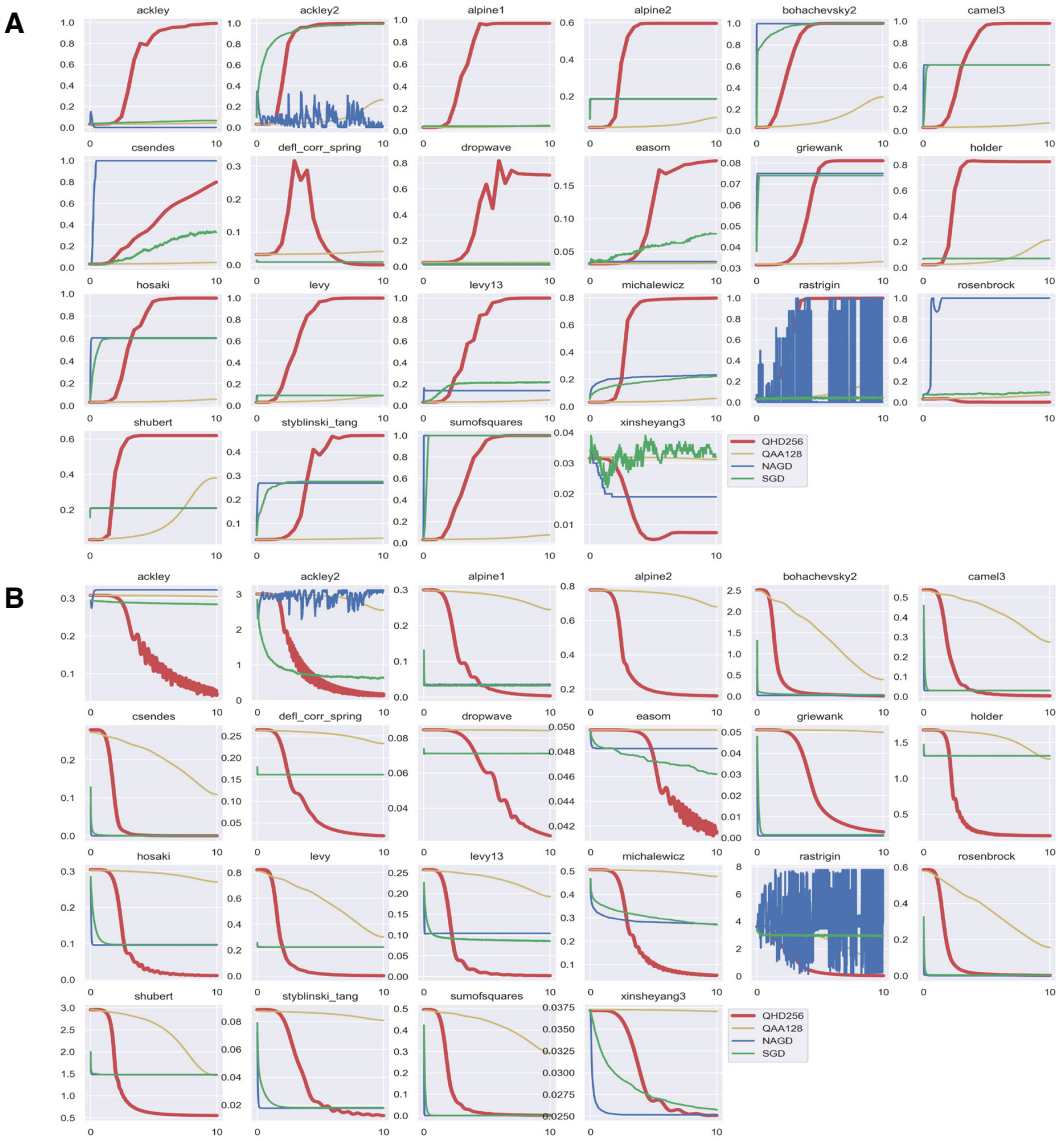


Figure 4.2: The performance of four tested algorithms for all 22 problems. **A:** success probability, **B:** loss curve (expected function value). In both panels, the x data in each subplot is the evolution time $t \in [0, 10]$.

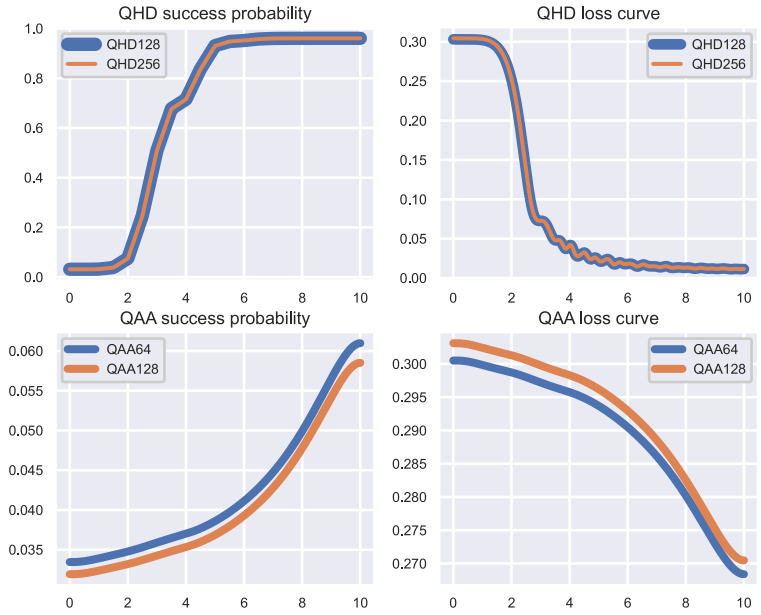


Figure 4.3: Performance of QHD and QAA in different resolutions (QHD: 128, 256; QAA: 64, 128), illustrated with Hosaki function. The performance of QHD is not affected by the resolution, while QAA does worse when the resolution is higher.

Chapter 5: Hardware-efficient quantum simulation of sparse matrices

Chapter summary. Many promising quantum applications depend on the efficient quantum simulation of an exponentially large sparse Hamiltonian, a task known as sparse Hamiltonian simulation, which is fundamentally important in quantum computation. Although several theoretically appealing quantum algorithms have been proposed for this task, they typically require a black-box query model of the sparse Hamiltonian, rendering them impractical for near-term implementation on quantum devices. In this chapter, we propose a technique named *Hamiltonian embedding*. This technique simulates a desired sparse Hamiltonian by embedding it into the evolution of a larger and more structured quantum system, allowing for more efficient simulation through hardware-efficient operations. We conduct a systematic study of this new technique and demonstrate significant savings in computational resources for implementing prominent quantum applications. As a result, we can now experimentally realize quantum walks on complicated graphs (e.g., binary trees, glued-tree graphs), quantum spatial search, and the simulation of real-space Schrödinger equations on current trapped-ion and neutral-atom platforms. Given the fundamental role of Hamiltonian evolution in the design of quantum algorithms, our technique markedly expands the horizon of implementable quantum advantages in the NISQ era.

5.1 Introduction

Quantum technology has achieved significant milestones, notably demonstrating quantum advantage over classical computers [99, 100], and the realization of error-corrected digital qubits [101]. Meanwhile, various quantum algorithms targeting practical applications have emerged, including solving large-scale linear systems [36], simulating linear and nonlinear dynamics [102, 34], and finding optimal solutions for mathematical optimization problems [40]. At the core of these quantum applications lies a fundamental subroutine named *sparse Hamiltonian simulation*, i.e., to compute the matrix function $f(A) = e^{-iAt}$ for a given sparse Hermitian matrix A. Over the past few decades, sparse Hamiltonian simulation has been a central topic in quantum computation research, and a rich variety of quantum algorithms (e.g., [103, 104, 105, 106, 107, 38]) have been developed for this task. While these algorithms achieve exponential quantum speedups over known classical ones, their implementation requires executing deep quantum circuits. The absence of fault tolerance in near-term devices hinders our ability to utilize these advanced quantum algorithms for solving classically intractable computational tasks in science and business [108].

Delving into the low-level implementation of quantum simulation algorithms reveals two prominent factors that contribute to the depth of quantum circuits. First, almost all sparse Hamiltonian simulation algorithms require a sophisticated quantum input model for the sparse matrix A, such as quantum oracles for sparse matrices [109, 103, 110], QRAM [111], or block-encodings [112, 38]. Implementing these input models usually introduces a significant gate count, even in relatively simple cases [113]. Second, in mapping a quantum algorithm to executable quantum gates, typical circuit compilation routines begin with representing the algorithm using

standardized gates, such as CNOT and Toffoli. The gate sequence is further decomposed into native 1- and 2-qubit gates [114, 115, 116]. Failing to fully leverage the native programmability of the target quantum hardware, this compilation workflow often leads to significant practical overheads.

In this chapter, we would like to explore an alternative approach to sparse Hamiltonian simulation that exploits both the sparsity structure of the input data and the resource efficiency within the underlying quantum hardware. To this end, we first introduce a Hamiltonian-based model of quantum algorithms that encompass all *hardware-efficient* operations in a specific quantum computer, as detailed in Section 5.2. Unlike quantum circuits, which act as a mathematical model agnostic to the underlying hardware, our new Hamiltonian model integrates low-level machine information to minimize the cost of end-to-end implementation. Based on this new model, we develop a technique named *Hamiltonian embedding* in Section 5.3, which enables us to simulate a sparse Hamiltonian as part of a larger quantum evolution that can be efficiently simulated on quantum hardware. With this technique, we can build input models directly with quantum Hamiltonians and perform sparse Hamiltonian simulations with extremely limited quantum resources. In Section 5.4, we showcase an interesting application of Hamiltonian embedding for simulating real-space quantum dynamics. This technique will be used for implementing Quantum Hamiltonian Descent with transverse-field Ising simulators, see Chapter 6.

5.2 Hardware-efficient Hamiltonian models of quantum computer

Our Hamiltonian model of quantum computers is motivated by the low-level control logic of quantum hardware. In general, physically realizable quantum platforms, such as transmon

qubits [117, 118], trapped ions [119, 120], neutral atoms [121], are described as quantum systems whose evolution is governed by a quantum Hamiltonian involving 1- and 2-body interactions. For example, the emulation of analog quantum simulators, such as quantum annealers [122, 123] and Rydberg atom arrays [124, 125], is governed by a system Hamiltonian composed of 1- and 2-local Pauli operators. In the case of digital quantum computers, their native continuously parameterized 1- and 2-qubit gates are typically specified by effective generating Hamiltonians. We propose to use the following quantum Hamiltonian to model hardware-efficient quantum operations:

$$H(t) = \sum_j \alpha_j(t) H_j + \sum_{j,k} \beta_{j,k}(t) H_{j,k}, \tag{5.1}$$

where H_j (or $H_{j,k}$) is a Hamiltonian on qubit j (or j and k) that corresponds to a native operation in specific hardware, $\alpha_j(t)$ and $\beta_{j,k}(t)$ are time-dependent functions. This Hamiltonian model can represent any implementable quantum circuits through piecewise-constant α_j and $\beta_{j,k}$.[1] On the other hand, any unitary operations generated by the Hamiltonian $H(t)$ with smooth time dependence can be readily implemented by a sequence of native gates via product formulas [126].

Trapped ion devices. The IonQ Aria-1 quantum computer natively supports the GPi gate, GPi2 gate [127], virtual-Z gate, and arbitrary angle MS (Mølmer-Sørenson) gate [128]. The hardware-efficient Hamiltonian model for the IonQ Aria-1 device is composed of the following 1- and 2-qubit components:

$$H_j \colon aX + bY + cZ, \quad H_{j,k} \colon (\cos(\phi_1)X_j + \sin(\phi_1)Y_j) \otimes (\cos(\phi_2)X_k + \sin(\phi_2)Y_k), \tag{5.2}$$

[1] An implementable quantum circuit is composed of a sequence of native 1- and 2-qubit gates. Each execution of a native gate can be represented as a rectangular control signal in the time-dependent functions.

where a, b, c, ϕ_1, ϕ_2 can be any real-valued scalars.

Neutral atom arrays. The QuEra Aquila quantum computer is an analog quantum simulator. It allows users to program certain parameters, including the Rabi frequency, local detuning, and atom-atom distance, in the effective Hamiltonian describing the Rydberg atom arrays. More details on the QuEra device are provided in Appendix D.2. We can formulate the abstract model for the QuEra Aquila device using the following 1- and 2-qubit component Hamiltonians:

$$H_j \colon aX + bY + cZ, \quad H_{j,k} \colon \alpha Z \otimes Z, \tag{5.3}$$

where a, b, c are real-valued scalars. The parameter α is engineered by Rydberg interactions and thus must be non-negative.

5.3 Quantum simulation using Hamiltonian embedding

Notation. Let $\mathcal{H}$ be a Hilbert space, and $A \colon \mathcal{H} \to \mathcal{H}$ be a linear operator over $\mathcal{H}$. Let $\mathcal{S}$ be a subspace of $\mathcal{H}$, and $\mathcal{S}^\perp$ be the orthogonal complement of $\mathcal{S}$ such that $\mathcal{H} = \mathcal{S} \oplus \mathcal{S}^\perp$. We denote $P_\mathcal{S}$ (or $P_{\mathcal{S}^\perp}$) as the projection onto $\mathcal{S}$ (or $\mathcal{S}^\perp$). We write $A|_\mathcal{S} \coloneqq P_\mathcal{S} A P_\mathcal{S}$ as the restriction of A in the subspace $\mathcal{S}$. $\mathrm{Herm}(\mathbb{C}^n)$ represents the space of all n-by-n complex Hermitian matrices.

General formulation

To illustrate the idea of Hamiltonian embedding, we consider a simple example. Suppose that we have two (time-independent) Hamiltonian operators A and H, and H contains A as a diagonal block such that $H = \mathrm{diag}(A, *)$, where $*$ represents another irrelevant diagonal block.

Then, the time evolution generated by H is also block-diagonal,

$$e^{-iHt} = \begin{bmatrix} e^{-iAt} & 0 \\ 0 & * \end{bmatrix}, \tag{5.4}$$

where the upper left block is the time evolution of A. In other words, we can simulate a target Hamiltonian A by embedding it into a larger block-diagonal Hamiltonian H. Generalizing this intuition to "approximately" block-diagonal Hamiltonians, we arrive at the following formal definition of Hamiltonian embedding.

Let A be an n-dimensional Hermitian operator and H be a q-qubit operator.

Definition 4 (Hamiltonian embedding)**.** *Let $\eta, \epsilon > 0$ be positive scalars. We say H is a (q, η, ϵ)-**embedding** of A if there exists a subspace $\mathcal{S} \subset \mathbb{C}^{2^q}$ and a unitary operator U such that*

1. *$P_{\mathcal{S}}(U^\dagger H U)P_{\mathcal{S}^\perp} = 0$, i.e., $U^\dagger H U$ is block-diagonal in $\mathcal{S}$ and $\mathcal{S}^\perp$,*
2. *$\|I - U\| \leq \eta$, where I is the identity operator in $\mathbb{C}^{2^q}$,*
3. *$\|(U^\dagger H U)|_{\mathcal{S}} - A\| \leq \epsilon$, where $(\cdot)|_{\mathcal{S}} \coloneqq P_{\mathcal{S}}(\cdot)P_{\mathcal{S}}$.*

*We call the subspace $\mathcal{S}$ as the **embedding subspace**.*

By the definition, the operator H is approximately block-diagonal (up to a minor basis change U) with respect to the embedding subspace $\mathcal{S}$ and its orthogonal complement $\mathcal{S}^\perp$. The target Hamiltonian A is embedded in the upper left block of $U^\dagger H U$ up to an additive error ϵ. Clearly, a q-qubit Hamiltonian H is a $(q, 0, 0)$-embedding of itself. For sufficiently small η and ϵ, we show that the embedding Hamiltonian H simulates the time evolution generated by the target Hamiltonian A.

Theorem 7 (Hamiltonian simulation with Hamiltonian embedding)**.** *Suppose that H is a (q, η, ϵ)-embedding of A. Then, for a fixed evolution time $t \geq 0$, we have that*

$$\left\| \left(e^{-iHt}\right) \Big|_{\mathcal{S}} - e^{-iAt} \right\| \leq (2\eta\|H\| + \epsilon)t. \tag{5.5}$$

Proof. We define $\mathscr{E}(t) \coloneqq e^{-iHt} - U^\dagger e^{-iHt} U$, and it follows that

$$\dot{\mathscr{E}}(t) = (-iH)e^{-iHt} - U^\dagger(-iH)e^{-iHt}U = (-iH)\mathscr{E}(t) - i\left[H - \tilde{H}\right] U^\dagger e^{-iHt}U, \tag{5.6}$$

where $\tilde{H} = U^\dagger H U$. By the variation-of-parameter formula, we have

$$\mathscr{E}(t) = -i\int_0^t e^{-iH(t-\tau)}\left[H - \tilde{H}\right] e^{-i\tilde{H}\tau}\,\mathrm{d}\tau, \tag{5.7}$$

and it follows that

$$\|P_{\mathcal{S}}\mathscr{E}(t)P_{\mathcal{S}}\| \leq \|(H - \tilde{H})P_{\mathcal{S}}\|t = \|[H, U]P_{\mathcal{S}}\|t.$$

We denote $H_0 = P_{\mathcal{S}}\tilde{H}P_{\mathcal{S}}$. By the definition of Hamiltonian embedding, $\tilde{H}$ is block-diagonal and $\|H_0 - A\| \leq \epsilon$. Then, we have

$$\|P_{\mathcal{S}}U^\dagger e^{-iHt}UP_{\mathcal{S}} - e^{-iAt}\| = \|e^{-iH_0t} - e^{-iAt}\| \leq \|H_0 - A\|t \leq \epsilon t.$$

It follows from the triangle inequality that

$$\| \left(e^{-iHt}\right) \Big|_{\mathcal{S}} - e^{-iAt}\| \leq \|P_{\mathcal{S}}\mathscr{E}(t)P_{\mathcal{S}}\| + \|P_{\mathcal{S}}U^\dagger e^{-iHt}UP_{\mathcal{S}} - e^{-iAt}\| \leq (\|[H,U]P_{\mathcal{S}}\| + \epsilon)\, t. \tag{5.8}$$

Note that $\|[H,U]\| = \|[H,I]-[H,(I-U)]\| \leq 2\eta\|H\|$, which implies the error bound (5.5). □

In addition, we find that a complicated Hamiltonian embedding can be built from simpler ones through the four composing rules, including addition, multiplication, composition, and tensor product.

Lemma 4 (Addition)**.** *Let $A_1, A_2 \in$ Herm$(\mathbb{C}^n)$. For $j = 1, 2$, we suppose that H_j is a (q, η, ϵ_j)-embedding of A_j with unitary transformation U and embedding subspace $\mathcal{S}$. Then, $H = H_1 + H_2$ is a $(q, \eta, \epsilon_1 + \epsilon_2)$-embedding of $A = A_1 + A_2$ with the same unitary transformation U and embedding subspace $\mathcal{S}$.*

Proof. Since the unitary U block-diagonalizes both H_1 and H_2, it also block-diagonalizes the $H = H_1 + H_2$. Moreover, by the triangle inequality, $\|(U^\dagger HU)_{\mathcal{S}} - (A_1 + A_2)\| \leq \|(U^\dagger H_1U)_{\mathcal{S}} - A_1\| + \|(U^\dagger H_2U)_{\mathcal{S}} - A_2\| \leq \epsilon_1 + \epsilon_2$. □

Lemma 5 (Multiplication)**.** *Let H be a (q, η, ϵ)-embedding of A with unitary transformation U and embedding subspace $\mathcal{S}$. Then, for any real number α, αH is a $(q, \eta, |\alpha|\epsilon)$-embedding of αA with the same unitary transformation U and embedding subspace $\mathcal{S}$.*

Proof. Due to the linearity of matrix multiplication, we have that

$$\|(U^\dagger \alpha HU)_{\mathcal{S}} - \alpha A\| = |\alpha|\|(U^\dagger HU)_{\mathcal{S}} - A\| \leq |\alpha|\epsilon.$$

□

Lemma 6 (Composition)**.** *For $j = 1, 2$, let H_j be a $(q_j, \eta_j, \epsilon_j)$-embedding of A_j with unitary transformation U_j and embedding subspace $\mathcal{S}_j$, respectively. Then, $H = H_1 \otimes I + I \otimes H_2$ is a $(q_1 + q_2, \eta_1 + \eta_2, \epsilon_1 + \epsilon_2)$-embedding of $A = A_1 \otimes I + I \otimes A_2$ with unitary transformation $U = U_1 \otimes U_2$ and embedding subspace $\mathcal{S} = \mathcal{S}_1 \otimes \mathcal{S}_2$.*

Proof. In the full Hilbert space $\mathcal{H}' = \mathbb{C}^{2^{q_1+q_2}}$, we have the embedding subspace $\mathcal{S} = \mathcal{S}_1 \otimes \mathcal{S}_2$, so its orthogonal complement is

$$\mathcal{S}^\perp = (\mathcal{S}_1^\perp \otimes \mathcal{S}_2) \oplus (\mathcal{S}_1 \otimes \mathcal{S}_2^\perp) \oplus (\mathcal{S}_1^\perp \otimes \mathcal{S}_2^\perp).$$

First, we check the rotated Hamiltonian $\tilde{H} \coloneqq U^\dagger H U$ is block-diagonal in $\mathcal{S}$ and $\mathcal{S}^\perp$. We observe that $P_{\mathcal{S}^\perp} = P_{\mathcal{S}_1^\perp \otimes \mathcal{S}_2} + P_{\mathcal{S}_1 \otimes \mathcal{S}_2^\perp} + P_{\mathcal{S}_1^\perp \otimes \mathcal{S}_2^\perp}$. It is readily verified that

$$P_{\mathcal{S}_1^\perp \otimes \mathcal{S}_2} \left(U^\dagger H U\right) P_{\mathcal{S}_1 \otimes \mathcal{S}_2} = \left(P_{\mathcal{S}_1^\perp} \tilde{H}_1 P_{\mathcal{S}_1}\right) \otimes I + \left(P_{\mathcal{S}_1^\perp} P_{\mathcal{S}_1}\right) \otimes \left(P_{\mathcal{S}_2} \tilde{H}_2 P_{\mathcal{S}_2}\right) = 0.$$

Similarly, we can show that

$$P_{\mathcal{S}_1 \otimes \mathcal{S}_2^\perp} \left(U^\dagger H U\right) P_{\mathcal{S}_1 \otimes \mathcal{S}_2} = 0, \quad P_{\mathcal{S}_1^\perp \otimes \mathcal{S}_2^\perp} \left(U^\dagger H U\right) P_{\mathcal{S}_1 \otimes \mathcal{S}_2} = 0,$$

which implies $P_{\mathcal{S}^\perp} \tilde{H} P_{\mathcal{S}} = 0$, i.e., $\tilde{H}$ is block-diagonal. Also, for the unitary $U = U_1 \otimes U_2$, we have

$$\|I - U_1 \otimes U_2\| \leq \|(I - U_1) \otimes I\| + \|U_1 \otimes (I - U_2)\| \leq \eta_1 + \eta_2.$$

Finally, we check the approximation error,

$$\begin{aligned}
&\|P_{\mathcal{S}}U_1^\dagger \otimes U_2^\dagger(H_1 \otimes I + I \otimes H_2)U_1 \otimes U_2 P_{\mathcal{S}} - (A_1 \otimes I + I \otimes A_2)\| \\
&= \left\| \left[(U_1^\dagger H_1 U_1)|_{\mathcal{S}_1} - A_1\right] \otimes I_{\mathcal{S}_2} + I_{\mathcal{S}_1} \otimes \left[(U_2^\dagger H_2 U_2)|_{\mathcal{S}_2} - A_2\right] \right\| \leq \epsilon_1 + \epsilon_2.
\end{aligned}$$

□

Lemma 7 (Tensor product)**.** *For $j = 1, 2$, let H_j be a $(q_j, \eta_j, \epsilon_j)$-embedding of A_j with unitary transformation U_j and embedding subspace $\mathcal{S}_j$, respectively. Then, $H = H_1 \otimes H_2$ is a $(q_1 + q_2, \eta_1 + \eta_2, \|A_1\|\epsilon_2 + \|A_2\|\epsilon_1 + \epsilon_1\epsilon_2)$-embedding of $H = A_1 \otimes A_2$ with unitary transformation $U = U_1 \otimes U_2$ and embedding subspace $\mathcal{S} = \mathcal{S}_1 \otimes \mathcal{S}_2$.*

Proof. Similar to Rule 3, we can check that $P_{\mathcal{S}^\perp}(U^\dagger H_1 \otimes H_2 U)P_{\mathcal{S}} = 0$ and $\|I - U_1 \otimes U_2\| \leq \eta_1 + \eta_2$. For $j = 1, 2$, we have that $\|(U_j^\dagger H_j U_j)|_{\mathcal{S}_j}\| \leq \|A_j\| + \epsilon_j$. Using this fact, we can estimate the approximation error:

$$\begin{aligned}
&\|P_{\mathcal{S}}U_1^\dagger \otimes U_2^\dagger(H_1 \otimes H_2)U_1 \otimes U_2 P_{\mathcal{S}} - (A_1 \otimes A_2)\| = \|(U_1^\dagger H_1 U_1)|_{\mathcal{S}_1} \otimes (U_2^\dagger H_2 U_2)|_{\mathcal{S}_2} - A_1 \otimes A_2\| \\
&\leq \|[A_1 - (U_1^\dagger H_1 U_1)|_{\mathcal{S}_1}] \otimes A_2\| + \|(U_1^\dagger H_1 U_1)|_{\mathcal{S}_1} \otimes [A_2 - (U_2^\dagger H_2 U_2)|_{\mathcal{S}_2}]\| \\
&\leq \|A_2\|\epsilon_1 + \|A_1\|\epsilon_2 + \epsilon_1\epsilon_2.
\end{aligned}$$

□

Perturbative Hamiltonian embedding

As we have seen, a target Hamiltonian A can be simulated with a block-diagonal embedding Hamiltonian. However, identifying such a corresponding block-diagonal embedding could

be non-trivial for many sparse Hamiltonians A arising from real-world applications. Here, we give an explicit construction of Hamiltonian embedding based on perturbation theory.

First, we assume there is a q-qubit quantum operator Q and a subspace $\mathcal{S}$ such that $Q|_{\mathcal{S}} = A$. We express the operator Q in a block-matrix form by projecting it down to the subspaces $\mathcal{S}$ and $\mathcal{S}^{\perp}$, respectively,

$$Q = \begin{bmatrix} A & R^{\dagger} \\ R & B \end{bmatrix}, \tag{5.9}$$

where $A = P_{\mathcal{S}} Q P_{\mathcal{S}}$, $R = P_{\mathcal{S}^{\perp}} Q P_{\mathcal{S}}$, $B = P_{\mathcal{S}^{\perp}} Q P_{\mathcal{S}^{\perp}}$. Since the off-diagonal block R is not necessarily zero, we can not directly simulate A by evolving the Hamiltonian Q because in general $P_{\mathcal{S}} e^{-iQt} P_{\mathcal{S}} \neq e^{-iAt}$. This means that a quantum state initialized in the subspace $\mathcal{S}$ could be driven out of this subspace in the course of quantum evolution.

To suppress the leakage from the embedding subspace $\mathcal{S}$, we introduce another q-qubit operator H^{pen} as the *penalty Hamiltonian*. We assume H^{pen} has an n-fold degenerate ground-energy subspace $\mathcal{S}$ with the ground energy being zero, i.e., the least eigenvalue of H^{pen} is zero. For a fixed positive number $g > 0$, we define the following quantum operator,

$$H = gH^{\mathrm{pen}} + Q. \tag{5.10}$$

Similarly, this new operator H can be expressed in a block-matrix form (we denote $C = P_{\mathcal{S}^{\perp}} H^{\mathrm{pen}} P_{\mathcal{S}^{\perp}}$

and $G = B + gC$):

$$H = \begin{bmatrix} A & R^\dagger \\ R & B + gC \end{bmatrix} = \underbrace{\begin{bmatrix} A & 0 \\ 0 & G \end{bmatrix}}_{\text{diagonal}} + \underbrace{\begin{bmatrix} 0 & R^\dagger \\ R & 0 \end{bmatrix}}_{\text{off-diagonal}}, \tag{5.11}$$

Schrieffer-Wolff transformation

We define $\Delta_0 \coloneqq \lambda_{\min}(B) - \lambda_{\max}(A)$ as the spectral distance between the two diagonal blocks in Q, and $\Delta \coloneqq \lambda_{\min}(G) - \lambda_{\max}(A)$ being the spectral distance between the two diagonal blocks in H. Note that the spectral gaps Δ_0 and Δ are not necessarily positive. By Weyl's inequality, we have

$$\Delta \geq g\lambda_1 + \Delta_0,$$

where $\lambda_1 > 0$ is the minimum eigenvalue of C (recall that $\mathcal{S}$ is the ground-energy subspace of H^{pen}).

By choosing sufficiently large $g > 0$, we can make $\Delta > 0$ and $\Delta \sim g$. For large Δ such that $\|R\|/\Delta \ll 1$, the off-diagonal part in H can be regarded as a perturbation and we can invoke the Schrieffer-Wolff transformation [129, 130].

Theorem 8 (Schrieffer-Wolff transformation)**.** *Define $\kappa \coloneqq \|R\|/\Delta$. We assume that $\Delta > 0$ and $\kappa < 1/2$. Then, there exists a unitary transformation U such that $\|I - U\| \leq 2\sqrt{2}\kappa$ and $U^\dagger H U$ is block-diagonal. Moreover, if $\kappa < 1/16$, there exists an absolute constant $c > 0$ such that*

$$\left\| \left(U^\dagger H U\right)_{\mathcal{S}} - A \right\| \leq c\|R\|\kappa. \tag{5.12}$$

In other words, H is a $(q, 2\sqrt{2}\kappa, c\|R\|\kappa)$-embedding of A if $\kappa < 16$.

Proof. We define $H_0 = \begin{bmatrix} A & 0 \\ 0 & G \end{bmatrix}$ and $V = \begin{bmatrix} 0 & R \\ R^\dagger & 0 \end{bmatrix}$, so $H = H_0 + V$. Note that $\|V\| = \|R\|$. When $\kappa = \|V\|/\Delta < 1/2$, there is a unique Schrieffer-Wolff transformation U such that $U(H_0 + V)U^\dagger$ is block-diagonal (see [130] for detailed discussions). By [129, Section 1], there exists an *angle* operator Θ such that $\|I - U\| = 2\|\sin\frac{1}{2}\Theta\|$. Let $\mathcal{T} = U\mathcal{S}$ be the rotated subspace, we also have that $\|P_\mathcal{S} - P_\mathcal{T}\| = \|\sin\Theta\|$. On the other hand, by [130, Lemma 3.1], we can estimate the distance between $\mathcal{S}$ and $\mathcal{T}$:

$$\|P_\mathcal{S} - P_\mathcal{T}\| \leq 2\kappa.$$

Since we have $\kappa < 1/2$, the angle operator is bounded by $\pi/2$ [129]. Note that for any angle $|\theta| < \pi/2$, we have

$$\left|\sin\frac{1}{2}\theta\right| = \sqrt{\frac{1}{2}(1-\cos\theta)} = \sqrt{\frac{1}{2}(1-\sqrt{1-\sin^2\theta})} \leq \frac{\sqrt{2}}{2}|\sin\theta|,$$

which implies that $\|\sin\frac{1}{2}\Theta\| \leq \frac{\sqrt{2}}{2}\|\sin\Theta\|$. It turns out that

$$\|I - U\| \leq \sqrt{2}\|P_\mathcal{S} - P_\mathcal{T}\| \leq 2\sqrt{2}\kappa.$$

We define the effective low-energy Hamiltonian $H_{\text{eff}} \coloneqq P_\mathcal{S}U^\dagger HUP_\mathcal{S}$. The effective Hamiltonian H_{eff} allows a Taylor series expansion [130, 131]:

$$H_{\text{eff}} = \sum_{j=1}^{\infty} H_{\text{eff},j}. \tag{5.13}$$

We define $H_{\text{eff}}(k) = \sum_{j=1}^{k} H_{\text{eff},j}$. When $\kappa < 1/16$, the series (5.13) converge absolutely and there exists an absolute constant $c > 0$ such that

$$\|H_{\text{eff}} - H_{\text{eff}}(k)\| \leq c\frac{\|V\|^{k+1}}{\Delta^k} = c\|R\|\kappa^k.$$

Note that $H_{\text{eff}}(1) = A$. Therefore, we prove that $\|H_{\text{eff}} - A\| \leq c\|R\|\kappa$. □

By Theorem 8, a bigger g implies a more accurate Hamiltonian embedding H. The idea is that a large parameter g suppresses the leakage of the quantum evolution from the embedding subspace $\mathcal{S}$ to its orthogonal complement $\mathcal{S}^{\perp}$. Based on this intuition, we refer to g as the *penalty coefficient* and the operator H^{pen} as the *penalty Hamiltonian*.

In a perturbative Hamiltonian embedding with penalty coefficient $g > 0$, the error characterization parameters are $\eta \leq 2\sqrt{2}\kappa \sim \frac{1}{g}$, $\epsilon \leq c\|R\|\kappa \sim \frac{\|R\|}{g}$. By choosing a large enough g, the error parameters η and ϵ can be made arbitrarily small. On the other hand, when H^{pen} is fast-forwardable, there exist quantum algorithms (e.g., continuous qDRIFT [132]) that can efficiently simulate the Hamiltonian embedding even for large g. Therefore, for simplicity, we will not explicitly specify η and ϵ when discussing a perturbative Hamiltonian embedding in the rest of this paper.

Constructing perturbative Hamiltonian embeddings

All the basic rules of constructing Hamiltonian embeddings can be reformulated for perturbative Hamiltonian embedding. We omit the proof as it is similar to those in Section 5.3.

Proposition 2 (Rules for building perturbative Hamiltonian embedding)**.**

1. *(Addition) For $j = 1, 2$, suppose $H_j = gH^{\mathrm{pen}} + Q_j$ is an embedding of A_j with the same embedding subspace $\mathcal{S}$. Then, $H = gH^{\mathrm{pen}} + Q_1 + Q_2$ is an embedding of $A = A_1 + A_2$ with embedding subspace $\mathcal{S}$.*

2. *(Multiplication) Suppose $H = gH^{\mathrm{pen}} + Q$ is an embedding of A with embedding subspace $\mathcal{S}$. Then, for any real number α, $H' = gH^{\mathrm{pen}} + \alpha Q$ is an embedding of $A' = \alpha A$.*

3. *(Composition) For $j = 1, 2$, suppose that the q_j-qubit operator $H_j = gH_j^{\mathrm{pen}} + Q_j$ is an embedding of A_j with the embedding subspace $\mathcal{S}_j$ being the ground-energy subspace of H_j^{pen}. Then, $H = gH^{\mathrm{pen}} + Q$ with*

$$H^{\mathrm{pen}} = H_1^{\mathrm{pen}} \otimes I + I \otimes H_2^{\mathrm{pen}},\ Q = Q_1 \otimes I + I \otimes Q_2 \tag{5.14}$$

is an embedding of $A = A_1 \otimes I + I \otimes A_2$ with embedding subspace $\mathcal{S} = \mathcal{S}_1 \otimes \mathcal{S}_2$.

4. *(Tensor product) For $j = 1, 2$, suppose that $H_j = gH_j^{\mathrm{pen}} + Q_j$ is an embedding of A_j with the embedding subspace $\mathcal{S}_j$ being the ground-energy subspace of H_j^{pen}. Then, $H = gH^{\mathrm{pen}} + Q_1 \otimes Q_2$ with*

$$H^{\mathrm{pen}} = H_1^{\mathrm{pen}} \otimes I + I \otimes H_2^{\mathrm{pen}} \tag{5.15}$$

is an embedding of $H = A_1 \otimes A_2$ with embedding subspace $\mathcal{S} = \mathcal{S}_1 \otimes \mathcal{S}_2$.

Improved simulation error analysis

By Theorem 8, the Hamiltonian embedding $H = gH^{\mathrm{pen}} + Q$ is a (q, η, ϵ)-embedding of A, where $\eta \sim 1/g$, $\epsilon \sim \|R\|/g$. Invoking Theorem 7, we know that the simulation error using perturbative Hamiltonian embedding is of the order $\mathcal{O}(\|H\|t/g)$. By leveraging the particular operator splitting structure as shown in (5.11), we can prove a tighter error bound, which is of the order $\mathcal{O}(\|R\|t/g)$.

Theorem 9 (Quantum simulation with perturbative embedding)**.** *Let all the notations be the same as above. We assume $\Delta > 0$ and $\kappa < 1/2$. Then, for any fixed $t > 0$, we have*

$$\|P_{\mathcal{S}} e^{-iHt} P_{\mathcal{S}} - e^{-iAt}\| \leq 4\sqrt{2}\kappa \|R\| t. \tag{5.16}$$

Proof. Let $H_0 = P_{\mathcal{S}} H P_{\mathcal{S}} = \mathrm{diag}(A, 0)$, and we define:

$$\mathscr{A}(t) = P_{\mathcal{S}} e^{-iHt} P_{\mathcal{S}}, \quad \mathscr{B}(t) = P_{\mathcal{S}} e^{-iH_0 t} P_{\mathcal{S}}. \tag{5.17}$$

We also define the $\mathscr{E}(t) \coloneqq \mathscr{A}(t) - \mathscr{B}(t)$. Note that $\mathscr{E}(0) = 0$. We find that

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathscr{A}(t) &= P_{\mathcal{S}} \left(-iH\right) \left(P_{\mathcal{S}} + P_{\mathcal{S}^\perp}\right) e^{-iHt} P_{\mathcal{S}} = -iH_0 \mathscr{A}(t) - iRe^{-iHt} P_{\mathcal{S}}, \\
\frac{\mathrm{d}}{\mathrm{d}t}\mathscr{B}(t) &= P_{\mathcal{S}} \left(-iH_0\right) e^{-iH_0 t} P_{\mathcal{S}} = -iH_0 \mathscr{B}(t).
\end{aligned}$$

Therefore, we have $\dot{\mathscr{E}}(t) = \dot{\mathscr{A}}(t) - \dot{\mathscr{B}}(t) = -iH_0 \mathscr{E}(t) - iRe^{-iHt} P_{\mathcal{S}}$. By the variation-of-

parameter formula [133, Theorem 4.9], we represent the function $\mathscr{E}(t)$ as the time integral:

$$\mathscr{E}(t) = -i \int_0^t \exp(-iH_0(t-\tau)) R \exp(-iH\tau) P_{\mathcal{S}} \, \mathrm{d}\tau, \tag{5.18}$$

so we can estimate the error (recall that $R = P_{\mathcal{S}} H P_{\mathcal{S}^\perp}$):

$$\begin{aligned} \|\mathscr{E}(t)\| &\leq \int_0^t \|P_{\mathcal{S}} H P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}\| \, \mathrm{d}\tau \leq \int_0^t \|R P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}\| \, \mathrm{d}\tau \\ &\leq \int_0^t \|R\| \left(4\sqrt{2}\kappa\right) \, \mathrm{d}\tau = 4\sqrt{2}\kappa \|R\| t. \end{aligned}$$

In the second step, we use $P_{\mathcal{S}^\perp} = P_{\mathcal{S}^\perp}^2$. In the second to last step, we invoke Lemma 8. □

Lemma 8. *Assume that $\kappa < 1/2$ and let $\tau \geq 0$ be an arbitrary real number. Then, we have*

$$\|P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}\| \leq 4\sqrt{2}\kappa. \tag{5.19}$$

Proof. By Theorem 8, there exists a Schrieffer-Wolff transformation U such that

$$U^\dagger H U = \begin{bmatrix} H_{\text{eff}} & 0 \\ 0 & * \end{bmatrix},$$

where $H_{\text{eff}} = P_{\mathcal{S}} U^\dagger H U P_{\mathcal{S}}$. We remark that $P_{\mathcal{S}^\perp} U^\dagger \exp(-iH\tau) U P_{\mathcal{S}} = 0$. To see this, note that U block-diagonalizes H, so does $\exp\left(-iUHU^\dagger\tau\right)$:

$$P_{\mathcal{S}^\perp} U \exp(-iH\tau) U^\dagger P_{\mathcal{S}} = P_{\mathcal{S}^\perp} \exp\left(-iUHU^\dagger\tau\right) P_{\mathcal{S}} = 0.$$

Now, we bound the cross term $P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}$ by,

$$\begin{aligned}\|P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}\| &= \|P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}} - P_{\mathcal{S}^\perp} U^\dagger \exp(-iH\tau) U P_{\mathcal{S}}\| \\ &\leq \| \exp(-iH\tau) - U^\dagger \exp(-iH\tau) U\| = \|U \exp(-iH\tau) - \exp(-iH\tau) U\|.\end{aligned}$$

Then, by Theorem 8, we conclude that

$$\|P_{\mathcal{S}^\perp} \exp(-iH\tau) P_{\mathcal{S}}\| \leq \|(U-I) \exp(-iH\tau) - \exp(-iH\tau)(U-I)\| \leq 2\|U-I\| \leq 4\sqrt{2}\kappa.$$

□

Quantum simulation of perturbative embedding

The perturbative Hamiltonian embedding $H^{\mathrm{ebd}} = gH^{\mathrm{pen}} + Q$ often comes with a large penalty coefficient g. This could lead to significant simulation error if one uses a standard quantum algorithm (e.g., a product formula) to simulate H^{ebd}. In this section, we discuss two types of quantum simulation protocols (product formulas and qDRIFT) for perturbative Hamiltonian embedding. It turns out that qDRIFT is friendly to near-term devices with minimal overheads in terms of g.

First, we consider the standard first-order Trotter formula:

$$e^{-iH^{\mathrm{ebd}}t} \approx U_1(t) := \prod_{k=1}^{r} e^{-iQt/r} e^{-igH^{\mathrm{pen}}t/r}. \tag{5.20}$$

By [126], the Trotter error scales like

$$\|e^{-iH^{\mathrm{ebd}}t} - U_1(t)\| \leq O\left(\frac{g[H^{\mathrm{pen}}, Q]t^2}{r}\right). \tag{5.21}$$

To simulate the embedding Hamiltonian up to an additive error ϵ, we need to choose the Trotter number

$$r = O\left(\frac{g[H^{\mathrm{pen}}, Q]t^2}{\epsilon}\right). \tag{5.22}$$

Similarly, the second-order Trotter-Suzuki formula requires the Trotter number

$$r = O\left(\frac{\Gamma t^{3/2}}{\epsilon^{1/2}}\right), \tag{5.23}$$

where $\Gamma = \left(\frac{g}{12}\|[Q,[Q,H^{\mathrm{pen}}]]\| + \frac{g^2}{24}\|[H^{\mathrm{pen}},[H^{\mathrm{pen}},Q]]\|\right)^{1/2}$.

We can improve the first-order Trotter formula by randomization [134], which turns out to be effectively a second-order product formula:

$$\|e^{-iH^{\mathrm{ebd}}t} - U_{1,\mathrm{rand}}(t)\| \leq O\left(\frac{\Lambda^3 t^3}{r^2}\right), \tag{5.24}$$

where $U_{1,\mathrm{rand}}(t)$ is the randomized first-order Trotter formula, and $\Lambda = \max\{g\|H^{\mathrm{pen}}\|, \|Q\|\}$. It follows that we need the Trotter number

$$r = O\left(\frac{\Lambda^{3/2} t^{3/2}}{\epsilon^{1/2}}\right) \tag{5.25}$$

in order to achieve an ϵ-accurate simulation of H'. A second-order (deterministic) Trotter formula will require that $r = O(gt^{3/2}\epsilon^{-1/2})$, but potentially have a better pre-factor in terms of the commutators of H^{pen} and Q [126].

A major drawback of the aforementioned quantum simulation protocols we discussed above is that their simulation error has explicit dependence on the penalty coefficient g, which could be very huge in practice. To address this issue, we consider quantum simulation in the interaction picture. For a Hamiltonian $H = A + B$, we transform it to the interaction picture:

$$H_I(t) = e^{iAt} B e^{-iAt}, \tag{5.26}$$

and consider the Schrödinger equation

$$i\frac{\mathrm{d}}{\mathrm{d}t} |\psi_I(t)\rangle = H_I(t) |\psi_I(t)\rangle , \tag{5.27}$$

subject to $|\psi_I(0)\rangle = |\psi_0\rangle$. One can easily verify that the state vector $|\psi(t)\rangle = e^{-iHt} |\psi_0\rangle$ can be recovered by

$$|\psi(t)\rangle = e^{-iAt} |\psi_I(t)\rangle . \tag{5.28}$$

Based on this observation, digital quantum simulation protocols for time-dependent Hamiltonian that has L^1-norm scaling can be used to simulate $H_I(t)$ to achieve an error scaling only depends on $\|B\|$, given that A can be fast-forwardable.

In our construction of Hamiltonian embedding, H^{pen} is often fast-forwardable, so we can

leverage the interaction-picture quantum simulation protocols to simulate

$$H_I^{\mathrm{ebd}}(t) = e^{igH^{\mathrm{pen}}t} Q e^{-igH^{\mathrm{pen}}t} \tag{5.29}$$

and achieve g-independent error scaling. There are a handful of quantum algorithms that can achieve l^1-norm scaling for interaction picture Hamiltonian, e.g., truncated Dyson series [112, 132], continuous qDRIFT [132], qHOP [135], etc. Unfortunately, the truncated Dyson series method and the qHOP method require complicated control flows and advanced input models (e.g., block-encodings) that are infeasible for near-term quantum computers. On the other hand, continuous qDRIFT relies on a simple product formula and a classical stochastic sampling protocol, which is possible to implement on near-term devices. The details of qDRIFT are presented in Algorithm 3.

Algorithm 3: Quantum simulation by qDRIFT.

input : An initial state $|\psi_0\rangle$, number of samples M, Trotter number K, total evolution time T.

output: An estimate of the measurement result $\tilde{E}$.

Let $\Delta t = T/K$;
for $j \leftarrow 1$ **to** M **do**
  for $k \leftarrow 1$ **to** K **do**
    Sample a uniformly random number $\xi_k \in [(k-1)\Delta t, k\Delta t]$;
    Propagate the quantum state

$$|\psi_k\rangle = e^{-iH_I^{\mathrm{ebd}}(\xi_k)\Delta t} |\psi_{k-1}\rangle = e^{igH^{\mathrm{pen}}\xi_k} e^{-iQ\Delta t} e^{-igH^{\mathrm{pen}}\xi_k} |\psi_{k-1}\rangle .$$

  Measure the quantum state $|\psi_K\rangle$ with the observable $M_I = e^{igH^{\mathrm{pen}}T} O e^{-igH^{\mathrm{pen}}T}$ and compute $m_j = \langle\psi_K| M_I |\psi_K\rangle$;
Compute the sample mean $\tilde{E} = \frac{1}{M}\sum_{j=1}^{M} m_j$.

We can prove that when M, K are large enough,

$$\langle\psi_0| e^{iH^{\mathrm{ebd}}T} O e^{-iH^{\mathrm{ebd}}T} |\psi_0\rangle \approx \tilde{E}. \tag{5.30}$$

More precisely, by [132], the Trotter number required by continuous qDRIFT to simulate the interaction-picture Hamiltonian $H_I'(t)$ is

$$r = O\left(\frac{\|Q\|^2 t^2}{\epsilon}\right). \tag{5.31}$$

Notably, the error bound in (5.31) is independent of the penalty coefficient g, which allows us to simulate the embedding Hamiltonian H^{ebd} with large g.

Hamiltonian embedding of sparse matrices

For Hermitian matrices with certain special sparsity structures, we can construct explicit Hamiltonian embeddings for them. These embeddings involve particular *embedding schemes*, which encode the original sparse matrices using qubits. In what follows, we provide six different Hamiltonian embedding schemes for three types of sparse Hermitian matrices. It's important to note that many of these embedding schemes are already documented in existing literature, albeit under different names. For details discussions, see [42, Appendix B].

1. **Band matrix**: an n-by-n matrix A is a band matrix of bandwidth d if $A_{i,j} = 0$ for any $i, j = 1, \ldots, n$ such that $|i - j| > d$.

2. **Banded circulant matrix**: an n-by-n matrix A is a banded circulant matrix of bandwidth d if *(i)* in the first row, $A_{1,j} = 0$ for any $j = d + 2, \ldots, n - d$, and *(ii)* all row vectors

are composed of the same elements and each row vector is rotated one element to the right relative to the preceding row vector.

3. **s-sparse matrices**: an n-by-n matrix A is s-sparse if each row/column of A has at most s non-zero elements.

In Table 5.1, we list all the embedding schemes discussed in this paper that can be applied to simulate at least one type of sparse Hermitian matrices. Among our embedding schemes, the unary and one-hot encodings are well-known [136, 137, 138, 139, 140], while the others have not been systematically studied in the existing literature. Note that, except for the last item ("Penalty-free one-hot"), all embeddings are perturbative Hamiltonian embeddings and thus require a penalty Hamiltonian. The "Max. weight" column shows the maximal weight of the Pauli operators involved in the corresponding embedding Hamiltonian (d represents the bandwidth of the target Hamiltonian A). For perturbative Hamiltonian embedding, the simulation error can be made arbitrarily small by increasing the penalty coefficient g. Thus, for simplicity, we do not explicitly specify the tuple (q, η, ϵ) for each Hamiltonian embedding listed in the table.

Embedding scheme	**Applications**	**Max. weight**
Unary	band	$\max(d, 2)$
Antiferromagnetic	band	$\max(d, 2)$
Circulant unary	banded circulant	$\max(d, 2)$
Circulant antiferromagnetic	banded circulant	$\max(d, 2)$
One-hot	band, banded circulant, s-sparse	2
Penalty-free one-hot	band, banded circulant, s-sparse	2

Table 5.1: Six Hamiltonian embeddings for sparse Hamiltonian simulation

A notable feature of these embedding schemes is that their maximal weight depends on the special sparsity pattern of the target Hamiltonian A. In particular, for band matrices (or banded circulant matrices) with bandwidth $d \leq 2$, the corresponding unary/antiferromagnetic

(or circulant unary/circulant antiferromagnetic) embeddings have a maximal weight of 2, which means these embedding Hamiltonians can be directly simulated using native gates. This is also true for one-hot/penalty-free one-hot embeddings that encode arbitrary s-sparse matrices.

We also note that all these schemes utilize only an exponentially small subspace of the full Hilbert space for embedding. Therefore, direct use of these schemes to build Hamiltonian embeddings does not lead to meaningful quantum speedups in Hamiltonian simulation. To achieve exponential quantum speedups, one strategy is to first decompose the target sparse Hamiltonian using the *composition* and *tensor product* rules (as detailed in Section 5.3), and then use the embedding schemes listed in Table 5.1 to construct elementary building blocks of size $O(1)$. Using this strategy, we can build Hamiltonian embeddings for symmetric/Hermitian matrices that arise from interesting quantum applications, such as graph theory and differential equations, with logarithmic quantum resources [42].

Now, we provide some concrete examples of the embedding of a small sparse matrix. We consider a 5-by-5 tridiagonal matrix A given by

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}, \tag{5.32}$$

which is the Laplacian matrix of a 5-node chain graph. In Table 5.2, we list the operators Q and H^{pen} for embedding A using various embedding schemes. Here, $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and

$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ are the Pauli-X and Pauli-Z matrices, and $\hat{n} = \frac{1}{2}(I - Z)$. The overall embedding Hamiltonian is $H = gH^{\mathrm{pen}} + Q$, where $g > 0$ is a sufficiently large penalty coefficient. The embedding subspace $\mathcal{S}$ depends on the embedding scheme. For sufficiently large $g > 0$, Theorem 9 implies that $e^{-iAt} \approx P_{\mathcal{S}} e^{-iH_A t} P_{\mathcal{S}}$.

Embedding scheme	Q	H^{pen}
Unary	$-\hat{n}_1 + \hat{n}_4 + \sum_{j=1}^{4} X_j$	$-\sum_{j=1}^{3} Z_{j+1}Z_j + Z_1 - Z_4$
Antiferromagnetic	$-\hat{n}_1 - \hat{n}_4 + \sum_{j=1}^{4} X_j$	$\sum_{j=1}^{3} Z_{j+1}Z_j + Z_1 + Z_4$
One-hot	$\left(-\hat{n}_1 - \hat{n}_5 - 2\sum_{j=2}^{4} \hat{n}_j\right) + \sum_{j=1}^{4} X_{j+1}X_j$	$\left(\sum_{j=1}^{5} \hat{n}_j - 1\right)^2$
Penalty-free one-hot	$\left(-\hat{n}_1 - \hat{n}_5 - 2\sum_{j=2}^{4} \hat{n}_j\right) + \sum_{j=1}^{4} (X_{j+1}X_j + Y_{j+1}Y_j)$	0

Table 5.2: Hamiltonian embeddings of the tridiagonal matrix A in (5.32).

While there could be several possible Hamiltonian embedding schemes for a fixed target Hamiltonian A, we remark that there is no single criterion to determine which one would perform the best. We provide a few aspects to compare various Hamiltonian embeddings in practice. First, the Hamiltonian embedding must match the hardware programmability. For example, the antiferromagnetic embedding scheme fits well with Rydberg atom arrays. Second, we want to use the Hamiltonian embedding with minimal simulation error. The simulation error could come from the perturbative construction of an embedding and/or a specific Hamiltonian simulation algorithm (e.g., a Trotter formula, continuous qDRIFT, etc.).

Discussions

Quantum speedups and applicability of Hamiltonian embedding

In theory, a Hamiltonian embedding can be constructed for any sparse matrix, as detailed in Section 7. However, for large matrices, this construction process may incur an exponential cost, casting doubt on the feasibility of achieving quantum speedup in such cases. Fortunately, we managed to identify various scenarios, including high-dimensional graphs created through graph product operations and specific linear differential operators, where the Hamiltonian embedding can be constructed using quantum resources that scale logarithmically in the input size. This leads to exponential quantum speedups using our methodology.

A key feature of Hamiltonian embedding is that it directly harnesses hardware-efficient operations to build the input model, which significantly reduces the quantum resources needed in Hamiltonian simulation tasks and quantum algorithms based on these. This technique enables us to implement several experiments on existing open-access cloud-based quantum computers, showcasing interesting quantum applications (see [42]). In contrast, Hamiltonian simulation methods using traditional query models (such as quantum oracles or block-encodings) are impractical on these quantum computers. Even a single implementation of the query model would deplete the available quantum resources [113].

It is worth noting that our Hamiltonian embedding technique can also be adapted for analog quantum simulators, unlike existing sparse Hamiltonian simulation methods which are tailored exclusively for gate-based quantum computers. An experimental demonstration with Rydberg atom arrays is provided in Section 5.4. This adaptability broadens the possibilities for analog

quantum computation.

Connection to quantum Hamiltonian complexity

Hamiltonian embedding is closely related to previously studied notions of simulating one Hamiltonian by another Hamiltonian, including the isometry-based definition of *simulation* in [131] used to study the complexity of 2-local stoquastic Hamiltonians, and *Hamiltonian encodings* used to show the universality of certain spin-lattice models [141, 142].

The simulation introduced in [131] quantifies how close are two quantum Hamiltonians in terms of their low-energy spectrum. It is defined as an isometry transformation and its application has been limited to stoquastic local Hamiltonians [143], while our Hamiltonian embedding applies to a broader class of sparse matrices.

A Hamiltonian encoding (aka, encoding transformation) is a map that encodes a Hamiltonian H into some other Hamiltonian H'. Specifically, this map needs to fulfill a few basic requirements, including the preservation of locality, spectrum, and real-linearity [141]. Hamiltonian encoding is more general than simulation since an encoding does not need to be an isometry. The construction of Hamiltonian encodings heavily utilizes perturbative gadgets [144, 145, 146], a theoretical tool originally developed to prove hardness results in Hamiltonian complexity theory.

Compared to [141], our definition of Hamiltonian embedding (Definition 4) more closely resembles the *simulation* defined in [131], in the sense that both do not impose any locality-related restriction on the encoding/embedding maps. Technically speaking, our perturbative Hamiltonian embedding can be viewed as an explicit construction of perturbative gadgets, but two major dif-

ferences distinguish our work from prior arts. First, existing techniques are almost exclusively applied to (local) quantum Hamiltonians arising from many-body physics (i.e., bosons, fermions, and local spin/qudit Hamiltonians), while our focus is on the simulation of sparse matrices. Second, Hamiltonian encodings often involve complicated basis change, while our Hamiltonian embedding preserves the energy spectrum and dynamical evolution without introducing a nontrivial basis change. This allows us to measure the embedded system using a subset of computational basis. However, this favorable feature is achieved at a cost of universality. The construction technique in [141] can be applied to arbitrary local Hamiltonians, while we only provide explicit constructions of Hamiltonian embedding for certain families of sparse Hermitian matrices.

5.4 Applications to simulating real-space quantum dynamics

In this section, we discuss how to apply Hamiltonian embedding to simulate real-space quantum dynamics without introducing coherent black-box query models. This approach also leads to a resource-efficient implementation of Quantum Hamiltonian Descent on analog quantum simulators, as detailed in Chapter 6.

The real-space quantum dynamics are governed by the time-dependent Schrödinger equation over the d-dimensional Euclidean space,

$$i\frac{\partial}{\partial t}\Psi(t,x) = \left[-\frac{1}{2}\nabla^2 + f(x)\right]\Psi(t,x), \tag{5.33}$$

$$\text{subject to} \quad \Psi(0,x) = \Psi_0(x), \tag{5.34}$$

where $\Psi(t,x) : [0,T] \times \mathbb{R}^d \to \mathbb{C}$ is the quantum wave function, $f\colon \mathbb{R}^d \to \mathbb{R}$ is a potential field,

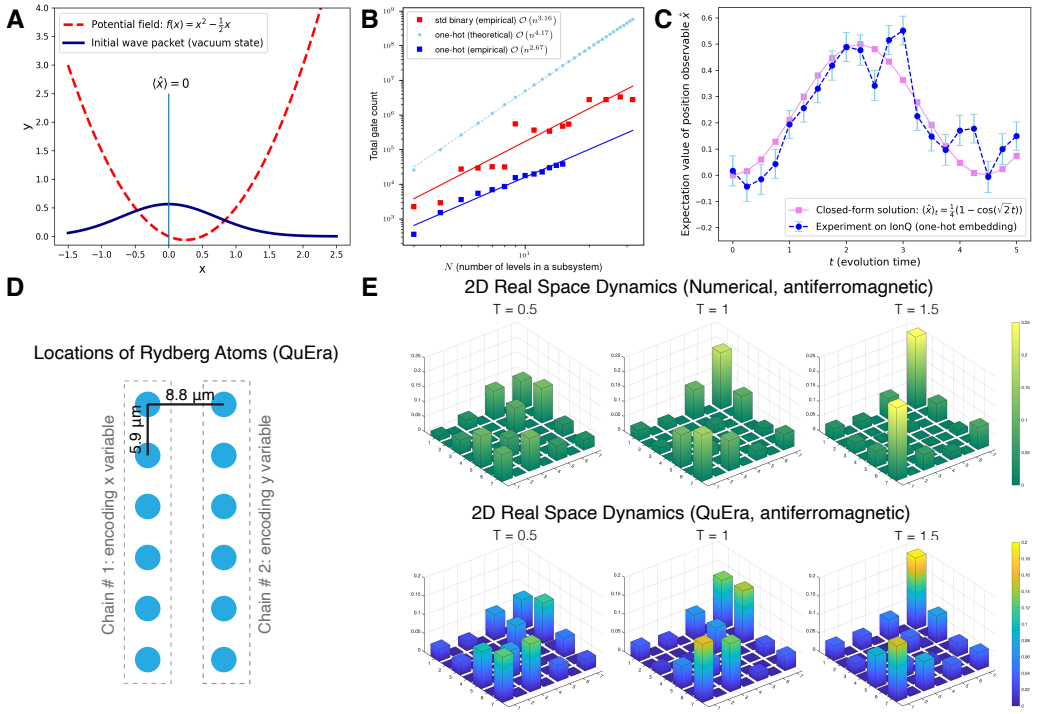


Figure 5.1: **Simulating real-space quantum dynamics. A.** The potential field and initial state in the IonQ experiment. **B.** Resource count of simulating real-space quantum dynamics using an N-level truncated Hamiltonian. The empirical data suggests that one-hot embedding has a better asymptotic scaling than the standard binary encoding. **C.** Expectation value of the position observable $\hat{x}$ depicted as a function of evolution time t. Experiment data obtained from IonQ Aria-1. **D.** Locations of the 12 Rydberg atoms in the experiment of simulating quantum dynamics in a 2D space. **E.** Measuring the 2D real-space quantum dynamics at $T = 1, 1.5, 2$. Top: numerical simulation; Bottom: experiment results obtained on QuEra Aquila.

and the initial state is given by $\Psi_0(x)$.

Although physically relevant systems are of dimension 3, high-dimensional Schrödinger equations find ubiquitous applications for simulating multi-particle systems in condensed matter physics [147] and quantum chemistry [148, 149]. Also, some quantum algorithms for numerical optimization require the simulation of Schrödinger equations in high-dimensional Euclidean spaces [57, 85, 40]. Several quantum algorithms for real-space quantum simulation have been proposed, e.g., [150, 151, 88, 87, 135, 76, 152], while most of them require fully fault-tolerant quantum computers that are currently out of reach. Recently, Chang et al. [153] utilize variants of

Gray code to reformulate the continuous-space Schrödinger equation as spin Hamiltonians, leading to exponentially better space complexity over classical methods. However, the spin operators in [153] involve 3-body interaction terms and thus require further decomposition in the implementation. In our experiment, the different choices of embedding schemes allow us to represent the Schrödinger equation using 1- and 2-body spin operators, eliminating the Pauli compilation overhead.

We showcase the flexibility and versatility of the Hamiltonian embedding technique by implementing the real-space quantum simulation task on two different experimental platforms: a trapped-ion quantum processor (IonQ Aria-1) and a programmable Rydberg atom array (QuEra Aquila).

On the trapped-ion processor, we simulate a 1-dimensional Schrödinger equation with a quadratic potential field and a Gaussian initial state,

$$f(x) = x^2 - \frac{1}{2}x, \quad \Psi_0(x) = \left(\frac{1}{2\pi}\right)^{1/4} e^{-\frac{x^2}{4}},$$

as illustrated in Figure 5.1A. We use a truncated Fock space method with $N = 5$ levels to discretize the Hamiltonian operator and obtain a finite-dimensional sparse Hamiltonian. Next, we employ the penalty-free one-hot embedding to build the corresponding embedding Hamiltonian

$$H_{\text{real-space}} = \frac{1}{2}Q_{\hat{p}^2}^{\text{one-hot}} - \frac{1}{2}Q_{\hat{x}^2}^{\text{one-hot}} + \frac{1}{2}Q_{\hat{x}}^{\text{one-hot}}, \tag{5.35}$$

where $Q_{\hat{p}^2}^{\text{one-hot}}$, $Q_{\hat{x}^2}^{\text{one-hot}}$, and $Q_{\hat{x}}^{\text{one-hot}}$ are all 2-local Hamiltonians. More details of the truncation method and embedding are discussed in Appendix D.1.

In Figure 5.1B, we estimate and compare the gate counts required in the circuit implementation for the standard binary encoding and the penalty-free one-hot embedding. The extrapolation based on empirical data shows that our Hamiltonian embedding implementation uses fewer elementary gates and has slightly better asymptotic scaling in terms of system size, while the theoretical (i.e., worst-case) asymptotic scaling of our embedding does not show an advantage.

We simulate the embedding Hamiltonian $H_{\text{real}-\text{space}}$ on the IonQ Aria-1 processor and compute the expectation value of the position observable $\hat{x}$ as a function of time as shown Figure 5.1C. Notably, computing $\hat{x}$ only requires measurements in the x-basis when using Hamiltonian embedding. The same measurements using the standard binary code would require performing measurements in several different bases corresponding to the Pauli decomposition of $\hat{x}$, which demonstrates another advantage of Hamiltonian embedding. The experimental data matches the closed-form solution and we observe a full oscillation from $t = 0$ to $t = 5$. More details regarding the experiment setup and results are provided in Appendix D.1. Furthermore, we list in Table 5.3 the empirical resource usage for simulating the real-space Schrödinger equation with the one-hot code as well as an estimate of the resources required by the standard binary code. Although we consider only a 1-dimensional case in this example, the one-hot code demonstrates a significant advantage in gate count. In particular, the one-hot code only requires a single 1-qubit gate (for state preparation), while the standard binary code requires over 1800 single-qubit gates, which would not be feasible on current quantum hardware.

We also use programmable Rydberg atom arrays to simulate a 2-dimensional Schrödinger equation. First, we apply the finite difference method to discretize the Hamiltonian operator,

Encoding/embedding	# of qubits	# of 1-qubit gates	# of 2-qubit gates
Standard binary	3	1826	220
Penalty-free one-hot	5	1	154

Table 5.3: 1- and 2-qubit gate counts for simulating the real-space Schrödinger equation with $N = 5$ levels. The accuracy is chosen to correspond to the Trotter number $r = 11$ for the one-hot code as used in the real-machine experiment.

yielding a finite-dimensional Hamiltonian,

$$\hat{H} = -\frac{1}{2}(D \otimes I + I \otimes D) + U, \tag{5.36}$$

where D is an N-by-N tridiagonal matrix representing the finite-difference discretization of the second-order differential operator $\frac{\partial^2}{\partial x^2}$, I is the N-by-N identity matrix and U is a N^2-by-N^2 diagonal matrix corresponding to the potential field $f(x, y)$. The native Hamiltonian of Rydberg atom arrays allows us to form a Hamiltonian embedding of (5.36) using the antiferromagnetic scheme, while other schemes (such as unary) are not possible because the Rydberg interaction coefficients must be positive. The desired Hamiltonian embedding can be realized by arranging the atoms into two chains as shown in Figure 5.1D, where each chain represents an individual continuous variable (x for chain 1, y for chain 2). Note that the lack of local detuning in QuEra Aquila poses a significant restriction on the shape of the potential field. In our experiment, the *effective* potential field U and the penalty Hamiltonian are both engineered by the pairwise Rydberg interactions between atoms. As shown in Figure 5.1E, the QuEra-implemented quantum simulation results have good agreement with the numerical simulation. The quantum distributions are obtained using 1000 shots/measurements per time step (for $T = 0.5, 1, 2$). More details of the experiment are presented in Appendix D.2.

Chapter 6: Analog implementation of Quantum Hamiltonian Descent

Chapter summary. Analog quantum computing represents another significant paradigm in quantum computation. It addresses computational tasks by mimicking physical quantum systems, a process known as analog quantum simulation. In contrast to gate-based (digital) quantum computers, analog quantum computers are generally easier to control and scale. In this chapter, we introduce an implementation of Quantum Hamiltonian Descent with Ising-type analog quantum simulators (i.e., Quantum Ising Machines). By leveraging the Hamiltonian embedding technique proposed in Chapter 5, we propose a resource-efficient numerical scheme that implements Quantum Hamiltonian Descent to solve box-constrained nonlinear optimization problems. The proposed implementation method is deployed with the D-Wave quantum computer to solve up to 75-dimensional nonconvex quadratic programming problems. We empirically observe that the D-Wave-implemented Quantum Hamiltonian Descent outperforms a selection of state-of-the-art gradient-based classical solvers and the standard quantum adiabatic algorithm, based on the time-to-solution (TTS) metric.

6.1 Computational model and problem formulation

Among various types of analog quantum simulators, we will be focusing on an important class of simulators whose evolutions are effectively described by quantum Hamiltonians in the

Ising form:

$$H(t) = -\frac{A(t)}{2}\left(\sum_j \sigma_x^{(j)}\right) + \frac{B(t)}{2}\left(\sum_j h_j \sigma_z^{(j)} + \sum_{j>k} J_{j,k}\sigma_z^{(j)}\sigma_z^{(k)}\right), \tag{6.1}$$

where the operator $\sigma_\alpha^{(j)}$ for $\alpha = x, y, z$ stands for Pauli operator acting on the j-th qubit, $A(t)$ and $B(t)$ are time-dependent functions. Notable examples in this class include superconducting qubits spin glass simulators [154] and Rydberg atoms [155]. The technology of building such quantum simulators has seen a great advancement in the past decades, leading to several cloud-based commercial quantum processors (e.g., D-Wave, QuEra, etc). In order to facilitate further theoretical discussions, we propose a formal abstraction called the *quantum Ising machine* for quantum simulators with Ising-type Hamiltonian.

Definition 5. *A **Quantum Ising Machine** embodies an n-qubit quantum register which permits the following operations:*

1. ***Initialization**: the quantum register is initialized to certain quantum state, e.g., the all-zero state $|0\rangle^{\otimes n}$ or the uniform superposition state $|+\rangle^{\otimes n} = |\psi_0\rangle = \frac{1}{\sqrt{2^n}}\sum_{b\in\{0,1\}^n}|b\rangle$.*

2. ***Simulation**: the evolution of the quantum register is described by the Schrodinger equation,*

$$ih\frac{\mathrm{d}}{\mathrm{d}t}|\psi_t\rangle = H(t)|\psi_t\rangle, \tag{6.2}$$

   where h is the Planck constant, and the Hamiltonian $H(t)$ is given by (6.1). We assume $A(t)$, $B(t)$, h_j, $J_{j,k}$ are programable parameters.

3. ***Measurement**: the quantum register is measured in the computational basis, i.e., we can*

compute $|\langle b|\psi_t\rangle|^2$ *with* $b \in \{0,1\}^n$ *at certain time* $t \geq 0$.

In practice, it is often convenient to divide h on both sides of (6.2) and consider time-dependent functions $A(t)/h$ and $B(t)/h$. Usually, the time-dependent functions are given in giga-Hertz (1GHz = 10^9Hz); accordingly, the simulation time is given in nano-second (1ns=10^{-9}s).

Problem formulation.

We consider the following box-constrained optimization problem,

$$\min_x f(x_1,\ldots,x_n) = \underbrace{\sum_{i=1}^{n} g_i(x_i)}_{\text{univariate part}} + \underbrace{\sum_{j=1}^{m} p_j(x_{k_j})q_j(x_{\ell_j})}_{\text{bivariate part}}, \tag{6.3a}$$

$$s.t.\ L_i \leq x_i \leq U_i, \forall i \in \{1,\ldots,n\}, \tag{6.3b}$$

where $x_1,\ldots,x_n$ are n variables subject to the box constraint $x_i \in [L_i, U_i] \subset \mathbb{R}$ for each $i = 1,\ldots,n$, and the indices $k_j, \ell_j \in \{1,\ldots,n\}$ and $k_j \neq \ell_j$ for each $j = 1,\ldots,m$. The functions $g_i(x_i)$, $p_j(x_{k_j})$, and $q_j(x_{\ell_j})$ are real univariate differentiable functions defined on $\mathbb{R}$. We assume access to an abstract analog quantum simulator named **Quantum Ising Machine**, as detailed in Definition 5.

Note that the univariate part in (6.3a) has at most n terms because we can always combine separate univariate functions of a fixed variable x_i into a single one. However, there is no upper bound for the integer m (i.e., the number of bivariate terms). It is generally impossible to combine a sum of products into a single product form. For example, we can not find two univariate functions $p(x)$ and $q(y)$ such that $p(x)q(y) = \sin(x)y + xe^y$.

The problem (6.3) can be used to model several common classes of optimization problems

(with box constraints), including linear programming, quadratic programming, and polynomial optimization. We limit the appearance of trivariate parts or higher because of the current quantum hardware restrictions, while QHD can deal with arbitrary functions with powerful enough quantum devices.

6.2 Implementing QHD via Hamiltonian embedding

In this section, we discuss how to implement Quantum Hamiltonian Descent with Quantum Ising Machines. The key idea is to embed the QHD dynamics into a quantum evolution that is simulatable on a Quantum Ising Machine. There are two major steps in the implementation, namely, *spatial discretization* and *Hamiltonian embedding*.

Spatial discretization

First, we need to perform spatial discretization of the QHD Hamiltonian (which is an unbounded operator) so that it can be described by a finite-dimensional quantum system. Given a nonlinear optimization in the form of (6.3), the QHD Hamiltonian reads the following,

$$H(t) = e^{\varphi_t}\left(-\frac{1}{2}\Delta\right) + e^{\chi_t}\left(\sum_{i=1}^{n} g_i(x_i) + \sum_{j=1}^{m} p_j(x_{k_j})q_j(x_{\ell_j})\right),$$

which acts on any L^2-integrable functions over the feasible set $\Omega = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \colon L_i \leq x_i \leq U_i, \forall i = 1, \ldots, n\}$. Here, for simplicity, we assume the feasible set is the unit box, i.e., $L_i = 0$ and $U_i = 1$ for all $i = 1, \ldots, n$. We utilize the centered finite difference scheme to discretize this differential operator. Suppose that we divide each dimension of the unit box Ω using N quadrature points $\{0, h, \ldots, (N-2)h, 1\}$ (where $h = 1/(N-1)$), the resulting

discretized QHD Hamiltonian is an N^n-dimensional operator of the form,

$$\hat{H}(t) = e^{\varphi_t}\left(-\frac{1}{2}L_d\right) + e^{\chi_t}F_d, \tag{6.4}$$

where (assuming $k_j < \ell_j$ for all $j = 1, \ldots, m$)

$$L_d = \sum_{i=1}^{n} I \otimes \cdots \otimes \underbrace{L}_{\text{the } i\text{-th operator}} \otimes \ldots I,$$

$$F_d = \sum_{i=1}^{n} I \otimes \cdots \otimes \underbrace{D(g_i)}_{\text{the } i\text{-th operator}} \otimes \ldots I + \sum_{j=1}^{m} I \otimes \cdots \otimes \underbrace{D(p_j)}_{\text{the } k_j\text{-th operator}} \otimes \cdots \otimes \underbrace{D(q_j)}_{\text{the } \ell_j\text{-th operator}} \otimes \ldots I.$$

Here, I is the N-dimensional identity operator, L and $D(g)$ are N-dimensional matrices given by (g is a differentiable function defined on $[0, 1]$ and $g_i \coloneqq g(ih)$ for $i = 0, \ldots, N-1$),

$$L = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ \ldots & \ldots & \ldots & \ldots \\ & 1 & -2 & 1 \\ & & 1 & -2 \end{bmatrix}, \quad D(g) = \begin{bmatrix} g_0 & & & \\ & g_1 & & \\ \ldots & \ldots & \ldots & \ldots \\ & & g_{N-2} & \\ & & & g_{N-1} \end{bmatrix}. \tag{6.5}$$

The tridiagonal L matrix corresponds to the finite difference discretization of the second-order differential operator $\frac{d^2}{dx^2}$, and the diagonal matrix $D(g)$ corresponds to the finite difference discretization of the univariate function $g(x)$. Note that L has a global phase $-2/h^2$, i.e., $L = L' - 2/h^2$ with L' only contains the off-diagonal part of L. Since the global phase does not affect the quantum evolution (therefore, the result of the QHD algorithm), we replace L with L' in the

rest of the discussion.

Hamiltonian embedding

The discretized QHD Hamiltonian, as described in (6.4), is a Hermitian matrix with an explicit tensor product decomposition structure. This particular structure allows us to leverage the Hamiltonian embedding technique to construct a surrogate Hamiltonian $\tilde{H}(t)$ such that the QHD algorithm (i.e., simulating the Hamiltonian $\hat{H}(t)$) can be executed by simulating $\tilde{H}(t)$. In our case, the surrogate Hamiltonian $\tilde{H}(t)$ is an Ising-type quantum Hamiltonian that involves at most nN qubits and $\max(n, m)N$ two-body interaction terms. This means $\tilde{H}(t)$ can be efficiently simulated on current quantum computers, including IonQ's trapped ion systems and D-Wave's quantum annealer.

To construct the Hamiltonian embedding of $\hat{H}(t)$, the first step is to build the Hamiltonian embeddings of the N-by-N matrices L' and $D(g)$ (for arbitrary differentiable function g). Both are sparse matrices so we can utilize the *unary* and *one-hot* embedding schemes provided in Chapter 5.

We denote the integer r to be the number of qubits used to embed an N-dimensional matrix. Then, we write $\mathcal{E}^{(i)}[A]$ as a Hamiltonian embedding of an N-by-N Hermitian matrix A acting on sites $(i-1)r, (i-1)r+1, \ldots, ir-1$, where $i = 1, \ldots, n$. Then, using the rules of building Hamiltonian embeddings (see Section 5.3), we obtain a nr-qubit Hamiltonian that embeds the discretized QHD Hamiltonian $\hat{H}(t)$,

$$\tilde{H}(t) = e^{\varphi_t}\left(-\frac{1}{2}\sum_{i=1}^{n}\mathcal{E}^{(i)}[L']\right) + e^{\chi_t}\left(\sum_{i=1}^{n}\mathcal{E}^{(i)}[D(g_i)] + \sum_{j=1}^{m}\mathcal{E}^{(k_j)}[D(p_j)]\mathcal{E}^{(\ell_j)}[D(q_j)]\right). \quad (6.6)$$

Example 1 (One-hot embedding for x_1x_2)**.** *We give a simple example for the Hamiltonian embedding of the discretized QHD Hamiltonian when the objective function is* $f(x_1, x_2) = x_1x_2$*. This objective only involves a single bivariate term with* $p(x) = q(x) = x$*. We use the one-hot embedding with* $N = r = 3$*. The operators* $\mathbf{X}_k$*,* $\mathbf{Y}_k$*, and* $\mathbf{n}_k$ *are the Pauli-X, Pauli-Y, and number operator acting at site* k*, respectively,*

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

Then, the Hamiltonian embeddings of L' *and* $D(x)$ *are,*

$$\mathcal{E}[L'] = \frac{1}{2h^2}\left(\mathbf{X}_0\mathbf{X}_1 + \mathbf{X}_1\mathbf{X}_2 + \mathbf{Y}_0\mathbf{Y}_1 + \mathbf{Y}_1\mathbf{Y}_2\right), \quad \mathcal{E}[D(x)] = \mathbf{n}_0 + \frac{1}{2}\mathbf{n}_1,$$

respectively. As a result, the full Hamiltonian embedding reads

$$\begin{aligned}\tilde{H}(t) =& \frac{2e^{\varphi_t}}{h^2}\left(\mathbf{X}_0\mathbf{X}_1 + \mathbf{X}_1\mathbf{X}_2 + \mathbf{X}_3\mathbf{X}_4 + \mathbf{X}_4\mathbf{X}_5 + \mathbf{Y}_0\mathbf{Y}_1 + \mathbf{Y}_1\mathbf{Y}_2 + \mathbf{Y}_3\mathbf{Y}_4 + \mathbf{Y}_4\mathbf{Y}_5\right) + \\ & e^{\chi_t}\left(\mathbf{n}_0 + \frac{1}{2}\mathbf{n}_1\right)\left(\mathbf{n}_3 + \frac{1}{2}\mathbf{n}_4\right).\end{aligned}$$

6.3 Hamming embedding for quadratic programming

In the previous section, we discussed a general recipe for implementing QHD via Hamiltonian embedding to solve box-constrained nonlinear optimization problems of the form (6.3). In this section, we introduce a specific Hamiltonian embedding technique that works exclusively for box-constrained quadratic programming (QP) problems, as formulated below,

$$\text{minimize} \qquad f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{b}^\top \mathbf{x}, \tag{6.7a}$$

$$\text{subject to} \qquad \mathbf{0} \preccurlyeq \mathbf{x} \preccurlyeq \mathbf{1}, \tag{6.7b}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\mathbf{0}$ and $\mathbf{1}$ are n-dimensional vectors of all zeros and all ones, respectively. QP problems are the simplest case of nonlinear programming and they appear in almost all major fields in the computational sciences [1, 156]. Despite their simplicity and ubiquity, non-convex QP problems (i.e., ones in which the Hessian matrix $\mathbf{Q}$ is indefinite) are known to be NP-hard [45] in general.

As we have shown, the implementation of QHD on Quantum Ising Machines eventually boils down to the construction of Hamiltonian embeddings for the square matrices L (or L') and $D(g)$ (see (6.5) for more details). In a quadratic programming problem, we will only need to consider identity and quadratic functions, i.e., $g(x) = x$ and $g(x) = x^2$. In these cases, the finite difference discretization of the two functions yields the following diagonal matrices,

$$D(x) = \sum_{j=0}^{r} \frac{j}{r} |j\rangle \langle j|, \qquad D(x^2) = \sum_{j=0}^{r} \frac{j^2}{r^2} |j\rangle \langle j|, \tag{6.8}$$

where r is the number of quadrature points used for the discretization of each continuous variable.

Now, we introduce a specific embedding scheme named **Hamming encoding** (or **Hamming embedding**) that can be used to construct the Hamiltonian embedding of $D(x)$, $D(x^2)$, and an approximate version of L'.

Definition 6 (Hamming states)**.** *The Hilbert space of r qubits is $\widetilde{\mathcal{H}} = \mathbb{C}^{2^r}$ with the standard*

computational basis $\{|b\rangle : b \in \{0,1\}^r\}$. *The Hamming weight of a computational basis* $|b\rangle$ *is the number of ones in the bit-string* b*, denoted by* $|b|$.

Given an integer $j = 0, 1, ..., r$*, we define the* j*-th* ***Hamming state*** *as the uniform superposition over all computational basis of Hamming weight* j*, i.e.,*

$$|H_j\rangle := \frac{1}{\sqrt{C_j}} \sum_{|b|=j} |b\rangle \,, \tag{6.9}$$

where $C_j = \binom{r}{j}$.

Lemma 9. *Hamming states are of norm* 1 *and mutually orthogonal, i.e., for* $j, k = 0, 1, ..., r$*,*

$$\langle H_j|H_k\rangle = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases} \tag{6.10}$$

Proof. By the definition of Hamming states,

$$\langle H_j|H_k\rangle = \frac{1}{\sqrt{C_j C_k}} \sum_{|b|=j,|b'|=k} \langle b|b'\rangle \,.$$

If $j \neq k$, we have $\langle b|b'\rangle = 0$ so that $\langle H_j|H_k\rangle = 0$. If $j = k$, $\langle H_j|H_k\rangle = (\sum_{|b|=j} 1)/C_j = 1$. □

We consider the subspace $\mathcal{S} \subset \widetilde{\mathcal{H}}$ spanned by all $(r+1)$ Hamming states. By Lemma 9, the set of Hamming states form an orthonormal basis of $\mathcal{S}$.

Lemma 10. *The uniform superposition state* $|s\rangle = |+\rangle^{\otimes r} = \frac{1}{\sqrt{2^r}} \sum_{b\in\{0,1\}^r} |b\rangle$ *is in the subspace*

$\mathcal{S}$. *In particular, we have*

$$|s\rangle = \sum_{j=0}^{r-1} \sqrt{\frac{C_j}{2^r}} \, |H_j\rangle \, . \tag{6.11}$$

Proof.

$$\sum_{j=0}^{r-1} \sqrt{\frac{C_j}{2^r}} \, |H_j\rangle = \sum_{j=0}^{r-1} \sqrt{\frac{C_j}{2^r}} \left(\frac{1}{\sqrt{C_j}} \sum_{|b|=j} |b\rangle \right) = \frac{1}{\sqrt{2^r}} \sum_{j=0}^{r-1} \sum_{|b|=j} |b\rangle = |s\rangle \, .$$

□

Lemma 10 shows that a Quantum Ising Machine can be directly initialized to the subspace $\mathcal{S}$. Also, it shows that the superposition state $|s\rangle$ is a *discrete Gaussian state* in the Hamming basis.

Lemma 11. *We denote* $S_x = \sum_{j=0}^{r-1} \sigma_x^{(j)}$ *and* $S_z = \sum_{j=0}^{r-1} \sigma_z^{(j)}$. *The subspace* $\mathcal{S}$ *is an invariant subspace of both* S_x *and* S_z. *Furthermore,*

$$S_x|_{\mathcal{S}} = \sum_{j=0}^{r-1} \sqrt{(j+1)(r-j)} \Big(\, |H_{j+1}\rangle \langle H_j| + |H_j\rangle \langle H_{j+1}| \, \Big), \tag{6.12}$$

$$S_z|_{\mathcal{S}} = \sum_{j=0}^{r-1} (r-2j) \, |H_j\rangle \langle H_j| \, . \tag{6.13}$$

Proof. Note that $\sigma_x^{(j)}$ acts on a computational basis state $|b\rangle$ by flipping the j-th qubit while keeping all other qubits unchanged. Therefore, S_x is a sum of all single qubit flips. Then

$$S_x \, |H_0\rangle = \sum_j \sigma_x^{(j)} \left|0^{\otimes r}\right\rangle = \sum_{|b|=1} |b\rangle = \sqrt{r} \, |H_1\rangle \, ,$$

and similarly,

$$S_x \left|H_r\right\rangle = \sum_j \sigma_x^{(j)} \left|1^{\otimes r}\right\rangle = \sum_{|b|=r-1} \left|b\right\rangle = \sqrt{r}\left|H_{n-1}\right\rangle .$$

Let $j \in \{1, 2, \ldots, r-1\}$. For any bit-string $|y\rangle$ of Hamming weight $|y| = j-1$, there are $r-j+1$ bit-strings of Hamming weight j that are mapped to $|y\rangle$ by a single bit flip. Similarly, for any bit-string $|z\rangle$ of Hamming weight $|z| = j+1$, there are $j+1$ bit-strings of Hamming weight j that are mapped to $|z\rangle$ by a single bit flip. It follows that

$$\begin{aligned}
S_x \left|H_j\right\rangle &= \frac{r-j+1}{\sqrt{C_j}} \sum_{|y|=j-1} \left|y\right\rangle + \frac{j+1}{\sqrt{C_j}} \sum_{|z|=j+1} \left|z\right\rangle \\
&= (r-j+1)\frac{\sqrt{C_{j-1}}}{\sqrt{C_j}}\left|H_{j-1}\right\rangle + (j+1)\frac{\sqrt{C_{j+1}}}{\sqrt{C_j}}\left|H_{j+1}\right\rangle \\
&= \sqrt{\frac{(r-j+1)^2 r!j!(r-j)!}{r!(j-1)!(r-j+1)!}}\left|H_{j-1}\right\rangle + \sqrt{\frac{(j+1)^2 r!j!(r-j)!}{r!(j+1)!(r-j-1)!}}\left|H_{j+1}\right\rangle \\
&= \sqrt{j(n-j+1)}\left|H_{j-1}\right\rangle + \sqrt{(j+1)(n-j)}\left|H_{j+1}\right\rangle .
\end{aligned}$$

This proves the first part of the proposition.

Note that σ_z maps $|0\rangle$ to itself, and flips the phase of $|1\rangle$: $\sigma_z \left|1\right\rangle = -\left|1\right\rangle$. For any bit-string $|b\rangle$ of Hamming weight j, we have

$$S_z \left|b\right\rangle = \sum_j \sigma_z^{(j)} \left|b\right\rangle = (r-j)\left|b\right\rangle - j\left|b\right\rangle = (r-2j)\left|b\right\rangle ,$$

and it follows that $S_z \left|H_j\right\rangle = (r-2j)\left|H_j\right\rangle$ for any $j = 0, 1, ..., r$. This proves the second part of the proposition. □

We now summarize what we have proven so far:

Proposition 3. *Let all the notations the same as above. We have*

1. *The operator* $\left(\frac{1}{2} - \frac{1}{2r}S_z\right)$ *is a* $(r, 0, 0)$*-embedding of* $D(x)$ *defined in (6.8);*

2. *The operator* $\left(\frac{1}{2} - \frac{1}{2r}S_z\right)^2$ *is a* $(r, 0, 0)$*-embedding of* $D(x^2)$ *defined in (6.8);*

3. *The operator* $\frac{1}{r^{1/2}}S_x$ *is a* $(r, 0, 0)$*-embedding of* $\tilde{L}$ *defined as*

$$\tilde{L} = \sum_{j=0}^{r-1} \sqrt{(j+1)(r-j)/r}\left(|j+1\rangle\langle j| + |j\rangle\langle j+1|\right). \tag{6.14}$$

In the above Hamiltonian embeddings, the embedding subspace is $\mathcal{S} \subset \widetilde{\mathcal{H}}$ *spanned by all* $(r+1)$ *Hamming states.*

Proposition 3 gives a Hamiltonian embedding of the tridigonal matrix $\tilde{L}$ using the off-diagonal (i.e., X) part of the Quantum Ising Machine Hamiltonian (6.1), at the cost of using an exponentialy small subspace $\mathcal{S}$ of the full Hilbert space $\widetilde{\mathcal{H}} = \mathbb{C}^{2^r}$. This seems to be an unavoidable comprosmise due to the limited programmability of Quantum Ising Machine. A nice property of the Hamming embedding scheme is that it does not require any penalty Hamiltonian, so we do not need to choose the penalty coefficient and do post-selection when using it on real devices.

Note that, precisely speaking, the matrix $\tilde{L}$ does not match the exact discretization matrix L (or L') as given in (6.5). Given this discrepancy, however, empirical results show that Hamming embedding has really extrodinary performance for quadratic programming, as detailed in Section 6.4. By using the Hamming embedding as described in Proposition 3 and the composition rules (see Section 5.3), we can construct any Hamiltonian embedding to represent a quadratic

programming problem as in (6.7). This feature has already been integrated into the software `QHDOPT` as detailed in Chapter 7.

6.4 Real-machine experiments

In this section, we implement Quantum Hamiltonian Descent to solve the box-constrained quadratic programming problem defined in (6.7).

We implement QHD on the D-Wave system[1], which instantiates a QIM and allows for the control of thousands of physical qubits with decent connectivity [157]. While existing libraries of QP benchmark instances (e.g., [158]) are natural candidates for our empirical study, most of them can not be mapped to the D-Wave system because of its limited connectivity. We instead create a new test benchmark with 160 randomly generated QP instances in various dimensions (5, 50, 60, 75) whose Hessian matrices are indefinite and sparse and for which analog implementations are possible on the D-Wave machine (referred to as DW-QHD).

We compare DW-QHD with 6 other state-of-the-art solvers in our empirical study: DW-QAA (baseline QAA implemented on D-Wave), IPOPT [159], SNOPT [160], MATLAB's `fmincon` (with `SQP` solver), QCQP [161], and a basic Scipy `minimize` function (with `TNC` solver). In the two quantum methods (DW-QHD, DW-QAA), we discretize the search space $[0, 1]^d$ into a regular mesh grid with 8 cells per edge due to the limited number of qubits in the D-Wave machine. To compensate for the loss of resolution, we post-process the coarse-grained D-Wave results by the Scipy `minimize` function, which is a local gradient solver mimicking the descent phase of a higher-resolution QHD and only has mediocre performance by itself. The choice of classical solvers covers a variety of state-of-the-art optimization methods, including gradient-based local

[1] We access the D-Wave `advantage_system6.1` through Amazon Braket.

search (Scipy `minimize`), interior-point method (IPOPT), sequential quadratic programming (SNOPT, MATLAB), and heuristic convex relaxation (QCQP). Finally, to investigate the quality of the D-Wave machine in implementing QHD and QAA, we also classically simulate QHD and QAA for the 5-dimensional instances (Sim-QHD, Sim-QAA).[2]

We use the *time-to-solution* (TTS) metric [82] to compare the performance of solvers. TTS is the number of trials (i.e., initializations for classical solvers or shots for quantum solvers) required to obtain the correct global solution[3] up to 0.99 success probability:

$$\text{TTS} = t_f \times \left\lceil \frac{\ln(1-0.99)}{\ln(1-p_s)} \right\rceil, \tag{6.15}$$

where t_f is the average runtime per trial, and p_s is the success probability of finding the global solution in a given trial. We run 1000 trials per instance and compute the TTS for each solver.

In Figure 6.1, we show the distribution of TTS for different solvers.[4] In the 5-dimensional case (Figure 6.1A), Sim-QHD has the lowest TTS, and the quantum methods are generally more efficient than classical solvers. Note that with a much shorter annealing time ($t_f = 1\mu s$ for Sim-QHD and $t_f = 800\mu s$ for DW-QHD) Sim-QHD still does better than DW-QHD, indicating the D-Wave system is subject to significant noise and decoherence. Interestingly, Sim-QAA ($t_f = 1\mu s$) is worse than DW-QAA ($t_f = 800\mu s$), which shows QAA indeed has much slower convergence. In the higher dimensional cases (Figure 6.1B,C,D), DW-QHD has the lowest median TTS among all tested solvers. Despite the infeasibility of running Sim-QHD in high dimensions, our obser-

[2]Note that we numerically compute Sim-QHD and Sim-QAA for $t_f = 1\mu s$, which is much shorter than the time we set in the D-Wave experiment (in DW-QHD and DW-QAA, we choose $t_f = 800\mu s$.

[3]For each test instance, the global solution is obtained by Gurobi [81].

[4]We also provide spreadsheets that summarize the test results for the QP benchmark, see `https://github.com/jiaqileng/quantum-hamiltonian-descent/tree/main/plot/fig4/qp_data`.

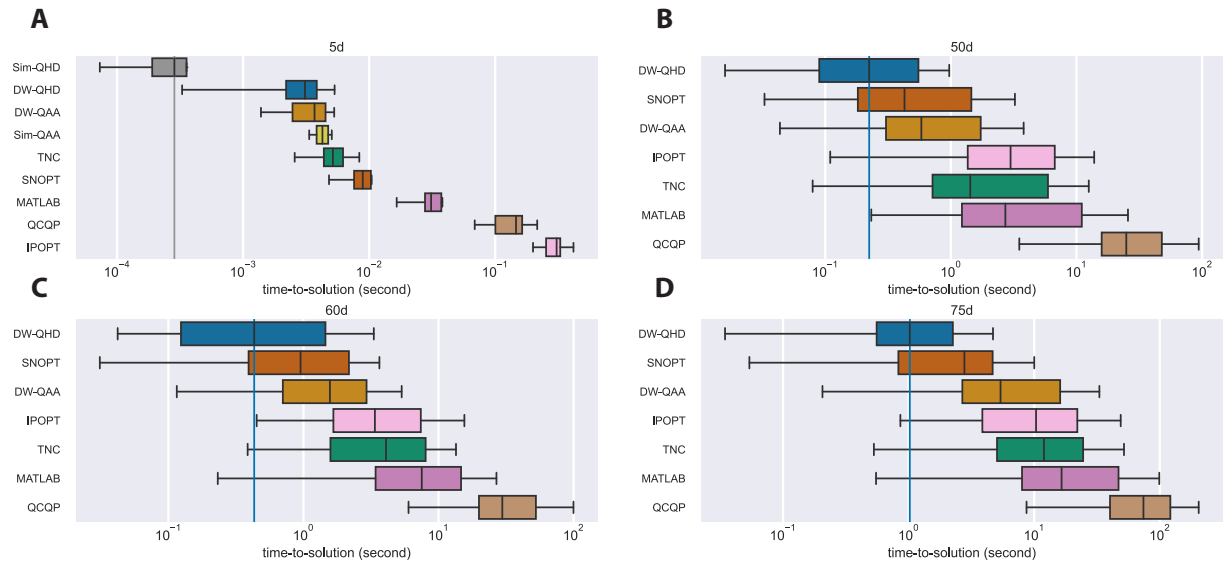


Figure 6.1: **Experiment results for quadratic programming problems.** Box plots of the time-to-solution (TTS) of selected quantum/classical solvers, gathered from four randomly generated quadratic programming benchmarks (**A**: 5-dimensional, **B**: 50-dimensional, **C**: 60-dimensional, **D**: 75-dimensional). The left and right boundaries of a box show the lower and upper quartiles of the TTS data, while the whiskers extend to show the rest of the TTS distribution. The median of the TTS distribution is shown as a black vertical line in the box. In each panel, the median line of the best solver extends to show the comparison with all other solvers.

vation in the 5-dimensional case suggests that an ideal implementation of QHD could perform much better than DW-QHD, and therefore all other tested solvers in high dimensions.

It is worth noting that DW-QHD does not outperform industrial-level nonlinear programming solvers such as Gurobi [81] and CPLEX [162]. In our experiment, Gurobi usually solves the high-dimensional QP problems with TTS no more than 10^{-2} s. These solvers approximate the nonlinear problem by potentially exponentially many linear programming subroutines and use a branch-and-bound strategy for a smart but exhaustive search of the solution. However, the restrictions of the D-Wave machine (e.g., programmability and decoherence) force us to test on very sparse QP instances, which can be efficiently solved by highly optimized industrial-level branch-and-bound solvers. On the other hand, we believe that QHD should be more appropri-

ately deemed as a quantum upgrade of classical GD, which would more conceivably replace the role of GD rather than the entire branch-and-bound framework in classical optimizers.

Chapter 7: Software design for quantum optimization

Chapter summary. In this chapter, we introduce an open-source, end-to-end software library named `QHDOPT` to facilitate the implementation of Quantum Hamiltonian Descent on real machines, including Quantum Ising Machines (such as the D-Wave quantum computer) and gate-based quantum computers (such as the IonQ quantum computer). `QHDOPT` eliminates the technical barrier to implementing quantum algorithms by offering an accessible interface and automatically maps tasks to various supported quantum backends. These features enable users, even those without prior knowledge or experience in quantum computing, to utilize the power of existing quantum devices for large-scale nonlinear and nonconvex optimization tasks. This software is available at `https://github.com/jiaqileng/QHDOPT`.

7.1 Review: the state of software for quantum optimization

Software packages are crucial for lowering the barrier to developing and implementing quantum programs across broad user communities. Upon examining the current landscape of quantum software for mathematical optimization, we observe that the majority of the software dedicated to quantum optimization focuses on addressing combinatorial and discrete optimization problems, with limited options available for continuous optimization.

Generally, a combinatorial optimization problem can be reformulated as a Quadratically

Unconstrained Binary Optimization (QUBO) problem, the solution of which is believed to be a promising application of quantum computing [163]. There is a rich collection of libraries for quantum and quantum-inspired optimization that can be employed to generate QUBO reformulations, including `QUBO.jl` [164], `Amplify` [165], `PyQUBO` [166], `qubovert` [167]. These QUBO problems can be tackled by several methods, such as quantum annealing, Quantum Approximate Optimization Algorithms (QAOA) [7], and other hybrid approaches [168]. D-Wave's `Ocean SDK` [169] enables users to interface with their direct QPU (i.e., quantum annealer) and hybrid solvers and retrieve results. QuEra's `Bloqade.jl` [170] is a high-level language for configuring programmable Rydberg atom arrays, which can be used to implement annealing-type quantum algorithms and discrete optimization problems like QUBO [171]. Los Alamos Advanced Network Science Initiative has also released a package named `QuantumAnnealing.jl` [172] for simulation and execution of quantum annealing. Besides, several software packages have been published for programming quantum circuits, including IBM's `Qiskit` [173], Google's `Cirq` [174], Amazon's `Braket SDK` [175], Microsoft's `Q#` [176], and Xanadu's `PennyLane` [177]. These tools can be used to deploy QAOA on gate-based quantum computers.

While there have been a few proposals for solving continuous optimization problems using quantum or hybrid computing devices such as photonic quantum computers [178] and coherent continuous variable machines [179], we are not aware of a software library customized for nonlinear continuous optimization problems. In practice, some nonlinear optimization problems, such as quadratic programming, may be reformulated as QUBO problems and handled by the aforementioned software tools. However, it remains unclear whether this approach could lead to robust quantum advantages.

7.2 Workflow of QHDOPT: Hamiltonian-Oriented Programming (HOP)

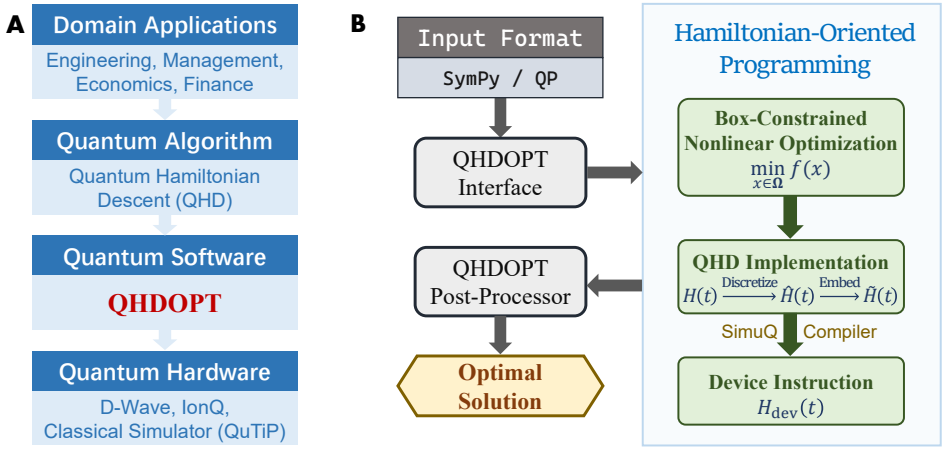


Figure 7.1: An Overview of QHDOPT. **A.** Building the stack of quantum computing for nonlinear optimization. **B.** The workflow of QHDOPT, inspired by the Hamiltonian-oriented programming paradigm.

QHDOPT exploits the quantum Hamiltonian descent (QHD) algorithm to solve nonlinear optimization problems. This quantum algorithm is formulated as a Hamiltonian simulation (i.e., simulating the evolution of a quantum-mechanical system), encompassing a novel abstraction of computation on quantum devices, which we call **Hamiltonian-Oriented Programming (HOP)**. In contrast to the conventional circuit-based quantum computation paradigm where theorists describe quantum algorithms in terms of quantum circuits, the HOP paradigm describes quantum algorithms as a single or a sequence of quantum Hamiltonian evolution. This new paradigm enables us to build a stack of quantum applications by leveraging the native programmability of quantum hardware in the development of quantum algorithms and software, as illustrated in Figure 7.1A.

In QHDOPT, we utilize SimuQ, a recent framework for programming and compiling quantum Hamiltonian systems by Peng et al. [180], as an intermediate layer for the programming

of QHD and leverage the `SimuQ` compiler to realize multi-backend compatibility. Through `SimuQ`, `QHDOPT` initially constructs a hardware-agnostic Hamiltonian representation of QHD (i.e., Hamiltonian embedding) that can be deployed on various quantum backends, including D-Wave devices, IonQ devices, and classical simulators via QuTiP [181]. The workflow of `QHDOPT`, as illustrated in Figure 7.1B, encompasses the following major steps:

- **Input format:** `QHDOPT` offers support for two Python-based input formats: the SymPy format for *symbolic* input and the QP format for *numerical* input (i.e., arrays). These two input formats enable users to define their target optimization problems both efficiently and with great flexibility. Figure 7.2 displays two Python code snippets for the two input formats.

- **Hamiltonian programming and compilation:** Once a nonlinear optimization problem $f(x)$ is defined using one of the supported input formats, `QHDOPT` will form a Hamiltonian description of the corresponding QHD algorithm, as described in Section 6.2. This Hamiltonian description serves as an intermediate layer in the compilation stack and is independent of the choice of the backend (i.e., hardware-agnostic).

- **Deployment on quantum devices:** When the Hamiltonian embedding of a given nonlinear optimization problem is built, it can be executed on a supported quantum backend by running `optimize()`. Currently, `QHDOPT` supports three backend devices for deployment, including classical simulators (e.g., QuTiP), IonQ, and D-Wave. The quantum measurement results are then retrieved from the executing backend in the form of bitstrings.

- **Classical post-processing**: Following the deployment step, `QHDOPT` implements a series of post-processing subroutines, including decoding the raw measurement results (i.e., bit-

strings) into low-resolution solutions and then refining these solutions via classical local optimizers. In the end, the refined solutions are returned to the users as final results.

```python
from qhdopt import QHD
from sympy import symbols, exp

x, y = symbols("x y")
f = y**1.5 - (y-0.75) * exp(4*x)
model = QHD.SymPy(f, [x, y])
```

(a) An example using the SymPy input format

```python
from qhdopt import QHD

Q = [[-8, 3],
     [3, -4]]
b = [3, -1]
model = QHD.QP(Q, b)
```

(b) An example using the QP input format

Figure 7.2: Input formats in QHDOPT

7.3 Examples using QHDOPT

Now, we exhibit two simple examples showcasing the use cases of QHDOPT.

- **Quadratic programming.** In Figure 7.3, we exemplify using QHDOPT to solve a quadratic programming problem with all three backends.[1] By default, `QHD.optimize()` automatically executes the Scipy TNC method to fine-tune the raw quantum measurement data. To switch to the Ipopt optimizer in the fine-tuning step, one can specify `post_processing_method="IPOPT"` in the setup.

- **Nonlinear programming.** In Figure 7.4a, we illustrate a sample code that runs QHDOPT to solve a nonlinear optimization problem involving a fractional power and an exponential function. The function is constructed using the SymPy input format and then deployed on the D-Wave quantum computer with the resolution parameter $r = 8$. We may set

[1] Note that the API key (not included in QHDOPT) is required to access cloud-based quantum computers such as D-Wave and IonQ.

`verbose=1` to print a detailed summary of this execution, including the best-so-far coarse and fine-tuned solutions, as well as a total runtime breakdown, as shown in Figure 7.4b.

```python
from qhdopt import QHD

# Using QP input format
Q = [[-2, 1],[1, -1]]
b = [3/4, -1/4]
model = QHD.QP(Q, b, bounds=(0,1))

# Deployment # 1: D-Wave (default embedding scheme: unary)
model.dwave_setup(8, api_key="DWAVE_API_KEY")
model.optimize()

# Deployment # 2: IonQ (default embedding scheme: one-hot)
model.ionq_setup(6, api_key="IONQ_API_KEY",
 ↪  time_discretization=30)
model.optimize()

# Model 3: QuTiP simulator (default embedding scheme: one-hot)
model.qutip_setup(6, post_processing_method="IPOPT")
model.optimize()
```

Figure 7.3: Solving a quadratic programming problem using `QHDOPT`

```python
from sympy import symbols, exp
from qhdopt import QHD

# Using SymPy input format
x, y = symbols("x y")
f = y**1.5 - exp(4*x) *
 ↪  (y-0.75)
model = QHD.SymPy(f, [x, y],
                  bounds=(0,1))

# Deploying QHD on the D-Wave
 ↪  device
model.dwave_setup(8,
 ↪  api_key="API_KEY")
model.optimize(verbose=1)
```

(a) Deploying QHD on D-Wave

```
* Coarse solution
    Minimizer: [1. 0.
     ↪  1.]
    Minimum: -1.0
* Fine-tuned solution
    Minimizer: [1. 0.
     ↪  1.]
    Minimum: -1.0
    Success rate: 1.0
* Runtime breakdown
    SimuQ compilation:
     ↪  0.000 s
    Backend QPU runtime:
     ↪  0.119 s
    Backend overhead
     ↪  time: 3.825 s
    Decoding time: 0.019
     ↪  s
    Fine-tuning time:
     ↪  0.161 s
* Total time: 4.124 s
```

(b) Execution summary

Figure 7.4: Solving a nonlinear optimization problem using `QHDOPT`. The backend overhead time includes network transmission time, queue time, etc.

7.4 Comparison with existing tools

In this section, we conduct an empirical study with QHDOPT to understand to what extent the noisy implementation of Quantum Hamiltonian Descent improves solution quality, compared with other classical local optimizers.

To this end, we have designed a benchmark with ten nonlinear optimization instances and evaluated the performance of QHDOPT using this benchmark. All test instances in this benchmark are nonconvex optimization problems with unit box constraints, as detailed in Table 7.1. The first 3 instances are high-dimensional sparse quadratic programming (QP) problems drawn from the benchmark devised in [40].[2] The rest 7 instances are nonlinear problems in the form of (6.3) that involve 2 to 3 continuous variables. These instances were generated in a largely random manner and each possesses multiple local solutions, making them particularly challenging for classical local solvers.

Test index	Problem description	Test index	Problem description
1	instance 0 in qp-50d-5s (50-dimensional QP)	6	$f(x,y) = y^{3/2} - e^{4x}\left(y - \frac{3}{4}\right)$
2	instance 0 in qp-60d-5s (60-dimensional QP)	7	$f(x,y,z) = x^2 - 3y^2 + 2\sin\left(\frac{3\pi}{2}x\right)\cos\left(\frac{3\pi}{2}y\right) + x - 3y$
3	instance 0 in qp-75d-5s (75-dimensional QP)	8	$f(x,y,z) = (2y-1)^2\left(z - \frac{2}{5}\right) - (2x-1)z + y\left(2x - \frac{3}{2}\right)^2$
4	$f(x,y) = -4x^2 + 3xy - 2y^2 + 3x - y$	9	$f(x,y,z) = 2e^{-x} * (2z-1)^2 - 3\left(2y - \frac{7}{10}\right)^2 e^{-z} + \log(x+1)\left(y - \frac{4}{5}\right)$
5	$f(x,y) = -2\left(x - \frac{1}{3}\right)^2 + y^2 - \frac{1}{3}y\log\left(3x + \frac{1}{2}\right) + 5\left(x^2 - y^2 - x - \frac{1}{2}\right)^2$	10	$f(x,y,z) = x\sin(2\pi y) - \frac{1}{2}\left(y - \frac{3}{10}\right)^2\cos(3\pi z) + \sin(4\pi x)$

Table 7.1: Ten test instances in the designed nonlinear optimization benchmark. All the test instances are nonconvex problems with unit box constraints $[0,1]^n$.

We evaluate QHDOPT on this benchmark using the D-Wave Advantage_system6.3 as the quantum backend. For a fair comparison, we use the unary embedding scheme for all instances,

[2]The QP benchmark introduced in [40] contains 160 nonconvex quadratic programming problems. The sizes of these instances vary, including dimensions of 5, 50, 60, and 75. The description of these test instances and their optimal solutions are available via https://umd.app.box.com/s/vq747fvjnt8qrkbxprexhoh44n0q9m0i/folder/180948948670.

including quadratic and non-quadratic problems. We test QHDOPT with two post-processing optimizers (i.e., Ipopt, Scipy-TNC) and measure the quantum runtime ("Quan. Rtime"), average post-processing time ("PP"), success probability ("SP"). Here, the quantum runtime refers to the QPU execution time reported by D-wave, in which the transmission time and the task queuing time are not included. A result x' is considered successful if the optimality gap $f(x') - f(x^*)$ is less than 0.001, where x^* is the best solution.[3] Using this data, we calculate the time-to-solution ("TTS"),

$$\text{TTS} = t_0 \times \left\lceil \frac{\ln(1 - 0.99)}{\ln(1 - p_s)} \right\rceil,$$

where t_0 is the (average) runtime per shot[4], and p_s is the success probability. TTS is interpreted as the total runtime required to achieve at least 0.99 success probability, with a lower TTS indicating better overall performance. As a comparison, we also test the two classical optimizers independently with random initialization and measure the average runtime ("Rtime"), success probability ("SP"), and the corresponding time-to-solution ("TTS"). These two classical optimizers are assessed as baselines to illustrate the quantum advantage introduced by the D-Wave-implemented quantum Hamiltonian descent (QHD). The collected data is displayed in Table 7.2.

Test	Quan.	QHD+Ipopt			QHD+TNC			Ipopt			TNC		
Index	Rtime	PP	SP	TTS	PP	SP	TTS	Rtime	SP	TTS	Rtime	SP	TTS
1	2.24e-3	1.31e+1	1.04e-1	5.48e+2	1.09e+0	7.00e-2	6.93e+1	5.26e+1	1.92e-1	1.14e+3	7.26e+0	7.00e-3	4.76e+3
2	2.28e-3	1.24e+1	7.10e-2	7.76e+2	1.49e+0	7.40e-2	8.93e+1	8.58e+1	6.90e-2	5.53e+3	6.82e+0	3.00e-3	1.05e+4
3	2.28e-3	9.69e+0	1.00e-3	4.46e+4	1.21e+0	1.00e-3	5.57e+3	1.52e+2	0.00e+0	N/A	1.16e+1	0.00e+0	N/A
4	1.15e-3	2.03e-1	9.96e-1	2.05e-1	3.19e-2	9.84e-1	3.68e-2	1.33e+1	5.72e-1	7.19e+1	2.82e-1	5.64e-1	1.57e+0
5	9.97e-4	3.16e-1	9.23e-1	5.69e-1	5.24e-2	9.12e-1	1.11e-1	1.29e+1	5.44e-1	7.56e+1	5.86e-1	5.15e-1	3.73e+0
6	1.08e-3	3.35e-1	9.40e-1	5.50e-1	2.96e-2	9.82e-1	3.52e-2	2.09e+1	6.78e-1	8.51e+1	6.85e-1	5.61e-1	3.83e+0
7	1.15e-3	2.04e-1	1.00e+0	2.05e-1	3.96e-2	9.99e-1	3.85e-2	2.99e+1	8.75e-1	6.63e+1	4.84e-1	8.79e-1	1.05e+0
8	1.25e-3	1.60e+0	7.98e-1	4.60e+0	1.28e-1	8.67e-1	2.95e-1	9.74e+0	7.54e-1	3.20e+1	5.31e-1	6.87e-1	2.10e+0
9	1.23e-3	3.64e-1	9.62e-1	5.14e-1	5.75e-2	9.82e-1	6.73e-2	1.43e+1	6.27e-1	6.67e+1	6.42e-1	6.23e-1	3.03e+0
10	9.49e-4	5.99e+0	8.60e-1	1.40e+1	1.73e-1	8.43e-1	4.33e-1	8.83e+1	3.05e-1	1.12e+3	1.54e+0	3.13e-1	1.88e+1

Table 7.2: Performance of QHDOPT and baseline methods, tested using the designed benchmark. The unit of time is second. In each row, the lowest time-to-solution is underlined.

[3] We use the best solution found by Ipopt as the best solution. For each test instance, we ran 1000 trials of Ipopt, each with a random initial guess within the feasible set.

[4] More precisely, a *shot* is defined as generating one quantum sample and performing subsequent post-processing on it.

We observe that QHDOPT with TNC as a post-processing subroutine always shows the lowest TTS among the other three methods. This is potentially a result of better initial guesses (i.e., QHD samples retrieved from the D-Wave device) and fast post-processing. As we can see, compared with Ipopt, TNC has a notably faster runtime, because TNC is a quasi-Newton method that does not need to solve the full Newton linear system in the iterations. An interesting finding is that the post-processing times ("PP", mainly for running classical fine-tuning subroutines) in the first two methods (QHD hybridized with classical local solvers) are significantly shorter than those in the last two methods (purely classical local solvers). This means that the solutions retrieved from the D-Wave device, though of low resolution and affected by the hardware noise, are still significantly better than random guesses. This implies that the QHD algorithm introduces a genuine quantum advantage into the workflow of QHDOPT, despite the current challenges of limited quantum resources and prevalent hardware noise.

Chapter 8: Conclusion

In this thesis, we have studied a quantum algorithm named Quantum Hamiltonian Descent (QHD). This is a quantum algorithm for continuous optimization problems, with exceptional performance for nonconvex problems.

First, we derive Quantum Hamiltonian Descent by quantizing a family of classical mechanical systems that are used to model accelerated gradient descent algorithms. As detailed in Chapter 2, the quantization is based on Feynman's path integral formulation of quantum mechanics. For convex optimization problems, Quantum Hamiltonian Descent inherits the same convergence rate as its classical counterpart while exhibiting robustness in time discretization. For nonconvex problems, the behavior of Quantum Hamiltonian Descent is radically different from any classical gradient method. This is because Quantum Hamiltonian Descent can be regarded as the path integral of all possible descent trajectories, including those prohibited by the high-energy barrier on the optimization landscape. The interference of paths gives rise to the quantum tunneling effect, which helps the quantum system escape from sub-optimal energy configurations and eventually reach the global minimum.

Next, in Chapter 3, through the study of Schrödinger operators, we further investigate the practical quantum advantage of Quantum Hamiltonian Descent for nonconvex optimization problems. Our main theoretical results are as follows:

1. Under mild assumptions, we prove that Quantum Hamiltonian Descent finds the global minimum of any nonconvex optimization problem, provided that the evolution time is long enough. However, this does not mean Quantum Hamiltonian Descent can efficiently solve any nonconvex optimization problems, as the evolution time could be exponentially long.

2. We show that Quantum Hamiltonian Descent can be used to find a marked item in a size-N structured database using $\tilde{O}(\sqrt{N})$ queries to the database. The idea is to reformulate the structured search problem as a nonconvex optimization problem with N local minima, with the unique global minimum corresponding to the marked item. This result can be interpreted as a quadratic query separation between quantum and classical computation.

3. By constructing a class of nonconvex optimization instances with exponentially many local minima, we demonstrate a super-polynomial performance separation between Quantum Hamiltonian Descent and several state-of-the-art classical optimizers. Specifically, Quantum Hamiltonian Descent can solve these optimization instances in polynomial time, while all the tested classical optimizers appear to require super-polynomial run time to solve the same set of problems up to a comparable quality.

Besides the theoretical analysis, we have proposed two approaches to implementing Quantum Hamiltonian Descent on digital and analog quantum computers. The digital implementation of Quantum Hamiltonian Descent, based on the standard product formula for simulating quantum evolution, has been introduced in Chapter 4. The analog implementation of Quantum Hamiltonian Descent relies on the Hamiltonian embedding technique, which admits quantum simulation of sparse Hamiltonian in analog and near-term quantum devices, as detailed in Chapter 5. Through the Hamiltonian embedding technique, we show that a Quantum Hamiltonian Descent

can be mapped to an Ising-type quantum Hamiltonian, which can be directly simulated in several analog quantum emulators. The detailed mapping procedures and a large-scale real-machine demonstration have been presented in Chapter 6.

Last but not least, we have developed a software library named `QHDOPT` that automatically deploys Quantum Hamiltonian Descent to solve nonlinear optimization problems on various quantum backend machines, including gate-based digital quantum computers (e.g., IonQ) and analog quantum computers (e.g., D-Wave). `QHDOPT` is designed following the Hamiltonian-Oriented Programming (HOP) paradigm, where the quantum Hamiltonian is the central object in building the stack of quantum software. The designing principles and some high-level discussions of `QHDOPT` are provided in Chapter 7. The source code can be found in the open-source repository: `https://github.com/jiaqileng/QHDOPT`.

This research opens up new avenues for accelerating the solution of nonconvex optimization problems by leveraging quantum computers. In particular, the development of real-machine implementation techniques and automated software toolchains has extended the scope of this research beyond purely theoretical investigations, leading to a viable pathway toward practical quantum advantage.

Appendix A: Quantum and classical algorithms for continuous optimization

We test four algorithms for the 22 two-dimensional non-convex optimization instances, including two quantum algorithms and two classical algorithms: (1) Quantum Hamiltonian Descent (QHD), (2) Quantum Adiabatic Algorithms (QAA), (3) Nesterov's accelerated gradient descent (NAGD), and (4) stochastic gradient descent (SGD). In our experiment, the two quantum algorithms are simulated numerically on classical computers so we can visualize the evolution of the probability distribution.

Quantum Hamiltonian Descent (QHD)

As a reminder, the QHD Hamiltonian (2.24) is

$$H(t) = e^{\varphi_t}\left(-\frac{1}{2}\nabla^2\right) + e^{\chi_t} f(x).$$

The QHD evolution from $t = 0$ to $t = T$ is simulated by iteratively applying the product formula,

$$|\Psi_{j+1}\rangle = \left[\exp\left(-isa_j(-\nabla^2/2)\right)\exp(-isb_j f)\right]|\Psi_j\rangle\,, \tag{A.1}$$

where $j = 0, ..., N-1$, $s = T/N$ is the stepsize, and $a_j = e^{\varphi_{js}}$, $b_j = e^{\chi_{js}}$. We choose the initial

state $|\Psi_0\rangle$ to be the the uniform random distribution over the domain $[0,1]^2$.

We observe that the operator f is diagonal in the position basis, and ∇^2 is diagonalized by the Fourier basis, i.e., $-\frac{1}{2}\nabla^2 = \mathbf{F}_s \mathbf{L} \left(\mathbf{F}_s\right)^{-1}$, where $\mathbf{F}_s$ is the shifted Fourier transform (SFT)[1], and $\mathbf{L}$ is a diagonal matrix with eigenvalues of $(-\nabla^2/2)$. These eigenvalues can be computed from the frequency number of a Fourier basis function [76, Eq.(26)]. This means that each iteration step can be implemented by

$$|\Psi_{j+1}\rangle = \left[\mathbf{F}_s \exp(-isa_j\mathbf{L}) \left(\mathbf{F}_s\right)^{-1} \exp(-isb_j f)\right] |\Psi_j\rangle \,. \tag{A.2}$$

In the literature, this method is often known as the pseudo-spectral method [182] and it is a standard numerical method for Schrödinger equations.

Experiment parameters. We choose the time-dependent functions φ_t and χ_t so that they are the same as in the ODE model of the Nesterov's accelerated gradient descent algorithm [30]. To avoid the singularity at $t = 0$ when performing numerical simulation, we slightly modify the first time-dependent function and the QHD Hamiltonian becomes

$$H(t) = \left(\frac{2}{s+t^3}\right)\left(-\frac{1}{2}\nabla^2\right) + 2t^3 f(x), \tag{A.3}$$

where s is the stepsize. We choose $s = 0.001$ for all test instances, and the total evolution time is $T = 10$, hence the maximal iteration number $N = T/s = 10^4$. We try two resolutions, 128 and 256, in the spatial discretization of QHD, i.e., we simulate QHD on a 128×128 and a 256×256 mesh grid over the unit square $[0,1]^2$. The initial state is the equal uniform superposition of all

[1] SFT is a close variant of the Fast Fourier Transform (FFT) and can be readily implemented from FFT.

points.

Quantum Adiabatic Algorithm (QAA)

Quantum Adiabatic Algorithm (QAA) [52] is a well-known quantum algorithm for general optimization problems. QAA is formulated as a quantum adiabatic evolution described by the Schrödinger equation ($0 \leq t \leq T$):

$$i\frac{\mathrm{d}}{\mathrm{d}t}\left|\psi(t)\right\rangle = \Big[\big(1-g(t)\big)H_0 + g(t)H_1\Big]\left|\psi(t)\right\rangle, \tag{A.4}$$

where $g(t)$ is the annealing schedule such that $g(0) = 0$ and $g(T) = 1$, H_0 is the initial Hamiltonian, and H_1 is the problem Hamiltonian that encodes the problem of interest. If we choose the initial state $|\psi(0)\rangle$ as the ground state of H_0 and let the evolution time T be large enough, the quantum state $|\psi(t)\rangle$ remains the ground state of the system in the whole evolution, and the final state $|\psi(T)\rangle$ corresponds to a solution to the problem of interest.

Conventionally, QAA is defined over discrete domains and thus applicable to discrete optimization problems.[2] The performance of QAA heavily depends on the choice of the initial Hamiltonian H_0. Prior to QHD, the standard approach for solving continuous problems using QAA is to adopt Radix-2 representation of real numbers (i.e., binary numeral system) in the continuous domain so that the original problem is converted to a discrete optimization problem defined over $\{0,1\}^N$, e.g., [98, 97]. In this case, the initial Hamiltonian is chosen to be the "sum

[2]Note that there is no direct or natural counterpart of gradient descent/QHD in discrete domains.

of Pauli-X" operator:

$$H_0 = -\sum_{j=1}^{N} \sigma_x^{(j)}, \tag{A.5}$$

whose ground state is the uniform superposition state, and H_1 is a diagonal matrix whose diagonal elements are evaluated from the objective functions. We will refer to this traditional approach of solving continuous problems using QAA as the **baseline QAA**, or simply QAA.

In our experiment, we simulate the baseline QAA using the leapfrog scheme for time integration. Leapfrog integrators are time-reversible and symplectic [183], so lend themselves well to the simulation of unitary dynamics.

The choice of the annealing schedule $g(t)$ has a huge impact on the performance of QAA. In practice, it is common to use the linear interpolation schedule $g(t) = \frac{t}{T}$, while sometimes it fails to achieve optimal results. For example, to achieve the quadratic quantum speedup for the unstructured search problem, one needs to use the local adiabatic schedule (see Figure A.1A) instead of the plain linear interpolation [184]. For our 2D test problems, however, we find that the linear interpolation schedule works better than the local adiabatic schedule, potentially due to the limited evolution time and the continuous nature of the original problem. In Figure A.1B, we show the success probability of the quantum adiabatic evolution (applied to the Levy function). It is clear that the local adiabatic schedule from [184] does not keep the state in the ground-energy subspace and it is outperformed by the standard linear interpolation schedule at $T = 10$. Actually, we also tried the prolonged evolution time $T = 100$ for the local adiabatic schedule but it is still worse than the linear interpolation one. For other test instances, we also observe similar behavior of the local adiabatic schedule. Therefore, we choose the linear interpolation schedule

in our numerical experiment.

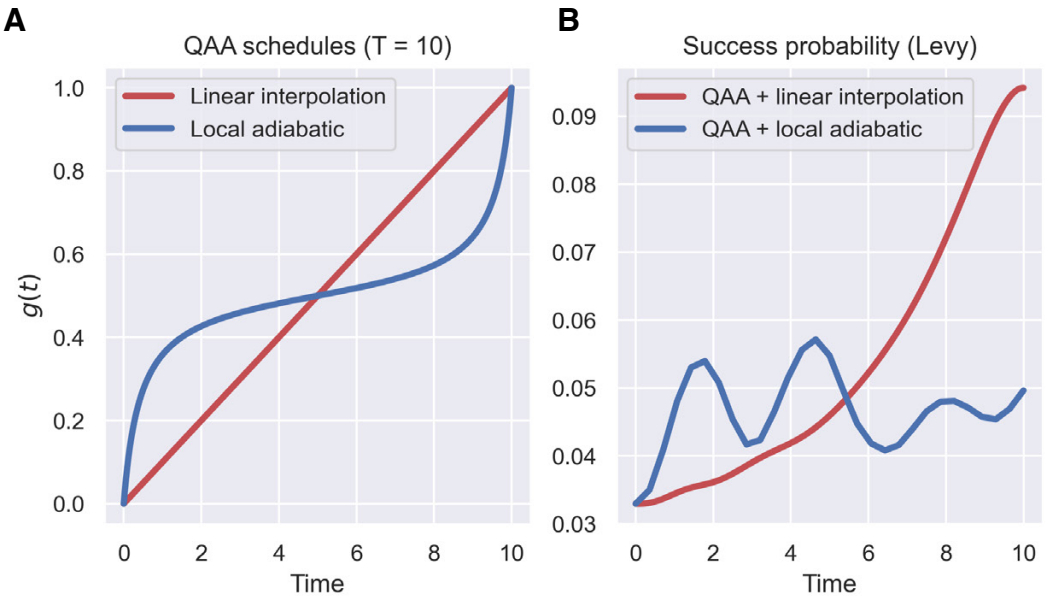


Figure A.1: **A:** the linear interpolation schedule and the local adiabatic schedule for $T = 10$. **B:** the success probability in the quantum adiabatic evolution. We choose spatial resolution 64 and simulate QAA by leapfrog integrator.

Remark 6. *Although QAA is usually applied to discrete problems, one can lift QAA to continuous domains by choosing H_0, H_1 over a continuous space. From this perspective, QHD can be regarded as a special version of this general definition of QAA (by choosing H_0 as the kinetic operator and H_1 as the potential operator). However, unlike the continuous-version (baseline) QAA, we have a more refined theoretical analysis of QHD because of the rich structures in the QHD Hamiltonian. Not as many theoretical results are known for QAA due to the weak structure of general discrete optimization problems. Meanwhile, we also observe that QHD converges much faster than the baseline QAA for continuous problems in our empirical study.*

Experiment parameters. We try two resolutions, 64 and 128, in the simulation of QAA. A higher resolution (e.g., 256) will make the problem too large for classical computers to solve in a rea-

sonable time.[3] Also, the discrete problem becomes significantly harder with more grid points and QAA needs a super long time to return a meaningful solution. To compare with the performance of QHD, we choose the total evolution time $T = 10$ for all instances. The initial state is the equal superposition of all grid points, and we choose the linear interpolation schedule $g(t) = t/T$.

Nesterov's accelerated gradient descent (NAGD)

In 1983, Nesterov proposed an accelerated gradient method [185]. It begins with initial guesses $x_0 = y_0$ and inductively computes the update sequence,

$$x_k = y_{k-1} - s\nabla f(y_{k-1}), \tag{A.6a}$$

$$y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1}). \tag{A.6b}$$

We will refer to this method as Nesterov's accelerated gradient descent algrothms, or NAGD, in what follows. It has been shown that, for any step size $s \leq 1/L$, where L is the Lipschitz constant of ∇f, this method achieves the convergence rate $O(k^{-2})$ for continuously differentiable and convex f. This rate is faster than the standard gradient descent.

In the continuous-time limit (i.e., $s \to 0$), NAGD is modeled as a second-order differential equation, $\ddot{X} + \frac{3}{t}\dot{X} + \nabla f(X) = 0$; see [30]. In the continuous-time model, the effective evolution time at the k-th step is $t_k = ks$. Equivalently, this system is described by the following

[3] Doubling the resolution in 2D therefore requires four times the storage (so four times the work per step) and eight times the number of steps.

Hamiltonian-mechanical system (where P is the momentum variable, X is the position variable):

$$H(X, P, t) = \frac{2}{t^3}\left(\frac{1}{2}|P|^2\right) + 2t^3 f(X). \tag{A.7}$$

Experiment parameters. We choose the stepsize $s = 0.001$ for all instances. We also fix the total effective evolution time $T = 10$, so the maximal number of iterations is $N = T/s = 10^4$. We draw 1000 initial points $x_0 = y_0$ uniformly random from the feasible domain $[0, 1]^2$ and compute the optimization updates paths by (A.6), respectively.

Stochastic Gradient Descent (SGD)

Stochastic gradient descent (SGD) is widely used in continuous non-convex optimization, and has been very successful in training large-scale systems such as neural networks. SGD is a gradient-based method and it updates each step with the rule:

$$x_{k+1} = x_k - s\tilde{\nabla} f(x_k), \tag{A.8}$$

where s is a fixed stepsize (or learning rate), $\tilde{\nabla} f$ is a noisy gradient evaluated from a single data point or a mini-batch.

SGD is a discrete-time algorithm. The continuous-time limit of SGD turns out to be a first-order stochastic differential equation (SDE), as derived in [186],

$$dX_t = -\nabla f(X_t)\mathrm{d}t + \sqrt{s}\mathrm{d}W_t. \tag{A.9}$$

With the SDE model, we can compare the evolution of the distribution in SGD to the evolution in QHD. In particular, we compute the effective evolution time in SGD at the k-th step by $t_k = sk$.

Experiment parameters. We model the noisy gradient in (A.8) by adding unit Gaussian noise to the exact gradient, i.e., $\tilde{\nabla} f = \nabla f + \mathcal{N}(0, 1)$. We set the stepsize $s = 0.001$ for all instances, and we stop the algorithm after 10^4 iteration steps, i.e., the total effective evolution time for SGD is $T = 10$. We draw 1000 initial points x_0 uniformly random from the feasible domain $[0, 1]^2$ and compute the optimization updates paths by (A.8), respectively.

Appendix B: Three phases in QHD

In Theorem 4, we have shown that there exist time-dependent parameters in QHD such that the convergence to global minimum is guaranteed, regardless of the shape of f. This is because the quantum state in QHD stays in the low-energy subspace in the evolution, and the low-energy subspace will eventually settle at the global minimizer of f. In this section, we introduce a more refined description of the global convergence in QHD. In particular, we look into the energy exchange between energy levels in the quantum evolution. We observe that there are **three phases** in QHD evolutions: namely, the **kinetic** phase, **global search** phase, and **descent** phase. We also develop a quantitative analysis for the three-phase picture of QHD. Serving as an explanative framework of QHD, the three-phase picture demystifies why QHD usually does better than QAA in solving non-convex problems.

B.1 Probability spectrum of QHD

In Section 3.2, we introduce the notion of low- and high-energy subspaces in the QHD evolution. We now generalize this idea and define the *probability spectrum* of QHD.

Let $H(t)$ be the QHD Hamiltonian. Suppose $E_n(t)$ and $|n;t\rangle$ are the n-th eigenvalue and eigenstate of $H(t)$, we can express the Hamiltonian as $H(t) = \sum_n E_n(t) |n;t\rangle \langle n;t|$. Here the integer n represents the energy levels of the system. Given a wave function $|\Psi_t\rangle$ at time

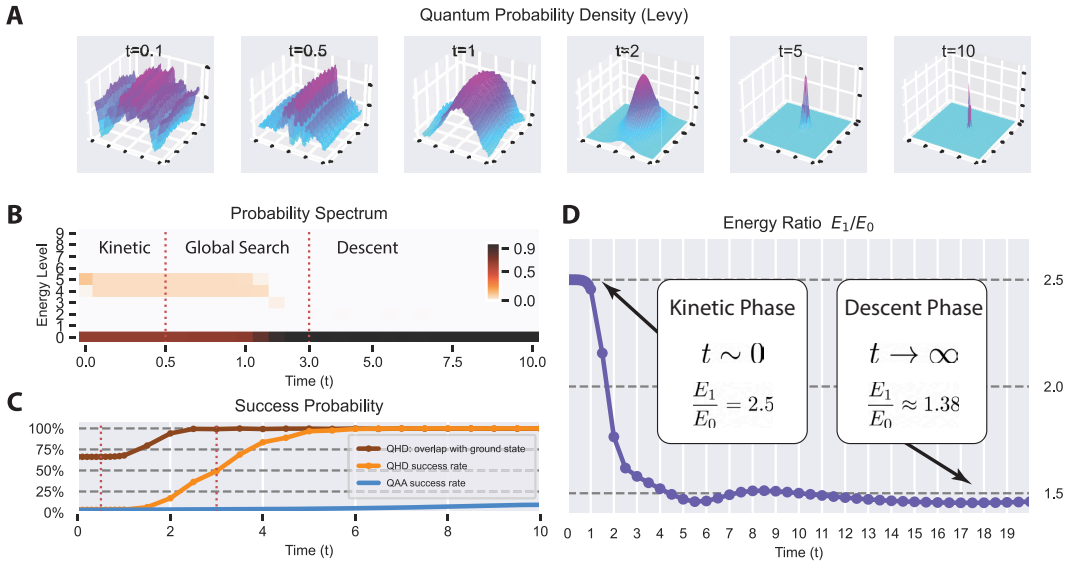


Figure B.1: The three-phase picture of QHD, illustrated with Levy function.

$t > 0$, it can be written as a superposition of eigenstates of $H(t)$: $|\Psi_t\rangle = \sum_n c_n(t)\,|n;t\rangle$, where $c_n(t) = \langle n;t|\Psi_t\rangle$ is the probability amplitude.

Definition 7 (Probability spectrum)**.** *The modulus squared of the amplitude (i.e., $|c_n(t)|^2$) represents the probability density of the wave function with respect to the energy eigenstates. We call the sequence $\{|c_n(t)|^2 = |\,\langle n;t|\Psi_t\rangle\,|^2\}_n$ as the probability spectrum of $|\Psi_t\rangle$ at time $t \geq 0$.*

Since the quantum wave function has unit L^2 norm, the probability spectrum of wave functions at any $t \geq 0$ sum up to 1: $\sum_n |c_n(t)|^2 = 1$. With this definition, the projection of $|\Psi_t\rangle$ onto the N-low-energy subspace of Hamiltonian $H(t)$ is the sum of the first N probability spectrum.

The probability spectrum of QHD is a dynamical property of the system, and we can use it to characterize different stages in the quantum evolution. In Figure B.1 (B), we numerically calculate the probability spectrum of QHD for the Levy function. We also plot the quantum wave packet at various t in Figure B.1 (A). The wave function data is from the numerical simulation of

QHD as in Section 2.5. The probability spectrum is depicted on a heatmap with the horizontal axis being the time span t and the vertical axis being the energy levels n. Each cell in the heatmap is colored by the corresponding numerical value of $|c_n(t)|^2 = |\langle n; t|\Psi_t\rangle|^2$.

We observe an interesting probability transition pattern in Figure B.1 (B). In the very beginning of QHD, there are two clusters in the probability spectrum: one low-energy cluster at $n = 0$ and one high-energy cluster at $n = 4, 5$. There are some probability exchange in the high-energy cluster but it does not interact with the low-energy one. After $t = 1$, the high-energy cluster starts to get absorbed into the low-energy subspace. As time goes by, the high-energy cluster evaporates and the quantum state completely sits in the low-energy subspace. The same trend is repeatedly observed in the probability spectrum of QHD (with other test functions). Motivated by the findings, we introduce the three-phase picture of QHD.

B.2 Three phases in QHD

We divide the course of QHD evolution into the following three phases based on the behavior of the wave function and the direction of energy/probability flows:

Kinetic phase. In the first phase of QHD, the wave function is of ample kinetic energy and it rapidly bounces within the whole search space (see $t = 0.1, 0.5$ in Figure B.1 (A)). While the majority of probability spectrum is in the low-energy subspace, a mid- or high-energy cluster in the probability spectrum prevails. The two energy clusters co-exist and almost do not interact. We call this phase the *kinetic* phase of QHD because it is characterized by the mobility of wave functions (as a result of the dominating kinetic energy term).

Global search phase. In the second phase of QHD, the kinetic energy in the system starts to drain out. The wave function becomes less oscillatory and shows a selectivity toward the global minimum of f (see $t = 1, 2$ in Figure B.1 (A)). In the probability spectrum, the high-energy cluster in the wave function is driven toward the low-energy subspace, and this trend does not reverse. We call this phase the *global search* phase because the quantum system manages to locate the global minimum of f after the screening of the whole search domain in the first phase. This phase separates QHD from other classical gradient methods.

Descent phase. In the last phase of QHD, the wave function settles and becomes increasingly concentrated near the global minimizer of f (see $t = 5, 10$ in Figure B.1 (A)). The wave function stays in the low-energy subsapce. In this phase, the quantum evolution enters the *semi-classical regime* in which the potential energy term dominates. QHD starts to behave like classical gradient descent as it converges to the global minimizer x^*. We call this phase the *descent* phase of QHD.

B.3 Quantitative analysis

The probability spectrum of QHD witness a structural change only in the global search phase, in which a significant probability transition from the high-energy subspace to the low-energy subspace takes place. While in the kinetic phase and descent phase of QHD, we do not see an outstanding shift in the probability spectrum. This phenomenon can be explained by quantitative arguments.

Suppressed probability transition in the kinetic and descent phase

First, we focus on the underlying mechanism in QHD that discourages the probability transition in the kinetic and descent phase. For simplicity, we assume the dimension $d = 1$ in the following discussion, while it can be readily generalized to arbitray finite dimensions.

The QHD Hamiltonian takes the form $H(t) = e^{\varphi_t}\left(-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right) + e^{\chi_t} f(x)$. Suppose the Hamiltonian is written as $H(t) = \sum_n E_n(t) \left|n;t\right\rangle \left\langle n;t\right|$ where $E_n(t)$, $\left|n;t\right\rangle$ are eigenvalues and eigenstates of the system at time t. As in the proof of Theorem 4, we represent the wave function as $\left|\Psi_t\right\rangle = \sum_n c_n(t) e^{i\theta_n(t)} \left|n;t\right\rangle$, where $\theta_n(t) = -\int_0^t E_n(s)\,\mathrm{d}s$. Recall from (3.20) that the amplitudes $c_n(t)$ are determined by the equations:

$$\dot{c}_n(t) = \underbrace{-c_n(t)\left\langle n;t\right| \left[\frac{\partial}{\partial t}\left|n;t\right\rangle\right]}_{\text{Part I.}} - \underbrace{\sum_{k\neq n} c_k(t) e^{i(\theta_k-\theta_n)} \frac{\left\langle n;t\right| \dot{H} \left|k;t\right\rangle}{E_k(t) - E_n(t)}}_{\text{Part II.}}.$$

As we have shown, Part I does not contribute to the change in amplitudes because $\left\langle n;t\right| \left[\frac{\partial}{\partial t}\left|n;t\right\rangle\right]$ is pure imaginary. A further understanding of the amplitude evolution amounts to investigating Part II in the equation.

In the kinetic phase, it expected that the kinetic operator $\left(-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right)$ plays a dominant role in the system, while the potential operator $f(x)$ is minor. In this case, QHD behaves like free particle evolution:

$$H^{kinetic}(t) \propto \left(-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right). \tag{B.1}$$

We then approximate the eigenstates of $H(t)$ by those of the kinetic operator. Suppose the QHD

is simulated in the unit interval $[0, 1]$ (with vanishing boundary conditions $\Psi(0, t) = \Psi(1, t) = 0$), eigenstates of the kinetic operator are sine waves: $|n; t\rangle \propto \sin(n\pi x)$. Therefore, we can compute the inner product in Part II:

$$\langle n; t| \dot{H} |k; t\rangle \propto \dot{\varphi}_t e^{\varphi_t} \int_0^1 \sin(n\pi x) \left(-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right) \sin(k\pi x)\, \mathrm{d}x = 0, \tag{B.2}$$

for $k \neq n$. Given the non-degeneratcy of $H(t)$, i.e., $E_k(t) - E_n(t) \neq 0$, we find that Part II is actually very close to zero in the kinetic phase. This fact implies that there is no amplitude exchange in the kinetic phase, which is consistent with our numerical evidence.

In the descent phase, we do not see significant amplitude exchange in the probability spectrum as well. However, the underlying mechanisms are quite different. In this phase, QHD runs into the *semi-classical* regime as the potential operator becomes the major contributor to the quantum dynamics. As shown in Section 3.2, the eigenstates of the QHD Hamiltonian can be approximated by those of a harmonic oscillator:

$$H^{descent}(t) \approx \hat{H}_{HO}(t) = \hat{K}(t) + \hat{V}(t), \tag{B.3}$$

where $\hat{K}(t) = e^{\varphi_t}\left(-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right)$ and $\hat{V}(t) = e^{\chi_t}\frac{f''(x^*)}{2}(x - x^*)^2$. As a result, $\dot{H}(t) \approx \dot{\varphi}_t\hat{K} + \dot{\chi}_t\hat{V}$ and we use Lemma 3 to compute the inner product between adjacent energy levels in Part II:

$$\langle n; t| \dot{H} |n \pm 1; t\rangle = \dot{\varphi}_t \langle n; t| \dot{K} |n \pm 1; t\rangle + \dot{\chi}_t \langle n; t| \dot{V} |n \pm 1; t\rangle = 0, \tag{B.4}$$

while the contributions from non-adjacent energy levels are minor due to the wider energy gaps (i.e., a bigger denominator $(E_k - E_n)$). We conclude that, in the descent phase, Part II in the

amplitude equation is very small so there is little energy exchange in QHD.

Energy ratio and phase transitions

We consider the energy ratio $E_1(t)/E_0(t)$, i.e., the first excited energy $E_1(t)$ to the ground energy $E_0(t)$. The energy ratio serves as a good indicator of the phase transitions in QHD; meanwhile, it also sheds light on the mechanism of probability transition in the global search phase. In what follows, we assume QHD is defined for two-dimensional problems so that we can refer to our experiment results in Section 2.5.

We begin with two extreme cases. At the very beginning of the kinetic phase ($t = 0$), the system Hamiltonian is dominated by the kinetic term $(-\nabla^2/2)$. This means that we can ignore the structure of f and push the Hamiltonian to the kinetic limit, $H(0) \approx -\nabla^2/2$. We denote the eigenstates of the kinetic operator $(-\nabla^2/2)$ as $|\psi_{\boldsymbol{n}}\rangle$, where the index is $\boldsymbol{n} = (n_x, n_y)$ with $n_x, n_y = 1, 2, ...$:

$$|\psi_{\boldsymbol{n}}\rangle = \sqrt{2}\sin(n_x\pi x)\sin(n_y\pi y),$$

and correspondingly, the eigenvalues are $E_{\boldsymbol{n}} = (n_x^2 + n_y^2)\pi^2/2$. In particular, the ground energy is $E_0 = \pi^2$ and the first excited energy (with degeneracy) is $E_1 = 5\pi^2/2$. The energy ratio is $E_1/E_0 = 2.5$. Clearly, the deviation of the energy ratio from the kinetic limit 2.5 indicates a transition from the kinetic phase to the global search phase; see Figure B.1 (D).

In the descent phase, the Schrödinger operator $H(t)$ enters the semi-classical regime so it can be effectively approximated by a quantum harmonic oscillator (see Section 3.2). The

eigenvalues of two-dimensional harmonic oscillators are

$$E_{\boldsymbol{n}} = \frac{\hbar}{2}\left[(n_1+1)\omega_1 + (n_2+1)\omega_2\right], \tag{B.5}$$

with $n_1, n_2 = 0, 1, 2, ...$[1] Therefore, the ground energy is $E_0 = \hbar(\omega_1+\omega_2)/2$ and the first excited energy (assuming $\omega_1 \geq \omega_2$) is $E_1 = \hbar(\omega_1+3\omega_2)/2$. Hence, the energy ratio at the semi-classical limit is

$$\lim_{t\to\infty} \frac{E_1}{E_0} = \frac{\omega_1+3\omega_2}{\omega_1+\omega_2}. \tag{B.6}$$

The convergence of the energy ratio to (B.6) indicates the transition from the global search phase to the descent phase; see Figure B.1 (E).

In the global search phase, the energy ratio changes and it has different values than the two extreme cases. Usually, it decreases from the initial value 2.5 and gradually converges to the semi-classical limit. Intuitively, the shrinking energy gap between E_0 and E_1 makes the probability transition between energy levels much easier. This can be a reason why the high-energy cluster is absorbed into the low-energy cluster in the global search phase.

In fact, from the energy ratio, we can theoretically predict the times at which phase transition happens. For example, we can approximately predict the two phase-transition times in QHD for Levy functions (see Figure B.1 (B)): the energy ratio significantly deviates from the initial value 2.5 after $t = 0.5$, which marks the first phase transition (kinetic to global search); after $t = 5$, the energy ratio curve starts to converge to the semi-classical limit, indicating the

[1] The $\hbar$, ω here are determined by the time-dependent parameters and the neighborhood information of x^*. Details can be found in Section 3.2.

termination of the global search phase.

Energy ratio for Levy function. The Levy function (for dimension 2) is defined by

$$f(w_1, w_2) = \sin(\pi w_1)^2 + (w_1 - 1)^2 \left[1 + 10\sin(\pi w_1 + 1)^2\right] + (w_2 - 1)^2 \left[1 + \sin(2\pi w_2)^2\right], \tag{B.7}$$

where $w_j = 1 + (x_j - 1)/4$ for $j = 1, 2$. When used as an optimization test problem, the Levy function is usually evaluated on the square $[-10, 10]^2$. It has a unique global minimum $f(1, 1) = 0$. The Hessian matrix of the Levy function at $(1, 1)$ is a diagonal matrix with two positive eigenvalues: $\lambda_1 = (\pi^2 + 1 + 10\sin(1)^2)/8$, and $\lambda_2 = 1/8$. Therefore, the second-order Taylor expansion (quadratic model) of f at the global minimizer $x^* = (1, 1)$ is:

$$f(x_1, x_2) \approx \frac{1}{2}\omega_1^2(x_1 - 1)^2 + \frac{1}{2}\omega_2^2(x_2 - 1)^2, \tag{B.8}$$

where $\omega_1 = \sqrt{\lambda_1} \approx 1.4979$, $\omega_2 = \sqrt{\lambda_2} \approx 0.3536$. By (B.6), the energy ratio at the semi-classical limit is $E_1/E_0 \approx 1.3819$, which is quite close to the numerical results shown in Figure B.1 (D).

B.4 Comparison with QAA

In our experiment, we test two quantum algorithms (QHD and QAA) and two classical algorithms (NAGD and SGD). Although both quantum algorithms are formulated as continuous-time quantum evolutions, QHD appears to outperform QAA in most test problems. The three-phase picture of QHD lends us a pivot to understand why QHD is better than QAA in solving non-convex optimization problems.

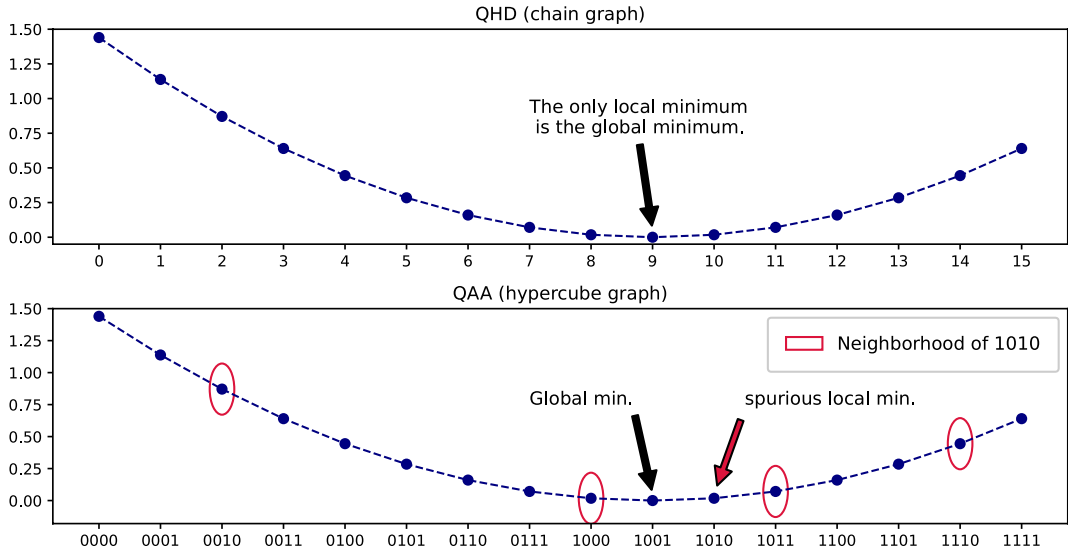


Figure B.2: The hypercube search structure in QAA twists the landscape of f. **Upper:** in QHD (problem discretized as an optimization on a one-dimensional chain graph), each node has two neighborhood nodes, and there is only one local minimum. **Lower:** in QAA (problem discretized as an optimization on a 4-dimensional hypercube graph), the node `1010` becomes a spurious local minimizer as it has the least function value compared to its neighborhood. This means QAA maps a *convex continuous* problem to a *non-convex discrete* problem.

The key difference between the two quantum algorithms is that they have different "kinetic" operators in the system Hamiltonian. The kinetic operator $(-\nabla^2/2)$ in QHD comes from the Euclidean distance function in the Bregman-Lagrangian framework. This differential operator stands for the *quantized* Euclidean distance and it allows QHD to *see* the continuous structure of f – the continuity of f is defined by the Euclidean topology in $\mathbb{R}^d$. When the spatial discretization is applied, the finite difference discretization of the kinetic operator $(-\nabla^2/2)$ makes it resemble the graph Laplacian of a regular lattice. However, in QAA, the "kinetic operator" is the initial Hamiltonian H_0 (see (A.5)), interpreted as the adjacency matrix of the hypercube graph. This means QAA regards the task as a discrete optimization problem on a hypercube graph. This misinterpretation of the underlying topology of continuous optimization problems leads to several serious consequences:

1. QAA twists the optimization landscape of f, making the problem much harder to solve. It is shown in Figure B.2 that, even for a convex f, the hypercube search graph in QAA creates many spurious local minima in the problem. This makes the discretized problem much more challenging than the original continuous problem.

2. The performance of QAA heavily depends on the resolution in the discretization. Since the naive discretization in QAA does not preserve the continuous structure of the original problem, a higher resolution can result in a harder problem, thus even worse performance in QAA. For example, when applied to the Hosaki function, the performance of QAA becomes significantly worse when we choose a higher resolution (128), while QHD does not suffer from this issue, see Figure 4.3. We observe similar patterns in other instances as well.

3. Compared to QHD, QAA does not have the kinetic phase. In the kinetic phase, QHD averages the initial wave function over the whole feasible domain, thus reducing the risk of poor random initializations. In QAA, however, the quantum state is supposed to remain the ground state of the Hamiltonian, to there is no preparatory/warm-up stage.

4. QAA does not have a descent phase, either. The semi-classical approximation theory fails to hold in QAA so that the minimal spectral gap (i.e., $\lim_{0\le t\le T} E_1(t) - E_0(t)$) can be arbitrarily small when the spatial discretization stepsize shrinks.[2] In contrast, the energy gaps in QHD is insensitive to spatial discretization. Therefore, we can not use fast-varying time-dependent functions in QAA since they will drive the quantum state out of the ground-

[2]To see this, note that $\lim_{0\le t\le T} E_1(t) - E_0(t) \le E_1(T) - E_0(T)$. At $t = T$, the QAA Hamiltonian is just the problem Hamiltonian H_P, which is a discretization of the continuous function f. Note that the spectral gap in H_P corresponds to the resolution in the discretization of f.

energy subspace. This fact prevents us from achieving faster convergence in QAA.

Appendix C: Super-polynomial quantum-classical performance separation

C.1 The spectral gap of asymmetric double well: quartic polynomials

In this section, we use a numerical example to illustrate the idea of *well-formed* asymmetric double well as we discussed in Section 3.4.1. We consider the following 1-parameter family of Schrödinger operators

$$\hat{H}(\lambda) = \frac{1}{\lambda}\left(-\frac{1}{2}\nabla^2\right) + \lambda w(x), \tag{C.1}$$

where $w(x)\colon \mathbb{R} \to \mathbb{R}$ is a one-dimensional double-well function given by a degree-4 polynomial:

$$w(x) = x^4 - \left(x - \frac{1}{32}\right)^2 - c, \tag{C.2}$$

where $c \approx -0.296$ such that the global minimum of $w(x)$ is zero.

In Figure C.1A, we plot the graph of the function w between $x = -1$ and $x = 1$. This function has two local minima:

$$x_1 \approx -0.722,\ w(x_1) = 0,$$

$$x_2 \approx 0.691,\ w(x_2) = 0.088.$$

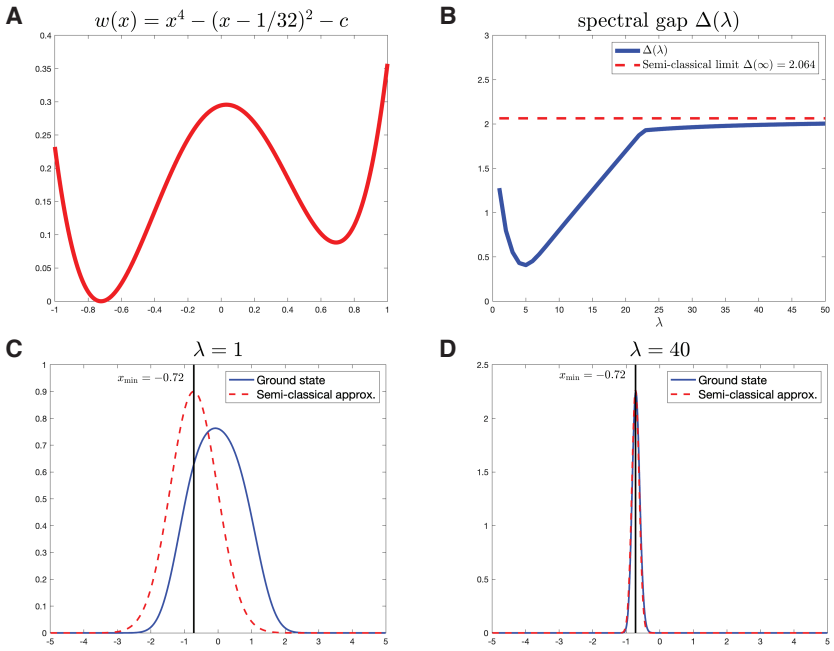


Figure C.1: **The spectral gap and ground state of asymmetric double well. A.** Plot of the double well function $w(x)$. **B.** The spectral gap $\Delta(\lambda)$ of the Hamiltonian operator $\hat{H}(\lambda)$ as a function of λ. **C, D.** The blue curve (or the red dashed curve, resp.) represents the ground state (or the harmonic approximation of the ground state, resp.) of the Hamiltonian operator $\hat{H}(\lambda)$ for $\lambda = 1, 40$. The x axis represents the one-dimensional space and the y axis shows the amplitude of the quantum states. When λ becomes larger, the harmonic approximation becomes more accurate.

For any $\lambda > 0$, the spectral theory implies that the spectrum of $\hat{H}(\lambda)$ is discrete. We denote the spectrum of the Hamiltonian $\hat{H}(\lambda)$ as $\{E_0(\lambda) < E_1(\lambda) \leq \dots\}$. For large λ, the first few eigenvalues of $\hat{H}(\lambda)$ can be computed using the semi-classical approximation. Note that w has a positive Hessian at the global minimum: $w''(x_1) \approx 4.5092$. Let $\gamma^2 = w''(x_1)$. For $\lambda \gg 1$, by Theorem 2, we have that

$$E_0(\lambda) = \frac{1}{2}\gamma + \mathcal{O}(\lambda^{-1}), \quad E_1(\lambda) = \frac{3}{2}\gamma + \mathcal{O}(\lambda^{-1}), \tag{C.3}$$

which implies that the spectral gap in the large λ limit is

$$\lim_{\lambda\to\infty} \Delta(\lambda) = \gamma \approx 2.064. \tag{C.4}$$

We employ numerical methods to compute the spectral gap of the operator $\hat{H}(\lambda)$. Our numerical results confirm our estimate of the spectral gap in the limit of large λ. In Figure C.1B, we show the spectral gap $\Delta(\lambda) = E_1(\lambda) - E_0(\lambda)$ as a function of λ. For large λ, it is clear that the spectral gap eventually converges to the predicted value γ (indicated by the red dashed line). We also find that the spectral gap $\Delta(\lambda)$ achieves its minimum $\Delta_{\min} \approx 0.408$ at around $\lambda = 5$. For $\lambda > 5$, the spectral gap $\Delta(\lambda)$ starts to increase and gradually approaches the semi-classical limit. At first glance, it is a bit surprising to see that the spectral gap $\Delta(\lambda)$ has a lower bound for all $\lambda > 0$ even when the potential function w is nonconvex. This means that the minimal spectral gap $\Delta_{\min}$ is actually an intrinsic property that only depends on the geometry of the double-well potential w but not the interpolating parameter λ.

C.2 Sub-Gaussian ground states

Probability toolbox

Lemma 12. *Let $X \in \mathbb{R}$ be a random variable such that $\mathbb{E}[e^{X^2/K}] \leq 2$, where $K > 0$ is a real number. Then, for any $s \in \mathbb{R}$, we have*

$$\mathbb{E}[e^{sX}] \leq e^{2s^2K}. \tag{C.5}$$

Proof. This is a standard result of sub-Gaussian random variables. Here, we give the proof for the completeness of this paper. By Markov's inequality, we have

$$\Pr[|X| > t] = \Pr\left[e^{X^2/K} > e^{t^2/K}\right] \leq \frac{\mathbb{E}[e^{X^2/K}]}{e^{t^2/K}} \leq 2e^{-t^2/K}.$$

With the above inequality, we can estimate the k-th moment of X. For any positive integer $k \geq 1$,

$$\begin{aligned}\mathbb{E}[|X|^k] &= \int_0^\infty \Pr\big[|X|^k > t\big]\,\mathrm{d}t = \int_0^\infty \Pr\big[|X| > t^{1/k}\big]\,\mathrm{d}t \\ &\leq 2\int_0^\infty e^{-t^{2/k}/K}\,\mathrm{d}t = K^{k/2}k\Gamma(k/2),\end{aligned}$$

where $\Gamma(z) \coloneqq \int_0^\infty t^{z-1}e^{-t}\,\mathrm{d}t$ is the Gamma function.

For any $s \in \mathbb{R}$, we use the Taylor expansion of the exponential function e^{sX} and apply the dominated convergence theorem,

$$\begin{aligned}\mathbb{E}[e^{sX}] &\leq 1 + \sum_{k=1}^\infty \frac{s^k\mathbb{E}[|X|^k]}{k!} \leq 1 + \sum_k \frac{(s^2K)^{k/2}k\Gamma(k/2)}{k!} \\ &= 1 + \sum_{k=1}^\infty \frac{(s^2K)^k 2k\Gamma(k)}{(2k)!} + \sum_{k=1}^\infty \frac{(s^2K)^{k+1/2}(2k+1)\Gamma(k+1/2)}{(2k+1)!} \\ &\leq 1 + \left(2 + \sqrt{s^2K}\right)\sum_{k=1}^\infty \frac{(s^2K)^k k!}{(2k)!} \leq 1 + \left(1 + \sqrt{\frac{s^2K}{4}}\right)\sum_{k=1}^\infty \frac{(s^2K)^k}{k!} \leq e^{2s^2K}.\end{aligned}$$

Note that we use the inequality $2(k!)^2 \leq (2k)!$ in the second-to-last step. □

Lemma 13. *Let $X \in \mathbb{R}$ be a random variable such that $\mathbb{E}[e^{X^2/K}] \leq \alpha$ with $\alpha > 1$. Then, we*

have

$$\mathbb{E}[e^{X^2/(\alpha-1)K}] \leq 2. \tag{C.6}$$

Proof. Let $r > 1$ be a real-valued parameter. Using the Fubini theorem,

$$\mathbb{E}[e^{X^2/rK}] = 1 + \sum_{k=1}^{\infty} \frac{\mathbb{E}[(X^2/rK)^k]}{k!} \leq 1 + \frac{1}{r} \sum_{k=1}^{\infty} \frac{\mathbb{E}[(X^2/K)^k]}{k!} \leq 1 + \frac{\alpha - 1}{r}.$$

Therefore, if we choose $r = \alpha - 1$, we end up with $\mathbb{E}[e^{X^2/(\alpha-1)K}] \leq 2$. □

Theorem 10 (Theorem 1, [187])**.** *Let $A \in \mathbb{R}^{m \times n}$ be a matrix, and let $\Sigma = A^T A$. Suppose that $x = (x_1, \ldots, x_n)$ is a random vector such that, for some $\sigma \geq 0$,*

$$\mathbb{E}\left[e^{r^T x}\right] \leq e^{\|r\|^2 \sigma^2/2} \tag{C.7}$$

for all $r \in \mathbb{R}^d$. Then, for all $c > 0$,

$$\Pr\left[\|Ax\|^2 > \sigma^2 \left(\mathrm{Tr}(\Sigma) + 2\sqrt{\mathrm{Tr}(\Sigma^2)c} + 2\|\Sigma\|c\right)\right] \leq e^{-c}. \tag{C.8}$$

Sub-Gaussian estimate

Lemma 14 (Sub-Gaussian ground state)**.** *Let w be a well-formed asymmetric double well. Let $\phi_\lambda(x)$ be the ground state of the Hamiltonian*

$$\hat{H}(\lambda) = \frac{1}{\lambda}\left(-\frac{1}{2}\nabla^2\right) + \lambda w(x).$$

Then, for any $s \in \mathbb{R}$, we have

$$\mathbb{E}_{X\sim|\phi_\lambda|^2}\left[e^{s(X-x^*)}\right] \leq e^{s^2\sigma^2/2}, \tag{C.9}$$

where $\sigma^2 = \beta/\lambda$, and β is a constant that only depends on w.

Proof. Note that the ground state of the Hamiltonian $\hat{H}(\lambda)$ is exactly the same as

$$2\lambda\hat{H}(\lambda) = -\nabla^2 + 2\lambda^2 w(x),$$

where the effective potential function is $\lambda^2 w(x)$. By [62, Theorem 3.4], there is a constant $\alpha > 0$ such that

$$\int_{\mathbb{R}} e^{d_0(x)}|\phi_\lambda(x)|^2 \,\mathrm{d}x \leq \alpha, \tag{C.10}$$

where $d_0(x)$ is the Agmon distance between x and 0 (see [62, Definition 3.2]). It is well known that the Agmon distance scales as the square root of the potential function, i.e.,

$$d_0(x) \sim (\lambda^2 w(x))^{1/2} \text{ as } |x| \to \infty. \tag{C.11}$$

By Definition 2, the function $w(x)$ grows as fast as a quartic function, which means there exists a positive constant c' such that $w(x) \geq c'(x - x^*)^4$. It turns out that

$$d_0(x) \geq c''\lambda(x - x^*)^2,$$

where c'' is an absolute constant. It follows from (C.10) that

$$\int_{\mathbb{R}} e^{c''\lambda(x-x^*)^2}|\phi_\lambda(x)|^2\,\mathrm{d}x \le \alpha. \tag{C.12}$$

Let $X \sim |\phi_\lambda(x)|^2$ and $\xi := X - x^*$, we rewrite (C.12) as $\mathbb{E}[e^{c''\lambda\xi^2}] \le \alpha$. Without loss of generality, we assume $\alpha > 1$. Then, Lemma 13 implies that $\mathbb{E}[e^{c''\lambda\xi^2/(\alpha-1)}] \le 2$. Now, we invoke Lemma 12 to obtain that

$$\mathbb{E}[e^{s\xi}] \le e^{s^2\sigma^2/2}, \tag{C.13}$$

where $\sigma^2 = \beta/\lambda$ with $\beta = 4(\alpha-1)/c''$. Note that β only depends on our choice of w. □

C.3 Rotational invariance and high-dimensional convexity

When the function w is a well-formed asymmetric double well, we can apply the adiabatic theorem for unbounded Hamiltonian (i.e., Theorem 3) to prepare the ground state of the Schrödinger operator

$$\hat{H}(\lambda) = \frac{1}{\lambda}\left(-\frac{1}{2}\nabla^2\right) + \lambda w(x)$$

for very large λ. Meanwhile, condition 2 in Definition 2 ensures that the ground state of $\hat{H}(\lambda)$ is sub-Gaussian (for details, see Lemma 14). These nice properties will allow us to find an approximate solution to the optimization problem $\min_x w$.

Provided with a pre-fixed parameter λ_f (which we will discuss soon), we define a time-

dependent function

$$\lambda(t) = e^{2t - \log(\lambda_f)}, \tag{C.14}$$

Clearly, for $0 \le t \le \log(\lambda_f)$, we have that $1/\lambda_f \le \lambda(t) \le \lambda_f$.

Lemma 15. *Let $w\colon \mathbb{R} \to \mathbb{R}$ be a well-formed asymmetric double well. For $0 \le t \le \log(\lambda_f)$, let $H(t) \coloneqq \hat{H}(\lambda(t))$. Denote $|\phi(t)\rangle$ as the ground state of $H(t)$ for $t \in [0, \log(\lambda_f)]$. Let $|\psi^\epsilon(t)\rangle$ be the solution to the Schrödinger equation (3.9) governed by $H(t)$ with the initial condition $|\psi^\epsilon(0)\rangle = |\phi(0)\rangle$. Then, for any $0 \le t \le \log(\lambda_f)$, we have that*

$$\|\psi^\epsilon(t) - \tilde{\phi}(t)\| \le \epsilon\theta t, \tag{C.15}$$

where θ is an absolute constant that only depends on f. Here, $\left|\tilde{\phi}(t)\right\rangle$ only differs from $|\phi(t)\rangle$ by a global phase, i.e.,

$$\tilde{\phi}(t) = e^{i\alpha(t)}\phi(t), \tag{C.16}$$

where $\alpha(t)$ is a real-valued function.

Proof. First, we show that $H(t)$ satisfies all conditions specified in Theorem 3. Since w is non-negative, $\hat{H}(\lambda) \ge 0$ for all positive λ. The gappedness of w guarantees a uniform lower bound on the spectral gap of $H(t)$. We claim that

$$\dot{H}^2 \le 4\rho + 4H^2. \tag{C.17}$$

By the definition of $H(t)$, we have that

$$\dot{H} = \frac{\dot{\lambda}}{\lambda}\left(\frac{1}{2\lambda}\frac{\mathrm{d}^2}{\mathrm{d}x^2} + \lambda w\right) = 2\left(\frac{1}{2\lambda}\frac{\mathrm{d}^2}{\mathrm{d}x^2} + \lambda w\right). \tag{C.18}$$

For any test function $\psi \in C_0^\infty(\mathbb{R})$, we consider the inner product:

$$\begin{aligned}\langle \psi, [4H^2 - \dot{H}^2]\psi\rangle &= 4\langle \psi, H^2\psi\rangle - \langle \psi, (H')^2\psi\rangle \\ &= -4\left(\langle \psi, (w\psi)''\rangle + \langle \psi, w\psi''\rangle\right) \\ &= -4\left(\langle \psi, w''\psi\rangle + 2\langle \psi, w'\psi'\rangle + 2\langle \psi, w\psi''\rangle\right).\end{aligned}$$

Using integral by parts, we have that

$$\langle \psi, w\psi''\rangle = \langle w\psi, \psi''\rangle = -\langle w'\psi, \psi'\rangle - \langle w\psi', \psi'\rangle = -\langle \psi, w'\psi'\rangle - \langle \psi', w\psi'\rangle, \tag{C.19}$$

which implies that

$$\langle \psi, [4H^2 - \dot{H}^2]\psi\rangle = -4\langle \psi, w''\psi\rangle + 4\langle \psi', w\psi'\rangle. \tag{C.20}$$

By Definition 2, w is non-negative and ρ-smooth, so we have $\langle \psi', w\psi'\rangle \geq 0$ and $w'' \leq \rho$. Therefore, we have that

$$\langle \psi, [4H^2 - \dot{H}^2]\psi\rangle \geq -4\rho, \tag{C.21}$$

which implies (C.17). With all conditions satisfied, we invoke Theorem 3 and obtain that

$$\| \left(U^\epsilon(t) - U_a^\epsilon(t)\right) \phi(0) \| \leq \epsilon\theta t, \tag{C.22}$$

where θ is a constant that only depends on w. Note that

$$U^\epsilon(t) \left|\phi(0)\right\rangle = \left|\psi^\epsilon(t)\right\rangle, \tag{C.23}$$

and $U_a^\epsilon(t) \left|\phi(0)\right\rangle$ is always in the ground-energy subspace of $H(t)$, we denote

$$\tilde{\phi}(t) = U_a^\epsilon(t) \left|\phi(0)\right\rangle \tag{C.24}$$

which differs from $\left|\phi(t)\right\rangle$ by a global phase because the ground-energy subspace of $H(t)$ is 1-dimensional. Substituting (C.23) and (C.24) to (C.22), we complete the proof. □

Let $d \geq 1$ be an integer and $w(x)$ be a well-formed asymmetric double well. We define the d-dimensional objective function F:

$$F(x_1, \ldots, x_d) = \sum_{k=1}^{d} w(x_k).$$

If the 1D model function $w(x)$ has two local minima (e.g., a double-well potential), the function $F(\boldsymbol{x})$ has 2^d local minima. Let U be an arbitrary d-by-d orthogonal matrix, we define a new

function $F_U(\boldsymbol{x}) \coloneqq F(U\boldsymbol{x})$. Now, we consider the time-dependent Hamiltonian:

$$H(t) = \frac{1}{\lambda(t)}\left(-\frac{1}{2}\nabla^2\right) + \lambda(t)F_U(\boldsymbol{x}), \tag{C.25}$$

where the function $\lambda(t)$ is defined in (C.14).

Proposition 4. *For $0 \leq t \leq \log(\lambda_f)$, let $H(t)$ be defined as in (C.25). Denote $|\Phi(t)\rangle$ as the ground state of $H(t)$ for $t \in [0, \log(\lambda_f)]$. Let $|\Psi^\epsilon(t)\rangle$ be the solution to the Schrödinger equation (3.9) governed by $H(s)$ with the initial condition $|\Psi^\epsilon(0)\rangle = |\Phi(0)\rangle$. Then, for any $0 \leq t \leq \log(\lambda_f)$, we have that*

$$\|\Psi^\epsilon(t) - \tilde{\Phi}(t)\| \leq \epsilon\theta dt, \tag{C.26}$$

where θ is an absolute constant that only depends on $w(x)$, and $\left|\tilde{\Phi}(t)\right\rangle$ only differs from $|\Phi(t)\rangle$ by a global phase.

Proof. We consider the following Hamiltonian operator with the separable objective function $F(\boldsymbol{x})$,

$$H_0(t) = \frac{1}{\lambda(t)}\left(-\frac{1}{2}\nabla^2\right) + \lambda(t)F(\boldsymbol{x}). \tag{C.27}$$

For $t \in [0, \log(\lambda_f)]$, let $|\Psi_0^\epsilon(t)\rangle$ be the solution to the Schrödinger equation,

$$i\epsilon\,|\Psi_0^\epsilon(t)\rangle = H_0(t)\,|\Psi_0^\epsilon(t)\rangle\,. \tag{C.28}$$

If the initial state $\Psi_0^\epsilon(\boldsymbol{x}, 0) = \Psi^\epsilon(U^\top\boldsymbol{x}, 0)$, since the Laplacian is invariant under rotation, it is

clear that

$$\Psi_0^\epsilon(\boldsymbol{x}, t) = \Psi^\epsilon(U^\top \boldsymbol{x}, t) \tag{C.29}$$

for all $t \in [0, \log(\lambda_f)]$. Similarly, the ground state of $H_0(t)$ is given by

$$\Phi_0(U^\top \boldsymbol{x}, t) = \Phi(U^\top \boldsymbol{x}, t), \tag{C.30}$$

where $\Phi(\boldsymbol{x}, t)$ is the ground state of $H(t)$. Therefore, to show (C.26), it suffices to prove that there is a state $\left|\tilde{\Phi}_0(t)\right\rangle$ that differs from $|\Phi_0(t)\rangle$ by a global phase such that

$$\|\Psi_0^\epsilon(t) - \tilde{\Phi}_0(t)\| \leq \epsilon\theta dt. \tag{C.31}$$

Next, we prove (C.31) using the triangle inequality. Note that the Hamiltonian operator $H_0(t)$ is separable in the sense that

$$H_0(t) = \sum_{k=1}^{d} H_0^{(k)}(t), \tag{C.32}$$

where for $k = 1, \ldots, d$, the operator $H_0^{(k)}(t)$ is given by

$$H_0^{(k)}(t) \coloneqq \frac{1}{\lambda(t)} \left(-\frac{1}{2} \frac{\partial^2}{\partial x_k^2} \right) + \lambda(t) w(x_k). \tag{C.33}$$

If we denote the ground state of $H_0^{(k)}(t)$ as $|\phi_k(t)\rangle$, the separability implies that the ground state of $H_0(t)$ is given by the product of the 1D ground states $\Phi_0(t) = \prod_{k=1}^{d} \phi_k(t)$. Similarly, the

solution $|\Psi_0^\epsilon(t)\rangle$ is a product state given by $\Psi_0^\epsilon(t) = \prod_{k=1}^d \psi_k^\epsilon(t)$, where each $\psi_k^\epsilon(t)$ solves the 1D Schrödinger equation $i\epsilon \frac{\mathrm{d}}{\mathrm{d}t} |\psi_k^\epsilon(t)\rangle = H_0^{(k)}(t) |\psi_k^\epsilon(t)\rangle$. By Lemma 15, we have that

$$\|\psi_k^\epsilon(t) - e^{i\alpha_k(t)}\phi_k(t)\| \leq \epsilon\theta t, \tag{C.34}$$

where each $\alpha_k(t)$ is a real-valued global phase. We define

$$\tilde{\Phi}_0(t) \coloneqq e^{i\sum_{k=1}^d \alpha_k(t)} \prod \prod_{k=1}^d \psi_k^\epsilon(t) = e^{i\sum_{k=1}^d \alpha_k(t)} \Phi_0(t). \tag{C.35}$$

Then, by applying the triangle inequality d times, we end up with

$$\|\Psi_0^\epsilon(t) - \tilde{\Phi}_0(t)\| \leq \sum_{k=1}^d \|\psi_k^\epsilon(t) - e^{i\alpha_k(t)}\phi_k(t)\| \leq \epsilon\theta dt. \tag{C.36}$$

□

C.4 Initial state preparation

An important step in Algorithm 1 is to prepare the ground state of the initial Hamiltonian $H(0)$. Since the function f grows as $\Theta(|x|^4)$, we can choose a large enough constant M such that the quantum dynamics in $\mathbb{R}^d$ can be faithfully simulated over a compact domain $\Omega \coloneqq [-M, M]^d$ (with periodic boundary conditions). We denote $V(\boldsymbol{x})$ as the truncated objective function $f(\boldsymbol{x})$ on Ω. Then, it is sufficient to prepare the ground state of the operator

$$H(0) = \lambda_f \left(-\frac{1}{2}\nabla^2\right) + \frac{1}{\lambda_f} V.$$

Since we choose $\lambda_f \gg 1$, the potential operator V/λ_f can be regarded as a perturbative term in the initial Hamiltonian $H(0)$. Meanwhile, the ground state of the kinetic operator $\left(-\frac{1}{2}\nabla^2\right)$ over the periodic domain Ω is a constant function that is easily prepared on a quantum computer as a uniform superposition state. Therefore, we may prepare the initial state of $H(0)$ from the ground state of the kinetic operator via a quantum adiabatic evolution.

More concretely, we can simulate a time-dependent Schrödinger equation with the following Hamiltonian,

$$\tilde{H}(t) = -\frac{1}{2}\nabla^2 + \frac{a(t)}{\lambda_f^2}V,$$

where $a(t)\colon [0,T] \to [0,1]$ is a monotonically increasing function such that $a(0) = 0$, $a(T) = 1$. Note that the ground state of $\tilde{H}(0)$ is a constant function and that of $\tilde{H}(T)$ is the desired initial state in QHD. To ensure the quantum state stays in the ground-energy subspace, the evolution time T usually depends on a polynomial of $1/\Delta$, where Δ is the minimal spectral gap of $\tilde{H}(t)$ for $0 \le t \le T$. For any $t \in [0,T]$, the potential term $\frac{a(t)}{\lambda_f^2}V$ is perturbative so we can derive a constant lower bound for the spectral gap using the standard perturbation theory [62]. This means the adiabatic evolution time T is independent of the problem dimension d.

The query complexity of this quantum evolution is typically proportional to the simulation time T and the L^∞-norm of the perturbation V/λ_f^2. As V is the sum of d identity 1D model problems $w(x) \sim |x - x^*|^4$, we have $\|V\|_\infty \sim dM^4$. Together with $\lambda_f \sim d$, we have that $\|V\|_\infty/\lambda_f^2 \sim M^4/d$, where M is a large constant. It turns out that the initial state of $H(0)$ is prepared with $\mathcal{O}(1/d)$ queries to O_f and an additional $\mathcal{O}(d)$ 1- and 2-qubit gates. Both the query complexity and the gate complexity are significantly lower than the upper bounds claimed in Theorem 5.

C.5 Time rescaling

to leverage the quantum simulation algorithm in Theorem 11, we need to rescale the time in the Schrödinger equation (3.57). In the following lemma, we give the rescaled quantum dynamics whose final state is precisely the same as in (3.57).

Lemma 16 (Time rescaling)**.** *The final state* $|\Psi^\epsilon(t_f)\rangle$ *of the quantum dynamics in Algorithm 1 can be obtained by simulating the following Schrödinger equation for* $\frac{1}{\epsilon\lambda_f} \le s \le \frac{\lambda_f}{\epsilon}$,

$$i\frac{\mathrm{d}}{\mathrm{d}s}|\Psi(s)\rangle = \frac{1}{2}\left[-\frac{1}{2}\nabla^2 + \varphi(s)f\right]|\Psi(s)\rangle\,, \tag{C.37}$$

where

$$\varphi(s) \coloneqq \frac{1}{\left(-\epsilon s + \lambda_f + 1/\lambda_f\right)^2} \in [1/\lambda_f^2, \lambda_f^2]. \tag{C.38}$$

Proof. To simulate the QHD dynamics as given in (3.57), we consider the change of variable,

$$\xi = \lambda_f + \frac{1}{\lambda_f} - \frac{1}{\lambda(t)} = \lambda_f\left(1 - e^{-2t}\right) + \frac{1}{\lambda_f}. \tag{C.39}$$

We see that $\xi(t)$ is an increasing function in t such that $\xi(0) = \frac{1}{\lambda_f}$, $\xi(t_f) = \lambda_f$. Moreover, we can express t in terms of ξ:

$$t(\xi) = -\frac{1}{2}\log\left(-\frac{\xi}{\lambda_f} + 1 + \frac{1}{\lambda_f^2}\right). \tag{C.40}$$

Using the change of variable $t \mapsto \xi(t)$ and we define $|\Psi^\epsilon(\xi)\rangle = |\Psi^\epsilon(\xi(t))\rangle$, the Schrödinger

equation (3.57) becomes

$$i\epsilon\frac{\mathrm{d}}{\mathrm{d}\xi}\left|\Psi^{\epsilon}(\xi)\right\rangle = \frac{1}{2}\left[-\frac{1}{2}\nabla^2 + \varphi(\xi)f\right]\left|\Psi^{\epsilon}(\xi)\right\rangle, \tag{C.41}$$

where

$$\varphi(\xi) \coloneqq \frac{1}{\left(-\xi + \lambda_f + 1/\lambda_f\right)^2} \in [1/\lambda_f^2, \lambda_f^2]. \tag{C.42}$$

We can also absorb the parameter ϵ into the Hamiltonian by dilating the time scale $s \coloneqq \xi/\epsilon \in J \coloneqq [\frac{1}{\epsilon\lambda_f}, \frac{\lambda_f}{\epsilon}]$. Let $\varphi(s) \coloneqq \varphi(\epsilon s)$, we end up with the effective dynamics described by (C.37). □

C.6 Quantum simulation of Schrödinger equations

In Quantum Hamiltonian Descent [40], a major subroutine is to simulate the Schrödinger equation, a task known as *quantum simulation*. Quantum simulation is a prominent application of quantum computation, and several efficient quantum algorithms for simulating real-space quantum dynamics have been proposed, including [150, 151, 87, 76, 135]. In this paper, we employ a quantum simulation algorithm due to Childs, Leng, Li, Liu, and Zhang [76]. The complexity of this algorithm has near-optimal dependence in the dimension d and accuracy η by leveraging the pseudo-spectral representation and interaction-picture quantum simulation. We consider the Schrödinger equation over the time interval $[t_0, t_1]$ for a given time-dependent potential $V(x,t)$,

$$i\frac{\partial}{\partial t}\left|\Psi(x,t)\right\rangle = \left[-\frac{1}{2}\nabla^2 + V(x,t)\right]\left|\Psi(x,t)\right\rangle, \tag{C.43}$$

where we specify $\Omega = [-M, M]^d$ for a sufficiently large M and $V(x,t)\colon \Omega \times [t_0, t_1] \to \mathbb{R}$ is a time-dependent potential function, $\Psi(x,t)\colon \Omega \times [t_0, t_1] \to \mathbb{C}$ is the wave function subject to certain initial condition and the periodic boundary condition.

In [76, Theorem 8], the complexity of simulating (C.43) involves an additional parameter g' that depends on the regularity of the wave function $\Psi(x,t)$. We slightly improve this result by assuming the initial condition is *analytic*, which is the case in QHD.

Theorem 11. *Suppose the potential field $V(x,t)$ is bounded, smooth in x and t, and periodic in x. Moreover, we assume the initial data $\Psi_0(x)$ is analytic on Ω and V is L-Lipschitz in t. We define $\|V\|_{\infty,1} \coloneqq \int_{t_0}^{t_1} \|V(\cdot,t)\|_\infty \,\mathrm{d}t$. Then, the Schrödinger equation (C.43) can be simulated for time $t \in [t_0, t_1]$ up to accuracy η with the following cost:*

1. *Queries to the quantum evaluation oracle O_V: $\mathcal{O}\left(\|V\|_{\infty,1} \frac{\log(\|V\|_{\infty,1}/\eta)}{\log\log(\|V\|_{\infty,1}/\eta)}\right)$,*
2. *1- and 2-qubit gates:*

$$\mathcal{O}\left(\|V\|_{\infty,1}\left(\mathrm{poly}(z) + \log^{2.5}\left(L\|V\|_{\infty,1}/\eta\right) + d\log\log(1/\eta)\right)\frac{\log(\|V\|_{\infty,1}/\eta)}{\log\log(\|V\|_{\infty,1}/\eta)}\right).$$

Note that the quantum simulation algorithm in [76] requires two quantum oracles other than O_V, namely, the *inverse change-of-variable* oracle O_{inv} and the *max-norm* oracle O_{norm} (see Lemma 5 [76]). In QHD, the potential function $V(x,t) \coloneqq \varphi(t)V(x)$, where $\varphi(t)$ is a time-dependent function described by a closed-form formula (see (C.38)) and $V(x)$ is given by O_V. In this case, the two oracles O_{inv} and O_{norm} are efficiently implemented without querying the function V.

Lemma 17. *Let $\Psi(x,t)$ denote the exact solution of (C.43) and $\widetilde{\Psi}(x,t)$ denote the approximated*

solution by the Fourier spectral method (truncated up to frequency n).[1] *We assume that the initial data $\Psi_0(x)$ is periodic and analytic in $x \in \Omega$. Then, for any integer $n \geq 1$, the error from the Fourier spectral method satisfies*

$$\max_{x,t} |\Psi(x,t) - \widetilde{\Psi}(x,t)| \leq 2r^{n/2+1}, \tag{C.44}$$

where $0 < r < \min(\frac{1}{2}, \frac{1}{At})$, A is an absolute constant that only depends on $\Psi_0(x)$.

Proof. We give the proof in one dimension, as the same argument is readily generalized to arbitrary finite dimensions. We assume the initial data $\Psi_0(x)$ is periodic over $[0, 2\pi]$. The analyticity implies that, for any $x \in [0, 2\pi]$, there is a constant C such that

$$\left|\Psi_0^{(k)}(x)\right| \leq C^{k+1}(k!). \tag{C.45}$$

Therefore, the function $\Psi_0(x)$ admits an analytic continuation in the strip

$$\Gamma_0 = \left\{ z \in \mathbb{C} : |z - x| < \frac{1}{C}, x \in [0, 2\pi] \right\}.$$

Due to [188, Proposition 1], there is an absolute constant A such that for any $t \in [0, T]$,

$$\left\|\Psi^{(k)}(\cdot, t)\right\| \leq At \left\|\Psi_0^{(k)}\right\| \leq AC^{k+1}t(k!), \tag{C.46}$$

which implies that the wave function $\Psi(x,t)$ is periodic and analytic for any finite t. The strip on

[1] More details on the Fourier spectral method can be found in Section 2.2 (in particular Lemma 1) in [76].

which $\Psi(x,t)$ admits an analytic continuation is

$$\Gamma_t = \left\{ z \in \mathbb{C} : |z - x| < \frac{1}{ACt}, x \in [0, 2\pi] \right\}.$$

Suppose that the function $\Psi(x,t)$ allows an exact, infinite trigonometric polynomial representation (see [76, Lemma 16]),

$$\Psi(x,t) = \sum_{k=0}^{\infty} c_k(t) e^{ikx}. \tag{C.47}$$

Let $\tilde{\Psi}(t,x)$ be the truncated Fourier series up to $k = n/2$, then the error from the Fourier spectral method satisfies

$$|\Psi(x,t) - \tilde{\Psi}(x,t)| \leq \sum_{k=n/2+1}^{\infty} |c_k(t)|. \tag{C.48}$$

Meanwhile, the function $\Psi(x,t)$ has an analytic continuation defined by $\Psi(x,t) = \sum_{k=0}^{\infty} c_k(t) z^k$ for $z \in \Gamma_t$. By Cauchy's integral formula, for any simply connected curve γ on the strip Γ_t, we have that

$$c_k(t) = \frac{1}{k!} \int_{\gamma} \Psi^{(k)}(z,t) \mathrm{d}z. \tag{C.49}$$

Together with (C.46), it turns out that $|c_k(t)| \leq r^k$ for some $0 < r < \frac{1}{At}$. Therefore, the error

from the Fourier spectral method satisfies

$$|\Psi(x,t) - \tilde{\Psi}(x,t)| \leq \sum_{k=n/2+1}^{\infty} |c_k(t)| \leq \frac{r^{n/2+1}}{1-r}. \tag{C.50}$$

Moreover, if we force $r < 1/2$, the error is bounded by $2r^{n/2+1}$. □

To ensure that the simulation error is bounded by η, we may choose the truncation number

$$n = \left\lceil 2\left(\frac{\log(4/\eta)}{\log(1/r)} - 1\right)\right\rceil \leq \mathcal{O}\left(\log(1/\eta)\right).$$

Next, by plugging this truncation number in Equation (113) in [76], we prove Theorem 11.

C.7 Auxiliary lemmas

Lemma 18. *Suppose that ψ_1, ψ_2 are two unit vectors in $\mathcal{H}$ such that*

$$\|\psi_1 - \psi_2\| \leq \delta. \tag{C.51}$$

Let O be a bounded operator on $\mathcal{H}$ such that $\|O\| \leq 1$. Then, we have

$$|\langle\psi_1| O |\psi_1\rangle - \langle\psi_2| O |\psi_2\rangle| \leq 2\delta. \tag{C.52}$$

Proof. By the triangle inequality,

$$|\langle\psi_1| O |\psi_1\rangle - \langle\psi_2| O |\psi_2\rangle| \leq |(\langle\psi_1| - \langle\psi_2|)O |\psi_1\rangle| + |\langle\psi_2| O(|\psi_1\rangle - |\psi_2\rangle)| \leq 2\delta.$$

□

Appendix D: Simulating quantum dynamics in the real space via Hamiltonian embedding

In this appendix, we consider the simulation of quantum dynamics in the real space. The quantum dynamics is described by the Schrödinger equation over $\mathbb{R}^d$:

$$i\frac{\partial}{\partial t}\Psi = \left[-\frac{1}{2}\nabla^2 + f(x)\right]\Psi(t,x), \tag{D.1}$$

where $\Psi(t,x)\colon [0,T]\times\mathbb{R}^d \to \mathbb{C}$ is the quantum wave function. The Schrödinger equation is subject to an initial condition $\Psi(0,x) = \Psi_0(x)$.

D.1 Experiments on IonQ

Problem formulation

Lemma 19. *Consider the 1D Schrödinger equation with quadratic potential field ($a > 0$):*

$$i\frac{\partial}{\partial t}\Psi(t,x) = \left[-\frac{1}{2}\frac{\partial^2}{\partial x^2} + \left(\frac{1}{2}ax^2 + bx\right)\right]\Psi(t,x), \tag{D.2}$$

subject to a Gaussian initial state

$$\Psi(0,x) = \left(\frac{1}{2\pi\sigma^2}\right)^{1/4} e^{-\frac{x^2}{4\sigma^2}}. \tag{D.3}$$

Then, the expectation value of the position observable $\hat{x}$ is given by

$$\langle \hat{x} \rangle_t = \frac{b}{a}\left(\cos\left(\sqrt{a}t\right) - 1\right). \tag{D.4}$$

Proof. Since the potential field is quadratic, we invoke the Ehrenfest theorem to obtain the following Hamilton's equation for $\langle \hat{x} \rangle_t$ and $\langle \hat{p} \rangle_t$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\langle \hat{x} \rangle_t = \langle \hat{p} \rangle_t, \tag{D.5}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\langle \hat{p} \rangle_t = -\langle f \rangle_t = -a\langle \hat{x} \rangle_t - b, \tag{D.6}$$

where the initial data are given by

$$\langle \hat{x} \rangle_0 = \langle \hat{p} \rangle_0 = 0. \tag{D.7}$$

Therefore, the closed-form formula of $\langle \hat{x} \rangle_t$ is obtained by solving the system of ODEs. □

Fock space truncation method

We can rewrite the real-space Schrödinger operator using the position operator $\hat{x}$ and the momentum operator $\hat{p} = -i\frac{\partial}{\partial x}$:

$$H = \frac{1}{2}\sum_{j=1}^{d} \hat{p}_j^2 + f(\hat{x}_1, \ldots, \hat{x}_d). \tag{D.8}$$

We define two operators:

$$\hat{a} = \frac{1}{\sqrt{2}}(i\hat{p} + \hat{x}), \quad \hat{a}^\dagger = \frac{1}{\sqrt{2}}(-i\hat{p} + \hat{x}). \tag{D.9}$$

The operator $\hat{a}^\dagger$ is the raising/creation operator, and $\hat{a}$ is the lowering/annihilation operator. They are infinite-dimensional quantum operators with eigenstates being the number basis (corresponding to the eigenstates of quantum harmonic oscillators):

$$\hat{a}\,|0\rangle = 0,\ \hat{a}\,|n\rangle = \sqrt{n}\,|n-1\rangle \ \text{(for } n \geq 1), \tag{D.10}$$

$$\hat{a}^\dagger\,|n\rangle = \sqrt{n+1}\,|n+1\rangle \ \text{(for } n \geq 0). \tag{D.11}$$

We can express the creation and annihilation operators in the standard matrix form,

$$\hat{a}^\dagger = \begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ & \sqrt{2} & 0 & & \\ & & \sqrt{3} & 0 & \\ & & & \cdots & \cdots \end{bmatrix}, \; \hat{a} = \begin{bmatrix} 0 & 1 & & & \\ & 0 & \sqrt{2} & & \\ & & 0 & \sqrt{3} & \\ & & & 0 & 2 \\ & & & \cdots & \cdots \end{bmatrix}. \tag{D.12}$$

Using the ladder operators, we can rewrite the position and momentum operators:

$$\hat{x} = \frac{1}{\sqrt{2}} \left(\hat{a}^\dagger + \hat{a} \right), \; \hat{p} = \frac{i}{\sqrt{2}} \left(\hat{a}^\dagger - \hat{a} \right). \tag{D.13}$$

In the matrix form, the two operators are infinite-dimensional tri-diagonal matrices.

$$\hat{x} = \frac{1}{\sqrt{2}} \sum_{j=0}^{\infty} \sqrt{j+1} \left(|j\rangle \langle j+1| + |j+1\rangle \langle j| \right), \tag{D.14}$$

$$\hat{p} = \frac{i}{\sqrt{2}} \sum_{j=0}^{\infty} \sqrt{j+1} \left(- |j\rangle \langle j+1| + |j+1\rangle \langle j| \right). \tag{D.15}$$

Similarly, we can compute the quadratic operators:

$$\hat{p}^2 = \frac{1}{2} \sum_{j=0}^{\infty} (2j+1) |j\rangle \langle j| - \frac{1}{2} \sum_{j=0}^{\infty} \sqrt{(j+1)(j+2)} \left(|j\rangle \langle j+2| + |j+2\rangle \langle j| \right), \tag{D.16}$$

$$\hat{x}^2 = \frac{1}{2} \sum_{j=0}^{\infty} (2j+1) |j\rangle \langle j| + \frac{1}{2} \sum_{j=0}^{\infty} \sqrt{(j+1)(j+2)} \left(|j\rangle \langle j+2| + |j+2\rangle \langle j| \right), \tag{D.17}$$

which are infinite-dimensional matrices with bandwidth 2.

To simulate the real-space quantum operator (D.8) on a quantum computer, we need to truncate the Fock space (spanned by the number basis) up to finite dimension N. Then, we can construct Hamiltonian embeddings of the truncated quantum operator for real-machine implementation.

By truncating Fock space up to N levels, we have the truncated operators,

$$\hat{x} = \frac{1}{\sqrt{2}} \sum_{j=0}^{N-2} \sqrt{j+1} \left(|j\rangle \langle j+1| + |j+1\rangle \langle j| \right), \tag{D.18}$$

$$\hat{p}^2 = \frac{1}{2} \sum_{j=0}^{N-1} (2j+1) |j\rangle \langle j| - \frac{1}{2} \sum_{j=0}^{N-3} \sqrt{(j+1)(j+2)} \left(|j\rangle \langle j+2| + |j+2\rangle \langle j| \right), \tag{D.19}$$

$$\hat{x}^2 = \frac{1}{2} \sum_{j=0}^{N-1} (2j+1) |j\rangle \langle j| + \frac{1}{2} \sum_{j=0}^{N-3} \sqrt{(j+1)(j+2)} \left(|j\rangle \langle j+2| + |j+2\rangle \langle j| \right). \tag{D.20}$$

Therefore, if we use the unary embedding, the encoding Hamiltonians are $(N-1)$-qubit operators:

$$Q_{\hat{x}}^{\text{unary}} = \frac{1}{\sqrt{2}} \sum_{j=1}^{N-1} \sqrt{j} X_j, \tag{D.21}$$

$$Q_{\hat{p}^2}^{\text{unary}} = \sum_{j=1}^{N-1} \hat{n}_j - \frac{1}{2} \sum_{j=1}^{N-2} \sqrt{j(j+1)} X_{j+1} X_j, \tag{D.22}$$

$$Q_{\hat{x}^2}^{\text{unary}} = \sum_{j=1}^{N-1} \hat{n}_j + \frac{1}{2} \sum_{j=1}^{N-2} \sqrt{j(j+1)} X_{j+1} X_j. \tag{D.23}$$

It is worth noting that we already discard the global phase in the embedding operators.

Similarly, if we choose the one-hot embedding, the encoding Hamiltonians are N-qubit

operators:

$$Q_{\hat{x}}^{\text{one-hot}} = \frac{1}{\sqrt{2}} \sum_{j=1}^{N-1} \sqrt{j} X_{j+1} X_j, \tag{D.24}$$

$$Q_{\hat{p}^2}^{\text{one-hot}} = \sum_{j=1}^{N} (j - \frac{1}{2}) \hat{n}_j - \frac{1}{2} \sum_{j=1}^{N-2} \sqrt{j(j+1)} \frac{(X_{j+2} X_j + Y_{j+2} Y_j)}{2}, \tag{D.25}$$

$$Q_{\hat{x}^2}^{\text{one-hot}} = \sum_{j=1}^{N} (j - \frac{1}{2}) \hat{n}_j + \frac{1}{2} \sum_{j=1}^{N-2} \sqrt{j(j+1)} \frac{(X_{j+2} X_j + Y_{j+2} Y_j)}{2}. \tag{D.26}$$

Again, the global phase is discarded.

With these embedding operators, we can use the rules in Section 5.3 to construct an embedding of the real-space Schrödinger operator with a quadratic potential field.

To simulate the N-level truncated Hamiltonian, we require a large penalty coefficient g. With a large g, the norm of the embedding Hamiltonian becomes so huge that the standard product formula requires a significant number of Trotter steps that are infeasible for near-term devices. To address this issue, we transform the embedding operators to the interaction picture and use the quantum simulation algorithm `qDRIFT` (for details, see Algorithm 3).

Experiment setup and result

We perform experiments of the Schrödinger equation in one-dimensional real space using the IonQ Aria-1 processor. We use the penalty-free one-hot embedding to simulate the real-space Schrödinger equation with quadratic potential $f(x) = x^2 - \frac{1}{2}x$. The embedding Hamiltonian is then given by

$$H_{\text{real-space}} = \frac{1}{2} Q_{\hat{p}^2}^{\text{one-hot}} - \frac{1}{2} Q_{\hat{x}^2}^{\text{one-hot}} + \frac{1}{2} Q_{\hat{x}}^{\text{one-hot}}, \tag{D.27}$$

as described in Appendix D.1.

We truncate the Hamiltonian to $N = 5$ levels, corresponding to 5 qubits using the one-hot embedding. The initial state is the first encoded state of the one-hot code, prepared using a single X rotation gate. We perform quantum simulation using the randomized first-order Trotter-Suzuki formula for evolution times up to $T = 5$ with $r = 11$ Trotter steps. For each circuit, the total gate count is 154 two-qubit gates in addition to a single one-qubit gate for preparing the initial state. To compute the expected position observable $\hat{x}$, we measure in x-basis using 1500 shots. For computing the expected kinetic energy $\frac{1}{2}\hat{p}^2$, we also measure in the computational basis using 500 shots.

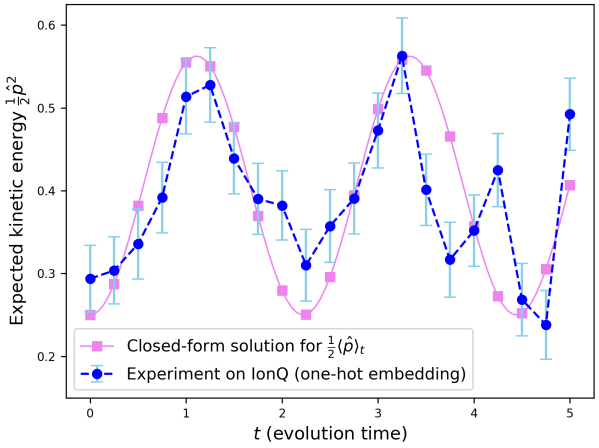


Figure D.1: Expectation value of kinetic energy $\frac{1}{2}\hat{p}^2$ depicted as a function of evolution time t. The violet curve denotes the analytically computed expected kinetic energy with squares drawn at the same evolution times as the experiment data for comparison. Blue dots denote experimental results obtained from the IonQ Aria-1 processor using 1500 shots for x-basis measurements and 500 shots for z-basis measurements for each data point. Error bars denote standard error.

In Figure 5.1C, we plot the expected value of the position observable. In violet, we plot the closed form solution, given by $\langle\hat{x}\rangle_t = \frac{1}{4}(1 - \cos(\sqrt{2}t))$. In blue, we plot the real-machine data obtained from the IonQ processor. The experimentally observed values show a sinusoidal trend

consistent with the analytically predicted values.

In addition, we present the expected kinetic energy $\frac{1}{2}\langle p^2 \rangle$ in Figure D.1. The analytical solution, shown in violet, is evaluated as $\frac{1}{2}\langle p^2 \rangle = -\frac{5}{16}\cos\left(\sqrt{2}t\right)^2 + \frac{9}{16}$. The experimental data is depicted by blue dots, showing a sinusoial trend consistent with the analytically predicted solution.

Resource analysis

We conduct a resource analysis for performing a simulation of the real-space Schrödinger equation via the truncated Fock space method. The task is to perform quantum simulation for a fixed evolution time T. To be consistent with the real-machine experiment, we fix the evolution time to be $T = 5$ and the potential function is $f(x) = \frac{1}{2}ax^2 + bx$ with $a = 2$ and $b = -1/2$. For this task, we apply the randomized first-order Trotter-Suzuki product formula.

For the standard binary encoding, we start with the desired quantum operator with $2^{\lceil \lg N \rceil}$ levels, in order to preserve the structure of the problem. To simulate the operator with N levels, we only use the first N basis states as the encoding subspace. To this end, we remove all matrix entries that contribute to interactions between the encoding subspace and the orthogonal complement. We use Qiskit to compute the Pauli decomposition of the resulting matrix and convert the circuit to single-qubit Pauli rotations and XX-rotation gates.

In Figure 5.1B, we plot estimates of the total gate count required for the standard binary code and the penalty-free one-hot embedding. For standard binary, we observe an asymptotic gate complexity of $O(n^{3.16})$. For the one-hot embedding, the complexity is estimated as $O(n^{2.67})$, showing a modest advantage in the asymptotic scaling. In addition, we observe an order of

magnitude advantage in the constant factor for the one-hot embedding.

D.2 Experiments on QuEra

Finite difference method

We now discuss how to simulate two-dimensional Schrödinger equations on the QuEra machine. Consider the following partial differential equation defined on the unit square $\Omega = [0,1]^2$,

$$i\frac{\partial}{\partial t}\Psi(t,\boldsymbol{x}) = \left(-\frac{1}{2}\nabla^2 + V(\boldsymbol{x})\right)\Psi(t,\boldsymbol{x}), \tag{D.28}$$

where $\Psi(t,\boldsymbol{x})\colon [0,t]\times\Omega\to\mathbb{C}$ is the quantum wave function with Dirichlet boundary conditions (i.e., $\Psi(t,\boldsymbol{x})=0$ for $\boldsymbol{x}\in\partial\Omega$). $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplacian operator, and $V(\boldsymbol{x})\colon \Omega\to\mathbb{R}$ is the potential field.

In our QuEra experiment, we simulate the Schrödinger equation (D.28) using the finite difference method. Suppose that we discretize the unit interval $[0,1]$ into N pieces with grid points $\{z_k = k/(N-1)\colon k=0,\ldots,N-1\}$. For any function $u(x)$ defined on $[0,1]$ with vanishing boundary condition $u(0)=u(1)=0$, we can discretize u on the mesh $\{z_k\}$ and represent it as an N-dimensional vector:

$$\vec{u} = [u(z_0),\ldots,u(z_{N-1})]^\top.$$

Using the central finite difference approximation, i.e., $u''(x)\approx\frac{u(x+h)-2u(x)+u(x-h)}{h^2}$, we can ap-

proximate the function u'' by a new N-dimensional vector $D\vec{u}$ (with $h = 1/(N-1)$), where the matrix D is given by

$$D = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ \cdots & \cdots & \cdots & \cdots & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & \end{bmatrix}. \tag{D.29}$$

The matrix D is often called the finite difference discretization of the differential operator $\frac{\partial^2}{\partial x^2}$. Note that the matrix D is tri-diagonal. We can use either unary embedding or antiferromagnetic embedding for the matrix D (for details, see [42, Appendix B]).

Similarly, the finite difference method can be applied to the 2D Schrödinger equation (D.28). The discretized quantum Hamiltonian reads,

$$\hat{H} = -\frac{1}{2}(D \otimes I) + I \otimes D) + U, \tag{D.30}$$

where I is the N-by-N identity matrix and U is a N^2-by-N^2 diagonal matrix such that

$$U\,|z_j\rangle\,|z_k\rangle = V(z_j, z_k)\,|z_j\rangle\,|z_k\rangle\,.$$

Neutral-atom simulator and antiferromagnetic embedding

The QuEra quantum simulator is based on neutral atoms. The atoms are placed individually and deterministically on a two-dimensional plane by optical tweezers [189, 190]. On the

QuEra quantum simulator, a qubit is realized by an atom with an internal ground state $|0\rangle$ and an excited Rydberg state $|1\rangle$. The Rydberg atoms are long-lived and can be coherently controlled by external laser pulses. The evolution of the neutral atoms is described by the Schrödinger equation $i\hbar\frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle = \hat{H}(t)|\psi(t)\rangle$, where the system Hamiltonian is given by

$$\frac{\hat{H}(t)}{\hbar} = \sum_j \left(\frac{\Omega_j(t)}{2}\hat{X}_j - \Delta_j(t)\hat{n}_j\right) + \sum_{j<k}\frac{C_6}{|\mathbf{r}_j - \mathbf{r}_k|^6}\hat{n}_j\hat{n}_k. \tag{D.31}$$

In the Hamiltonian, $\hat{X}_j = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\hat{n}_j = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ are the Pauli-X operator and the number operator acting on the j-th qubit, respectively. $\Omega_j(t)$, $\Delta_j(t)$ denotes the Rabi frequency and local detuning of the driving laser field on qubit j. C_6 is the Rydberg interaction constant that depends on the particular Rydberg atom used. $\mathbf{r}_j$ denotes the position vector of the j-th qubit. In our experiment, we use the neutral-atom analog quantum computer fabricated by QuEra Computing Inc. In each simulation, we can initiate no more than 256 atoms, and the Rydberg interaction constant is $C_6 = 862690 \times 2\pi\ \mathrm{MHz}\cdot\mu\mathrm{m}^6$.

In the system Hamiltonian (D.31), we have the atom-atom interactions $\hat{n}_j\hat{n}_k$ with positive coefficients. This structure is very suitable for antiferromagnetic embedding. In Lemma 20, we show how to engineer a penalty Hamiltonian for antiferromagnetic embedding by tuning the local detuning of neutral atoms.

Lemma 20. *Given an integer $N > 1$, let $n = N - 1$. Define the penalty Hamiltonian:*

$$\hat{H}_{\mathrm{pen}} = \sum_{1\le j<k\le n}\frac{C_6}{|\mathbf{r}_j - \mathbf{r}_k|^6}\hat{n}_j\hat{n}_k - \sum_{j=1}^{n}\Delta_j\hat{n}_j, \tag{D.32}$$

where the atoms are placed on a straight line with equal distance $r > 0$, i.e., $\mathbf{r}_j = (0, (j-1)r)$, and the local detuning values are:

$$\Delta_j = \begin{cases} \frac{C_6}{r^6}\left(\sum_{m=1}^{\lfloor (n+1-j)/2 \rfloor} \frac{1}{(2m-1)^6} + \sum^{\lfloor j/2 \rfloor} \frac{1}{(2m)^6}\right), & (\textit{for odd } j) \\ \frac{C_6}{r^6}\left(\sum_{m=1}^{\lfloor (n-j)/2 \rfloor} \frac{1}{(2m-1)^6} + \sum^{\lfloor (j-1)/2 \rfloor} \frac{1}{(2m)^6}\right). & (\textit{for even } j) \end{cases} \tag{D.33}$$

Then, the penalty Hamiltonian (D.32) has an N-fold degenerate ground-energy subspace $\mathcal{S}$ which is spanned by all the n-bit antiferromagnetic code (for details, see [42, Appendix B]).

Proof. We use $|w_k\rangle$ to denote the antiferromagnetic code for $k = 1, \ldots, N$. Define the following two operators

$$H'_Z = \sum_{1 \le j < k \le n} \frac{C_6}{|\mathbf{r}_j - \mathbf{r}_k|^6} \hat{n}_j \hat{n}_k, \quad H''_Z = -\sum_{j=1}^{n} \Delta_j \hat{n}_j,$$

where $\mathbf{r}_j = (0, (j-1)r)$, and we consider an ansatz

$$\hat{H}_{\text{pen}} = H'_Z + H''_Z.$$

Since the Hamiltonian H'_Z is in antiferromagnetic ordering, its low-energy subspace is spanned by antiferromagnetic code. However, these low-energy states are not degenerate. To get the N-degenerate ground-energy subspace, we solve a linear system to compute Δ_j in $\hat{H}_{\text{pen}}$.

First, we try to match the energy of $|w_1\rangle$ and $|w_2\rangle$. This requires the following identity holds:

$$\langle w_1 | \hat{H}_{\text{pen}} | w_1 \rangle = \langle w_2 | \hat{H}_{\text{pen}} | w_2 \rangle . \tag{D.34}$$

Note that $w_1 = 0101\ldots$ and $w_2 = 1101\ldots$, and the 2-tensor $\hat{n}_j\hat{n}_k = |11\rangle_{jk}\langle 11|_{jk}$ is visible only when the atoms j and k are simultaneously excited, so we have that,

$$\langle w_2| H_Z' |w_2\rangle - \langle w_1| H_Z' |w_1\rangle = \sum_{m=1}^{\lfloor n/2\rfloor} \frac{C_6}{r^6(2m-1)^6}.$$

Meanwhile, we have $\langle w_2| H_Z'' |w_2\rangle - \langle w_1| H_Z'' |w_1\rangle = -\Delta_1$. To have the identity (D.34), we must have

$$\Delta_1 = \frac{C_6}{r^6}\sum_{m=1}^{\lfloor n/2\rfloor} \frac{1}{(2m-1)^6}.$$

Similarly, we can compute Δ_2. Comparing $w_2 = 1101\ldots$ and $w_3 = 1001\ldots$, we have

$$\langle w_2| H_Z^{(1)} |w_2\rangle - \langle w_3| H_Z^{(1)} |w_3\rangle = \frac{C_6}{r^6} + \sum_{m=1}^{\lfloor n/2\rfloor-1} \frac{C_6}{r^6(2m)^6}.$$

Also, we have $\langle w_2| H_Z'' |w_2\rangle - \langle w_3| H_Z'' |w_3\rangle = -\Delta_2$, it follows that

$$\Delta_2 = \frac{C_6}{r^6}\left(\frac{1}{1^6} + \sum_{m=1}^{\lfloor n/2\rfloor-1} \frac{1}{(2m)^6}\right).$$

We can compute all local detuning Δ_j following the same procedure. It turns out that if we choose Δ_j as in (D.33), the ground-energy subspace of $\hat{H}_{\text{pen}}$ is indeed spanned by the codeword states $|w_j\rangle$ for $j = 1, \ldots, N$. □

Experiment setup and result

We perform experiments of simulating the 2D Schrödinger equation using the QuEra Aquila processor. The programmable quantum Hamiltonian on QuEra Aquila reads [191],

$$\frac{\hat{H}(t)}{\hbar} = \frac{\Omega(t)}{2}\left(\sum_j e^{i\phi(t)}\left|0\right\rangle\left\langle 1\right|_j + e^{-i\phi(t)}\left|1\right\rangle\left\langle 0\right|_j\right) - \Delta(t)\left(\sum_j \hat{n}_j\right) + \sum_{j<k}\frac{C_6}{|\mathbf{r}_j - \mathbf{r}_k|^6}\hat{n}_j\hat{n}_k, \tag{D.35}$$

where $\Omega(t)$, $\phi(t)$, and $\Delta(t)$ denote the Rabi frequency, laser phase, and the detuning of the driving laser field on atom (qubit) j coupling the two states $|0\rangle$ (ground state) and $|1\rangle$ (Rydberg state). It is worth noting that all programmable control parameters are *global*.

Hamiltonian embedding and the engineering of the potential field. We use the antiferromagnetic embedding the simulate the quantum dynamics. Ideally, we want to use Lemma 20 to engineer a perfect antiferromagnetic embedding of the discretized Laplacian operator $-\frac{1}{2}\left(D \otimes I + I \otimes D\right)$ (see (D.30)). However, currently (as of the fall of 2023), the QuEra Aquila processor does not support user-specified local detuning (see (D.35)). The lack of local detuning poses a significant restriction on the Hamiltonian we can engineer on the QuEra Aquila processor.

In our QuEra experiment, we follow the intuition of Lemma 20 and arrange 12 neutral atoms as in Figure 5.1D. The 12 neutral atoms are grouped into 2 vertical chains. Each chain has 6 atoms. The (sub-)system Hamiltonian for each chain reads

$$H_0 = \sum_{1\le j<k\le 6}\frac{C_6}{(k-j)^6 r^6}\hat{n}_j\hat{n}_k - \Delta\sum_{j=1}^{6}\hat{n}_j, \tag{D.36}$$

where $r = 5.9\,\mu\mathrm{m}$ is the distance between two adjacent atoms, and Δ is a global detuning.

It is clear that the actual Hamiltonian (D.36) is similar to the penalty Hamiltonian specified in (D.32), except that we can not specify individual detuning Δ_j for each atom. In this case, the codeword subspace of antiferromagnetic embedding

$$\mathcal{S} = \{|010101\rangle, |010100\rangle, |010110\rangle, |010010\rangle, |011010\rangle, |001010\rangle, |101010\rangle\}$$

coincide with the low-energy subspace of H_0 but not exactly the ground-energy subspace. This non-degeneracy can be realized as an *effective* potential field. Namely, we may regard the operator (D.32) as the *desired* penalty Hamiltonian (as in (D.32)) plus an operator that embeds *a* potential field (although this potential field is not very smooth and it does not correspond to a familiar closed-form function). Similarly, the system Hamiltonian of the two atom chains (Figure 5.1D) can be regarded as the penalty Hamiltonian for 2D antiferromagnetic embedding plus an effective potential operator. In Figure D.2A, we plot the effective 2D potential function at $\Delta = 5 \times 10^7$ rad/s. The potential is shown as a discretized function on the 2D mesh $\{(z_j, z_k) : j, k = 1, \ldots, N\}$.

Initial state preparation. The QuEra Aquila machine is initialized to the all-zeros state. This default initial state is not in the embedding subspace of the antiferromagnetic embedding. In our experiment, we use a *3-stage* state preparation subroutine [125] to prepare an initial state that has a significant overlap (around 73%) with the embedding subspace. Since the initial all-zeros state is the ground state when $\Omega = 0\,\mathrm{MHz}$ and $\Delta < 0\,\mathrm{MHz}$, this state preparation pulse is quasi-adiabatic and prepares a state with high overlap with the antiferromagnetic encoding subspace.

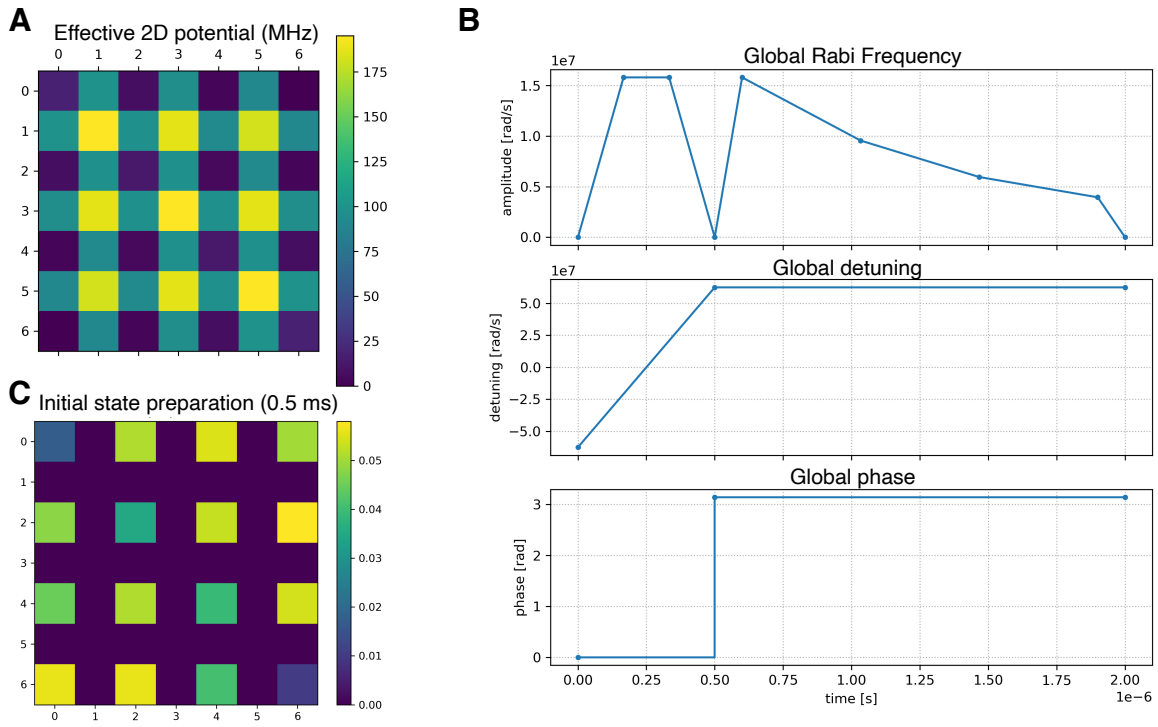


Figure D.2: Simulating real-space dynamics on QuEra Aquila. **A.** The effective potential field engineered by Rydberg interactions. The unit of the potential is MHz. The potential field is normalized such that the minimal potential is 0.

In our pulse, we use a trapezoidal pulse for the Rabi frequency with maximum frequency $\Omega_{\max} = 15.8\,\mathrm{MHz}$ and a linear ramp for the detuning from $-50\,\mathrm{MHz}$ to $50\,\mathrm{MHz}$. The duration of initial state preparation is $0.5\ \mu$s. The initial state preparation pulses are shown in Figure D.2C (see the first $0.5\ \mu$s). The engineered initial state (e.g., the quantum state at $t = 0.5\ \mu$s) is shown in Figure D.2B.

Time-rescaling and parameters for analog quantum simulation. On the QuEra Aquila device, all parameters are specified with a physical unit. We conducted our experiments through Amazon Braket, where the unit of the Rabi frequency Ω is radians per second (rad/s) and the evolution is in the unit of second (s) [192, Section 1.3]. Note that $1\ \mathrm{rad/s} = \frac{1}{2\pi}\ \mathrm{Hz}$.

We can establish a relation between these QuEra physical parameters with the abstract

unitless model Hamiltonian (D.30) via time-rescaling. We assume the quantum simulation is restricted to the unit square $[0,1]^2$ (with Dirichlet boundary conditions), and each dimension is discretized into 8 pieces. Therefore, the discretized differential operator is D in (D.29) with $h = 1/8$. Therefore, simulating a physical Hamiltonian (on QuEra Aquila) $Q = -\frac{\Omega(t)}{2}\sum_j X_j$ for time T (seconds) is equivalent to simulating the Hamiltonian $Q = -\varphi(t)\frac{1}{2h^2}\sum_j X_j$ (i.e., the antiferromagnetic embedding of the discretized Laplacian $\varphi(t)D/2h^2$) for (unitless) time

$$T_{\text{eff}} = 10^6 \cdot T,$$

where $\varphi(t) = \frac{h^2\Omega(t)}{2\pi\cdot 10^6}$. In our experiments, we run the simulation on QuEra Aquila for $T = 0.5, 1, 1.5$ μs (plus an initial state preparation duration of 0.5 μs). Effectively, we are simulating the real-space quantum dynamics for $T_{\text{eff}} = 0.5, 1, 1.5$.

Post-selection. The global Rabi frequency corresponds to the time-dependent function (i.e., $\varphi(t)$) in the kinetic energy. We gradually decrease the global Rabi frequency (see Figure D.2B for the control pulses with total evolution time $T = 2.0$ μs) so the quantum state will eventually find the low-energy configuration regarding the effective potential field. We consider a measured sample *legit* if it is in the embedding subspace. In our experiment on QuEra Aquila, approximately half of the samples in the measurement are legit. In Figure 5.1E, we plot the distribution of the legit samples for evolution time $T = 0.5, 1, 1.5$ (corresponding to total physical evolution time $T = 1, 1.5, 2$ μs).

Bibliography

[1] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[2] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311, 1984.

[3] Yuri Manin. Computable and uncomputable. *Sovetskoye Radio*, 1980.

[4] Richard P Feynman. Simulating physics with computers. *Int. j. Theor. phys*, 21, 1982.

[5] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[6] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.

[7] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014. arXiv:1411.4028.

[8] Matthew B Hastings. The power of adiabatic quantum computation with no sign problem. *Quantum*, 5:597, 2021.

[9] András Gilyén, Matthew B Hastings, and Umesh Vazirani. (sub) exponential advantage of adiabatic quantum computation with no sign problem. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1357–1369, 2021.

[10] Ruslan Shaydulin, Changhao Li, Shouvanik Chakrabarti, Matthew DeCross, Dylan Herman, Niraj Kumar, Jeffrey Larson, Danylo Lykov, Pierre Minssen, Yue Sun, et al. Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem. *arXiv preprint arXiv:2308.02342*, 2023.

[11] Fernando GSL Brandao and Krysta M Svore. Quantum speed-ups for solving semidefinite programs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 415–426. IEEE, 2017.

[12] Joran Van Apeldoorn and András Gilyén. Improvements in quantum sdp-solving with applications. In *Proceedings of 46th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2019. arXiv:1804.05058.

[13] Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M Svore, Xiaodi Wu, et al. Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning. *Leibniz international proceedings in informatics*, 2019.

[14] Joran Van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum SDP-solvers: Better upper and lower bounds. *Quantum*, 4:230, 2020.

[15] Iordanis Kerenidis and Anupam Prakash. Quantum gradient descent for linear systems and least squares. *Physical Review A*, 101(2):022316, 2020.

[16] Brandon Augustino, Giacomo Nannicini, Tamás Terlaky, and Luis F Zuluaga. Quantum interior point methods for semidefinite optimization. *Quantum*, 7:1110, 2023.

[17] Baihe Huang, Shunhua Jiang, Zhao Song, Runzhou Tao, and Ruizhe Zhang. A faster quantum algorithm for semidefinite programming via robust IPM framework. 2022. arXiv:2207.11154.

[18] Joran van Apeldoorn and András Gilyén. Quantum algorithms for zero-sum games. 2019. arXiv:1904.03180.

[19] Adam Bouland, Yosheb M Getachew, Yujia Jin, Aaron Sidford, and Kevin Tian. Quantum speedups for zero-sum games via improved dynamic gibbs sampling. In *International Conference on Machine Learning*, pages 2932–2952. PMLR, 2023.

[20] Mohammadhossein Mohammadisiahroudi, Ramin Fakhimi, and Tamás Terlaky. Efficient use of quantum linear system algorithms in interior point methods for linear optimization. 2022. arXiv:2205.01220.

[21] Simon Apers and Sander Gribling. Quantum speedups for linear programming via interior point methods. 2023. arXiv:2311.03215.

[22] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, 2015.

[23] Olivia Di Matteo, Vlad Gheorghiu, and Michele Mosca. Fault-tolerant resource estimation of quantum random-access memories. *IEEE Transactions on Quantum Engineering*, 1:1–13, 2020.

[24] B David Clader, Alexander M Dalzell, Nikitas Stamatopoulos, Grant Salton, Mario Berta, and William J Zeng. Quantum resources required to block-encode a matrix of classical data. *IEEE Transactions on Quantum Engineering*, 3:1–23, 2022.

[25] Samuel Jaques and Arthur G Rattew. QRAM: A survey and critique. 2023. arXiv:2305.10310.

[26] Brandon Augustino, Jiaqi Leng, Giacomo Nannicini, Tamás Terlaky, and Xiaodi Wu. A quantum central path algorithm for linear optimization. 2023. arXiv:2311.03977.

[27] Chenyi Zhang and Tongyang Li. Quantum lower bounds for finding stationary points of nonconvex functions. In *International Conference on Machine Learning*, pages 41268–41299. PMLR, 2023.

[28] Patrick Rebentrost, Maria Schuld, Leonard Wossnig, Francesco Petruccione, and Seth Lloyd. Quantum gradient descent and Newton's method for constrained polynomial optimization. *New Journal of Physics*, 21(7):073023, 2019.

[29] Keren Li, Shijie Wei, Pan Gao, Feihao Zhang, Zengrong Zhou, Tao Xin, Xiaoting Wang, Patrick Rebentrost, and Guilu Long. Optimizing a polynomial function on a quantum processor. *npj Quantum Information*, 7(1):16, 2021.

[30] Weijie Su, Stephen P Boyd, and Emmanuel J Candes. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. In *NIPS*, volume 14, pages 2510–2518, 2014.

[31] Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

[32] Jon Jacobsen, Owen Lewis, and Bradley Tennis. Approximations of continuous newton's method: an extension of cayley's problem. *Electronic Journal of Differential Equations (EJDE)[electronic only]*, 2007:163–173, 2007.

[33] Andrew M Childs and Joshua Young. Optimal state discrimination and unstructured search in nonlinear quantum mechanics. *Physical Review A*, 93(2):022314, 2016.

[34] Jin-Peng Liu, Herman Øie Kolden, Hari K Krovi, Nuno F Loureiro, Konstantina Trivisa, and Andrew M Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proceedings of the National Academy of Sciences*, 118(35):e2026805118, 2021.

[35] A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. 1995. arXiv:quant-ph/9511026.

[36] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[37] Andrew M Childs, Robin Kothari, and Rolando D Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6):1920–1950, 2017.

[38] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019.

[39] Richard P Feynman, Albert R Hibbs, and Daniel F Styer. *Quantum mechanics and path integrals*. Courier Corporation, 2010.

[40] Jiaqi Leng, Ethan Hickman, Joseph Li, and Xiaodi Wu. Quantum Hamiltonian Descent, 2023. arXiv:2303.01471.

[41] Jiaqi Leng, Yufan Zheng, and Xiaodi Wu. A quantum-classical performance separation in nonconvex optimization, 2023. arXiv:2311.00811.

[42] Jiaqi Leng, Joseph Li, Yuxiang Peng, and Xiaodi Wu. Expanding hardware-efficiently manipulable hilbert space via Hamiltonian embedding, 2024. arXiv:2401.08550.

[43] Samuel Kushnir, Jiaqi Leng, Yuxiang Peng, Lei Fan, and Xiaodi Wu. Qhdopt: A software for nonlinear optimization with quantum Hamiltonian decent, 2024.

[44] E Weinan. Machine learning and computational mathematics, 2020. arXiv:2009.14596.

[45] Samuel Burer and Adam N Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97–106, 2012.

[46] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[47] Prateek Jain, Purushottam Kar, et al. Non-convex optimization for machine learning. *Foundations and Trends in Machine Learning*, 10(3-4):142–363, 2017.

[48] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[49] Jiasi Chen and Xukan Ran. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019.

[50] Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer Relu neural networks. In *International Conference on Machine Learning*, pages 4433–4441. PMLR, 2018.

[51] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2016. arXiv:1609.04747.

[52] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science*, 292(5516):472–475, 2001.

[53] Sergio Boixo, Troels F Rønnow, Sergei V Isakov, Zhihui Wang, David Wecker, Daniel A Lidar, John M Martinis, and Matthias Troyer. Evidence for quantum annealing with more than one hundred qubits. *Nature physics*, 10(3):218–224, 2014.

[54] Shouvanik Chakrabarti, Andrew M Childs, Tongyang Li, and Xiaodi Wu. Quantum algorithms and lower bounds for convex optimization. *Quantum*, 4:221, 2020.

[55] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Convex optimization using quantum oracles. *Quantum*, 4:220, 2020.

[56] Tongyang Li, Shouvanik Chakrabarti, and Xiaodi Wu. Sublinear quantum algorithms for training linear and kernel-based classifiers. In *International Conference on Machine Learning*, pages 3815–3824. PMLR, 2019.

[57] Chenyi Zhang, Jiaqi Leng, and Tongyang Li. Quantum algorithms for escaping from saddle points. *Quantum*, 5:529, 2021.

[58] Michael Betancourt, Michael I Jordan, and Ashia C Wilson. On symplectic optimization, 2018. arXiv:1802.03653.

[59] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[60] Chris J Maddison, Daniel Paulin, Yee Whye Teh, Brendan O'Donoghue, and Arnaud Doucet. Hamiltonian descent methods, 2018. arXiv:1809.05042.

[61] Gheorghe Nenciu. Linear adiabatic theory. Exponential estimates. *Communications in mathematical physics*, 152(3):479–496, 1993.

[62] Peter D Hislop and Israel Michael Sigal. *Introduction to spectral theory: With applications to Schrödinger operators*, volume 113. Springer Science & Business Media, 2012.

[63] M Van den Berg. On condensation in the free-boson gas and the spectrum of the Laplacian. *Journal of Statistical Physics*, 31:623–637, 1983.

[64] Mark S Ashbaugh and Rafael Benguria. Optimal lower bound for the gap between the first two eigenvalues of one-dimensional Schrödinger operators with symmetric single-well potentials. *Proceedings of the American Mathematical Society*, 105(2):419–424, 1989.

[65] Shing-Tung Yau. Nonlinear analysis in geometry, monographies de lenseignement mathématique, vol. 33. *L'Enseignement Mathématique, Geneva*, page 54, 1986.

[66] Ben Andrews and Julie Clutterbuck. Proof of the fundamental gap conjecture. *Journal of the American Mathematical Society*, 24(3):899–916, 2011.

[67] David J Griffiths and Darrell F Schroeter. *Introduction to Quantum Mechanics (Third Edition)*. Cambridge University Press, 2018.

[68] Shing-Tung Yau. Gap of the first two eigenvalues of the Schrödinger operator with non-convex potential, 2009. arXiv:0902.2253.

[69] EM Harrell. Double wells, comm. math. phys. 75. 1980.

[70] Barry Simon. Semiclassical analysis of low lying eigenvalues. i. non-degenerate minima: Asymptotic expansions. In *Annales de l'IHP Physique théorique*, volume 38, pages 295–308, 1983.

[71] Tosio Kato. On the adiabatic theorem of quantum mechanics. *Journal of the Physical Society of Japan*, 5(6):435–439, 1950.

[72] Stefan Teufel. *Adiabatic perturbation theory in quantum dynamics*. Springer Science & Business Media, 2003.

[73] Evgeny Mozgunov and Daniel A Lidar. Quantum adiabatic theorem for unbounded Hamiltonians with a cutoff and its application to superconducting circuits. *Philosophical Transactions of the Royal Society A*, 381(2241):20210407, 2023.

[74] JJ Sakurai and Napolitano J. *Modern Quantum Mechanics (Second Edition)*. Addison-Wesley, 2011.

[75] Andrew M Childs and Jeffrey Goldstone. Spatial search by quantum walk. *Physical Review A*, 70(2):022314, 2004.

[76] Andrew M. Childs, Jiaqi Leng, Tongyang Li, Jin-Peng Liu, and Chenyi Zhang. Quantum simulation of real-space dynamics. *Quantum*, 6:860, 2022. arXiv:2203.17006.

[77] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *Journal of the ACM (JACM)*, 60(6):1–39, 2013.

[78] Yang Xiang, DY Sun, W Fan, and XG Gong. Generalized simulated annealing algorithm and its application to the thomson model. *Physics Letters A*, 233(3):216–220, 1997.

[79] David J Wales and Jonathan PK Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

[80] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.

[81] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[82] Troels F Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V Isakov, David Wecker, John M Martinis, Daniel A Lidar, and Matthias Troyer. Defining and detecting quantum speedup. *Science*, 345(6195):420–424, 2014.

[83] Division of Information Technology of University of Maryland. Zaratan HPC cluster. `https://hpcc.umd.edu/hpcc/zaratan.html`. Accessed: 2023-08-16.

[84] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[85] Yizhou Liu, Weijie J Su, and Tongyang Li. On quantum speedups for nonconvex optimization via quantum tunneling walks. *Quantum*, 7:1030, 2023.

[86] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[87] Dong An, Di Fang, and Lin Lin. Time-dependent unbounded Hamiltonian simulation with vector norm scaling. *Quantum*, 5:459, 2021.

[88] Ian D. Kivlichan, Nathan Wiebe, Ryan Babbush, and Alán Aspuru-Guzik. Bounding the costs of quantum simulation of many-body physics in real space. *Journal of Physics A: Mathematical and Theoretical*, 50(30):305301, 2017.

[89] 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.

[90] Naomichi Hatano and Masuo Suzuki. Finding exponential product formulas of higher orders. In *Quantum annealing and other optimization methods*, pages 37–68. Springer, 2005.

[91] 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.

[92] Andrew M Childs, Jin-Peng Liu, and Aaron Ostrander. High-precision quantum algorithms for partial differential equations. *Quantum*, 5:574, 2021.

[93] Andrew Macgregor Childs. *Quantum information processing in continuous time*. PhD thesis, Massachusetts Institute of Technology, 2004.

[94] Momin Jamil and Xin-She Yang. A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194, 2013.

[95] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved March 29, 2022, from `http://www.sfu.ca/~ssurjano`.

[96] Ali R. Al-Roomi. Unconstrained Single-Objective Benchmark Functions Repository, 2015. `https://www.al-roomi.org/benchmarks/unconstrained`.

[97] Jeffrey Cohen, Alex Khan, and Clark Alexander. Portfolio optimization of 60 stocks using classical and quantum algorithms, 2020. arXiv:2008.08669.

[98] Thomas Potok et al. Adiabatic quantum linear regression. *Scientific reports*, 11(1):1–10, 2021.

[99] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[100] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020.

[101] Dolev Bluvstein, Simon J Evered, Alexandra A Geim, Sophie H Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, et al. Logical quantum processor based on reconfigurable atom arrays. *Nature*, pages 1–3, 2023.

[102] Dong An, Andrew M Childs, and Lin Lin. Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters, 2023. arXiv:2312.03916.

[103] 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.

[104] Dominic W. Berry and Andrew M. Childs. Black-box Hamiltonian simulation and unitary implementation. *Quantum Info. Comput.*, 12(1–2):29–62, 1 2012.

[105] Dominic W Berry, Andrew M Childs, Richard Cleve, Robin Kothari, and Rolando D Somma. Exponential improvement in precision for simulating sparse Hamiltonians. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 283–292, 2014.

[106] Dominic W Berry, Andrew M Childs, Richard Cleve, Robin Kothari, and Rolando D Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Physical review letters*, 114(9):090502, 2015.

[107] Guang Hao Low and Isaac L Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Physical review letters*, 118(1):010501, 2017.

[108] Michael E Beverland, Prakash Murali, Matthias Troyer, Krysta M Svore, Torsten Hoeffler, Vadym Kliuchnikov, Guang Hao Low, Mathias Soeken, Aarthi Sundaram, and Alexander Vaschillo. Assessing requirements to scale to practical quantum advantage, 2022. arXiv:2211.07629.

[109] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 20–29, 2003.

[110] Andrew M Childs and Robin Kothari. Simulating sparse Hamiltonians with star decompositions. In *Theory of Quantum Computation, Communication, and Cryptography: 5th Conference, TQC 2010, Leeds, UK, April 13-15, 2010, Revised Selected Papers 5*, pages 94–103. Springer, 2011.

[111] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical review letters*, 100(16):160501, 2008.

[112] Guang Hao Low and Nathan Wiebe. Hamiltonian simulation in the interaction picture. 2018. arXiv:1805.00675.

[113] Daan Camps, Lin Lin, Roel Van Beeumen, and Chao Yang. Explicit quantum circuits for block encodings of certain sparse matrices, 2022. arXiv:2203.10236.

[114] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023.

[115] Anders Sørensen and Klaus Mølmer. Quantum computation with ions in thermal motion. *Physical review letters*, 82(9):1971, 1999.

[116] S. A. Caldwell, N. Didier, C. A. Ryan, E. A. Sete, A. Hudson, P. Karalekas, R. Manenti, M. P. da Silva, R. Sinclair, E. Acala, N. Alidoust, J. Angeles, A. Bestwick, M. Block, B. Bloom, A. Bradley, C. Bui, L. Capelluto, R. Chilcott, J. Cordova, G. Crossman, M. Curtis, S. Deshpande, T. El Bouayadi, D. Girshovich, S. Hong, K. Kuang, M. Lenihan, T. Manning, A. Marchenkov, J. Marshall, R. Maydra, Y. Mohan, W. O'Brien, C. Osborn, J. Otterbach, A. Papageorge, J.-P. Paquette, M. Pelstring, A. Polloreno, G. Prawiroatmodjo, V. Rawat, M. Reagor, R. Renzas, N. Rubin, D. Russell, M. Rust, D. Scarabelli, M. Scheer, M. Selvanayagam, R. Smith, A. Staley, M. Suska, N. Tezak, D. C. Thompson, T.-W. To, M. Vahidpour, N. Vodrahalli, T. Whyland, K. Yadav, W. Zeng, and C. Rigetti. Parametrically activated entangling gates using transmon qubits. *Phys. Rev. Appl.*, 10:034050, 9 2018.

[117] Andreas Wallraff, David I Schuster, Alexandre Blais, Luigi Frunzio, R-S Huang, Johannes Majer, Sameer Kumar, Steven M Girvin, and Robert J Schoelkopf. Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics. *Nature*, 431(7005):162–167, 2004.

[118] Michel H Devoret and Robert J Schoelkopf. Superconducting circuits for quantum information: an outlook. *Science*, 339(6124):1169–1174, 2013.

[119] Rainer Blatt and Christian F Roos. Quantum simulations with trapped ions. *Nature Physics*, 8(4):277–284, 2012.

[120] Christopher Monroe, Wes C Campbell, L-M Duan, Z-X Gong, Alexey V Gorshkov, Paul W Hess, Rajibul Islam, Kihwan Kim, Norbert M Linke, Guido Pagano, et al. Programmable quantum simulations of spin systems with trapped ions. *Reviews of Modern Physics*, 93(2):025001, 2021.

[121] Loïc Henriet, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. Quantum computing with neutral atoms. *Quantum*, 4:327, September 2020.

[122] Mark W Johnson, Mohammad HS Amin, Suzanne Gildert, Trevor Lanting, Firas Hamze, Neil Dickson, Richard Harris, Andrew J Berkley, Jan Johansson, Paul Bunyk, et al. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198, 2011.

[123] Richard Harris, Mark W Johnson, T Lanting, AJ Berkley, J Johansson, P Bunyk, E Tolkacheva, E Ladizinsky, N Ladizinsky, T Oh, et al. Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor. *Physical Review B*, 82(2):024511, 2010.

[124] Sepehr Ebadi, Tout T Wang, Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Dolev Bluvstein, Rhine Samajdar, Hannes Pichler, Wen Wei Ho, et al. Quantum phases of matter on a 256-atom programmable quantum simulator. *Nature*, 595(7866):227–232, 2021.

[125] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuletić, and M. D. Lukin. Quantum optimization of maximum independent set using Rydberg atom arrays. *Science*, 376(6598):1209–1215, 2022.

[126] 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.

[127] Kenneth Wright, Kristin M Beck, Sea Debnath, JM Amini, Y Nam, N Grzesiak, J-S Chen, NC Pisenti, M Chmielewski, C Collins, et al. Benchmarking an 11-qubit quantum computer. *Nature communications*, 10(1):5464, 2019.

[128] IonQ. Getting started with native gates, 2023. `https://ionq.com/docs/getting-started-with-native-gates`.

[129] Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[130] Sergey Bravyi, David P DiVincenzo, and Daniel Loss. Schrieffer–Wolff transformation for quantum many-body systems. *Annals of physics*, 326(10):2793–2826, 2011.

[131] Sergey Bravyi and Matthew Hastings. On complexity of the quantum Ising model. *Communications in Mathematical Physics*, 349(1):1–45, 2017.

[132] Dominic W Berry, Andrew M Childs, Yuan Su, Xin Wang, and Nathan Wiebe. Time-dependent Hamiltonian simulation with L^1-norm scaling. *Quantum*, 4:254, 1 2020.

[133] Anthony W Knapp. *Basic real analysis*. Springer Science & Business Media, 2007.

[134] Andrew M Childs, Aaron Ostrander, and Yuan Su. Faster quantum simulation by randomization. *Quantum*, 3:182, 2019.

[135] Dong An, Di Fang, and Lin Lin. Time-dependent Hamiltonian simulation of highly oscillatory dynamics and superconvergence for Schrödinger equation. *Quantum*, 6:690, 2022.

[136] Nicholas Chancellor. Domain wall encoding of discrete variables for quantum annealing and QAOA. *Quantum Science and Technology*, 4(4):045004, 2019.

[137] Stuart Hadfield, Zhihui Wang, Bryan O'gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, 2019.

[138] 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.

[139] Nicolas PD Sawaya. mat2qubit: A lightweight pythonic package for qubit encodings of vibrational, bosonic, graph coloring, routing, scheduling, and general matrix problems, 2022. arXiv:2205.09776.

[140] Matthias Christandl, Nilanjana Datta, Tony C Dorlas, Artur Ekert, Alastair Kay, and Andrew J Landahl. Perfect transfer of arbitrary states in quantum spin networks. *Physical Review A*, 71(3):032312, 2005.

[141] Toby S Cubitt, Ashley Montanaro, and Stephen Piddock. Universal quantum Hamiltonians. *Proceedings of the National Academy of Sciences*, 115(38):9497–9502, 2018.

[142] Leo Zhou and Dorit Aharonov. Strongly universal Hamiltonian simulators, 2021. arXiv:2102.02991.

[143] Sergey Bravyi, David P Divincenzo, Roberto I Oliveira, and Barbara M Terhal. The complexity of stoquastic local Hamiltonian problems, 2006. arXiv:quant-ph/0606140.

[144] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local Hamiltonian problem. *SIAM journal on computing*, 35(5):1070–1097, 2006.

[145] Roberto Oliveira and Barbara M Terhal. The complexity of quantum spin systems on a two-dimensional square lattice, 2005. arXiv:quant-ph/0504050.

[146] Stephen P Jordan and Edward Farhi. Perturbative gadgets at arbitrary orders. *Physical Review A*, 77(6):062329, 2008.

[147] Ryan Babbush, Nathan Wiebe, Jarrod McClean, James McClain, Hartmut Neven, and Garnet Kin-Lic Chan. Low-depth quantum simulation of materials. *Physical Review X*, 8(1):011044, 2018.

[148] Ivan Kassal, Stephen P Jordan, Peter J Love, Masoud Mohseni, and Alán Aspuru-Guzik. Polynomial-time quantum algorithm for the simulation of chemical dynamics. *Proceedings of the National Academy of Sciences*, 105(48):18681–18686, 2008.

[149] Ryan Babbush, Dominic W Berry, Jarrod R McClean, and Hartmut Neven. Quantum simulation of chemistry with sublinear scaling in basis size. *npj Quantum Information*, 5(1):92, 2019.

[150] Stephen Wiesner. Simulations of many-body quantum systems by a quantum computer, 1996. arXiv:quant-ph/9603028.

[151] Christof Zalka. Efficient simulation of quantum systems by quantum computers. *Fortschritte der Physik: Progress of Physics*, 46(6-8):877–879, 1998.

[152] Shi Jin, Xiantao Li, and Nana Liu. Quantum simulation in the semi-classical regime. *Quantum*, 6:739, 2022.

[153] Chia Cheng Chang, Kenneth S McElvain, Ermal Rrapaj, and Yantao Wu. Improving Schrödinger equation implementations with gray code for adiabatic quantum computers. *PRX Quantum*, 3(2):020356, 2022.

[154] R Harris, Y Sato, AJ Berkley, M Reis, F Altomare, MH Amin, K Boothby, P Bunyk, C Deng, C Enderud, et al. Phase transitions in a programmable quantum spin glass simulator. *Science*, 361(6398):162–165, 2018.

[155] Mark Saffman, Thad G Walker, and Klaus Mølmer. Quantum information with Rydberg atoms. *Reviews of modern physics*, 82(3):2313, 2010.

[156] Zdenek Dostál. *Optimal quadratic programming algorithms: with applications to variational inequalities*, volume 23. Springer Science & Business Media, 2009.

[157] D-Wave Systems Inc. D-Wave solver documentation, 2021. QPU-Specific Characteristics.

[158] Fabio Furini, Emiliano Traversi, Pietro Belotti, Antonio Frangioni, Ambros Gleixner, Nick Gould, Leo Liberti, Andrea Lodi, Ruth Misener, Hans Mittelmann, et al. QPLIB: a library of quadratic programming instances. *Mathematical Programming Computation*, 11(2):237–265, 2019.

[159] Yoshiaki Kawajir, Carl Laird, and A Wachter. Introduction to Ipopt: A tutorial for downloading, installing, and using Ipopt. Technical report, Tech. Rep., Carnegie Mellon University, 2006.

[160] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[161] Jaehyun Park and Stephen Boyd. General heuristics for nonconvex quadratically constrained quadratic programming, 2017. arXiv:1703.07870.

[162] IBM ILOG CPLEX. V12. 1: User's manual for CPLEX. *International Business Machines Corporation*, 46(53):157, 2009.

[163] Rodolfo A Quintero and Luis F Zuluaga. QUBO formulations of combinatorial optimization problems for quantum computing devices. In *Encyclopedia of Optimization*, pages 1–13. Springer, 2022.

[164] Pedro Maciel Xavier, Pedro Ripper, Tiago Andrade, Joaquim Dias Garcia, Nelson Maculan, and David E Bernal Neira. QUBO.jl: A Julia ecosystem for quadratic unconstrained binary optimization, 2023. arXiv:2307.02577.

[165] Y Matsuda. Research and development of common software platform for Ising machines. In *2020 IEICE General Conference*, 2020. Fixstars Amplify SDK.

[166] Mashiyat Zaman, Kotaro Tanahashi, and Shu Tanaka. PyQUBO: Python library for mapping combinatorial optimization problems to QUBO form. *IEEE Transactions on Computers*, 71(4):838–850, 2021. https://github.com/recruit-communications/pyqubo.

[167] Joseph T. Iosue. Qubovert, 2022. https://github.com/jtiosue/qubovert.

[168] Yoshihisa Yamamoto, Kazuyuki Aihara, Timothee Leleu, Ken-ichi Kawarabayashi, Satoshi Kako, Martin Fejer, Kyo Inoue, and Hiroki Takesue. Coherent Ising machines—optical neural networks operating at the quantum limit. *npj Quantum Information*, 3(1):49, 2017.

[169] D-Wave Inc. D-Wave Ocean SDK, 2023. https://github.com/dwavesystems/dwave-ocean-sdk.

[170] QuEra Computing Inc. Quera Bloqade.jl, 2024. https://github.com/QuEraComputing/Bloqade.jl.

[171] Minh-Thi Nguyen, Jin-Guo Liu, Jonathan Wurtz, Mikhail D Lukin, Sheng-Tao Wang, and Hannes Pichler. Quantum optimization with arbitrary connectivity using Rydberg atom arrays. *PRX Quantum*, 4(1):010316, 2023.

[172] Zachary Morrell and Carleton Coffrin. QuantumAnnealing.jl, 2022. https://github.com/lanl-ansi/QuantumAnnealing.jl.

[173] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, F Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, et al. Qiskit: An open-source framework for quantum computing. *Accessed on: Mar*, 16, 2019. https://docs.quantum.ibm.com/.

[174] The Cirq Developers. Cirq, 2019. https://github.com/quantumlib/Cirq.

[175] Amazon Web Services. Amazon Braket Python SDK, 2024. https://github.com/amazon-braket/amazon-braket-sdk-python.

[176] Kartik Singhal, Kesha Hietala, Sarah Marshall, and Robert Rand. Q# as a quantum algorithmic language, 2022. arXiv:2206.03532.

[177] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M Sohaib Alam, Guillermo Alonso-Linaje, B AkashNarayanan, Ali Asadi, et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations, 2018. arXiv:1811.04968.

[178] Guillaume Verdon, Juan Miguel Arrazola, Kamil Brádler, and Nathan Killoran. A quantum approximate optimization algorithm for continuous problems, 2019. arXiv:1902.00409.

[179] Farhad Khosravi, Ugur Yildiz, Artur Scherer, and Pooya Ronagh. Non-convex quadratic programming using coherent optical networks, 2022. arXiv:2209.04415.

[180] Yuxiang Peng, Jacob Young, Pengyu Liu, and Xiaodi Wu. SimuQ: A framework for programming quantum Hamiltonian simulation with analog compilation. *Proc. ACM Program. Lang.*, 8(POPL), 1 2024.

[181] J Robert Johansson, Paul D Nation, and Franco Nori. QuTiP: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760–1772, 2012.

[182] John P Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

[183] Alice Quillen. Phy411 lecture notes part 7 – integrators, 2018. PDF.

[184] Jérémie Roland and Nicolas J Cerf. Quantum search by local adiabatic evolution. *Physical Review A*, 65(4):042308, 2002.

[185] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate O(1/kˆ2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[186] Bin Shi, Weijie J Su, and Michael I Jordan. On learning rates and Schrödinger operators, 2020. arXiv:2004.06977.

[187] Daniel Hsu, Sham Kakade, and Tong Zhang. A tail inequality for quadratic forms of subgaussian random vectors. 2012.

[188] Jean Bourgain. On growth of Sobolev norms in linear schrödinger equations with smooth time dependent potential. *Journal d'Analyse Mathématique*, 77(1):315–348, 1999.

[189] Henning Labuhn, Daniel Barredo, Sylvain Ravets, Sylvain De Léséleuc, Tommaso Macrì, Thierry Lahaye, and Antoine Browaeys. Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum Ising models. *Nature*, 534(7609):667–670, 2016.

[190] Vincent Lienhard, Sylvain de Léséleuc, Daniel Barredo, Thierry Lahaye, Antoine Browaeys, Michael Schuler, Louis-Paul Henry, and Andreas M Läuchli. Observing the space- and time-dependent growth of correlations in dynamically tuned synthetic Ising models with antiferromagnetic interactions. *Physical Review X*, 8(2):021070, 2018.

[191] QuEra Computing Inc. Bloqade documentation, 2023. Bloqade.jl.

[192] Jonathan Wurtz, Alexei Bylinskii, Boris Braverman, Jesse Amato-Grill, Sergio H Cantu, Florian Huber, Alexander Lukin, Fangli Liu, Phillip Weinberg, John Long, et al. Aquila: Quera's 256-qubit neutral-atom quantum computer (version 1.0). Technical report, 2023. 2306.11727.