ABSTRACT

Title of Dissertation: QUANTUM ALGORITHMS FOR SCIENTIFIC COMPUTATION

Muhammad Junaid Aftab
Doctor of Philosophy, 2026

Dissertation Directed by: Dr. Konstantina Trivisa
Department of Mathematics

Dr. Haizhao Yang
Department of Mathematics

This thesis presents a mathematical analysis of several quantum algorithms and their applications in scientific computation. It is organized into two parts.

Part I investigates quantum algorithms for the quantum simulation problem, with particular emphasis on methods based on Lie–Trotter–Suzuki formulas, commonly known as product formulas. It develops a rigorous analysis of a digital quantum simulation algorithm: the multiproduct formula, a high-order method constructed from low-order product formulas. First, it provides a detailed characterization of the error scaling of the time-independent multiproduct formula for simulating quantum dynamics in finite-dimensional quantum systems. This analysis establishes both *commutator scaling* and a near-optimal dependence on the simulation time and target precision. The results show that the multiproduct formula achieves a polynomial speedup with respect to system size and evolution time, together with an exponential improvement in

the dependence on precision. The thesis then extends the multiproduct formula to the simulation of *non-unitary dynamics* by combining it with a recently proposed quantum algorithm for dissipative linear differential equations. The resulting algorithm is shown to exhibit *commutator scaling*, a property that is particularly important for the quantum simulation of differential equations. The performance of both algorithms is also examined in various application cases.

Part II proposes Chebyshev polynomial expansions as a unifying principle for a range of quantum algorithms and applications. The approach exploits the direct correspondence between Chebyshev polynomials and the parameters of quantum signal processing, thereby enabling the straightforward implementation of Chebyshev polynomials on a quantum computer. This observation is used to design fixed-parameter quantum circuits for function approximation. The method is applied to the approximation of functions in two classes of function spaces: (i) high-dimensional functions in the Korobov function space, a structured subspace of Sobolev spaces well suited to high-dimensional approximation and integration and (ii) countably parametric holomorphic maps, which arise naturally in applications such as uncertainty quantification and parametric partial differential equations. Finally, the thesis introduces a quantum algorithm for the non-uniform quantum Fourier transform based on a low-rank factorization technique derived from Chebyshev expansions. This method overcomes limitations of algorithms that rely on uniform grids and enables efficient quantum spectral transforms on non-uniform grids. These results demonstrate the versatility of Chebyshev polynomials across a broad range of quantum computational tasks.

QUANTUM ALGORITHMS FOR SCIENTIFIC COMPUTATION

by

Muhammad Junaid Aftab

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
2026

Advisory Committee:
Dr. Konstantina Trivisa, Chair/Advisor
Dr. Haizhao Yang, Co-Advisor
Dr. Jonathan Rosenberg
Dr. Runzhou Tao
Dr. Andrew Childs, Dean's Representative

© Copyright by
Muhammad Junaid Aftab
2026

Dedication

To my parents, my wife, and my son, whose love, sacrifices, and unwavering support made this work possible.

Acknowledgments

I am deeply grateful to my thesis advisors, Dr. Konstantina Trivisa and Dr. Haizhao Yang, for their invaluable guidance and support throughout this work.

Dr. Trivisa guided Part I of this thesis with care, patience, and encouragement. She created a supportive space for discussion, helped me grow intellectually, and consistently pushed me to think more clearly about my work. I am also deeply grateful for her nomination of me for the MathQuantum Graduate Fellowship, supported by the Math Quantum RTG grant DMS-2231533, which provided essential support during my graduate studies. Dr. Yang guided Part II of this thesis with generosity and openness. He encouraged me to explore different research directions and supported the flexibility I needed as my interests developed. His project ideas, thoughtful feedback, and steady encouragement helped create an environment in which I could pursue independent research with confidence.

I am also grateful to Dr. Rosenberg for his mentorship on my work in topological phases of matter. Although this work is not directly included in the present thesis, his ideas and perspective have had a strong influence on my research program. I especially appreciate his organization of the RIT on Geometry and Physics, which gave me the opportunity to explore connections between topology, geometry, and quantum computation. His expertise in K-theory and non-commutative geometry, along with his guidance and many helpful discussions, has played an important role in shaping my broader research program.

I am grateful to Dr. Andrew Childs and Dr. Runzhou Tao for serving on my thesis committee and for taking the time to read this manuscript and offer thoughtful feedback. I also thank Dr. Maria Cameron for her mentorship during my first year and for her support as a member of my preliminary examination committee. I would also like to thank the department

staff—Jessica Sadler, Ruth Yun, Jemma Natanson, and Trystan Denhard—for their steady administrative support throughout my time in the program. I am especially grateful to the AMSC Graduate Committee, and to former director Dr. Howard Elman, for admitting me to the interdisciplinary AMSC program. The flexibility of the program allowed me to pursue research at the intersection of pure and applied mathematics.

I owe my deepest gratitude to my family. My parents have been a constant source of love, encouragement, and faith in me throughout this journey. They gave me the freedom to pursue what I cared about and to find my own path, even when that path was not always clear. Their patience, guidance, and steady support have carried me through many academic and personal challenges. It is difficult to put into words how much I owe them. Their love and sacrifices have shaped who I am, and this work would not have been possible without them.

Most of all, I am grateful to my wife, Fizza Aslam, and to our son, Nael. When I began graduate school, I could not have known all the challenges that lay ahead. Fizza stood by me through every difficult period with patience, love, and encouragement, and her support helped me keep going when things felt overwhelming. The birth of Nael during the final year of my PhD brought a new kind of joy, purpose, and perspective to my life. This thesis is as much a reflection of their love and support as it is of my own work.

Table of Contents

Dedication ii

Acknowledgements iii

Table of Contents v

List of Tables vii

List of Figures viii

List of Abbreviations viii

Chapter 1: Introduction 1
- 1.1 Organization . . . 2
- 1.2 Notation . . . 2
- 1.3 Part I: Quantum Simulation via Multi-Product Formulas . . . 8
- 1.4 Part II: Chebyshev-Based Quantum Computation . . . 20

Chapter 2: Quantum Computation 30
- 2.1 Why Lie Theory? . . . 30
- 2.2 Lie Groups and Lie Algebras . . . 31
- 2.3 Lie-Trotter & Baker-Campbell-Hausdorff Formulas . . . 40
- 2.4 Quantum Theory . . . 45
- 2.5 Gate-Based Model . . . 49
- 2.6 Quantum Algorithms . . . 54

I Quantum Simulation via Multi-Product Formulas 63

Chapter 3: Multi-Product Formula with Commutator Scaling 64
- 3.1 Introduction . . . 64
- 3.2 Multi-Product Formula . . . 70
- 3.3 Multi-Product Formula Based on Second-Order Product Formula . . . 74
- 3.4 Applications . . . 87
- 3.5 Multi-Product Formula Based on High Order Product Formulas . . . 98
- 3.6 Conclusion . . . 101

Chapter 4: Linear Combination of Hamiltonian Simulation with Commutator Scaling 104
- 4.1 Introduction . . . 104
- 4.2 Multi-Product Formulas . . . 109
- 4.3 Linear Combination Of Hamiltonian Simulation . . . 110

4.4 Pre-Quadrature Error Analysis . . . 113
4.5 Post-Quadrature Complexity Analysis . . . 124
4.6 Extensions to Non-Homogeneous and Time-Dependent Cases . . . 150
4.7 Applications . . . 160
4.8 Conclusion . . . 177

II Chebyshev-Based Quantum Computation 180

Chapter 5: Chebyshev-Based Function Approximation with Quantum Circuits 181
5.1 Introduction . . . 181
5.2 Implementing Linear Combination of Chebyshev Polynomials . . . 186
5.3 Approximating Korobov Functions . . . 192
5.4 Approximating Countably Parametric Holomorphic Functions . . . 207
5.5 Conclusion . . . 219

Chapter 6: Non-Uniform Quantum Fourier Transform 223
6.1 Introduction . . . 223
6.2 Classical Algorithm for Non-Uniform Discrete Fourier Transform . . . 226
6.3 Quantum Algorithm for Non-Uniform Discrete Fourier Transform . . . 231
6.4 Error Analysis . . . 250
6.5 Numerical Results . . . 263
6.6 Conclusion . . . 273

Appendix A: Omitted Proofs in Chapter 2 276

Appendix B: Omitted Proofs in Chapter 3 278

Bibliography 295

List of Tables

1.1 Comparison of representative quantum algorithms for time-independent homogeneous linear differential equations. Here, α_A denotes a norm or block-encoding normalization of A and μ_Q^{gen} and μ_Q^{loc} are the quadrature-dependent commutator scales appearing in the general and local/extensive MPF estimates, respectively. 18

3.1 The ratio in Equation (3.82) is computed for various values of k (rows) and N (columns). 96

6.1 Comparison of κ and $\log(1 + K\kappa)$ for two γ regimes for $N = 2^6$, $\epsilon = 10^{-12}$. . . . 266
6.2 Statevector error for fixed $n = 4$ and $r = 4$ as p increases. 269
6.3 State-vector error for fixed $n = 4$ and $q = 2$. The left table varies p with $m = 2$ fixed, while the right table varies m with $p = 2$ fixed. The last column shows the ratio of consecutive rows. 269
6.4 Numerical validation of the $M_{\vec{s}}$ block encoding for $n = 5$. 273

List of Figures

2.1 Bloch sphere representation of a single-qubit state $|\psi\rangle$. The computational basis states $|0\rangle$ and $|1\rangle$ define the z-axis. The angles θ and ϕ denote the spherical coordinates of $|\psi\rangle$. 50
2.2 A generic single-qubit quantum circuit. The qubit $|\psi\rangle$ passes through a sequence of unitary gates U_1, U_2, U_3 and is then measured. 51
2.3 Single-qubit Pauli gates σ_X, σ_Y, and σ_Z represented in circuit notation. 51
2.4 A generic 3-qubit quantum circuit illustrating the application of single- and multi-qubit gates, with measurements performed on selected qubits at the end of the circuit. 52
2.5 Quantum oracle for multiplication and addition implementing $|x\rangle|y\rangle|z\rangle \mapsto |x\rangle|y\rangle|xy+z\rangle$. 55
2.6 Quantum circuit U_{LCU} implementing the LCU algorithm. 58
2.7 Quantum circuit implementing the Hadamard test with the LCU operator. The measurement of the ancilla qubit yields information about $\langle\psi| U_{\mathrm{LCU}} |\psi\rangle$. 59
2.8 Quantum circuit representation of $V_{\vec{\varphi}}(x)$ illustrating the sequence of phase rotations and signal operators used in quantum signal processing. 62

5.1 The circuit diagram for the quantum circuit $U_r(x)$ in Lemma 5.2.1. 189
5.2 The 1-dimensional nodal basis for V_3 consists of functions $\phi_{3,i}$, $1 \leq i \leq 7$. 193
5.3 The 1-dimensional hierarchical basis vectors for W_1, W_2 and W_3 are plotted. . . 194

6.1 Qubit ordering convention used throughout this work: computational qubits (top), quantum register (middle), and ancilla qubits (bottom). 232
6.2 Quantum circuit implementing $U_{\vec{v}_r}$ for $r \geq 1$ in the special case $n = 5$. 238
6.3 Quantum circuit implementing $O_{\vec{s}}^{(m)}$. 240
6.4 Quantum circuit implementing the operator $U_{\vec{u}_r} = \sum_{q=0}^{K-1} \alpha'_{qr} U_{qr}$. We have abbreviated $\vec{t} - \vec{s}/N$ by $\vec{y}$. 245
6.5 $\|(F_{\mathrm{II}} - \widetilde{F}_{\mathrm{II}}^{(K)})\|_2$ as a function of K for $N = 2^n$, $n = 2, \ldots, 10$, with accuracy $\epsilon = 10^{-10}$. For each N and node geometry, the error is averaged over 20 independent complex Gaussian test vectors. K^* denotes the minimal K required to achieve the prescribed tolerance. 264
6.6 Comparison of actual and predicted values of $\arccos(x)$ for odd p values ($p = 1, 3, 5$). The corresponding mean absolute errors are 0.5195115053, 0.2251069658, and 0.0645826060, respectively. 267
6.7 Comparison of computed and predicted outputs for different parameter of the quantum circuit $U_{v_{\vec{r}}}$ for $n \in \{4, 5, 6\}$ and $r \in \{2, 3, 4\}$ with fixed $p = 5$. 268
6.8 Comparison of computed and predicted outputs for different parameters of the quantum circuit U_q for $n \in \{4, 5, 6\}$ and $q \in \{0, 1, 2\}$ with $m = 4$ and $p = 5$. . . 270
6.9 Comparison of the computed and predicted for different parameters of the quantum circuit $U_{u_{\vec{r}}}$ for $n \in \{4, 5, 6\}$ and $r \in \{0, 1, 2\}$ with $m = 4$ and $p = 5$. 271

Chapter 1: Introduction

Quantum computation has emerged as a powerful paradigm for addressing problems that are intractable for classical computers. By exploiting intrinsically quantum-mechanical phenomena such as superposition and entanglement, quantum algorithms provide fundamentally new computational approaches and may significantly outperform classical methods for a broad class of problems. In scientific computing, the ability to process and manipulate information in high-dimensional Hilbert spaces enables novel algorithmic strategies. Quantum algorithms therefore offer the potential to efficiently simulate complex physical systems, perform large-scale and high-dimensional linear algebra, and accelerate computational tasks across the applied sciences.

This thesis develops and analyzes quantum algorithms for scientific computation, with an emphasis on mathematically grounded techniques and rigorous performance guarantees. It is organized into two main parts, each devoted to a distinct set of algorithmic tools and application domains in applied mathematics.

In Part I, we propose and analyze a quantum simulation algorithm. Using techniques from Lie theory and applied analysis, we study the algorithm in both time-independent and time-dependent settings. We establish rigorous error bounds and performance guarantees, and discuss several applications of the proposed approach. In Part II, we use Chebyshev polynomial approximation on quantum computers to derive approximation rates for quantum circuits over

a range of function spaces. We also develop a quantum algorithm for the non-uniform discrete Fourier transform based on a low-rank matrix factorization using Chebyshev polynomials.

1.1 Organization

The remainder of this chapter summarizes the main results of the thesis. Section 1.2 provides a brief overview of the notation used throughout this work. Section 1.3 reviews the quantum simulation problem and highlights our results on quantum simulation based on the multi-product formula. Section 1.4 summarizes our contributions to the approximation theory of quantum circuits and to non-uniform quantum Fourier transforms based on Chebyshev polynomials.

1.2 Notation

This section introduces the notation used throughout the thesis. Technical background will be provided in subsequent chapters. The purpose of this section is solely to establish notation used consistently throughout the thesis.

1.2.1 Standard Notation

We use the conventions $0! := 1$ and $0^0 := 1$. Let $\mathbb{R}$, $\mathbb{C}$, $\mathbb{N}$ and $\mathbb{Z}$ an denote the sets of real numbers, complex numbers, natural numbers and integers, respectively. We also write $\mathbb{N}$ as $\mathbb{Z}^+$. For $M \in \mathbb{N}$, let $\mathbb{Z}_M^d := \{0, \ldots, M-1\}^d$ be the d-fold Cartesian product of the cyclic group, $\mathbb{Z}_M$. All logarithms are natural logarithms unless otherwise stated. Let $\mathbb{T}$ denote the 1-dimensional torus (i.e., the unit circle) identified with the interval $[0, 1]$ with its endpoints identified. Let

S_k denotes the symmetric group on k letters, and for each $\sigma = (\sigma(1), \cdots, \sigma(k)) \in S_k$, d_σ is the number of descents in σ which is the number of positions $i \in \{1, 2, \ldots, n-1\}$ such that $\sigma(i+1) < \sigma(i)$. The symbol $\overleftarrow{\prod_k}$ denotes a product in which the elements are arranged with increasing indices from right to left. Similarly, the notation $\overrightarrow{\prod_k}$ is defined to denote a product with elements ordered with increasing indices from left to right. Specifically, we have

$$\overleftarrow{\prod_{i=1,\cdots,k}} A_i = A_k \cdots A_1, \qquad \overrightarrow{\prod_{i=1,\cdots,k}} A_i = A_1 \cdots A_k. \tag{1.1}$$

We also denote as $\overleftarrow{\prod_k}$ as $\prod_{i=k}^{1}$. Similarly, we also denote $\overrightarrow{\prod_k}$ as $\prod_{i=1}^{k}$. For a subset $A \subseteq \mathbb{R}, \mathbb{C}$, $\mathbb{1}_A(x)$ denotes the indicator function. If $f, g : \mathbb{N} \to \mathbb{R}^+$ are non-negative functions, we use the following standard notation from complexity theory:

1. $f(n) = \mathcal{O}(g(n))$ if and only if there exists a constant $C > 0$ and an integer $N \in \mathbb{N}$ such that $f(n) \leq Cg(n)$ for all $n \geq N$.

2. $f(n) = \Omega(g(n))$ if and only if there exists a constant $C > 0$ and an integer $N \in \mathbb{N}$ such that $f(n) \geq Cg(n)$ for all $n \geq N$.

3. $f(n) = \Theta(g(n))$ if and only if $f(n) = \mathcal{O}(g(n))$ and $f(n) = \Omega(g(n))$.

For $d \geq 1$, a multi-index $\vec{\alpha}$ is an element of $\mathbb{N}^d$. For a given multi-index $\vec{\alpha}$, the ℓ^1-norm $\|\vec{\alpha}\|_1$ and the ℓ^∞-norm $\|\vec{\alpha}\|_\infty$ are defined in the usual way (see Section 1.2.2). The special multi-index $\vec{1}$ is defined as $\vec{1} = (1, \cdots, 1)$. Similarly, the special multi-index $\vec{2}$ is defined as $\vec{2} = (2, \cdots, 2)$. We will consider the following component-wise arithmetic operations:

$$\vec{\alpha} \cdot \vec{\beta} := (\alpha_1\beta_1, \ldots, \alpha_d\beta_d), \qquad b \cdot \vec{\alpha} := (b\alpha_1, \ldots, b\alpha_d), \qquad c^{\vec{\alpha}} := (c^{\alpha_1}, \ldots, c^{\alpha_d}), \tag{1.2}$$

for $b, c \in \mathbb{R}$. Let $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. We will also consider multi-indicies $\vec{\nu} = (\nu_j)_{j\in\mathbb{N}} \in \mathbb{N}_0^{\mathbb{N}}$ of possibly infinite length. The notation $\operatorname{supp}\vec{\nu} \subseteq \mathbb{N}$ stands for the support of the multi-index, i.e. $\operatorname{supp}\vec{\nu} = \{j \in \mathbb{N} \mid \nu_j \neq 0\}$. The support size of $\vec{\nu}$ is $\|\vec{\nu}\|_0 := \#(\operatorname{supp}(\vec{\nu})) \leq \infty$. $\|\vec{\nu}\|_1$ and $\|\vec{\nu}\|_\infty$ are defined as above. We will use the (countable) set $\mathscr{F}$ of "finitely supported" multi-indices $\mathscr{F} := \{\vec{\nu} \in \mathbb{N}_0^{\mathbb{N}} \mid \|\vec{\nu}\|_1 < \infty\}$. For $\vec{\nu} \in \mathscr{F}$, we denote $\vec{y}^{\vec{\nu}} := \prod_{j\in\mathbb{N}} y_j^{\nu_j}$ and $\vec{\nu}! := \prod_{j\in\mathbb{N}} \nu_j!$. Formally, the infinite products contain only finitely many nontrivial factors, with the conventions $0! := 1$ and $0^0 := 1$. A partial order, $\leq$, on $\mathscr{F}$ is defined as $\vec{\mu} \leq \vec{\nu}$ if and only if $\mu_j \leq \nu_j$ for all $j \in \mathbb{N}$. A subset $\Lambda \subseteq \mathscr{F}$ is called downward closed ("d.c." for short), if $\vec{\nu} = (\nu_j)_{j\in\mathbb{N}} \in \Lambda$ implies $\vec{\mu} = (\mu_j)_{j\in\mathbb{N}} \in \Lambda$ for all $\vec{\mu} \leq \vec{\nu}$. A subset $\Lambda \subseteq \mathscr{F}$ is called anchored if $\vec{e}_j \in \Lambda$ implies that $\{\vec{e}_1, \vec{e}_2, ..., \vec{e}_j\} \subseteq \Lambda$ for all $j \in \mathbb{N}$.

1.2.2 Linear Algebra Notation

All vector spaces considered are finite-dimensional and defined over either $\mathbb{R}$ or $\mathbb{C}$. Vectors are denoted by bold lowercase Roman letters, or by unbold lowercase Roman letters with a $\vec{\cdot}$ symbol on top. Both notations will be used interchangeably. Linear operators—represented as matrices—are denoted by uppercase Roman letters. $\mathsf{M}(n, \mathbb{R})$ denotes the set of $n \times n$ matrices with entries in $\mathbb{R}$. $\mathsf{M}(n, \mathbb{C})$ is defined similarly. The following standard linear algebra notation, used throughout this thesis, is summarized below.

1. The operator $\operatorname{diag}(\cdot)$ denotes a diagonal matrix whose diagonal entries are given by the components of the vector argument. I denotes the identity matrix of appropriate dimension.

2. The symbol $\top$ denotes the transpose of a real matrix, while $*$ or $\dagger$ denotes the conjugate

transpose (Hermitian adjoint) of a complex matrix.

3. The notation $A \succeq 0$ denotes that A is a positive semi-definite matrix.

4. For matrices A_1, A_2 in $\mathbb{R}^{n\times n}$ or $\mathbb{C}^{n\times n}$, their commutator is denoted by $[A_1, A_2] = A_1A_2 - A_2A_1$. For $m \geq 2$, $[A_1A_2\cdots A_m]$ denotes the nested commutator $[A_1, [A_2, \cdots, [A_{m-1}, A_m]\cdots]]$. If $A_1, \ldots, A_{m-1} = A$ and $A_m = B$, we abbreviate this nested commutator as $\mathrm{ad}_A^{m-1}(B)$.

5. For a vector $\vec{a} = (a_1, \ldots, a_n)$ in $\mathbb{R}^n$ or $\mathbb{C}^n$, $\|\vec{a}\|_p$ denotes its vector ℓ^p norm, defined by $\|\vec{a}\|_p = (\sum_{j=1}^n |a_j|^p)^{1/p}$, for $1 \leq p < \infty$. The ℓ^∞ norm is defined as $\|\vec{a}\|_\infty = \max_{1\leq j\leq n} |a_j|$.

6. For a vector $\vec{a} = (a_1, \ldots, a_n)$ in $\mathbb{R}^n$ or $\mathbb{C}^n$, if f is a function then $f(\vec{a})$ denotes the componentwise application of f to $\vec{a}$; that is, f is applied to each entry of $\vec{a}$ individually.

7. For a matrix A in $\mathbb{R}^{n\times n}$ or $\mathbb{C}^{n\times n}$, $\|A\|_{p,q}$ denotes the operator norm induced by the vector ℓ^p and ℓ^q norms, defined as $\|A\|_{p,q} = \sup_{\vec{x}\neq 0} \frac{\|A\vec{x}\|_q}{\|\vec{x}\|_p}$. We denote $\|A\|_p := \|A\|_{p,p}$. The norm $\|A\|_2$ is the spectral norm of A, which equals the largest singular value of A. When no subscript is used, $\|A\|$ will refer to the spectral norm.

8. For a matrix A in $\mathbb{R}^{n\times n}$ or $\mathbb{C}^{n\times n}$, $\|A\|_F$ denotes the Frobenius norm of A, defined as $\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |A_{ij}|^2}$. Equivalently, $\|A\|_F$ is the square root of the sum of the squares of all singular values of A.

9. For a matrix A in $\mathbb{R}^{n\times n}$ or $\mathbb{C}^{n\times n}$, $\|A\|_{\max}$ denotes the maximum absolute value among the entries of A is defined as $\|A\|_{\max} := \max_{1\leq i,j\leq n} |A_{ij}|$. Note that $\|A\|_{\max}$ is not a matrix norm.

10. For a matrices A, B in $\mathbb{R}^{n\times n}$ or $\mathbb{C}^{n\times n}$, the Hadamard product $A \odot B$ is defined as the element-wise multiplication of their entries: $(A \odot B)_{ij} = A_{ij}B_{ij}$ for $1 \leq i, j \leq n$.

11. $\|A\|_{L^1[a,b]}$ denotes the L^1-norm of a time dependent matrix $A(t)$ on $[a, b]$, defined as $\int_a^b \|A(s)\|\, ds$.

Chapter 3 uses some non-standard linear algebra notation, which we summarize below. In Section 3.2, we will frequently consider operators of the form $A = \sum_{\gamma_1,\ldots,\gamma_k} A_{\gamma_1,\ldots,\gamma_k}$, where each operator $A_{\gamma_1,\ldots,\gamma_k}$ acts on the underlying finite-dimensional vector space. In this case, $\|A\|_1$ denotes the 1-norm of the operator A, defined as follows. If A can be decomposed as $A = \sum_{\gamma_1,\ldots,\gamma_k} A_{\gamma_1,\ldots,\gamma_k}$, then $\|A\|_1 = \sum_{\gamma_1,\ldots,\gamma_k} \|A_{\gamma_1,\ldots,\gamma_k}\|$. Note that this is a sum of spectral norms, not the standard matrix 1-norm. For simplicity of notation, we continue to denote this norm by $\|A\|_1$. $|\!|\!|A|\!|\!|_1$ denotes the induced 1-norm $|\!|\!|A|\!|\!|_1 = \max_j \max_{\gamma_j} \sum_{\substack{\gamma_1,\ldots,\gamma_{j-1}\\ \gamma_{j+1},\ldots,\gamma_k}} \|A_{\gamma_1,\ldots,\gamma_k}\|$. Note that $|\!|\!|A|\!|\!|_1 \leq \|A\|_1$.

The formal solution to the matrix-valued linear differential equation $\frac{dU(t)}{dt} = -iH(t)U(t)$ for $T_0 \leq t \leq T$ is written as $U(T_0, T) = \exp_{\mathcal{T}}(-i\int_{T_0}^{T} H(s)ds)$, where $\exp_{\mathcal{T}}$ denotes the time-ordered exponential operator. We also denote the time-ordered exponential operator as $\mathcal{T}e$. The reader is referred to [DF84] for more details. In what follows, we set $T_0 = 0$ and write $U(T) := U(0, T)$ when convenient.

1.2.3 Analysis Notation

We consider the sets $\mathbb{R}^{\mathbb{N}}$ and $\mathbb{C}^{\mathbb{N}}$ endowed with the product topology. Any subset such as $[-1, 1]^{\mathbb{N}}$ is then understood to be equipped with the subspace topology. For $\epsilon \in (0, \infty)$ we write $B_\epsilon^{\mathbb{N}} := \prod_{j\in\mathbb{N}} B_\epsilon \subseteq \mathbb{C}^{\mathbb{N}}$, where B_ϵ is the open ball of radius ϵ in $\mathbb{C}$. We use the convention

$\hat{f}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(x) e^{-i\xi x} dx$ and $f(x) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(\xi) e^{i\xi x} d\xi$ for the Fourier transform and the inverse Fourier transform, respectively. Let be a $\Omega \subseteq \mathbb{R}^d$ measurable set endowed with the d-dimensional Lebesgue measure. For $1 \leq p \leq \infty$, we consider the function space of Lebesgue integrable functions on Ω. It is defined as $L^p(\Omega) = \{f : \Omega \to \mathbb{R} \,|\, f \text{ is Lebesgue measurable and } \|f\|_{L^p(\Omega)} < \infty\}$. The corresponding norm is given by

$$\|f\|_{L^p(\Omega)} := \begin{cases} \left(\int_\Omega |f(x)|^p \, dx\right)^{1/p}, & \text{if } p < \infty, \\ \text{ess sup}_{x \in \Omega} |f(x)|, & \text{if } p = \infty. \end{cases} \tag{1.3}$$

We also consider the weighted Lebesgue integrable functions. Let $w : \Omega \to (0, \infty)$ be a measurable weight function. For $1 \leq p < \infty$, we consider the weighted Lebesgue space on Ω, defined by $L^p_w(\Omega) = \left\{f : \Omega \to \mathbb{R} \;\middle|\; f \text{ is Lebesgue measurable and } \|f\|_{L^p_w(\Omega)} < \infty\right\}$. The corresponding norm is given by

$$\|f\|_{L^p_w(\Omega)} = \left(\int_U |f(x)|^p \, w(x) \, dx\right)^{1/p}. \tag{1.4}$$

We will also consider the function space of summable sequences. Let X be a set equipped with the counting measure. For $1 \leq p \leq \infty$, we consider function space of summable sequences on X. It is defined as $\ell^p(X) = \{\vec{\alpha} : X \to \mathbb{R} \,|\, f \text{ is Lebesgue summable and } \|\vec{\alpha}\|_{\ell^p(X)} < \infty\}$. For $p < \infty$, the corresponding norm is given by $\|\vec{\alpha}\|_{\ell^p(X)} = \left(\sum_{x \in X} |\alpha_x|^p\right)^{1/p}$. For $p = \infty$, the norm is defined in the ℓ^∞ case. Lastly, consider the function space of Sobolev functions on $\Omega \subseteq \mathbb{R}^d$. Let $k \in \mathbb{N}$ and $1 \leq p \leq \infty$. For each multi-index $\vec{\alpha} \in \mathbb{N}^k$ of length k, let $D^{\vec{\alpha}} f = D^{\alpha_1} \cdots D^{\alpha_k} f$ $\vec{\alpha}$-th weak derivative of f. The Sobolev space is defined as $W^{k,p}(\Omega) := \{f \in L^p(\Omega) \mid D^{\vec{\alpha}} f \in$

$L^p(\Omega)$ for all $\alpha \in \mathbb{N}^d$ with $\|\vec{\alpha}\|_1 \leq k\}$. The corresponding norm is given by

$$\|f\|_{W^{k,p}(\Omega)} := \begin{cases} \left(\sum_{\|\vec{\alpha}\|_1 \leq k} \|D^{\vec{\alpha}} f\|_{L^p(\Omega)}^p\right)^{1/p}, & \text{if } p < \infty, \\ \max_{\|\vec{\alpha}\|_1 \leq k} \|D^{\vec{\alpha}} f\|_{L^\infty(\Omega)}, & \text{if } p = \infty. \end{cases} \tag{1.5}$$

When $p = 2$, we denote $W^{k,2}(\Omega)$ as $H^k(\Omega)$ for $k \in \mathbb{N}$. Moreover, $H_0^k(\Omega) = \{f \in H^k(\Omega) \mid D^{\vec{\alpha}} f|_{\partial\Omega} = 0$ for all $\alpha \in \mathbb{N}^d$ with $\|\vec{\alpha}\|_1 \leq k-1\}$ is the subspace of functions for which all weak derivatives up to order $k-1$ vanish on $\partial\Omega$, the boundary of Ω.

1.3 Part I: Quantum Simulation via Multi-Product Formulas

The quantum simulation problem is widely regarded as one of the most significant applications of quantum computing, with the potential to demonstrate practical quantum advantage [Fey82; Fey86] in domains such as quantum chemistry, materials science, and condensed matter physics [JLP12; Bau+20; Cao+19; Bab+18]. In this work, quantum simulation refers to the computation of solutions to linear differential equations of the form

$$\frac{d}{dt}u(t) = -A(t)u(t) + b(t), \quad t \in [0, T], \tag{1.6}$$

where, for each $t \in [0, T]$, $A(t) \in \mathbb{C}^{N \times N}$ and $b(t) \in \mathbb{C}^N$. Such equations arise in broader quantum simulation settings,notably in the simulation of finite-dimensional open quantum systems. We therefore colloquially refer to Equation (1.6) as describing *non-unitary dynamics*. The solution

to Equation (1.6) can be expressed as

$$u(t) = \exp_{\mathcal{T}}\left(-\int_0^t A(s)\,ds\right)u(0) + \int_0^t \exp_{\mathcal{T}}\left(-\int_s^t A(s')\,ds'\right)b(s)\,ds, \tag{1.7}$$

where $\exp_{\mathcal{T}}$ denotes the time-ordered exponential operator. An important special case is the time-independent quantum simulation problem, where $A(t) \equiv A$ and $b(t) \equiv b$ are independent of time. In this setting, Equation (1.7) reduces to $u(t) = e^{-tA}u(0) + \int_0^t e^{-(t-s)A}b(s)\,ds$.

Example 1.3.1 (Hamiltonian simulation problem)**.** An important class of quantum simulation problems is the Hamiltonian simulation problem. In this case, $b(t) \equiv 0$ and $A(t) = -iH(t)$, where, for each $t \in [0,T]$, $H(t) \in \mathbb{C}^{N\times N}$ is Hermitian. Then Equation (1.6) reduces to

$$\frac{d}{dt}u(t) = -iH(t)u(t), \quad t \in [0,T]. \tag{1.8}$$

Equation (1.8) describes the quantum dynamics of a finite-dimensional closed quantum system and is one of the most promising applications of quantum computing. We therefore colloquially refer to Equation (1.8) as describing *unitary dynamics*. Its solution is $u(t) = \exp_{\mathcal{T}}(-i\int_0^t H(s)\,ds)u(0)$. If $H(t) \equiv H$ is time-independent, the problem is called time-independent Hamiltonian simulation, and the solution simplifies to $u(t) = e^{-itH}u(0)$.

The primary objective of the quantum simulation problem is to implement the solution of Equation (1.6) on a quantum computer.

Problem 1.3.2. Given $A(t) \in \mathbb{C}^{N\times N}$ and $b(t) \in \mathbb{C}^N$ for each $t \in [0,T]$, a maximum simulation time T, a target error ϵ, and an initial state $|u(0)\rangle$, construct a quantum algorithm V such that $\|V|u(0)\rangle - |u(T)\rangle\| \le \epsilon$, where $|u(T)\rangle$ solves Equation (1.6).

The quantum simulation problem may also require V to be ϵ-close to the exact solution operator in the spectral norm. This requirement is stronger than the state-preparation condition in Problem 1.3.2. We work with this stronger condition in Part I, which addresses Problem 1.3.2 by presenting quantum algorithms for (i) the time-independent Hamiltonian simulation problem Equation (1.8) and (ii) the more general quantum simulation problem Equation (1.6) based on the Multi-Product Formula (MPF).

1.3.1 Time-Independent Hamiltonian Simulation via Multi-Product Formulas

The primary objective of the time-independent Hamiltonian simulation problem is to design quantum algorithms that implement e^{-itH} on a quantum computer. The development of such algorithms dates back to Lloyd's seminal work [Llo96]. Since then, substantial progress has been made on quantum algorithms for this problem. As noted above, we study this problem in the setting of Multi-Product Formula (MPF) based quantum algorithms. We first review time-independent product formulas and multi-product formulas, and then describe our contributions.

Product Formulas

Consider a finite-dimensional closed quantum system with Hamiltonian $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$. Product formulas assume that the time evolution generated by each term H_γ, $U_\gamma(t) = e^{-itH_\gamma}$, can be implemented efficiently on a quantum computer. A generic product formula with Ξ factors approximates the exact time-evolution operator $U(t) = e^{-itH}$ by $U_p(t) = \prod_{\xi=1}^{\Xi} \prod_{\gamma=1}^{\Gamma} e^{a_{\xi,\gamma} t H_{\pi_\xi(\gamma)}}$, where $a_{\xi,\gamma} \in \mathbb{R}$ and $\pi_\xi \in S_\Gamma$ is a permutation. $U_p(t)$ is of order p if its one-step error satisfies $\|U_p(t) - U(t)\| = \mathcal{O}(t^{p+1})$. The first-order product formula follows from the Lie–Trotter formula

in Lie theory [Hal15]. Higher-order formulas can be constructed using Suzuki recursion [Suz85; Suz91]. In particular, the first-order product formula is $U_1(t) = \prod_{\gamma=1}^{\Gamma} e^{-i(t/r)H_\gamma}$, while the second-order product formula is $U_2(t) = \prod_{\gamma=\Gamma}^{1} e^{-i(t/2r)H_\gamma} \prod_{\gamma=1}^{\Gamma} e^{-i(t/2r)H_\gamma}$. We do not discuss higher-order product formulas further, since they are not used in this work.

Multi-Product Formulas

The multi-product formula (MPF) approximates quantum time evolution using linear combinations of product formulas. For a short time interval Δ, an MPF has the form $U_{\mathrm{MP}}(\Delta) = \sum_{j=1}^{M} a_j \, U_p(\Delta/k_j)^{k_j}$, where $a_j \in \mathbb{R}$ and $k_j \in \mathbb{Z}^+$. Since $U_{\mathrm{MP}}(\Delta)$ is constructed from a fixed p-th order product formula, we refer to it as a p-th-order-based MPF. Although MPFs are built from fixed-order product formulas, they can achieve arbitrarily high convergence order. The key idea is that the leading-order errors contributed by the individual terms can be approximately canceled through an appropriate choice of the coefficients a_j and integers k_j. Low, Kliuchnikov, and Wiebe exploit this idea in [LKW19] by constructing linear combinations of second-order product formulas that achieve convergence order $2m$ for any integer m, given by $U_{\mathrm{MP}}(\Delta) = \sum_{j=1}^{m} a_j U_2(\Delta/k_j)^{k_j} = e^{-iH\Delta} + \mathcal{O}(\Delta^{2m+1})$. This observation forms the starting point of our analysis of multi-product formulas. See Chapter 3 for further details on the time-independent multi-product formulas.

Our Contribution

Although [LKW19] proposed the use of multi-product formulas, their analysis does not provide a rigorous complexity analysis for implementation on a quantum computer. See Section 3.2

for a more detailed discussion. Our first contribution in Part I is a complete complexity analysis of the multi-product formula. The relevant results are given in Theorem 3.3.5, Corollary 3.3.6, Corollary 3.5.1, and Corollary 3.5.2. We next present a simplified version of Corollary 3.3.6 and briefly motivate its significance.

Result 1.3.3 (Informal version of Corollary 3.3.6)**.** Consider using MPF with $p = 2$ for simulating the Hamiltonian $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$. Then the number of queries to the base product formula is $\mathcal{O}\left(\mu T \text{poly}\log\left(\frac{\mu T}{\epsilon}\right)\right)$. Here T is the evolution time, ϵ is the tolerated error, and μ is a parameter with commutator scaling defined as

$$\mu = 2 \sup_{\substack{(j,l)\in 2\mathbb{Z}^+\times\mathbb{Z}^+: \\ j/2\geq l\geq 1}} \sup_{\substack{j_1,\cdots,j_l\in 2\mathbb{Z}^+: \\ j_1+\cdots+j_l=j}} \left(\prod_{\kappa=1}^{l} \alpha_{\text{comm},j_\kappa+1}\right)^{\frac{1}{j+l}}, \tag{1.9}$$

where $\alpha_{\text{comm},j}$ is defined as $\alpha_{\text{comm},j} = \sum_{\gamma_1,\cdots,\gamma_j=1}^{\Gamma} \|[H_{\gamma_1}, [H_{\gamma_2}, \cdots [H_{\gamma_j-1}, H_{\gamma_j}]]]\|$.

We briefly emphasize the significance of Result 1.3.3. A p-th order product formula has gate complexity [Chi+21] $\mathcal{O}\left(\frac{\alpha_{\text{comm},p}^{1/p} T^{1+1/p}}{\epsilon^{1/p}}\right)$. In contrast, quantum simulation algorithms based on quantum signal processing [LC17a] and quantum signal transformation [Gil+19b] achieve complexity $\mathcal{O}\left(\|H\| T + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right)$. This scaling in both T and ϵ improves on fixed-order product formulas and is asymptotically optimal [Ber+14]. Result 1.3.3 recovers, for the multi-product formula (MPF), the nearly linear dependence on the evolution time T and polylogarithmic dependence on the target precision ϵ originally shown in [LKW19]. Its main improvement is an $\mathcal{O}(\mu)$ query complexity, where μ depends only on commutators among the constituent Hamiltonians H_γ. In many practical settings, this commutator-based parameter is substantially smaller than $\mathcal{O}(\sum_\gamma \|H_\gamma\|)$, yielding a concrete advantage. Thus, Result 1.3.3 achieves nearly

optimal scaling in both T and ϵ while retaining the *commutator scaling* characteristic of product formulas. See Chapter 3 for details.

1.3.2 Time-Dependent Quantum Simulation via Multi-Product Formulas

The primary objective of the time-dependent quantum simulation problem for linear non-homogeneous differential equations in Equation (1.6) is to design quantum algorithms that implement $\exp_{\mathcal{T}}(-\int_0^t A(s)\,ds)$. Methods developed for time-dependent Hamiltonian simulation algorithms do not apply in this setting when $A(t) \in \mathbb{C}^{N\times N}$ may be non-Hermitian and $b(t) \neq 0$. However, a recent technique called Linear Combination of Hamiltonian Simulation (LCHS) [ALL23; ACL26; LS25] expresses the solution of a dissipative linear equation in Equation (1.6) as an integral over unitary evolutions, which can then be simulated using Hamiltonian simulation algorithms. We study this problem in the context of Multi-Product Formula (MPF) based quantum algorithms and the LCHS algorithm. We first briefly discuss the time-dependent MPF and the Linear Combination of Hamiltonian Simulation algorithms, and then describe our contributions.

1.3.2.1 Time-Dependent Product & Multi-Product Formulas

Consider a time-dependent Hamiltonian $H(t)$ for $t \in [0,T]$, decomposed as $H(t) = \sum_{\gamma=1}^{\Gamma} H_\gamma(t)$. For each γ, let $U_\gamma(T)$ denote the time-ordered exponential generated by $H_\gamma(t)$ on $[0,T]$. A generic time-dependent product formula approximates the exact time-evolution operator $U(T)$ as $U_p(T) = \prod_{\xi=1}^{\Xi}\prod_{\gamma=1}^{\Gamma} U_{\pi_\xi(\gamma)}(\beta_{\xi\gamma}T, (\beta_{\xi\gamma}+\alpha_{\xi\gamma})T)$, where π_ξ is a permutation in S_Γ and $\alpha_{\xi,\gamma}, \beta_{\xi,\gamma} \in [0,1]$ such that $\beta_{\xi\gamma}+\alpha_{\xi\gamma} \in [0,1]$. As before, we say that U_p is a pth-order

product formula if its single-step approximation error satisfies $\|U_p(T) - U(T)\| = \mathcal{O}(T^{p+1})$. In particular, the first-order and second-order time-dependent product formulas are given as

$$U_1(0,T) = \overleftarrow{\prod_{\gamma=1,\cdots,\Gamma}} U_\gamma(0,T), \quad U_2(0,T) = \overrightarrow{\prod_{\gamma=1,\cdots,\Gamma}} U_\gamma(T/2,T) \overleftarrow{\prod_{\gamma=1,\cdots,\Gamma}} U_\gamma(0,T/2). \tag{1.10}$$

In contrast to time-independent product formulas, the error scaling of time-dependent product formulas is less completely understood. Early work of Wiebe et al. [Wie+10] derived bounds for smoothly varying Hamiltonians based on 1-norm scaling. Subsequent studies [Ber+20; AFL21a; Ike+23] analyzed low-order time-dependent product formulas, some of which exhibit commutator scaling. A general treatment was later given by Childs et al. [CS19a], who established commutator scaling for smoothly varying Hamiltonians with finite-range interactions.

The MPF for time-dependent Hamiltonian simulation, constructed from the second-order time-dependent product formula, is given by

$$U_{\mathrm{MP}}(T) = \sum_{j=1}^{m} a_j \overleftarrow{\prod_{b=1,\cdots,b_j}} U_2\left(\frac{(b-1)T}{b_j}, \frac{bT}{b_j}\right). \tag{1.11}$$

An error analysis was recently given by Mizuta et al. [MIF24], who showed that time-dependent MPFs also exhibit commutator scaling. See Section 4.2 for further details on recent developments in MPFs that are not covered by the results in Chapter 3.

Linear Combination of Hamiltonian Simulation

An et al. first proposed the Linear Combination of Hamiltonian Simulation (LCHS) framework in [ALL23]. Their approach generalizes the Fourier representation of the two-sided exponential function

$$e^{-|x|} = \int_{\mathbb{R}} \frac{1}{\pi(1+k^2)} e^{-ikx} dk \tag{1.12}$$

to the matrix setting. Specifically, for a possibly time-dependent operator $A(t) = L(t) + iH(t)$, with Hermitian $L(t) \succeq 0$ and $H(t)$, they show that

$$\mathcal{T} e^{-\int_0^t A(s)\, ds} = \int_{\mathbb{R}} \frac{1}{\pi(1+k^2)} \mathcal{T} e^{-i\int_0^t (H(s)+kL(s))\, ds} dk, \quad t \in [0,T] \tag{1.13}$$

solves Equation (1.6) in the case $b(t) \equiv 0$. The case $b(t) \neq 0$ is then derived via an application of Duhamel's principle. The LCHS algorithm then proceeds by truncating the integral to $[-K, K]$ for some $K > 0$ and implementing it via the LCU algorithm, where each $U_k(t) := \mathcal{T} e^{-i\int_0^t (H(s)+kL(s))\, ds}$ is implemented via a time-dependent Hamiltonian simulation algorithm. See Section 4.3 for further details on recent developments in the LCHS algorithm.

Our Contribution

We briefly summarize our approach and main results in the time-independent homogeneous case. The non-homogeneous and time-dependent cases are discussed in Section 4.6. Let $A = L + iH$, where $L \succeq 0$ and L, H are Hermitian. The LCHS representation expresses the dissipative evolution e^{-AT} as an integral over unitary evolutions $e^{-AT} \approx \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(k) e^{-iG_k T}\, dk$. Here $k \in \mathbb{R}$ is the frequency variable, $\hat{f}$ is the kernel function and $G_k := H + kL$ is the Hamiltonian

simulated at frequency k. After applying a quadrature rule $Q = \{(k_i, w_i) : i \in \mathcal{I}_Q\}$ with nodes k_i and weights w_i, the integral is replaced by a finite sum of the form $W_Q^{\text{ideal}} = \sum_{i\in\mathcal{I}_Q} v_i e^{-iG_{k_i}T}$, where $v_i = \frac{w_i \hat{f}(k_i)}{\sqrt{2\pi}}$. Thus, the quadrature rule determines not only the discretization error but also the Hamiltonians that must be simulated. We replace each $e^{-iG_{k_i}T}$ with an MPF approximation. The post-quadrature error and complexity bounds are then determined by the quadrature radius $R_Q := \max_{i\in\mathcal{I}_Q} |k_i|$, which controls the range of Hamiltonians $H + kL$ appearing in the simulation, and by the LCU normalization factor $\alpha_Q := \sum_{i\in\mathcal{I}_Q} |v_i|$, which controls the success probability and amplitude-amplification overhead of the outer LCU procedure. Since LCHS is combined with Hamiltonian simulation via MPF, the cost also depends on a commutator scale μ_Q, which quantifies the MPF simulation difficulty of the quadrature-indexed Hamiltonian family $\{G_{k_i}\}_{i\in\mathcal{I}_Q}$. Our first result is a post-quadrature complexity estimate that makes this quadrature-dependent commutator scaling explicit.

Result 1.3.4 (Informal Version of Corollary 4.5.6)**.** Let $\epsilon > 0$. Suppose the kernel, quadrature rule, and MPF order are chosen so that the final implemented operator W_Q^{MPF} satisfies $\left\|W_Q^{\text{MPF}} - e^{-AT}\right\| \leq \epsilon$. The quadrature-indexed Hamiltonian family $\{G_{k_i}\}_{i\in\mathcal{I}_Q}$ admits a commutator-sensitive MPF bound with scale μ_Q, and choosing MPF order $m = O(\log(e + T/\epsilon))$ gives block-encoding complexity $\mathcal{O}\left(\mu_Q T F\left(\log(T/\epsilon)\right)\log^4(T/\epsilon)\right)$ and the normalized-state preparation complexity is $\mathcal{O}\left(\alpha_Q \frac{\|u(0)\|}{\|u(T)\|}\mu_Q T F\left(\log(T/\epsilon)\right)\log^4(T/\epsilon)\right)$. Here $F(m)$ records the order-dependence of the MPF.

The role of α_Q in Result 1.3.4 is important. In previous LCHS algorithms, the kernel and quadrature rule are typically fixed, so the corresponding LCU normalization is either bounded explicitly or absorbed into constants. In contrast, we keep α_Q explicit because the kernel and

quadrature choices also affect the commutator scale. Consequently, its effect must be evaluated jointly with α_Q and μ_Q. Result 1.3.4 is stated in a simplified informal form. More precise complexity estimates are given in Theorem 4.5.5 and Corollary 4.5.6. In Result 1.3.4, $F(m)$ and μ_Q are auxiliary functions that upper bound $\mu_{m,Q}$, which is defined in Equation (4.52). Here, we only note that $\mu_{m,Q}$ depends on the quantities $\Lambda_{m,Q} := \sum_{i\in\mathcal{I}_Q} |v_i|\Phi_m(k_i, N)$ and $\Phi^*_{m,Q} := \max_{i\in\mathcal{I}_Q} \Phi_m(k_i, N)$, where m denotes the MPF order and $\Phi_m(k, N)$ is an abstract MPF error profile for simulating G_k. Hence, Result 1.3.4 is abstract because it is phrased in terms of a general MPF error profile. In Sections 4.5.4 and 4.5.5, we specialize this result to two concrete settings:

The first setting is the general Hamiltonian setting discussed in Chapter 3. The analysis there associates to each pair (H, L) a commutator scale, denoted here by $\chi_J(H, L)$, which measures how rapidly the relevant nested commutators can grow. For a quadrature rule Q, this scale must be compared with the quadrature radius R_Q. Therefore, we choose ρ_Q satisfying $0 < \rho_Q < \frac{\chi_J(H,L)}{(J+1)^{1/J}\max\{1,R_Q\}}$. In this case, we have

$$\Phi_{m,\nu}(H, L) := \sum_{\substack{j\in 2\mathbb{Z}_+\\ j\geq 2m}} \sum_{l=1}^{m} \frac{\rho_Q^{j+l-(2m+1)}}{l!} \sum_{\substack{j_1,\ldots,j_l\in 2\mathbb{Z}_+\\ j_1+\cdots+j_l=j}} \sum_{\substack{\ell_1,\ldots,\ell_l\geq 0\\ 0\leq\ell_\kappa\leq j_\kappa+1\\ \ell_1+\cdots+\ell_l=\nu}} \prod_{\kappa=1}^{l} \mathcal{C}_{j_\kappa+1,\ell_\kappa}(H, L) \tag{1.14}$$

for $\nu \geq 0$. Here, $\mathcal{C}_{j_\kappa+1,\ell_\kappa}(H, L)$ is determined by nested commutators of length $j_\kappa + 1$ in H and L that contain exactly ℓ_κ occurrences of L. If the discrete moments are $M^Q_\nu := \sum_{i\in\mathcal{I}_Q} |v_i|\,|k_i|^\nu$ then $\Lambda_{m,Q} := \sum_{\nu\geq 0} \Phi_{m,\nu}(H, L)M^Q_\nu$. In this case, $F(m) \equiv 1$.

The second setting is the special class of q-local, g-extensive Hamiltonians, for which Mizuta [Miz26] proved an improved MPF estimate, as discussed in Chapter 3. We use this

Method	Query Complexity	Queries to $\lvert u(0)\rangle$
Original LCHS [ALL23]	$\widetilde{\mathcal{O}}\left(\left(\frac{\lVert u(0)\rVert}{\lVert u(T)\rVert}\right)^2 \alpha_A T/\epsilon\right)$	$\mathcal{O}\left(\frac{\lVert u(0)\rVert}{\lVert u(T)\rVert}\right)$
Improved LCHS [ACL26]	$\widetilde{\mathcal{O}}\left(\frac{\lVert u(0)\rVert}{\lVert u(T)\rVert}\alpha_A T\left(\log\left(\frac{1}{\epsilon}\right)\right)^{1+1/\beta}\right)$	$\mathcal{O}\left(\frac{\lVert u(0)\rVert}{\lVert u(T)\rVert}\right)$
Optimal LCHS [LS25]	$\mathcal{O}\left(\frac{\lVert u(0)\rVert}{\lVert u(T)\rVert}\alpha_A T\log\left(\frac{1}{\epsilon}\right)\right)$	$\mathcal{O}\left(\frac{\lVert u(0)\rVert}{\lVert u(T)\rVert}\right)$
LCHS–MPF (general case)	$\mathcal{O}\left(\frac{\alpha_Q\lVert u(0)\rVert}{\lVert u(T)\rVert}\mu_Q^{\mathrm{gen}} T\log^4\left(\frac{T}{\epsilon}\right)\right)$	$\mathcal{O}\left(\alpha_Q\frac{\lVert u(0)\rVert}{\lVert u(T)\rVert}\right)$
LCHS–MPF (local/extensive case)	$\mathcal{O}\left(\alpha_Q\frac{\lVert u(0)\rVert}{\lVert u(T)\rVert}\mu_Q^{\mathrm{loc}} T\log^6\left(\frac{T}{\epsilon}\right)\right)$	$\mathcal{O}\left(\alpha_Q\frac{\lVert u(0)\rVert}{\lVert u(T)\rVert}\right)$

Table 1.1: Comparison of representative quantum algorithms for time-independent homogeneous linear differential equations. Here, α_A denotes a norm or block-encoding normalization of A and μ_Q^{gen} and μ_Q^{loc} are the quadrature-dependent commutator scales appearing in the general and local/extensive MPF estimates, respectively.

estimate when it applies to $\{G_{k_i}\}_{i\in\mathcal{I}_Q}$. For $p_0 \in \mathbb{N}$, define

$$\mu_{m,p_0}(G_{k_i}) := \sup_{\substack{j,l\in\mathbb{N}\\ j\geq 2m,\ 1\leq l\leq \lfloor j/2\rfloor}} \left(\sum_{\substack{2\leq j_1,\ldots,j_l\leq p_0-1\\ j_1+\cdots+j_l=j}} \prod_{\kappa=1}^{l} \alpha_{\mathrm{comm},j_\kappa+1}(G_{k_i})\right)^{\frac{1}{j+l}}, \tag{1.15}$$

This locality-based estimate involves the quantities $\Lambda_{m,Q}(p_0) := \sum_{i\in\mathcal{I}_Q} |v_i|\,\mu_{m,p_0}(G_{k_i})^{2m+1}$ and $\Phi^*_{m,Q}(p_0) := \max_{i\in\mathcal{I}_Q}\mu_{m,p_0}(G_{k_i})^{2m+1}$. In this case, $F(m) = \mathcal{O}(\log^2 m)$.

Table 1.1 compares representative algorithms in the time-independent homogeneous setting. Compared with optimal LCHS, our algorithm replaces the norm-based Hamiltonian simulation cost with a commutator-sensitive MPF cost after quadrature. Thus, the method is useful when the family $H + kL$ has favorable commutator structure. The complexity also depends on quadrature-induced quantities such as R_Q and α_Q. Hence, the method is not a black-box improvement over LCHS. Rather, it is a structure-sensitive refinement. The tradeoff is that the dependence on T and $1/\epsilon$ is not linear and logarithmic, respectively, as in optimal LCHS.

Our second result concerns the choice of quadrature rule. Because the quadrature rule determines the nodes k_i and weights v_i, it affects both the discretization error and the MPF

cost through the quantities R_Q, α_Q, the discrete moments M_ν^Q, and the resulting weighted MPF profiles. We therefore compare quadrature rules in terms of these quantities, together with the quadrature error $E_{\mathrm{quad}}(Q)$ and the node count $|\mathcal{I}_Q|$.

Result 1.3.5 (Informal Version of Corollary 4.5.14 and Corollary 4.5.17)**.** For the uniform trapezoidal rule, if the mesh and tail-error budgets are balanced as $\epsilon_{\mathrm{mesh}} \asymp \epsilon_{\mathrm{tail}} \asymp \epsilon$, then the rule can be chosen so that $E_{\mathrm{quad}}(Q^{\mathrm{trap}}) = \mathcal{O}(\epsilon)$ with $|\mathcal{I}_{Q^{\mathrm{trap}}}| = \mathcal{O}\big(\log^2\big(\frac{1}{\epsilon}\big)\big)$ and $R_{Q^{\mathrm{trap}}} = \mathcal{O}\big(\log\big(\frac{1}{\epsilon}\big)\big)$. For the sinh–sinh rule, the rule can be chosen so that $E_{\mathrm{quad}}\big(Q^{\mathrm{sinh}}\big) = \mathcal{O}(\epsilon)$ with $\big|\mathcal{I}_{Q^{\mathrm{sinh}}}\big| = \mathcal{O}\big(\log\big(\frac{1}{\epsilon}\big)\log\log\big(\frac{1}{\epsilon}\big)\big)$ Both rules have the same leading quadrature-radius scaling and the same leading discrete-moment scaling. In particular, for the four-parameter (a, b, c, d) kernel family used in Equation (4.10), the LCU normalization satisfies

$$\alpha_Q = \begin{cases} \mathcal{O}(1), & a > 1, \\ \mathcal{O}(\log\log(1/\epsilon)), & a = 1, \end{cases} \tag{1.16}$$

provided that $a, b, d = \mathcal{O}(1)$ and $c = \Omega(\sqrt{\log(1/\epsilon)})$.

Thus, the sinh–sinh quadrature rule improves the quadrature-cardinality scaling due to its double-exponential decay rate. This reduction in the number of quadrature nodes directly improves the LCU implementation cost of the algorithm. We further demonstrate our analysis through three concrete classes of applications. See Chapter 4 for more details.

1.4 Part II: Chebyshev-Based Quantum Computation

Chebyshev polynomials play a central role in classical numerical analysis and approximation theory; see, for example, [Tre19; Riv20]. Chebyshev expansions provide a unifying framework for a broad class of algorithms, including spectral methods for partial differential equations [Boy01], polynomial chaos expansions [SZ19a], and fast transforms based on low-rank structure [RG17]. These classical developments motivate the central question addressed in Part II: to what extent can the approximation-theoretic advantages of Chebyshev polynomials be transferred to the quantum setting? Recent progress in quantum algorithms has revealed a close compatibility between Chebyshev polynomials and quantum circuit models. In particular, quantum signal processing [LC17b] and quantum singular value transformation [Gil+19a] provide mechanisms for implementing polynomial transformations of operators, with Chebyshev polynomials arising naturally as the associated basis functions. This suggests that Chebyshev expansions are not only useful from a classical perspective, but also well matched to the structure of quantum computation. Against this background, Part II develops a Chebyshev-based perspective on quantum computation for scientific and numerical problems. We use Chebyshev polynomial approximation as an organizing principle to study quantum algorithms for function approximation, high-dimensional parametric models, and spectral transforms. This viewpoint allows classical approximation theory to inform quantum circuit design systematically.

1.4.1 Chebyshev-Based Function Approximation with Quantum Circuits

The approximation of multivariate functions is a fundamental problem in numerical analysis. Given a function on a high-dimensional domain, the objective is to construct an approxima-

tion from a prescribed function class, typically polynomials or trigonometric series, such that the approximation error decays rapidly as the number of degrees of freedom increases. An analogous problem arises on a quantum computer: one seeks quantum circuits whose output encodes an accurate approximation of a target function, while retaining explicit control over circuit depth and gate complexity. We design quantum circuits for approximation in two function spaces.

Korobov Functions

Sobolev spaces provide a standard analytical framework for function approximation. Among them, Korobov spaces [Kor59; Smo63] play a distinguished role in high-dimensional settings, since they model mixed smoothness and anisotropy and arise naturally in approximation and numerical integration.

Definition 1.4.1. Let $2 \leq p \leq \infty$. The Korobov function space is defined as follows:

$$X^{2,p}([0,1]^d) = \left\{ f \in L^p([0,1]^d) : f|_{\partial[0,1]^d} = 0,\ D^{\vec{k}} f \in L^p([0,1]^d),\ \|\vec{k}\|_\infty \leq 2 \right\}. \tag{1.17}$$

Note that $X^{2,p}([0,1]^d)$ is a subspace of the Sobolev space $H^p([0,1]^d)$. The distinction lies in the multi-index conditions appearing in their definitions. Specifically, the Korobov space is characterized by the condition $\|\vec{k}\|_\infty \leq 2$, whereas the Sobolev space is characterized by the condition $\|\vec{k}\|_1 \leq 2$.

Example 1.4.2. In two dimensions ($d = 2$), a function f belongs to $X^{2,p}([0,1]^2)$ if and only if

$$\frac{\partial f}{\partial x_1},\quad \frac{\partial f}{\partial x_2},\quad \frac{\partial^2 f}{\partial x_1^2},\quad \frac{\partial^2 f}{\partial x_2^2},\quad \frac{\partial^2 f}{\partial x_1 \partial x_2},\quad \frac{\partial^3 f}{\partial x_1^2 \partial x_2},\quad \frac{\partial^3 f}{\partial x_1 \partial x_2^2},\quad \frac{\partial^4 f}{\partial x_1^2 \partial x_2^2} \in L^p([0,1]^2). \tag{1.18}$$

On the other hand, a function f belongs to $H^2([0,1]^2)$ if and only if

$$\frac{\partial f}{\partial x_1}, \quad \frac{\partial f}{\partial x_2}, \quad \frac{\partial^2 f}{\partial x_1^2}, \quad \frac{\partial^2 f}{\partial x_2^2}, \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} \in L^p([0,1]^2). \tag{1.19}$$

Hence, $X^{2,p}([0,1]^2) \subseteq H^2([0,1]^2)$.

Functions in Korobov spaces admit polynomial and trigonometric approximations whose convergence rates depend on structured smoothness rather than on the ambient dimension. This makes them a canonical model class for studying tractability and computational complexity in high-dimensional numerical problems. In particular, Korobov functions can be efficiently approximated using sparse grid decompositions [BG04]. We use this classical construction as the starting point for approximating Korobov functions on a quantum computer.

Countably Parametric Holomorphic Maps

Parametric models with countably many input variables arise naturally in uncertainty quantification, inverse problems, and the analysis of parametric partial differential equations, where system coefficients or forcing terms depend on infinitely many parameters. A standard analytical framework for such problems is holomorphic dependence on the parameters, which implies strong regularity and enables efficient approximation schemes; see, for example, [CDS10; CDS11; CCS13; CCS15; Zec18].

Example 1.4.3 (Response Surface of Parametric Elliptic PDEs)**.** We present a typical example of a countably parametric holomorphic map. Let $\Omega \subseteq \mathbb{R}^n$ be an open, bounded Lipschitz domain.

Given sufficiently regular functions f and $a : \Omega \to \mathbb{R}$, consider the diffusion equation

$$\begin{cases} \nabla_x \cdot \big(a(x)\nabla_x u\big) = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{1.20}$$

It is well known that, if a is continuous and bounded on $\overline{\Omega}$ and satisfies $a(x) \geq a_- > 0$ for $x \in \Omega$, then Equation (1.20) admits a unique weak solution $u \in H_0^1(\Omega)$. Assume that f and Ω are fixed. We are interested in the *solution map* $S : a \mapsto u$ on a suitable set of admissible coefficients a. To define such a set, introduce affine-parametric diffusion coefficients by

$$a(x; \vec{y}) := a_0(x) + \sum_{j \geq 1} y_j \psi_j(x), \tag{1.21}$$

where $|y_j| \leq 1$, $a_0, \psi_j \in C(\overline{\Omega})$, and $a_0(x) \geq a_{0,-} > 0$ for $x \in \Omega$. We further assume that there exists $\kappa \in (0,1)$ such that $\|\vec{b}\|_{\ell^1(\mathbb{N})} \leq \kappa a_{0,-}$, where $\vec{b} = (b_j)_{j \geq 1}$ and $b_j := \|\psi_j\|_{L^\infty(\Omega)}$. Typical examples of expansions of the form Equation (1.21) include Karhunen–Loève expansions of random field inputs, Fourier expansions, and wavelet expansions. Under these assumptions, standard PDE arguments imply that, for fixed $f \in H^{-1}(\Omega) = (H_0^1(\Omega))^*$ and every $\vec{y} \in \mathcal{U} := [-1,1]^{\mathbb{N}}$, Equation (1.20) admits a unique solution $u(\cdot, \vec{y}) \in H_0^1(\Omega)$. The parametric solution map $u : \mathcal{U} \to H_0^1(\Omega)$ is holomorphic with respect to the parameters $\vec{y}$ [CCS15]. Let $G \in H^{-1}(\Omega)$. Then $F = G \circ u$, that is, $F(\vec{y}) = G(u(\cdot, \vec{y}))$, is a countably parametric holomorphic map, also called the *response surface* of Equation (1.20) for the parametric input Equation (1.21) and quantity of interest G. Response maps of this type, with quantified holomorphic parameter dependence, arise for a wide range of elliptic and parabolic PDE models with uncertain inputs

in engineering and the sciences.

Countably parametric holomorphic maps admit convergent expansions in generalized polynomial chaos (gPC) bases, including Chebyshev polynomial systems. Under suitable assumptions, such as $(\vec{b}, \varepsilon)$-holomorphy with summable parameter weights, the associated gPC coefficient sequences belong to $\ell^p(\mathscr{F})$ for some $0 < p < 1$. This reflects a sparsity structure that is independent of the nominal parameter dimension and forms the basis for dimension-robust approximation rates obtained through best n-term truncations of the corresponding polynomial expansions. We use this observation as the starting point for approximating countably parametric holomorphic maps on a quantum computer.

Our Contribution

We establish the existence of quantum circuits for approximating high-dimensional and infinite-dimensional functions. The constructions rely on Chebyshev polynomial expansions implemented through quantum signal processing and linear-combination-of-unitaries algorithms; see Chapter 5 for details. Our first contribution in Part II establishes the existence of quantum circuits that approximate functions in $X^{2,p}([0,1]^d)$ and countably parameterized holomorphic maps. The result for $X^{2,p}([0,1]^d)$, with $p \in \{2, \infty\}$, is summarized below.

Result 1.4.4 (Informal version of Proposition 5.3.6)**.** Let $\epsilon \in (0,1)$ and $p \in \{2, \infty\}$. For $f \in X^{2,p}([0,1]^d)$, there exists a function, $g \in X^{2,p}([0,1]^d)$, defined on $[0,1]^d$ such that $\|f - g\|_{L^p([0,1]^d)} \leq \epsilon$. Moreover, there exists a quantum circuit, $U_{f,\varepsilon}$, that outputs g such that $U_{f,\varepsilon}$ has depth at most $\mathcal{O}(d\epsilon^{-(\frac{1}{2}+\frac{1}{d})}(2\log_2^{3/2}(\frac{1}{\epsilon}))^d)$ and the width at most $\mathcal{O}(2d + \epsilon^{-\frac{1}{d}}\log_2^{\frac{3}{2}}(\frac{1}{\epsilon}))$.

The estimates above are weaker than those in Proposition 5.3.6. These simplified bounds

follow from the sharper estimates by applying the inequality $W(x) \leq x$ for $x \geq 0$, where W denotes Lambert's W function. See Chapter 5 for details. We then extend the result to the case $2 < p < \infty$.

Result 1.4.5 (Informal version of Corollary 5.3.8)**.** Let $\epsilon \in (0,1)$ and $2 \leq p \leq \infty$. For $f \in X^{2,p}([0,1]^d)$, there exists a function, $g \in X^{2,p}([0,1]^d)$, defined on $[0,1]^d$ such that $\|f - g\|_{L^p([0,1]^d)} \leq \epsilon$. Moreover, there exists a quantum circuit, $U_{f,\varepsilon}$, that outputs g such that $U_{f,\varepsilon}$ has depth at most $\mathcal{O}\left(d(12\beta\log_2\beta)^{\alpha+\beta}\alpha^{\alpha+\beta}\epsilon^{-\frac{p}{2p-1}\left(1+\frac{1}{d}\right)}\log_2^{\alpha+\beta}\left(\frac{1}{\epsilon}\right)\epsilon^{-\frac{p}{d(2p-1)}}\right)$ and width most

$$\mathcal{O}\left(2d + (6\beta\log_2\beta)^{\alpha}\alpha^{\alpha}\epsilon^{-\frac{p}{d(2p-1)}}\log_2^{\alpha}\left(\frac{1}{\epsilon}\right)\right). \tag{1.22}$$

Here we have defined $\alpha = (3p-1)/(2p-1)$ and $\beta = \alpha(d-1)$.

We similarly derive approximation rates for countably parametric holomorphic maps.

Result 1.4.6 (Informal Version of Theorem 5.4.8 and Theorem 5.4.9)**.** Let $1 \leq d \leq \infty$ and $\mathscr{U} = [-1,1]^d$ and $f : \mathscr{U} \to \mathbb{R}$ in be an at most countably parametric holomorphic map.

1. If $d < \infty$, there are constants $C, \gamma > 0$ (depending on f and d) such that, for every $n \in \mathbb{N}$, there exists a quantum circuit, U_n, $\|f - \langle 0| U_n^\dagger \sigma_Z^{(1)} U_n |0\rangle\|_{L^\infty(\mathscr{U})} \leq C(f)e^{-\gamma n^{1/d}}$. Moreover, the quantum circuit has width $\mathcal{O}(d + \log_2(n))$ and depth $\mathcal{O}((n^{1/d}+n)d\log_2(n))$.

2. If $d = \infty$ and $\sup_{\vec{y}\in[-1,1]^{\mathbb{N}}}|f(\vec{y})| \leq 1$, there is a constant C (depending on f) such that, for every $n \in \mathbb{N}$, there exists a quantum circuit, U_n, such that bound $\|f - \langle 0| U_n^\dagger \sigma_Z^{(1)} U_n |0\rangle\|_{L^\infty(\mathscr{U})} \leq C(f)n^{-1/p+1}$. hold. Moreover, the quantum circuit has width $\mathcal{O}(\log_2 n)$ and depth $\mathcal{O}(n\log_2^2 n)$.

We briefly place these results in context. Our constructions extend those of [Yu+24] in two main respects. First, we construct *unparameterized* quantum circuits by exploiting the fact that

Chebyshev polynomials can be implemented using a predetermined set of parameters in quantum signal processing. Thus, function approximation via Chebyshev polynomials is a natural strategy on a quantum computer. Second, our examples include functions from Korobov spaces and countably parametric holomorphic maps, whereas [Yu+24] focuses primarily on function spaces defined by classical notions of continuity and smoothness on finite-dimensional domains.

1.4.2 Non-Uniform Quantum Fourier Transform

Fourier transforms are central computational primitives in numerical analysis and scientific computing, and their quantum analogue, the quantum Fourier transform (QFT), plays a foundational role in quantum algorithms. Prominent examples include Shor's factoring algorithm [Sho94] and Kitaev's quantum phase estimation algorithm [Kit95]. The efficiency and structural simplicity of the QFT make it particularly well suited to problems defined on uniformly spaced computational grids. Many practical applications, however, involve data sampled on non-uniform grids, for which the standard QFT is not directly applicable. This limitation motivates the development of a non-uniform quantum Fourier transform (NUQFT) that can accommodate irregular sampling while preserving quantum advantage.

Motivated by this connection, we extend the Chebyshev-based framework developed in the preceding sections to Fourier transforms on non-uniform grids. We present a quantum algorithm for the NUQFT that uses low-rank matrix factorizations induced by truncated Chebyshev expansions. This approach addresses the limitations of the standard QFT on irregular domains and illustrates how classical techniques based on Chebyshev polynomials can be systematically transferred to quantum spectral algorithms.

Non-Uniform Discrete Fourier Transform

Let $\{x_j\}_{j=0}^{N-1} \subseteq \mathbb{C}$ denote samples taken at non-uniform locations $\{t_j\}_{j=0}^{N-1} \subseteq \mathbb{T}$, viewed as a discrete representation of a function $f : \mathbb{T} \to \mathbb{C}$. The non-uniform discrete Fourier transform (NUDFT) of $\{x_j\}_{j=0}^{N-1}$, evaluated at non-uniform frequencies $\{\omega_k\}_{k=0}^{N-1} \subseteq \mathbb{R}$, is defined by

$$\vec{X} := \begin{pmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{pmatrix} = \begin{pmatrix} e^{-i\omega_0 t_0} & e^{-i\omega_0 t_1} & \cdots & e^{-i\omega_0 t_{N-1}} \\ e^{-i\omega_1 t_0} & e^{-i\omega_1 t_1} & \cdots & e^{-i\omega_1 t_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-i\omega_{N-1} t_0} & e^{-i\omega_{N-1} t_1} & \cdots & e^{-i\omega_{N-1} t_{N-1}} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix} := \overline{F}\vec{x}. \tag{1.23}$$

In the literature, Equation (1.23) is known as the Type–III NUDFT, and the associated matrix $\overline{F}$ is denoted by F_{III}. Two important special cases arise depending on whether the sampling locations and frequency points are uniform or non-uniform:

(I) **Type–I NUDFT**. The sequence $\{t_j\}_{j=0}^{N-1}$ is sampled uniformly at points $t_j = 2\pi j/N$, while the sequence $\{\omega_k\}_{k=0}^{N-1}$ may be sampled at non-uniform points. We denote the corresponding matrix $\overline{F}$ by F_{I}.

(II) **Type–II NUDFT**. The sequence $\{t_j\}_{j=0}^{N-1}$ may be sampled at non-uniform points, while the sequence $\{\omega_k\}_{k=0}^{N-1}$ is sampled uniformly with $\omega_k = k$ for $k = 0, \ldots, N-1$. We denote the corresponding matrix $\overline{F}$ by F_{II}.

We argue in Section 6.2 that it suffices to develop a quantum algorithm for the Type–II case, since the Type–I and Type–III cases can be reduced to it.

Our Contribution

Let $N = 2^n$ with $n \geq 1$. Antolín and Townsend [RT18] approximate the Type–II NUDFT matrix F_{II} by $F_{\mathrm{II}} \approx \sum_{r=0}^{K-1} D_{\vec{u}_r} F D_{\vec{v}_r}$, where F denotes the standard DFT matrix and $D_{\vec{u}_r}$ and $D_{\vec{v}_r}$ are diagonal matrices. See Section 6.2 for the general form required. Assuming oracle access to the sampling points $\{t_j\}_{j=0}^{2^n-1}$, we construct quantum circuits $U_{\vec{u}_r}$ and $U_{\vec{v}_r}$ that provide block encodings of $D_{\vec{u}_r}$ and $D_{\vec{v}_r}$. Combining these circuits with the standard block encoding of F and the linear-combination-of-unitaries (LCU) technique yields an implementation of $\sum_{r=0}^{K-1} D_{\vec{u}_r} F D_{\vec{v}_r}$. Our second contribution in Part II is explicit gate-count estimates sufficient to obtain an ϵ-accurate block encoding.

Result 1.4.7 (Asymptotic Version of Theorem 6.3.16 based on Corollary 6.4.5)**.** Let $\epsilon > 0$ and let $n \in \mathbb{N}$ with $N = 2^n$. There exists a quantum circuit, V_{II}, that implements an ϵ-accurate block-encoding of F_{II}. Let

$$L_{n,\epsilon} := n + \log(1/\epsilon), \quad \kappa := \max_j (1 + (y_j^*)^2)^{-1/2}, \tag{1.24}$$

where $y_j^* \in (-1, 1)$ are points determined by the non-uniform sampling points. The circuit uses $\mathcal{O}(L_{n,\epsilon} + \log(1 + \kappa L_{n,\epsilon}))$ qubits, $\mathcal{O}(n + \log L_{n,\epsilon})$ Hadamard gates, $\mathcal{O}(L_{n,\epsilon}^2 + \log(1 + \kappa L_{n,\epsilon}))$ CNOT gates, $\mathcal{O}(L_{n,\epsilon} + \log(1 + \kappa L_{n,\epsilon}))$ Toffoli and σ_X gates, and $\mathcal{O}(n^2 + L_{n,\epsilon}^2 + \log(1 + \kappa L_{n,\epsilon}))$ single-qubit controlled rotation gates. The circuit depth is $\mathcal{O}(L_{n,\epsilon} \log L_{n,\epsilon} + \log(1 + \kappa L_{n,\epsilon}))$.

The resource bounds in Result 1.4.7 depend on three parameters: the number of qubits n, the target accuracy ϵ, and the geometry-dependent quantity κ. The accuracy enters only through the logarithmic factor $\log(1/\epsilon)$, appearing additively with n in the composite parameter $L_{n,\epsilon}$.

Thus, increasing precision affects gate counts and circuit depth only polylogarithmically. The quadratic dependence on n arises from the quantum Fourier transform. All remaining gate counts scale as $L_{n,\epsilon} \log L_{n,\epsilon}$, except for a single $L_{n,\epsilon}^2$ term, up to additional logarithmic factors in κ. The conditioning parameter κ contributes only through terms of the form $\log(1 + \kappa L_{n,\epsilon})$. In particular, no resource scales polynomially in κ, and its contribution remains subdominant provided κ grows at most exponentially in n or $\log(1/\epsilon)$.

Furthermore, we provide explicit non-asymptotic gate-count estimates in Theorem 6.3.16, bounds for obtaining an ϵ-accurate block encoding in Theorem 6.4.4, and simplified complexity estimates in Corollary 6.4.5. These results are discussed in Chapter 6.

Chapter 2: Quantum Computation

Quantum computing is a computational paradigm that uses principles of quantum mechanics to perform tasks in ways that differ fundamentally from classical computation. This chapter provides the quantum-computing background needed for the results of this thesis. Section 2.1 explains the role of Lie theory in this background, and Section 2.2 introduces the essential notions of Lie groups and Lie algebras. Section 2.3 then presents the Lie–Trotter product formula and the Baker–Campbell–Hausdorff formula, both of which are used extensively in Part I. Section 2.4 gives an axiomatic formulation of finite-dimensional quantum mechanics in the language of Lie theory. Section 2.5 introduces the gate-based model of quantum computation, including single-qubit systems, quantum gates, multi-qubit systems, and multi-qubit gates. Finally, Section 2.6 discusses several quantum algorithms that play a central role in this thesis. For a more comprehensive introduction to quantum computing, see [NC02; Sch19].

2.1 Why Lie Theory?

Lie theory provides a fundamental framework for studying symmetry and has applications throughout mathematics, physics, and computer science. Although finite-dimensional complex linear algebra is often sufficient for describing finite-dimensional quantum systems, several important connections between Lie theory and quantum computing make the Lie-theoretic per-

spective useful. In particular, Lie-theoretic techniques have been used in the design and analysis of quantum algorithms; see, for example, [Rag+22; Kök+22; Rag+24; JN26]. Moreover, the algorithms studied in Chapter 3 and Chapter 4 rely critically on Lie-theoretic constructions. We therefore begin this chapter with a concise introduction to the relevant background. For more comprehensive treatments of Lie theory, see [Hal15; BT13; Sim96].

2.2 Lie Groups and Lie Algebras

This section introduces the basic notions of Lie groups and Lie algebras. To keep the discussion concrete, we focus on matrix Lie groups and matrix Lie algebras. This approach requires only linear algebra, group theory, and analysis, which we use freely throughout the chapter. We also include occasional remarks on the smooth-manifold perspective for readers who want a broader view of Lie theory.

2.2.1 Lie Groups

A Lie group is a group endowed with a smooth-manifold structure such that multiplication and inversion are smooth maps. Lie groups describe *continuous symmetries* and play a fundamental role in mathematics and theoretical physics. Although the general definition relies on smooth manifolds, we focus here on the more concrete setting of matrix Lie groups: Lie groups realized as closed subgroups of $\mathsf{GL}(n,\mathbb{C})$ or $\mathsf{GL}(n,\mathbb{R})$, the groups of invertible matrices over $\mathbb{C}$ and $\mathbb{R}$, respectively. Throughout this section, $\mathbb{K}$ denotes either $\mathbb{R}$ or $\mathbb{C}$.

Definition 2.2.1. A matrix Lie group is a closed subgroup G of $\mathsf{GL}(n,\mathbb{C})$. A Lie group homomorphism $\Phi : G \to H$ is a smooth map between matrix Lie groups that is also a group

homomorphism. That is, $\Phi(AB) = \Phi(A)\Phi(B)$ for all $A, B \in G$.

The closedness condition is primarily a technical requirement since most examples considered in this exposition satisfy it automatically. Key examples of matrix Lie groups include the following:

$$\mathsf{U}(n) = \{A \in \mathsf{GL}(n, \mathbb{C}) : A^*A = I_n\}, \tag{2.1}$$

$$\mathsf{SU}(n) = \{A \in \mathsf{U}(n) \mid \det(A) = 1\}, \tag{2.2}$$

$$\mathsf{SL}(n, \mathbb{R}) = \{A \in \mathsf{GL}(n, \mathbb{R}) : \det(A) = 1\}, \tag{2.3}$$

$$\mathsf{SO}(n) = \{A \in \mathsf{GL}(n, \mathbb{R}) : A^TA = I_n\}. \tag{2.4}$$

Example 2.2.2. $\mathsf{SU}(2)$ is a fundamental example of a matrix Lie group with direct relevance to quantum computing. If $A \in \mathsf{SU}(2)$, then by definition $A^*A = I_2$ and $\det(A) = 1$, so the columns of A form an orthonormal basis in $\mathbb{C}^2$. It follows that every element $A \in \mathsf{SU}(2)$ can be written in the form

$$A_{\alpha,\beta} = \begin{pmatrix} \alpha & -\overline{\beta} \\ \beta & \overline{\alpha} \end{pmatrix}, \quad \text{with} \quad |\alpha|^2 + |\beta|^2 = 1 \quad \alpha, \beta \in \mathbb{C}. \tag{2.5}$$

It can be checked that $\mathsf{SU}(2)$ is compact, connected, and diffeomorphic to $\mathbb{S}^3$ (see, e.g., [Hal15]). Moreover, $\mathsf{SU}(2)$ and $\mathsf{U}(2)$ are related by the following short exact sequence[1]:

$$1 \longrightarrow \mathsf{SU}(2) \to \mathsf{U}(2) \xrightarrow{\det} \mathsf{U}(1) \longrightarrow 1. \tag{2.6}$$

The map $s : \mathsf{U}(1) \longrightarrow \mathsf{U}(2)$ defined by $s(z) = \mathrm{diag}(z, 1)$ is a group homomorphism satisfying

[1] A short exact sequence is a sequence of group homomorphisms $1 \to A \xrightarrow{f} B \xrightarrow{g} C \to 1$ that is exact at each term, meaning f is injective, g is surjective, and $\mathrm{im}(f) = \ker(g)$.

$(\det \circ s) = \mathrm{Id}_{\mathsf{U}(1)}$. Hence, the short exact sequence in Equation (2.6) splits. That is, we have $\mathsf{U}(2) \cong \mathsf{SU}(2) \ltimes \mathsf{U}(1)$, where $\ltimes$ denotes the semidirect product of groups[2].

Remark 2.2.3. We briefly illustrate how a matrix Lie group can be viewed as a smooth manifold. This geometric perspective motivates several constructions that follow. Consider $G = \mathsf{SL}(2, \mathbb{R})$. Note that

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathsf{SL}(2, \mathbb{R}) \quad \iff \quad ad - bc = 1. \tag{2.7}$$

Consider the smooth function $f : \mathbb{R}^4 \to \mathbb{R}$ defined by $f(a, b, c, d) = ad - bc$. The level set $f^{-1}(1)$ coincides with the matrix Lie group $\mathsf{SL}(2, \mathbb{R})$. The partial derivatives of f with respect to a, b, c, d are $d, -c, -b, a$, respectively. Hence, at every point of $\mathsf{SL}(2, \mathbb{R}) \subseteq \mathbb{R}^4$, at least one partial derivative of f is nonzero. By the implicit function theorem (see, e.g., [Edw73]), $\mathsf{SL}(2, \mathbb{R})$ is a smooth submanifold of $\mathbb{R}^4$ of codimension 1, and hence has dimension 3. Thus, $\mathsf{SL}(2, \mathbb{R})$ may be regarded as a 3-dimensional smooth manifold embedded in $\mathbb{R}^4$[3]. For further details on the interplay between Lie groups and smooth manifolds, see [BT13].

Every Lie group has an associated Lie algebra, which captures the infinitesimal structure of the group and describes its linearized behavior near the identity element. Intuitively, a Lie algebra describes the infinitesimal symmetries that generate the continuous symmetries represented by the Lie group. We first give the *abstract* definition of a Lie algebra (Definition 2.2.4) and then specialize to Lie algebras arising from matrix Lie groups (Definition 2.2.9). We also recall that Lie algebras are conventionally denoted by lowercase Gothic (Fraktur) letters, such

[2] A semidirect product is a product of groups equipped with a group operation determined by an action of one group on the other. In what follows, we will not need the explicit form of the underlying group operation.

[3] It may be helpful to view this as analogous to a smooth surface embedded in $\mathbb{R}^3$.

as $\mathfrak{g}$ and $\mathfrak{h}$.

Definition 2.2.4. A finite-dimensional Lie algebra is a finite-dimensional $\mathbb{K}$-vector space, $\mathfrak{g}$, equipped with a map $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ such that $[x, y]$ is $\mathbb{K}$-bilinear, $[x, y] = -[y, x]$ for all $x, y \in \mathfrak{g}$ and satisfies the Jacobi identity $[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0$ for all $x, y, z \in \mathfrak{g}$. A Lie algebra homomorphism $\phi : \mathfrak{g} \to \mathfrak{h}$ is a linear map between Lie algebras such that, for all $x, y \in \mathfrak{g}$, we have $\phi([x, y]) = [\phi(x), \phi(y)]$.

The map $[\cdot,\cdot]$ is called the Lie bracket on $\mathfrak{g}$. Roughly speaking, the Lie bracket measures the non-commutativity of elements $x, y \in \mathfrak{g}$. Accordingly, a Lie algebra $\mathfrak{g}$ is called commutative if $[x, y] = 0$ for all $x, y \in \mathfrak{g}$.

Example 2.2.5. The following are examples of Lie algebras.

1. Let $\mathfrak{g} = \mathfrak{gl}(n, \mathbb{K})$, the vector space of $n \times n$ matrices over $\mathbb{K}$. Then $\mathfrak{g}$ is a Lie algebra with Lie bracket given by $[X, Y] = XY - YX$.

2. Let $\mathfrak{g} = \mathbb{R}^3$. Then $\mathfrak{g}$ is a Lie algebra with Lie bracket given by $[x, y] = x \times y$, where $x \times y$ denotes the cross product.

2.2.2 Lie Algebra of a Matrix Lie Group

Our goal in the remainder of this section is to associate to each matrix Lie group G a corresponding Lie algebra $\mathfrak{g}$. This association allows us to study many properties of the group through its Lie algebra, where linear-algebraic tools are available. We first discuss the matrix exponential.

Definition 2.2.6. Let $A \in \mathsf{M}(n, \mathbb{K})$. The matrix exponential of A, denoted e^A or $\exp(A)$, is

defined by the power series expansion $e^A := \sum_{m=0}^{\infty} \frac{A^m}{m!}$. Here A^0 is defined to be the identity matrix I_n.

Proposition 2.2.7 (Proposition 2.1, Proposition 2.4, Theorem 2.12 in [Hal15])**.** *Let* $A \in \mathsf{M}(n, \mathbb{K})$*. The series in defining* e^A *converges for all* $A \in \mathsf{M}(n, \mathbb{K})$*. Moreover, we have* $\det e^A = e^{\operatorname{Tr} A}$ *and* $\frac{d}{dt} e^{tA} = Ae^{tA} = e^{tA}A$*.*

One motivation for introducing the matrix exponential comes from linear differential equations, which play a central role in Part I. To illustrate this, consider the following initial-value problem defined by an autonomous system of differential equations:

$$\dot{A}(t) = XA(t), \quad A(0) = A_0. \tag{2.8}$$

Here $A(t) \in \mathbb{K}^n$ and $X \in \mathsf{M}(n, \mathbb{K})$. By analogy with the scalar case, the solution is $A(t) = e^{tX} A_0$. It can be readily verified that this solves Equation (2.8) using Proposition 2.2.7(3). More generally, one can consider the following initial-value problem defined by a non-autonomous system of differential equations:

$$\dot{A}(t) = X(t)A(t), \quad A(0) = A_0, \tag{2.9}$$

where $A(t) \in \mathbb{K}^n$ and $X(t) \in \mathsf{M}(n, \mathbb{K})$ for each t. We will encounter such time-dependent systems in Chapter 4. The formal solution to Equation (2.9) is given by $A(t) = \exp_{\mathcal{T}} \left(\int_0^t X(t')\, dt' \right) A_0$, where $\exp_{\mathcal{T}}$ denotes the time-ordered exponential operator. We do not discuss the time-ordered exponential in detail here, but note that Lie-theoretic methods have been extensively applied to such operators in both finite and infinite-dimensional settings. The reader is referred to [DF84]

for further details.

Remark 2.2.8. In infinite dimensions, extra care is required to define the exponential rigorously. Here, we focus on the finite-dimensional case, which suffices for the applications in Chapter 3 and Chapter 4.

We can now define the Lie algebra associated to a matrix Lie group.

Definition 2.2.9. Let G be a matrix Lie group. The Lie algebra of G, denoted $\mathfrak{g}$, is defined as $\mathfrak{g} = \{X \in \mathsf{M}(n, \mathbb{C}) \mid e^{tX} \in G \text{ for all } t \in \mathbb{R}\}$.

Remark 2.2.10. In what follows, if G is a matrix Lie group, we denote its elements by capital Roman letters A, B, C, etc. If $\mathfrak{g}$ is the Lie algebra of a matrix Lie group, we denote its elements by capital Roman letters X, Y, Z, etc. When the Lie group–Lie algebra relationship is not being emphasized, we freely use any capital Roman letter to denote a matrix.

We now have two seemingly different notions of Lie algebra. Proposition 2.2.11 resolves this ambiguity by showing that the Lie algebra of a matrix Lie group is an $\mathbb{R}$-Lie algebra in the sense of Definition 2.2.4, with Lie bracket defined by $[X, Y] = XY - YX$.

Proposition 2.2.11 (Theorem 3.20 in [Hal15])**.** *Let G be a matrix Lie group with Lie algebra $\mathfrak{g}$. If $X, Y \in \mathfrak{g}$, then $X + tY \in \mathfrak{g}$ for all $t \in \mathbb{R}$, $XY - YX \in \mathfrak{g}$ and $AXA^{-1} \in \mathfrak{g}$ for all $A \in G$.*

Remark 2.2.12. Definition 2.2.9 may seem unmotivated to readers unfamiliar with Lie groups and Lie algebras. To give some intuition, suppose G is a matrix Lie group and $X \in \mathfrak{g}$. Consider the smooth curve $\gamma_X(t) = e^{tX}$, which satisfies the differential equation $\gamma_X'(t) = X\gamma_X(t), \gamma_X(0) = I_n$. Thus, $\gamma_X(t)$ is the solution to the initial-value problem determined by $X \in \mathfrak{g}$, as described in Equation (2.8). Consequently, $\gamma_X(t)$ may be interpreted as an integral curve, or solution curve,

of this differential equation. Since $\gamma_X'(0) = X$, the tangent vector to the curve at the identity $\gamma_X(0) = I_n$ is precisely X. Therefore, elements of the Lie algebra $\mathfrak{g}$ can be identified with the tangent space[4] to the Lie group G at the identity element I_n. Since G is a smooth manifold, its tangent space at the identity, denoted $T_{I_n}G$, gives a linear approximation of G near the identity. The Lie algebra $\mathfrak{g}$ is precisely this tangent space $T_{I_n}G$. Hence, for a matrix Lie group G, there is a natural one-to-one correspondence:

$$\text{Lie Algebra of } G \longleftrightarrow \text{Tangent Space at } I_n. \tag{2.10}$$

The integral curve γ_X satisfies the identity $\gamma_X(s+t) = \gamma_X(s)\gamma_X(t)$ for all $s, t \in \mathbb{R}$. Such smooth curves are called one-parameter subgroups of G. It can be shown that one-parameter subgroups of a matrix Lie group are in bijective correspondence with elements of its Lie algebra [Hal15, Theorem 2.14]. Therefore, for a matrix Lie group G, we have a natural one-to-one correspondence:

$$\text{One-parameter subgroups of } G \longleftrightarrow \text{Lie Algebra of } G \longleftrightarrow \text{Tangent Space at } I_n. \tag{2.11}$$

The key takeaway from Remark 2.2.12 is that the Lie algebra of a matrix Lie group encodes the infinitesimal structure of the group at the identity. The exponential map then provides a bridge from this linearized data back to the nonlinear group structure. In particular, it allows one to reconstruct the group locally, and in some cases globally, from its Lie algebra by considering integral curves emanating from the identity. This correspondence is made precise below.

[4]The tangent space at a point of a manifold is the vector space of tangent vectors to curves passing through that point.

Proposition 2.2.13 (Corollary 3.47 in [Hal15]). *Let G be a matrix Lie group with Lie algebra $\mathfrak{g}$. The exponential map generates G_0, the connected component of the identity. In particular, if G is connected, then for every $A \in G$ there exist $X_1, \cdots, X_n \in \mathfrak{g}$ such that $A = e^{X_1} \cdots e^{X_n}$.*

Proposition 2.2.13 raises the question of whether every element of G is the exponential of a single element of $\mathfrak{g}$. In general, this is not true; for example, it fails for $G = \mathsf{SL}(2, \mathbb{R})$. However, there is an important special case in which the exponential map is surjective.

Proposition 2.2.14 (Corollary 11.10 in [Hal15]). *Let G be a compact, connected matrix Lie group. Then the exponential map is surjective.*

A Lie group homomorphism between two Lie groups naturally induces a corresponding map between their Lie algebras, called the differential or pushforward at the identity. This map preserves the Lie bracket and encodes the infinitesimal behavior of the homomorphism.

Proposition 2.2.15 (Theorem 3.28 in [Hal15]). *Let G, H be two matrix Lie groups with Lie algebras $\mathfrak{g}, \mathfrak{h}$ respectively. If $f : G \to H$ is a Lie group homomorphism, then there exists a unique $\mathbb{R}$-linear map $f_* : \mathfrak{g} \to \mathfrak{h}$ such that $f(e^X) = e^{f_*(X)}$ for all $X \in \mathfrak{g}$. In other words, the following diagram commutes:*

$$\begin{array}{ccc} \mathfrak{g} & \xrightarrow{f_*} & \mathfrak{h} \\ \Big\downarrow{\scriptstyle \exp} & & \Big\downarrow{\scriptstyle \exp} \\ G & \xrightarrow{f} & H \end{array} \tag{2.12}$$

Remark 2.2.16. There are several ways to justify the existence of the induced map f_*. From the viewpoint that Lie groups are smooth manifolds and Lie algebras are tangent spaces at the identity, the map f_* arises as the differential of f at the identity element, namely the linear approximation of f near the identity. Thus, Proposition 2.2.15 says that the exponential map

intertwines the Lie group homomorphism f with its induced Lie algebra homomorphism f_*.

Example 2.2.17. We compute the Lie algebras of $\mathsf{SL}(n,\mathbb{C}), \mathsf{U}(n), \mathsf{SU}(n)$ and $\mathsf{SO}(n)$.

1. Let $U(t) = e^{tX} \in \mathsf{SL}(n,\mathbb{C})$. Differentiating the equation $1 = \det e^{tX}$, we obtain

$$0 = \frac{d}{dt}\left(\det e^{tX}\right)\Big|_{t=0} = \frac{d}{dt}\left(e^{t\mathrm{Tr}X}\right)\Big|_{t=0} = (\mathrm{Tr}X)\, e^{t\mathrm{Tr}X}\Big|_{t=0} = \mathrm{Tr}X. \tag{2.13}$$

   Hence, the Lie algebra $\mathfrak{sl}(n,\mathbb{C})$ of $\mathsf{SL}(n,\mathbb{C})$ is given by $\mathfrak{sl}(n,\mathbb{C}) = \{X \in \mathsf{M}(n,\mathbb{C}) : \mathrm{Tr}X = 0\}$.

2. Let $U(t) = e^{tX} \in \mathsf{U}(n)$. Differentiating the equation $U(t)^*U(t) = I_n$, we obtain $0 = \frac{d}{dt}\left((e^{tX})^* e^{tX}\right)\big|_{t=0} = X^* + X$. Hence, the Lie algebra $\mathfrak{u}(n)$ of $\mathsf{U}(n)$ is given by $\mathfrak{u}(n) = \{X \in \mathsf{M}(n,\mathbb{C}) : X^* = -X\}$.

3. Note that $\mathsf{SU}(n) = \mathsf{U}(n) \cap \mathsf{SL}(n,\mathbb{C})$. Therefore, (1) and (2) imply that the Lie algebra $\mathfrak{su}(n)$ of $\mathsf{SU}(n)$ is given by $\mathfrak{su}(n) = \{X \in \mathfrak{u}(n) : \mathrm{Tr}X = 0\}$.

4. An argument similar to those in (2) and (3) shows that $\mathfrak{so}(n) = \{X \in \mathsf{M}(n,\mathbb{R}) : X + X^T = 0\}$.

We conclude this section by showing that $\mathfrak{so}(3)$ and $\mathfrak{su}(2)$ are isomorphic Lie algebras.

Proposition 2.2.18. *$\mathfrak{so}(3)$ and $\mathfrak{su}(2)$ are isomorphic Lie algebras.*

Proof. (Sketch) The following elements form a basis for the Lie algebra $\mathfrak{su}(2)$:

$$\tilde{\sigma}_x := \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}, \quad \tilde{\sigma}_y := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad \tilde{\sigma}_z := \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}. \tag{2.14}$$

Similarly, the following elements form a basis for the Lie algebra $\mathfrak{so}(3)$:

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

The corresponding isomorphism is given by the mapping $X \mapsto \frac{1}{2}\tilde{\sigma}_x, Y \mapsto \frac{1}{2}\tilde{\sigma}_y, Z \mapsto \frac{1}{2}\tilde{\sigma}_z$. □

It is well known that the Lie groups $\mathsf{SO}(3)$ and $\mathsf{SU}(2)$ are not isomorphic. In fact, $\mathsf{SU}(2)/\mathbb{Z}_2 \cong \mathsf{SO}(3)$ (see, e.g., [Hal15]). However, the Lie algebras of $\mathsf{SO}(3)$ and $\mathsf{SU}(2)$ are isomorphic. Hence, we say that $\mathsf{SO}(3)$ and $\mathsf{SU}(2)$ are *locally isomorphic*.

2.3 Lie-Trotter & Baker-Campbell-Hausdorff Formulas

In this section, we assume that the base field is $\mathbb{K} = \mathbb{C}$. We present the Lie–Trotter formula, which gives a method for approximating the exponential of a sum of operators, and the Baker–Campbell–Hausdorff (BCH) formula, which expresses the logarithm of a product of matrix exponentials in terms of iterated Lie brackets. These formulas are fundamental for understanding the relationship between Lie groups and their Lie algebras and play a crucial role in Chapter 3 and Chapter 4.

2.3.1 Matrix Logarithm

The matrix logarithm, where defined, serves as the inverse of the matrix exponential.

Definition 2.3.1. Let $X \in \mathsf{M}(n, \mathbb{C})$. The matrix logarithm of X, denoted $\log X$, is $\log X := \sum_{m=1}^{\infty} (-1)^{m+1} \frac{(X-I)^m}{m}$ whenever the series converges.

Definition 2.3.1 is motivated by the formal power series for the complex logarithm: $\log z = \sum_{m=1}^{\infty}(-1)^{m+1}\frac{(z-1)^m}{m}$. It is standard that this series defines a holomorphic function on the disk of radius 1 centered at $z = 1$. For all z with $|z-1| < 1$, we have $e^{\log z} = z$. For all u with $|u| < \log 2$, we have $|e^u - 1| < 1$ and $\log e^u = u$. This yields the following result:

Lemma 2.3.2 (Theorem 2.8, Proposition 2.9 in [Hal15]). *The matrix logarithm is defined and continuous for all $X \in \mathsf{M}(n,\mathbb{C})$ such that $\|X-I\| < 1$. For all $X \in \mathsf{M}(n,\mathbb{C})$ with $\|X-I\| < 1$, we have $e^{\log X} = X$. For all $X \in \mathsf{M}(n,\mathbb{C})$ with $\|X\| < \log 2$, we have $\|e^X - I\| < 1$ and $\log e^X = X$. Moreover, there exists a constant $c > 0$ such that for all $X \in \mathsf{M}(n,\mathbb{C})$ with $\|X\| < \frac{1}{2}$, we have $\|\log(I+X) - X\| \le c\|X\|^2$.*

2.3.2 Lie-Trotter Formula

With the matrix logarithm in hand, we now prove the Lie–Trotter formula. Since the proof is concise, we present it in detail. The argument is adapted from [Hal15].

Proposition 2.3.3 (**Lie-Trotter Formula**). *We have $e^{X+Y} = \lim_{m\to\infty}\left(e^{X/m}e^{Y/m}\right)^m$ for $X, Y \in \mathsf{M}(n,\mathbb{C})$.*

Proof. Multiplying the power series for $e^{X/m}$ and $e^{Y/m}$, all remaining terms beyond the linear terms have order $1/m^2$ or higher. Thus, $e^{X/m}e^{Y/m} = I_n + \frac{X}{m} + \frac{Y}{m} + \mathcal{O}\left(\frac{1}{m^2}\right)$. Since $e^{X/m}e^{Y/m} \to I_n$ as $m \to \infty$, $e^{X/m}e^{Y/m}$ lies in the domain of the logarithm for all sufficiently large m. By Lemma 2.3.2,

$$\log\left(e^{X/m}e^{Y/m}\right) = \log\left(I_n + \frac{X}{m} + \frac{Y}{m} + \mathcal{O}\left(\frac{1}{m^2}\right)\right) = \frac{X}{m} + \frac{Y}{m} + \mathcal{O}\left(\frac{1}{m^2}\right). \tag{2.16}$$

Exponentiating gives $e^{X/m}e^{Y/m} = e^{\frac{X}{m}+\frac{Y}{m}+\mathcal{O}\left(\frac{1}{m^2}\right)}$, and hence $(e^{X/m}e^{Y/m})^m = e^{X+Y+\mathcal{O}(1/m)}$. The result follows by continuity of the exponential. □

2.3.3 Baker-Campbell-Hausdorff Formula

The Baker–Campbell–Hausdorff (BCH) formula gives an explicit expression for a product of two matrix exponentials of Lie algebra elements as a single exponential. Specifically, for elements X, Y in a Lie algebra $\mathfrak{g}$, the BCH formula expresses $e^X e^Y$ as e^Z, where Z is given by a series involving X, Y, and their iterated Lie brackets. Beyond its foundational role in describing the local structure of Lie groups and their Lie algebras, the BCH formula has important applications in applied Lie theory, including quantum computation. Since the algorithm analyzed in Chapter 3 makes extensive use of the Baker–Campbell–Hausdorff formula, we discuss it in detail here. To simplify the nested commutator expressions appearing in the formula, we introduce two standard maps from Lie theory.

Definition 2.3.4. Let G be a matrix Lie group with Lie algebra $\mathfrak{g}$. Let $A \in G$ and $X \in \mathfrak{g}$. The map $\mathrm{Ad}_A : \mathfrak{g} \to \mathfrak{g}$ is the Lie algebra isomorphism given by $\mathrm{Ad}_A(X) = AXA^{-1}$. Similarly, $\mathrm{ad}_X : \mathfrak{g} \to \mathfrak{g}$ is the linear map given by $\mathrm{ad}_X(Y) = [X, Y]$.

Remark 2.3.5.] Both maps introduced in Definition 2.3.4 play an important role in the BCH formula. For any $X, Y \in \mathsf{M}(n, \mathbb{C})$, they are related by the identity $e^X Y e^{-X} = \mathrm{Ad}_{e^X}(Y) = e^{\mathrm{ad}_X}(Y)$ [Hal15, Proposition 3.35].

We now derive the BCH formula. Lemma 2.3.2 implies that, near the identity matrix in a matrix Lie group, the exponential map is locally invertible, with inverse given by the matrix logarithm. Hence, we denote by log the local inverse of exp. The key idea is to consider

the expression $Z(t) = \log\left(e^X e^{tY}\right)$. Clearly, $Z(0) = X$, and $Z(1)$ is the element satisfying $e^{Z(1)} = e^X e^Y$. The strategy is to find an analytic expression for $Z'(t)$ and then integrate from 0 to 1 to compute $Z(1)$. We first derive a formula for the derivative of a matrix exponential.

Lemma 2.3.6. *For all $X, Y \in \mathsf{M}(n, \mathbb{C})$, we have $\Delta(X, Y) := \frac{d}{dt} e^{X+tY}\big|_{t=0} = \left(e^X \circ \phi(\mathrm{ad}_X)\right)(Y)$, where $\phi(z) = \frac{1-e^{-z}}{z} = \sum_{m=0}^{\infty} \frac{(-1)^m}{(m+1)!} z^m$. More generally, if $Z(t)$ is a smooth matrix-valued function, then $\frac{d}{dt} e^{Z(t)} = e^{Z(t)}\, \phi(\mathrm{ad}_{Z(t)}) \left(\frac{dZ}{dt}\right)$.*

Proof. For $m \in \mathbb{N}$, we have

$$\Delta(X, Y) = \frac{d}{dt}\bigg|_{t=0} (e^{X/m + tY/m})^m, \tag{2.17}$$

$$= \sum_{k=0}^{m-1} (e^{X/m})^{m-k-1}\, \Delta(X/m, Y/m) (e^{X/m})^k, \tag{2.18}$$

$$= \frac{1}{m} (e^{X/m})^{m-1} \sum_{k=0}^{m-1} (e^{X/m})^{-k} \Delta(X/m, Y) (e^{X/m})^k, \tag{2.19}$$

$$= (e^{X/m})^{m-1} \frac{1}{m} \sum_{k=0}^{m-1} \exp\left(-\frac{\mathrm{ad}_X}{m}\right)^k \Delta(X/m, Y). \tag{2.20}$$

In the third equality, we used that $\Delta(X, Y)$ is $\mathbb{R}$-linear in Y. In the fourth equality, we used Remark 2.3.5. As $m \to \infty$, we have $(e^{X/m})^{m-1} \xrightarrow{m\to\infty} e^X$ and $\Delta(X/m, Y) \xrightarrow{m\to\infty} \Delta(0, Y) = Y$. Moreover, since ad_X is a linear operator, we have

$$\frac{1}{m} \sum_{k=0}^{m-1} \exp\left(-\frac{\mathrm{ad}_X}{m}\right)^k = \frac{1}{m} \sum_{k=0}^{m-1} \sum_{n=0}^{\infty} \frac{1}{n!} \left(-\frac{k\, \mathrm{ad}_X}{m}\right)^n, \tag{2.21}$$

$$= \sum_{n=0}^{\infty} \left[\frac{1}{m} \sum_{k=0}^{m-1} \left(\frac{k}{m}\right)^n\right] \frac{(-1)^n}{n!} (\mathrm{ad}_X)^n, \tag{2.22}$$

$$\xrightarrow{m\to\infty} \sum_{n=0}^{\infty} \left[\int_0^1 x^n dx\right] \frac{(-1)^n}{n!} (\mathrm{ad}_X)^n = \sum_{n=0}^{\infty} \frac{(-1)^n}{(n+1)!} (\mathrm{ad}_X)^n. \tag{2.23}$$

This proves the formula for $\Delta(X,Y)$. The formula for $d/dt\ e^{Z(t)}$ follows from the chain rule. $\square$

We now prove the Baker–Campbell–Hausdorff formula, which gives an explicit expression for the logarithm of a product of exponentials as an infinite series of nested commutators. Its integral representation links Lie algebras to Lie groups and underlies many constructions in mathematical physics and quantum mechanics.

Proposition 2.3.7. *Let G be a matrix Lie group with Lie algebra $\mathfrak{g}$. For $X, Y \in \mathfrak{g}$ sufficiently small, we have*

$$\log\left(e^X e^Y\right) = X + \int_0^1 g(e^{\operatorname{ad}_X} e^{t\operatorname{ad}_Y})(Y)\, dt, \tag{2.24}$$

$g(z) = \frac{\log z}{1-z^{-1}}$.

Proof. Consider the expression $Z(t) = \log\left(e^X e^{tY}\right)$, or equivalently, its exponential form $e^{Z(t)} = e^X e^{tY}$. We have $e^{-Z(t)}\frac{d}{dt}e^{Z(t)} = \left(e^X e^{tY}\right)^{-1}\frac{d}{dt}\left(e^X e^{tY}\right) = Y$. On the other hand, Lemma 2.3.6 implies that $e^{-Z(t)}\frac{d}{dt}e^{Z(t)} = \left(\frac{I-e^{-\operatorname{ad}_{Z(t)}}}{\operatorname{ad}_{Z(t)}}\right)\frac{dZ}{dt}$. Since X and Y are sufficiently small, $Z(t)$ is also sufficiently small, so the operator $\frac{I-e^{-\operatorname{ad}_{Z(t)}}}{\operatorname{ad}_{Z(t)}}$ is close to the identity and therefore invertible. Hence, $\frac{dZ}{dt} = \left(\frac{I-e^{-\operatorname{ad}_{Z(t)}}}{\operatorname{ad}_{Z(t)}}\right)^{-1}(Y)$. Since $e^{Z(t)} = e^X e^{tY}$, we have $\operatorname{Ad}_{e^{Z(t)}} = \operatorname{Ad}_{e^X}\operatorname{Ad}_{e^{tY}}$, $e^{\operatorname{ad}_{Z(t)}} = e^{\operatorname{ad}_X} e^{t\operatorname{ad}_Y}$ and $\operatorname{ad}_{Z(t)} = \log\left(e^{\operatorname{ad}_X} e^{t\operatorname{ad}_Y}\right)$. Therefore, we have

$$\frac{dZ}{dt} = \left(\frac{\log\left(e^{\operatorname{ad}_X} e^{t\operatorname{ad}_Y}\right)}{I-(e^{\operatorname{ad}_X} e^{t\operatorname{ad}_Y})^{-1}}\right)(Y) := g(e^{\operatorname{ad}_X} e^{t\operatorname{ad}_Y})(Y), \tag{2.25}$$

where $g(\cdot)$ is the function $g(z) = \frac{\log z}{1-z^{-1}}$. Since $Z(0) = X$, we obtain

$$\log\left(e^X e^Y\right) = Z(1) = X + \int_0^1 g(e^{\operatorname{ad}_X} e^{t\operatorname{ad}_Y})(Y)\, dt. \tag{2.26}$$

This completes the proof. □

Expanding the integrand in Equation (2.26) as a power series in iterated commutators gives the BCH formula as a formal infinite sum of nested Lie brackets of X and Y. A formal derivation can be found in [Hal15, Section 5.6], which gives the first few terms in the expansion of $\log\left(e^X e^Y\right)$:

$$\log\left(e^X e^Y\right) = X + Y + \frac{1}{2}[X,Y] + \frac{1}{12}[X,[X,Y]] + \frac{1}{12}[Y,[Y,X]] + \cdots, \tag{2.27}$$

where the higher-order terms involve increasingly nested commutators. This expansion shows how the noncommutativity of Lie algebra elements contributes to the structure of $\log\left(e^X e^Y\right)$.

2.4 Quantum Theory

The simplest classical system is represented by a point particle in $\mathbb{R}^3$. Quantization of this system is traditionally realized through the infinite-dimensional Hilbert space $L^2(\mathbb{R}^3)$, where $L^2(\mathbb{R}^3)$ is the space of square-integrable complex-valued functions on $\mathbb{R}^3$, equipped with the inner product $\langle f, g\rangle = \int_{\mathbb{R}^3} \overline{f(x)} g(x)\, dx$. Here, however, we adopt a simpler finite-dimensional framework, focusing on quantum systems modeled by finite-dimensional Hilbert spaces $\mathscr{H}$. A canonical example is the *spin* of the electron, an intrinsic quantum degree of freedom with no classical analogue. While classical rotations and angular momentum are described by the Lie group $\mathsf{SO}(3)$, quantum theory over complex vector spaces naturally identifies $\mathsf{SU}(2)$ as the relevant symmetry group. The action of $\mathsf{SU}(2)$ on $\mathbb{C}^2$ provides a representation of the electron's spin degrees of freedom. With this motivating example in mind, we proceed to the axiomatic

formulation of quantum theory.

Axiom 2.4.1 (Postulate I)**.** A finite-dimensional quantum system is modeled by a finite-dimensional complex Hilbert space $\mathscr{H}$.

Since we restrict our attention to finite-dimensional complex Hilbert spaces, every such space is isometrically isomorphic to $\mathbb{C}^n$ for some $n \geq 1$. Nevertheless, it is useful to adopt an intrinsic, basis-free definition of a Hilbert space $\mathscr{H}$.

Definition 2.4.2. A (finite-dimensional) complex Hilbert space, $\mathscr{H}$, is a (finite-dimensional) complex vector space equipped with a non-degenerate Hermitian inner product $\langle , \rangle : \mathscr{H} \times \mathscr{H} \to \mathbb{C}$ satisfying the following properties:

1. (Linearity in the second argument) For $x, y, y' \in \mathscr{H}$ and $\lambda \in \mathbb{C}$, we have $\langle x, y + y' \rangle = \langle x, y \rangle + \langle x, y' \rangle$ and $\langle x, \lambda y \rangle = \lambda \langle x, y \rangle$.

2. (Hermitian symmetry) For all $x, y \in \mathscr{H}$, we have $\langle y, x \rangle = \overline{\langle x, y \rangle}$,

3. (Positivity) For all $x \in \mathscr{H}$, we have $\langle x, x \rangle > 0$ for $x \neq 0$ and $\langle x, x \rangle = 0$ if and only if $x = 0$.

A quantum state of an n-dimensional quantum system is represented by a vector $v \in \mathbb{C}^n$ satisfying $\|v\| = 1$. Such a state is commonly written in Dirac notation as $|v\rangle$ and is called a *ket*. The Hermitian inner product $\langle \cdot, \cdot \rangle$ induces a canonical conjugate-linear isomorphism between the Hilbert space $\mathscr{H}$ and its dual space $\mathscr{H}^*$ via the map $\phi : \mathscr{H} \to \mathscr{H}^*$ mapping $|v\rangle \to \langle v, \cdot \rangle$. If a basis for $\mathscr{H}$ is fixed, a ket $|v\rangle$ can be represented by a column vector. The complex conjugate transpose of $|v\rangle$ is denoted by $\langle v|$ and is called a *bra*. Thus, $\langle v|$ represents the linear functional

$w \mapsto \langle v, w\rangle$. This notation is known as Dirac's bra-ket notation. If $\{|i\rangle\}_{i=1}^n$ is a basis for $\mathbb{C}^n$, then any quantum state in $\mathbb{C}^n$ can be expressed as $|v\rangle = \sum_{i=1}^n v_i |i\rangle$.

Axiom 2.4.3 (Postulate II)**.** The measurement postulate for a finite-dimensional closed quantum system with Hilbert space $\mathscr{H}$ can be summarized as follows:

1. An observable is a self-adjoint, or Hermitian, operator $T \in \mathsf{End}(\mathscr{H})$ satisfying $T = T^\dagger$. The possible measurement outcomes of T are its eigenvalues, counted without multiplicity.

2. If the system is in the state $|\psi\rangle \in \mathscr{H}$, the probability of obtaining an eigenvalue $\lambda \in \mathbb{R}$ is $\mathbb{P}(\lambda \,||\, \psi\rangle) = \frac{\langle\psi|P_\lambda|\psi\rangle}{\langle\psi|\psi\rangle}$, where P_λ is the orthogonal projection onto the λ-eigenspace of T. Upon obtaining the outcome λ, the post-measurement state becomes $P_\lambda|\psi\rangle$ up to normalization.

If a basis for $\mathscr{H}$ is fixed, the matrix representation of T, denoted O, satisfies $O^* = O$. For a unit-norm quantum state $|\psi\rangle$, Axiom 2.4.3 implies that the probability of obtaining the outcome λ is $\mathbb{P}(\lambda \,||\, \psi\rangle) = \langle\psi|\, P_\lambda \,|\psi\rangle$ and the corresponding expected value is $\mathbb{E}(O) = \langle\psi|\, O \,|\psi\rangle$. Axiom 2.4.3 implies that quantum measurement statistics derived from n-dimensional unit-norm quantum states remain unchanged if two states differ by a global phase. Consequently, if $\mathscr{H} \cong \mathbb{C}^n$, the physically distinct quantum states of an n-dimensional quantum system are represented by the complex projective space $\mathbb{CP}^{n-1}$ for $n \geq 2$. The space $\mathbb{CP}^{n-1}$ can be described as the quotient space $\mathbb{CP}^{n-1} := \mathbb{S}^{2n-1}/\mathbb{S}^1$, where $\mathbb{S}^{2n-1} \subseteq \mathbb{C}^n$ denotes the unit sphere, and for $z, w \in \mathbb{S}^{2n-1}$, $z \sim w$ in the quotient space if and only if $\lambda \in \mathsf{U}(1) \cong \mathbb{S}^1$. Nonetheless, by an abuse of notation, we continue to identify a quantum state of an n-dimensional quantum system with a unit vector in $\mathbb{C}^n$, keeping in mind that global phases are physically irrelevant.

Axiom 2.4.4 (Postulate III)**.** Given two distinct finite-dimensional quantum systems modeled by $\mathscr{H}_1$ and $\mathscr{H}_2$, the composite finite-dimensional quantum system is modeled by $\mathscr{H}_1 \otimes \mathscr{H}_2$.

The previous postulates describe the state space of finite-dimensional quantum systems and the measurement postulate. The final postulate specifies the unitary dynamics of quantum systems.

Axiom 2.4.5 (Postulate IV)**.** Let a finite-dimensional quantum system have state space $\mathscr{H}$. The evolution of the system from time t_1 to t_2 is described by a unitary propagator $U(t_2, t_1) \in \mathsf{U}(\mathscr{H})$ such that, if the system is in state $|\psi(t_1)\rangle$ at time t_1, then at time t_2 it is in the state $|\psi(t_2)\rangle = U(t_2, t_1)|\psi(t_1)\rangle$. The unitary propagator satisfies $U(t_3, t_1) = U(t_3, t_2)U(t_2, t_1)$ for all $t_1 \leq t_2 \leq t_3$.

The unitarity condition in Axiom 2.4.5 ensures that the norm of a quantum state is conserved over time. Let $\mathscr{H} = \mathbb{C}^2$. Since global phases are physically unobservable, and using the decomposition $\mathsf{U}(2) \cong \mathsf{SU}(2) \ltimes \mathsf{U}(1)$ (Example 2.2.2), the essential features of single-qubit quantum dynamics can be captured by matrices in $\mathsf{SU}(2)$. Specifically, every element of $\mathsf{U}(2)$ is equivalent, up to an overall global phase, to an element of $\mathsf{SU}(2)$. Thus, for the evolution of a single-qubit quantum system, it suffices to consider $\mathsf{SU}(2)$. A similar argument applies if $\mathscr{H} \cong \mathbb{C}^n$ for $n > 2$.

We further discuss Axiom 2.4.5 in the special case where quantum dynamics are generated by a time-independent operator. Let $\mathscr{H} \cong \mathbb{C}^n$ for $n \geq 2$. Since $\mathsf{SU}(n)$ is a compact and connected matrix Lie group, we can invoke Proposition 2.2.14 to argue that $U(t) = e^{tH}$ for some

$H \in \mathfrak{su}(n)$. Therefore, we have

$$|\psi_t\rangle = U|\psi_0\rangle \iff |\psi_t\rangle = e^{tH}|\psi_0\rangle \iff \frac{d}{dt}|\psi_t\rangle = H|\psi_t\rangle. \tag{2.28}$$

Physicists usually work with a Hermitian Hamiltonian, replacing the skew-Hermitian generator $H \in \mathfrak{su}(n)$ by $-iH$. With this convention, the matrix exponential is e^{-itH}. In this case, a slightly different version of Equation (2.28) reads:

$$|\psi_t\rangle = U|\psi_0\rangle \iff |\psi_t\rangle = e^{-itH}|\psi_0\rangle \iff \frac{d}{dt}|\psi_t\rangle = -iH|\psi_t\rangle. \tag{2.29}$$

The equation on the right-hand side of Equation (2.29) is equivalent to the Schrödinger equation, provided we adopt the convention that Planck's constant $\hbar$ is set to 1. The time-independent matrix H is called the *Hamiltonian* of the underlying closed finite-dimensional system, and it generates quantum dynamics via $U(t) = e^{-itH}$. If the Hamiltonian is time-dependent, the corresponding Schrödinger equation reads $i\frac{d}{dt}|\psi_t\rangle = H(t)|\psi_t\rangle$ and the formal solution to this equation is written as $U(T, T_0) = \exp_{\mathcal{T}}\left(-i\int_{T_0}^{T} H(s)ds\right)$, where $\exp_{\mathcal{T}}$ denotes the time-ordered exponential operator, for $T_0 \leq t \leq T$.

2.5 Gate-Based Model

We now discuss the gate-based model of quantum computation. In this model, information is represented by qubits, and each quantum gate is a unitary operator acting on the state space of one or more qubits, analogous to classical logic gates in conventional computation. A single-qubit quantum system is modeled in $\mathbb{C}^2$. A quantum state of such a system, called a *qubit* or

quantum bit, is a normalized vector in $\mathbb{C}^2$, i.e., $|v\rangle = (v_1, v_2)^\top$ with $\|v\| = 1$. In Dirac's bra-ket notation, a basis for $\mathbb{C}^2$ is written as

$$|0\rangle = \begin{pmatrix} 1 \\ \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ \\ 1 \end{pmatrix}. \tag{2.30}$$

The basis $\{|0\rangle, |1\rangle\}$ is called the *computational basis*. An arbitrary qubit can then be expressed as $|v\rangle = v_1 |0\rangle + v_2 |1\rangle$ with $|v_1|^2 + |v_2|^2 = 1$.

Remark 2.5.1. Quantum states of a single-qubit system are in bijection with $\mathbb{CP}^1$, which is topologically equivalent to the 2-sphere $\mathbb{S}^2$. This correspondence motivates the Bloch sphere representation of single-qubit states (Figure 2.1). We do not pursue this perspective further.

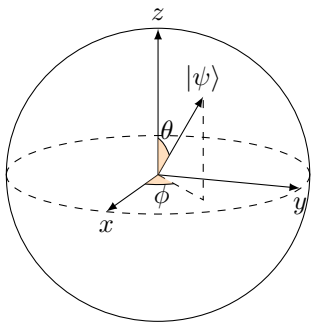


Figure 2.1: Bloch sphere representation of a single-qubit state $|\psi\rangle$. The computational basis states $|0\rangle$ and $|1\rangle$ define the z-axis. The angles θ and ϕ denote the spherical coordinates of $|\psi\rangle$.

A single-qubit quantum gate is a unitary operator on $\mathbb{C}^2$, represented up to global phase by an element of $\mathsf{SU}(2)$. A single-qubit quantum circuit is a sequence of such gates applied to a single qubit. The qubit is represented by a horizontal line, called a *wire*, and the gates are depicted as boxes acting on this wire. Measurements, if any, are typically shown at the end of the circuit. Single-qubit quantum circuits provide a visual and operational framework for designing, analyzing, and simulating quantum computations (Figure 2.2). Up to global phase,

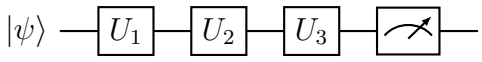


Figure 2.2: A generic single-qubit quantum circuit. The qubit $|\psi\rangle$ passes through a sequence of unitary gates U_1, U_2, U_3 and is then measured.

any $U \in \mathsf{U}(2)$ can be written as $U = e^{-itH}$, where $H \in i\mathfrak{su}(2)$. Using the physicists' convention of multiplying the basis elements of $\mathfrak{su}(2)$ by i, a basis for $i\mathfrak{su}(2)$ is given by the Pauli operators:

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.31}$$

The Pauli operators are unitary and therefore define three single-qubit gates, commonly denoted by $\sigma_X, \sigma_Y, \sigma_Z$ or X, Y, Z. The σ_X gate is the quantum analogue of the classical NOT gate, flipping the state of a qubit. The σ_Z gate is a phase-flip gate, which has no classical analogue and applies a relative phase of -1 to the $|1\rangle$ component of the qubit. The σ_Y gate can be expressed as $\sigma_Y = i\,\sigma_X\sigma_Z$, combining both bit-flip and phase-flip operations. In diagrammatic notation, these gates are represented in Figure 2.3. Using the matrix exponential, we can define

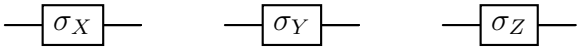


Figure 2.3: Single-qubit Pauli gates σ_X, σ_Y, and σ_Z represented in circuit notation.

single-qubit rotation gates about the X-axis, Y-axis, and Z-axis as follows:

$$e^{-i\theta\sigma_X} = \begin{pmatrix} \cos\theta & -i\sin\theta \\ -i\sin\theta & \cos\theta \end{pmatrix}, \quad e^{-i\theta\sigma_Y} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \quad e^{-i\theta\sigma_Z} = \begin{pmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{pmatrix}. \tag{2.32}$$

We denote the quantum gates $e^{-i\theta\sigma_X}$, $e^{-i\theta\sigma_Y}$, and $e^{-i\theta\sigma_Z}$ by $R_X(\theta)$, $R_Y(\theta)$, and $R_Z(\theta)$, respectively. The Hadamard gate, which creates an equal superposition of basis states, is represented

as follows:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \quad -\boxed{H}- \tag{2.33}$$

The simplest example of a composite quantum system is a k-qubit system. For $k \geq 1$, a k-qubit quantum state is an element of $\mathbb{C}^{2^k} := \underbrace{\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{k \text{ times}}$. For $k = 1$, we recover the single-qubit states discussed above. The computational basis for $\mathbb{C}^{2^k}$ is given by $\mathcal{B} = \{|j_1 \cdots j_k\rangle \mid j_i = 0, 1 \text{ for } i = 1, \cdots, k\}$. A k-qubit quantum gate is a unitary operator on $\mathbb{C}^{2^k}$, represented up to global phase by an element of $\mathsf{SU}(2^k)$. A k-qubit quantum circuit is a sequence of such gates applied to one or more qubits. As in the single-qubit case, multi-qubit circuits provide a visual and operational framework for designing, analyzing, and simulating quantum computations (Figure 2.4). For a k-qubit quantum circuit, the *width* is the number of qubits involved, while the *depth* is the maximum number of sequential operations acting on any single qubit. For example, in the 3-qubit circuit depicted in Figure 2.4, the width is 3 and the depth is 4. An

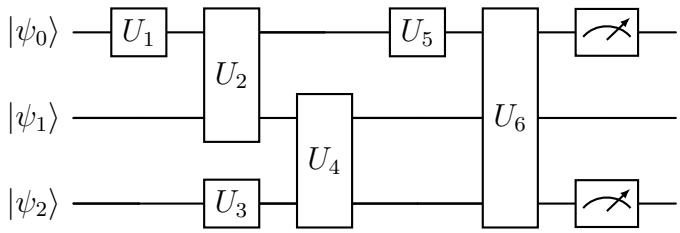


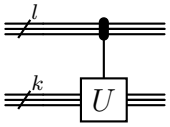
Figure 2.4: A generic 3-qubit quantum circuit illustrating the application of single- and multi-qubit gates, with measurements performed on selected qubits at the end of the circuit.

important example of a k-qubit gate is the k-fold tensor product of Hadamard gates:

$$H^{\otimes k} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes k} = \quad \overset{k}{\equiv}\boxed{H^{\otimes k}}\equiv \tag{2.34}$$

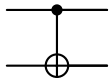
A generic k-qubit gate U, controlled on l qubits, is denoted as follows:

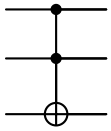
$$C^l U = \tag{2.35}$$



Important examples are the two-qubit CNOT gate and the three-qubit Toffoli, or CCNOT, gate:

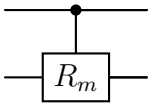
$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \quad \text{CCNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} = \tag{2.36}$$





Note that CNOT $= C\sigma_X$ and CCNOT $= C^2\sigma_X$. Single-qubit controlled rotations also play an important role in quantum computation. We consider one such example below.

$$CR_m = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-2\pi i/2^m} \end{pmatrix} = \tag{2.37}$$



2.6 Quantum Algorithms

Quantum algorithms use superposition, entanglement, and interference to solve certain computational problems more efficiently than classical methods. This section reviews the quantum algorithms and algorithmic techniques used throughout this thesis. We begin with quantum arithmetic primitives, which enable the efficient manipulation of numerical data stored in quantum registers. We then introduce frameworks for representing and manipulating operators, including block encoding and linear combinations of unitaries. Finally, we discuss the Quantum Fourier Transform (QFT) and Quantum Signal Processing (QSP), which are central tools in many quantum algorithms.

2.6.1 Quantum Arithmetic

We assume that all quantum algorithms take as input a fixed-precision binary number stored in an n-qubit quantum register, denoted $|x\rangle$. The leftmost m qubits encode the integer part, while the remaining $n-m$ qubits encode the fractional part. A leading qubit may be introduced, when necessary, to encode the sign of the number. Formally, $|x\rangle = |x^{(m-1)}\rangle \otimes \cdots \otimes |x^{(0)}\rangle \otimes |x^{(-1)}\rangle \otimes \cdots \otimes |x^{(m-n)}\rangle$, where each $x^{(j)} \in \{0,1\}$ and the number is represented as

$$x = \underbrace{\sum_{j=0}^{m-1} x^{(j)} 2^j}_{\text{integer part}} + \underbrace{\sum_{j=-1}^{-(n-m)} x^{(j)} 2^j}_{\text{fractional part}} . \tag{2.38}$$

Quantum algorithms for addition (see, e.g., [Dra00; Cuc+04; Dra+06]) and multiplication (see, e.g., [PRM17; Gid19; Ram+23]) have been extensively studied, with circuit complexities of

$\mathcal{O}(n)$ Toffoli and CNOT gates for addition and $\mathcal{O}(n^2)$ for multiplication. Since these algorithms will only be used in Chapter 6, and the algorithm considered there requires overall complexity $\Omega(n^2)$, it suffices to treat addition and multiplication as elementary primitives. See Figure 2.5 for a quantum oracle U_{AM} implementing multiplication and addition as a single quantum circuit.

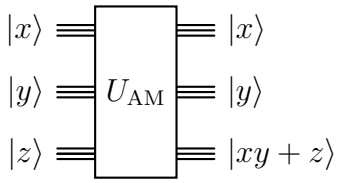


Figure 2.5: Quantum oracle for multiplication and addition implementing $|x\rangle|y\rangle|z\rangle \mapsto |x\rangle|y\rangle|xy+z\rangle$.

Recent work by Burge et al. [BBG24] introduces a reversible quantum algorithm for computing the arcsine function using a quantum adaptation of the CORDIC (COordinate Rotation DIgital Computer) method. The algorithm iteratively approximates $\arcsin(x)$ through a sequence of reversible rotations, enabling efficient implementation on quantum hardware. For a p-bit output, the algorithm requires $\mathcal{O}(p)$ qubits and $\mathcal{O}(p^2)$ CNOT gates, with circuit depth scaling as $\mathcal{O}(p \log p)$ and worst-case approximation error $\mathcal{O}(2^{-p})$. We will use this algorithm in Chapter 6.

Remark 2.6.1. The implementation of Burge et al. uses $\mathcal{O}(p)$ additional qubit registers as scratch space for intermediate variables. Since this overhead is absorbed into the asymptotic complexity, we write p for the ancilla register storing the final p-bit approximation whenever only the principal p ancillas are relevant.

2.6.2 Block Encoding

Quantum algorithms for matrix manipulation often assume that a possibly non-unitary matrix can be embedded as a block of a unitary matrix. The block-encoding framework provides a systematic way to perform linear-algebraic operations on a quantum computer. The key idea is to represent a subnormalized matrix A/α as the upper-left block of a unitary matrix. Since the block encoding need not be exact, we use the following definition:

Definition 2.6.2. Let $\alpha, \epsilon \in \mathbb{R}^+$. Given $A \in \mathbb{C}^{r\times c}$, we say $U \in \mathbb{C}^{d\times d}$ for $d \geq \max\{r, c\}$ is a (α, ϵ) block encoding of A if $\|A - \alpha(B_L^\dagger\, U\, B_R)\| \leq \epsilon$, where $B_L \in \mathbb{C}^{d\times r}$ and $B_R \in \mathbb{C}^{d\times c}$ are the first r and c columns of the identity matrix, respectively.

We will often take d, r, and c to be powers of two. In the square case, if $A \in \mathbb{C}^{2^n\times 2^n}$, we say that a unitary $U \in \mathbb{C}^{2^{n+a}\times 2^{n+a}}$ is a (α, a, ϵ)-block-encoding of A if $\|A-\alpha(\langle 0|^a \otimes I_{2^n})U(|0\rangle^a \otimes I_{2^n})\| \leq \epsilon$. This is the case considered below. For ease of notation, we write:

$$U = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix} \tag{2.39}$$

Example 2.6.3. Let us consider an explicit computation to clarify the notation. Let $A \in \mathbb{C}^{2\times 2}$ and $\alpha = 1$. In this case, we have

$$(\langle 0| \otimes I_2)\, U\, (|0\rangle \otimes I_2) = \begin{pmatrix} I_2 & 0 \end{pmatrix} \begin{pmatrix} A & C \\ B & D \end{pmatrix} \begin{pmatrix} I_2 \\ 0 \end{pmatrix} = \begin{pmatrix} I_2 & 0 \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = A. \tag{2.40}$$

The case $A \in \mathbb{C}^{2^s\times 2^s}$ with $s > 1$ and $a > 1$ is analogous.

We will use the following lemma in our analysis.

Lemma 2.6.4 (Lemma 53 in [Gil+19a]). *If U is an (α, a, δ)-block-encoding of an s-qubit operator A, and V is a (β, b, ϵ)-block-encoding of an s-qubit operator B, then $(I_b \otimes U)(I_a \otimes V)$ is an $(\alpha\beta, a+b, \alpha\epsilon+\beta\delta)$-block-encoding of AB, where I_a and I_b act on the ancilla qubits of U and V, respectively.*

We will also use a result on block encoding sparse matrices, which can be found in [Gil+19a, Lemma 48]. For our purposes, it suffices to restrict to sparse $0-1$ matrices. The corresponding specialization is stated below.

Lemma 2.6.5. *Let $A \in \mathbb{C}^{2^n \times 2^n}$ be a d_r-row-sparse and d_c-column-sparse 0-1 matrix. Suppose that we have access to the following $2(n+1)$-qubit oracles:*

$$O_r : |i\rangle|k\rangle \mapsto |i\rangle|r_{ik}\rangle, \quad 0 \le i < 2^n,\ 0 \le k < d_r, \tag{2.41}$$

$$O_c : |l\rangle|j\rangle \mapsto |c_{lj}\rangle|j\rangle, \quad 0 \le j < 2^n,\ 0 \le l < d_c, \tag{2.42}$$

where r_{ik} is the index of the k-th nonzero entry of the i-th row of A, with $r_{ik} = k + 2^n$ if there are fewer than k nonzero entries. Similarly, c_{lj} is the index of the l-th nonzero entry of the j-th column of A, with $c_{lj} = l + 2^n$ if there are fewer than l nonzero entries. Additionally, assume we have access to a one-bit oracle $O_A : |i\rangle|j\rangle|0\rangle \mapsto |i\rangle|j\rangle|A_{ij}\rangle$ for $0 \le i, j < 2^n$ and $A_{ij} \in \{0, 1\}$. Then one can implement a $(\sqrt{d_r d_c}, n+3, 0)$-block-encoding of A with a single use of O_r, O_c, two uses of O_A, and additionally $\mathcal{O}(n)$ one-qubit and two-qubit gates, while using $\mathcal{O}(1)$ ancilla qubits, which are discarded before the post-selection step.

The proof of Lemma 2.6.5 follows the same argument as the general result in [Gil+19a,

Lemma 48] and is therefore omitted.

2.6.3 Linear Combination of Unitaries

The Linear Combination of Unitaries (LCU) algorithm [CW12a] provides a framework for implementing linear combinations of unitary operators on a quantum computer. We review the algorithm in detail, as it will be used extensively in Chapter 3, Chapter 4, Chapter 5, and Chapter 6. Let $M \in \mathbb{N}$, and let $a_1, \ldots, a_M \in \mathbb{C}$ and $U_1, \ldots, U_M$ be k-qubit unitary operators. A linear combination of unitary operators need not be unitary. The LCU algorithm aims to implement the operator $U = \sum_{j=1}^{M} a_j U_j$. The algorithm assumes that the following two operators can be implemented. First, it requires access to state preparation (PREP) oracles $\mathrm{PREP}_{\mathrm{R}} |0\rangle = \frac{1}{\sqrt{\|\vec{a}\|_1}} \sum_{j=1}^{M} \sqrt{a_j} |j\rangle$ and $\mathrm{PREP}_{\mathrm{L}} |0\rangle = \frac{1}{\sqrt{\|\vec{a}\|_1}} \sum_{j=1}^{M} \overline{\sqrt{a_j}} |j\rangle$. where $\sqrt{\cdot}$ denotes the principal branch of the square root. Second, it assumes that a multi-qubit controlled gate, U_c, also called a select (SEL) oracle in the literature, $\mathrm{SEL} = \sum_{j=1}^{M} U_j \otimes |j\rangle\langle j|$ can be implemented. Assuming access to these operators, the LCU algorithm implements the unitary operator $U_{\mathrm{LCU}} = (\mathrm{PREP}_{\mathrm{L}}^{\dagger} \otimes I) \cdot \mathrm{SEL} \cdot (\mathrm{PREP}_{\mathrm{R}} \otimes I)$. See Figure 2.6 for the corresponding quantum circuit. If $|\psi\rangle$ denotes a k-qubit input state, $|0\rangle^{\log M}$ represents $\mathcal{O}(\log M)$ ancilla qubits,

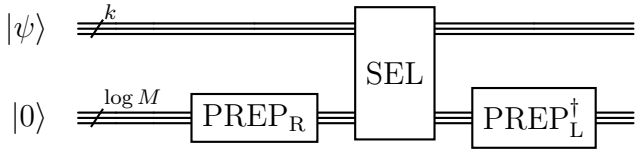


Figure 2.6: Quantum circuit U_{LCU} implementing the LCU algorithm.

and $|\perp\rangle$ denotes a possibly non-normalized state satisfying $(I_{2^k} \otimes |0\rangle^{\log M} \langle 0|^{\log M}) |\perp\rangle = 0$, we

have

$$U_{\mathrm{LCU}} |\psi\rangle |0\rangle^{\log M} = \frac{1}{\|\vec{a}\|_1} \left(\sum_{j=1}^{M} a_j U_j \right) |\psi\rangle |0\rangle^{\log M} + |\perp\rangle . \tag{2.43}$$

Thus, the LCU algorithm implements a $(\|\vec{a}\|_1, \log M, 0)$-block-encoding of $\sum_{j=1}^{M} a_j U_j$.

For a given quantum state $|\psi\rangle$, the primary quantity of interest is the expectation value $\langle\psi| U_{\mathrm{LCU}} |\psi\rangle$. This expectation can be evaluated efficiently using the Hadamard test [Cle+98]. The procedure takes as input the state $|\psi\rangle$ and the operator U_{LCU}, and outputs a random variable obtained from measuring an ancilla qubit. In particular, the real part $\mathrm{Re}\,\langle\psi| U_{\mathrm{LCU}} |\psi\rangle \in \mathbb{R}$ can be obtained by executing the quantum circuit illustrated in Figure 2.7. Moreover, we have

$$\langle 0| \langle\psi| U_{\mathrm{LCU}} |\psi\rangle |0\rangle = \frac{1}{\|\vec{a}\|_1} \langle\psi| \left(\sum_{j=1}^{M} a_j U_j \right) |\psi\rangle . \tag{2.44}$$

A minor modification of the Hadamard test can similarly estimate the imaginary part, $\mathrm{Im}\,\langle\psi| U_{\mathrm{LCU}} |\psi\rangle$.

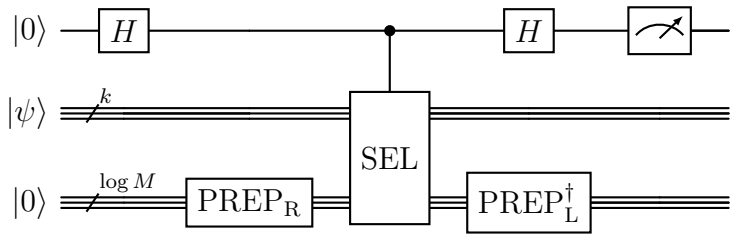


Figure 2.7: Quantum circuit implementing the Hadamard test with the LCU operator. The measurement of the ancilla qubit yields information about $\langle\psi| U_{\mathrm{LCU}} |\psi\rangle$.

Note that Equation (2.44) implies that the success probability scales as $O(1/\|\vec{a}\|_1^2)$. In particular, if $\|\vec{a}\|_1 = \Omega(c^n)$ for some $c > 1$, then the success probability decays exponentially. One can use amplitude amplification [Ber+15a] to boost the success probability to $1 - \mathcal{O}(\delta)$ for any $\delta > 0$. This requires $\mathcal{O}(\|\vec{a}\|_1/\delta)$ rounds of amplitude amplification.

2.6.4 Quantum Fourier Transform

Let $N \in \mathbb{N}$. The Discrete Fourier Transform (DFT) is the linear map $\mathrm{DFT}_N : \mathbb{C}^N \to \mathbb{C}^N$ whose (j,k)-entry is $(\mathrm{DFT}_N)_{jk} = \frac{1}{\sqrt{N}}\omega_N^{jk}$, where $\omega_N = e^{-2\pi i/N}$ is a primitive N-th root of unity and $0 \leq j,k \leq N-1$. In matrix notation, the DFT can be represented as follows:

$$\begin{pmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{pmatrix} = \frac{1}{\sqrt{N}} \begin{pmatrix} \omega_N^{0\cdot 0} & \omega_N^{0\cdot 1} & \cdots & \omega_N^{0\cdot N-1} \\ \omega_N^{1\cdot 0} & \omega_N^{1\cdot 1} & \cdots & \omega_N^{1\cdot N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \omega_N^{N-1\cdot 0} & \omega_N^{N-1\cdot 1} & \cdots & \omega_N^{N-1\cdot N-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix} \tag{2.45}$$

The Quantum Fourier Transform (QFT) is the quantum analogue of the Discrete Fourier Transform and implements DFT_N in Equation (2.45) as a quantum circuit, denoted by U_{QFT}. When $N = 2^n$, the algorithm realizes U_{QFT} through a sequence of single-qubit and two-qubit gates, based on the decomposition

$$U_{\mathrm{QFT}} |x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{-2\pi i xy/N} |y\rangle = \frac{1}{\sqrt{N}} \bigotimes_{\ell=0}^{n-1} \left(|0\rangle + e^{-2\pi i \sum_{m=1}^{n-\ell} x_{n-\ell-m}/2^m} |1\rangle \right). \tag{2.46}$$

The tensor-product form in Equation (2.46) is useful because it reveals a hierarchical dependence: the last output qubit depends on all n input qubits, whereas earlier output qubits depend on progressively fewer input qubits. This structure leads to a simple quantum circuit implementation, as detailed in Algorithm 1. In each of the n iterations in Algorithm 1, a Hadamard gate is applied to the current qubit, followed by controlled phase rotations acting on progressively

fewer qubits. This gives a total of $n + (n-1) + \cdots + 1 = \sum_{k=1}^{n} k = \frac{n(n+1)}{2} = \mathcal{O}(n^2)$ quantum gates. If $N = 2^n$, the QFT has computational complexity $\mathcal{O}(\log^2 N) = \mathcal{O}(n^2)$, in contrast to the classical FFT, which requires $\mathcal{O}(N \log N) = \mathcal{O}(n2^n)$ operations.

Algorithm 1: Quantum Fourier Transform (QFT) on an n-qubit register

Input: n-qubit quantum state $|x\rangle = |x_0 x_1 \cdots x_{n-1}\rangle$
Output: QFT-transformed quantum state $U_{\text{QFT}} |x\rangle$
for $\ell \leftarrow 0$ **to** $n-1$ **do**
    Apply Hadamard gate H to qubit $n-\ell-1$
    for $m \leftarrow 2$ **to** $n-\ell$ **do**
        Apply CR_m (Equation (2.37)) with qubit $n-\ell-m$ (control) and qubit $n-\ell-1$ (target)
return QFT transformed quantum state $U_{\text{QFT}} |x\rangle$

We will refer to the quantum Fourier transform algorithm in Chapter 3 and use it in Chapter 6.

2.6.5 Quantum Signal Processing

Quantum signal processing (QSP) is a framework for implementing polynomial transformations of scalars block-encoded in a unitary operator. In particular, QSP constructs 2×2 unitary matrices whose entries are complex polynomials of a real scalar $x \in [-1, 1]$, using compositions of single-qubit rotations and phase gates. It starts with a single-qubit unitary operator that block-encodes $x \in [-1, 1]$:

$$W(x) := \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} = e^{i \arccos(x) \sigma_X}. \tag{2.47}$$

QSP implements $V_{\vec{\varphi}}(x) = e^{i\varphi_0 \sigma_Z} W(x) e^{i\varphi_1 \sigma_Z} W(x) \cdots W(x) e^{i\varphi_\ell \sigma_Z}$ for $\vec{\varphi} = (\varphi_0, \ldots, \varphi_\ell)$. See Figure 2.8 for the corresponding quantum circuit. Polynomial functions realizable by such a quan-

tum circuit are characterized by the following result:

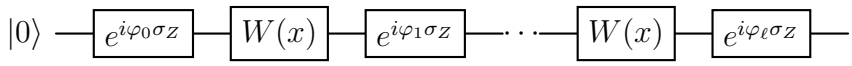


Figure 2.8: Quantum circuit representation of $V_{\vec{\varphi}}(x)$ illustrating the sequence of phase rotations and signal operators used in quantum signal processing.

Proposition 2.6.6 (Theorem 3 in [Gil+19a]). *Let $\ell \in \mathbb{N}$ and $x \in [-1, 1]$. There exists $\vec{\varphi} = (\varphi_0, \varphi_1, \ldots, \varphi_\ell) \in \mathbb{R}^{\ell+1}$ such that*

$$V_{\vec{\varphi}}(x) = e^{i\varphi_0\sigma_Z} \prod_{j=1}^{\ell} \left(W(x) e^{i\varphi_j\sigma_Z} \right) = \begin{pmatrix} p(x) & iq(x)\sqrt{1-x^2} \\ i\overline{q(x)}\sqrt{1-x^2} & \overline{p(x)} \end{pmatrix}. \tag{2.48}$$

if and only if $p, q \in \mathbb{C}[x]$ satisfy:

1. $\deg(p(x)) \leq \ell$ *and* $\deg(q(x)) \leq \ell - 1$,
2. $p(x)$ *has parity* $\ell \mod 2$ *and* $q(x)$ *has parity* $\ell - 1 \mod 2$,
3. *For all* $x \in [-1, 1]$, *we have* $|p(x)|^2 + (1 - x^2)|q(x)|^2 = 1$.

A polynomial has parity zero (respectively, one) if it is even (respectively, odd). By Proposition 2.6.6, we have $\langle 0| V_{\vec{\varphi}}(x) |0\rangle = p(x)$. Thus, the $(0, 0)$-matrix element of the QSP circuit realizes the polynomial $p(x)$. We will reference the quantum signal processing algorithm in Chapter 3 and Chapter 4, and use it in Chapter 5 and Chapter 6.

Part I

Quantum Simulation via Multi-Product Formulas

Chapter 3: Multi-Product Formula with Commutator Scaling

The results presented in this chapter are based on joint work with Dong An and Konstantina Trivisa [AAT24] and are included here with their permission. Junaid Aftab and Dong An contributed equally to all aspects of the work. Konstantina Trivisa served as our mentor. The code generated for parts of this chapter is available in [Aft25].

3.1 Introduction

Recall from Section 1.3.1 that the objective of time-independent Hamiltonian simulation is to prepare a quantum state $|u(T)\rangle$ evolving according to the time-independent Schrödinger equation

$$i\frac{d}{dt}u(t) = Hu(t), \quad t \in [0, T], \tag{3.1}$$

where $H \in \mathbb{C}^{N\times N}$ denotes the Hamiltonian of a finite-dimensional quantum system. The solution to Equation (3.1) is given explicitly by $|u(T)\rangle = e^{-iTH}|u(0)\rangle$. Accordingly, the central task in Hamiltonian simulation is to implement the unitary time-evolution operator e^{-iTH} efficiently. In what follows, we assume that H admits a decomposition $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$.

Product formulas assume that the time evolution generated by each term H_γ, namely $U_\gamma(t) = e^{-itH_\gamma}$ for $t \in [0, T]$, can be implemented efficiently on a quantum computer. This approach uses Lie–Trotter product formulas to recursively construct p-th-order approximations

to the full time-evolution operator e^{-iHT}. Substantial progress has been made on quantum algorithms based on product formulas, together with rigorous complexity analyses [Ber+07; Chi+18; CS19b; Chi+21; ŞS21; SHC21; AFL21b; YC22; Mor+22; Low+23]. Notably, Childs et al. [Chi+21] gave a rigorous error analysis of product formulas and showed that the gate complexity of implementing a p-th-order formula scales as $\mathcal{O}\left(\frac{\alpha_{\mathrm{comm},p+1}^{1/p} T^{1+1/p}}{\epsilon^{1/p}}\right)$, where $\alpha_{\mathrm{comm},p+1} = \sum_{j_1,\ldots,j_{p+1}=1}^{\Gamma} \left\|[H_{j_1},\ldots,H_{j_{p+1}}]\right\|$. Here $\alpha_{\mathrm{comm},p+1}$ denotes a sum of norms of nested commutators of length $p+1$ involving the terms H_γ, and ϵ is the target precision. The dependence on $\alpha_{\mathrm{comm},p+1}$ is commonly referred to as *commutator scaling*, reflecting the algebraic structure of the Hamiltonian terms. This property allows product-formula Hamiltonian simulation algorithms to achieve favorable, and in some cases optimal, system-size scaling for physically local Hamiltonians. Despite these advantages, product-formula methods have a fundamental limitation in their dependence on the evolution time T and target precision ϵ. The scaling is superlinear in T and superlogarithmic in $1/\epsilon$ for fixed p. Moreover, a p-th-order formula incurs substantial overhead, requiring approximately 5^p terms to approximate e^{-iTH}.

Recent work has introduced quantum algorithms that go beyond the traditional product-formula approach [BC12; CW12b; BCG14; Ber+15b; BCK15; LC17b; LC19; Gil+19b; COS19; Cam19; Low19; Che+20; Zen+25]. Prominent examples include qubitization-based algorithms developed by Low and Chuang [LC19] and quantum singular value transformation (QSVT), introduced by Gilyén et al. [Gil+19b]. These methods achieve query complexity $\mathcal{O}\left(\|H\|T + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right)$ with respect to black-box access to the Hamiltonian. This dependence on both T and ϵ improves upon that of product-formula methods of any fixed order and is asymptotically optimal, saturating known lower bounds in both parameters [Ber+14]. However, their complexity typically scales at least linearly with the spectral norm $\|H\|$ and does not exploit commutator scaling. Conse-

quently, in many physically relevant settings, these methods can have less favorable system-size dependence than product-formula methods.

The discussion above highlights a fundamental trade-off between favorable system-size dependence and favorable dependence on evolution time and precision, which must be considered when selecting an algorithm for simulating a given quantum system. Low, Kliuchnikov, and Wiebe [LKW19] proposed a Hamiltonian simulation algorithm based on the multi-product formula (MPF) that aims to combine commutator scaling with near-optimal dependence on time and precision. The MPF approximates the ideal evolution operator by a linear combination of product formulas, $e^{-iHt} \approx \sum_{j=1}^{M} a_j U_p(t/k_j)^{k_j}$, where $U_p(s)$ denotes a p-th-order product formula with time step s, referred to as the base sequence of the MPF. The order p of the base sequence is typically low, such as $p = 2$ or $p = 4$, to avoid the exponential overhead associated with high-order product formulas. They showed that implementing the MPF using the linear combination of unitaries (LCU) subroutine [CW12b] has complexity $\mathcal{O}((\sum_\gamma \|H_\gamma\|)T \operatorname{polylog}(T/\epsilon))$, which is near-optimal in both T and ϵ. Moreover, when T and ϵ are treated as constants, they showed that the MPF complexity depends on Hamiltonian terms appearing in nested commutators, thereby preserving commutator scaling.

However, [LKW19] does not provide a comprehensive complexity analysis of the MPF. First, it does not give a uniform complexity estimate that simultaneously captures commutator scaling and near-optimal dependence on time and precision. Specifically, [LKW19] presents two separate complexity estimates. The first achieves near-optimal dependence on time and precision but contains a Hamiltonian-dependent factor of order $\mathcal{O}(\sum_\gamma \|H_\gamma\|)$. The second accounts for commutator dependence but does not incorporate the dependence on time and precision. Second, the commutator dependence is only qualitative, and no rigorous MPF error bound with

an explicit expression for the nested-commutator contribution is provided. Consequently, the system-size dependence remains ambiguous when the MPF is applied to specific systems of practical relevance.

In this chapter, we address these shortcomings by establishing an improved complexity estimate for the MPF based on the second-order product formula, which we refer to as the second-order-based MPF, or 2nd-based MPF. Our analysis rigorously shows that the 2nd-based MPF simultaneously achieves explicit commutator scaling and near-optimal dependence on evolution time and target precision. Using this refined complexity estimate, we present several applications of the MPF. We further investigate the commutator-scaling behavior of MPFs constructed from higher-order product formulas and show that increasing the base order can yield additional improvements in system-size dependence.

3.1.1 Related Works

This chapter focuses on the complexity analysis of the MPF under its standard quantum implementation, namely the use of the linear combination of unitaries (LCU) algorithm to realize the linear combination of product formulas. In particular, we focus on MPF-based algorithms for fault-tolerant quantum computers. We first review related work outside this setting, including approaches designed for near-term or hybrid quantum–classical regimes. We then summarize and critically assess recent studies that provide error analyses for the MPF.

A common variant of the MPF methodology performs the linear-combination step classically, particularly when the goal is to estimate expectation values of observables. For example, [RWW24] proposes a polynomial-interpolation-based quantum algorithm for Hamiltonian simu-

lation. This approach improves accuracy by collecting simulation results from low-order product formulas at multiple time-step sizes and then applying polynomial interpolation to extrapolate the ideal zero-time-step limit. The interpolation uses Chebyshev nodes and Chebyshev polynomials, which provide favorable numerical stability. The primary motivation is to overcome inherent limitations of low-order Trotter formulas, such as their low convergence order, thereby closely aligning with the objectives of MPF-based algorithms.

Another common variant targets near-term and early fault-tolerant settings by modifying the implementation of the linear-combination step. In particular, [Fae+22] introduces a randomized MPF method for estimating observables based on importance sampling. In this approach, the MPF power parameters are sampled with probabilities proportional to their associated coefficients. For each sampled product-formula power, the corresponding observable is estimated independently on a quantum device, and the final estimate is obtained by classically averaging the measurement outcomes. By avoiding oblivious amplitude amplification, which is required in the standard LCU subroutine, the randomized MPF reduces circuit depth while achieving comparable error scaling, at the cost of additional classical repetitions. A complementary variant targets noisy hardware through a hybrid quantum–classical realization of the MPF. Specifically, [Vaz+23] proposes an implementation that avoids additional ancilla qubits and controlled operations, in contrast to fully quantum MPF constructions based on the LCU algorithm.

A related but distinct methodology is presented in [WW25], which uses Richardson extrapolation to mitigate several unfavorable features of low-order product-formula Hamiltonian simulation algorithms. This work follows a broader trend of using Richardson-extrapolation techniques to suppress both algorithmic and hardware-induced errors in Hamiltonian simula-

tion, as explored in prior studies such as [End+19].

Three recent works [Wat+24; ZRB24; WW25] attempt to provide error analyses of MPFs exhibiting commutator scaling. However, as we explain below, these studies do not fully address the challenges considered in this chapter. The work in [Wat+24] generalizes the MPF to time-dependent Hamiltonian simulation and shows that both time-independent and time-dependent MPFs exhibit commutator scaling. Nevertheless, the resulting commutator scaling remains implicit, reflecting a limitation similar to that of the original well-conditioned MPF formulation in [LKW19]. The analysis in [ZRB24] focuses on commutator error scaling for the hybrid quantum–classical MPF algorithm proposed in [Vaz+23]. Their results are restricted to the regime in which the number of terms M in the MPF sum satisfies $M \leq p + 1$ for a p-th-order base product formula, yielding only quadratic accuracy improvements over the base formula. Consequently, their analysis does not capture the rapid convergence properties of the well-conditioned MPF introduced in [LKW19]. The recent work [WW25] establishes commutator error scaling for the Richardson-extrapolation-based algorithm of [End+19], which is designed to mitigate algorithmic errors in product formulas. Similar to [ZRB24], this work considers hybrid implementations that produce classical observable estimates rather than fully quantum state preparation. This chapter addresses these gaps by presenting an MPF error analysis that explicitly leverages commutator scaling in the fault-tolerant setting.

3.1.2 Organization

The remainder of this chapter is organized as follows. Section 3.2 reviews the multi-product formula (MPF). Section 3.3 presents and proves our main result that the second-order-based

MPF admits an explicit commutator scaling. Section 3.4 discusses applications, and Section 3.5 extends the analysis to multi-product formulas based on higher-order product formulas. Section 3.6 provides concluding remarks and discusses future directions motivated by this work.

3.2 Multi-Product Formula

The approach of approximating high-order time evolution by a linear combination of low-order product formulas is known in the numerical analysis literature as a multi-product formula. For a short time step Δ, an MPF takes the form $U_{\mathrm{MP}}(\Delta) = \sum_{j=1}^{m} a_j U_p(\Delta/k_j)^{k_j}$, where $a_j \in \mathbb{R}$ and $k_j \in \mathbb{N}$. Since the MPF is constructed as a linear combination of repeated applications of a p-th-order product formula, we refer to it as a p-th-order-based MPF. Although an MPF is constructed from a fixed-order product formula, it can achieve arbitrarily high convergence order. The key idea is that the errors introduced by the individual terms in the linear combination can be approximately cancelled through a suitable choice of the coefficients a_j and powers k_j. In particular, Low, Kliuchnikov, and Wiebe exploit this idea in [LKW19] and, for any integer m, propose a linear combination of second-order product formulas of the form

$$U_{\mathrm{MP}}(\Delta) = \sum_{j=1}^{m} a_j U_2(\Delta/k_j)^{k_j} = e^{-iH\Delta} + \mathcal{O}(\Delta^{2m+1}), \quad m \in \mathbb{N}. \tag{3.2}$$

The parameters a_j and k_j are chosen to satisfy both the order conditions and the requirement that the local truncation error be of order $2m+1$. The analysis in [LKW19] is based on the observation that any symmetric product formula admits a formal BCH expansion [BCR00] of the form $U_2(\Delta) = e^{-iH\Delta + E_3\Delta^3 + E_5\Delta^5 + \cdots}$, where the operators E_ℓ depend on the Hamiltonian terms H_γ. We note that [LKW19] does not address convergence details for this formal BCH

expansion. A further formal Taylor expansion then implies that $U_2^{k_j}(\Delta/k_j)$ can be written as

$$U_2^{k_j}(\Delta/k_j) = e^{-iH\Delta} + \frac{\Delta^3}{k_j^2}\widetilde{E}_3(\Delta) + \frac{\Delta^5}{k_j^4}\widetilde{E}_5(\Delta) + \cdots . \tag{3.3}$$

Here the $\widetilde{E}_\ell(\Delta)$ are operators whose explicit forms are not specified in [LKW19], but which are independent of k_j and uniformly bounded in Δ. Thus, satisfying the order conditions requires cancellation of all lower-order terms, so the parameters must solve the linear system

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ k_1^{-2} & k_2^{-2} & \cdots & k_m^{-2} \\ \vdots & \vdots & \ddots & \vdots \\ k_1^{-2m+2} & k_2^{-2m+2} & \cdots & k_m^{-2m+2} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.4}$$

It is shown in [LKW19] that there exists a solution of Equation (3.4) such that

$$\|\vec{k}\|_1 = \mathcal{O}(m^2 \log m), \quad \|\vec{a}\|_1 = \mathcal{O}(\log m). \tag{3.5}$$

Hence, the cost of implementing Equation (3.2) via the linear combination of unitaries algorithm scales only polynomially in the convergence order. We review this implementation in Section 3.2.1.

Intuitively, the MPF should achieve high-order convergence while preserving commutator scaling, since the order conditions in Equation (3.4) cancel all lower-order terms and the MPF is exact when the Hamiltonian terms H_γ mutually commute. Indeed, [LKW19] proves two results that separately establish these two features.

Lemma 3.2.1 (Theorem 2 in [LKW19]). *Time evolution operator of a time-independent Hamiltonian simulation problem can be approximated by the 2nd-based MPF with error at most ϵ and probability at least $1 - \Omega(\epsilon)$, using $O(T\lambda \log^2(T\lambda/\epsilon))$ controlled-U_2 queries, where $\lambda = \sum_{\gamma=1}^{\Gamma} \|H_\gamma\|$.*

Lemma 3.2.2 (Theorem 3 in [LKW19]). *The error of a p-th-based MPF depends on Hamiltonian terms that all occur in commutators $[H_{\gamma_q}, [\ldots, [H_{\gamma_2}, H_{\gamma_1}]\ldots]]$ nested to depth $q > p$.*

While Lemma 3.2.1 establishes high-order convergence of the MPF with a low-order base sequence, its Hamiltonian dependence is expressed in terms of the 1-norm of the Hamiltonian. Commutator scaling is addressed separately in Lemma 3.2.2. However, it result does not provide an explicit, computable expression for this dependence. Thus, prior to our work, no unified result simultaneously established explicit commutator scaling and high-order convergence, leaving the practical advantages of the MPF somewhat ambiguous.

3.2.1 Quantum Implementation of Multi-Product Formula

A quantum algorithm for Hamiltonian simulation based on the MPF can be constructed using the linear combination of unitaries (LCU) algorithm discussed in Section 2.6.3. Suppose that we have access to the state-preparation oracles defined in Section 2.6.3. For the product formulas, we assume access to a select oracle SEL such that $\mathrm{SEL} = \sum_{j=1}^{m} U_p(\Delta/k_j)^{k_j} \otimes |j\rangle\langle j|$. According to the discussion in Section 2.6.3, for any input state $|\psi\rangle$, we have

$$(\mathrm{PREP}_{\mathrm{L}}^{\dagger} \otimes I) \cdot \mathrm{SEL} \cdot (\mathrm{PREP}_{\mathrm{R}} \otimes I) |\psi\rangle |0\rangle = \frac{1}{\|\vec{a}\|_1} \left(\sum_{j=1}^{M} a_j U_p(\Delta/k_j)^{k_j} \right) |\psi\rangle |0\rangle + |\perp\rangle , \tag{3.6}$$

where $|\perp\rangle$ denotes a possibly unnormalized state such that $(I \otimes |0\rangle\langle 0|)|\perp\rangle = 0$. The component associated with the ancilla state $|0\rangle$ encodes the desired MPF. However, direct post-selection on the ancilla state $|0\rangle$ can have success probability as small as $\mathcal{O}(1/\|\vec{a}\|_1^2)$. For long-time simulation up to time T, we divide the interval $[0, T]$ into r equally spaced segments. On each segment, we implement the MPF via LCU for time $\Delta = T/r$, as described above. To prevent the success probability from decaying exponentially, each step uses robust oblivious amplitude amplification [Ber+15b] to boost the stepwise success probability to $1 - \mathcal{O}(\epsilon/r)$. The overall success probability after r steps is then $1 - \mathcal{O}(\epsilon)$. The cost of implementing the MPF is determined by two main factors.

1. Each LCU implementation before post-selection requires one query to each of the oracles $\mathrm{PREP}_{\mathrm{L}}^{\dagger}$, $\mathrm{PREP}_{\mathrm{R}}$, and SEL.

2. To boost the success probability, we require $\mathcal{O}(\|\vec{a}\|_1)$ rounds of robust oblivious amplitude amplification, each of which uses $\mathcal{O}(1)$ LCU circuits.

The oracles $\mathrm{PREP}_{\mathrm{L}}^{\dagger}$ and $\mathrm{PREP}_{\mathrm{R}}$ can be constructed efficiently using $\mathcal{O}(m)$ gates. This cost is not dominant, since it is independent of the Hamiltonian dimension and m is small for the well-conditioned MPF. Thus, the computational complexity is dominated by the oracle U_c, which requires $\mathcal{O}(\|\vec{k}\|_1)$ queries to controlled U_2. To boost the success probability, $\mathcal{O}(\|\vec{a}\|_1)$ rounds of robust oblivious amplitude amplification are performed. Consequently, each time step requires $\mathcal{O}(\|\vec{a}\|_1\|\vec{k}\|_1)$ queries to U_2. Repeating this over r steps gives a total complexity of $\mathcal{O}(r\|\vec{a}\|_1\|\vec{k}\|_1)$. Although $\|\vec{a}\|_1$ and $\|\vec{k}\|_1$ are computed in Equation (3.5), a rigorous complexity analysis of the MPF that explicitly determines r was previously unavailable. We address this gap in Section 3.3.

3.3 Multi-Product Formula Based on Second-Order Product Formula

In this section, we establish a complexity analysis for the 2nd-based MPF. We begin by applying the multi-term Baker–Campbell–Hausdorff (BCH) formula to express U_2 as the exponential of a single Hamiltonian. To rigorously justify this step, we identify a sufficient convergence condition for the BCH formula under commutator scaling. We then combine the BCH formula with a high-order variation-of-parameters formula to derive a commutator-scaling error bound for the corresponding MPF and estimate its overall computational complexity.

3.3.1 BCH Formula and Convergence

We begin with the general form of the Baker–Campbell–Hausdorff (BCH) formula. Let $X_1, \ldots, X_\Gamma$ be arbitrary matrices. Following [ACC20], we write $e^{X_1}e^{X_2}\cdots e^{X_\Gamma} = e^Z$ for some matrix Z. The formal expansion of Z is given by

$$Z = \sum_{k=1}^{\infty} \sum_{\substack{i_1+\cdots+i_\Gamma=k,\\ i_j\geq 0}} \frac{1}{i_1!\cdots i_\Gamma!}\phi_k(\underbrace{X_1,\cdots,X_1}_{i_1},\cdots,\underbrace{X_\Gamma,\cdots,X_\Gamma}_{i_\Gamma}) \tag{3.7}$$

$$= X_1+\cdots+X_\Gamma+\sum_{k=2}^{\infty} \sum_{\substack{i_1+\cdots+i_\Gamma=k,\\ i_j\geq 0}} \frac{1}{i_1!\cdots i_\Gamma!}\phi_k(\underbrace{X_1,\cdots,X_1}_{i_1},\cdots,\underbrace{X_\Gamma,\cdots,X_\Gamma}_{i_\Gamma}), \tag{3.8}$$

where, for matrices $Y_1,\cdots,Y_k$,

$$\phi_k(Y_1,\cdots,Y_k) = \frac{1}{k^2}\sum_{\sigma\in S_k}(-1)^{d_\sigma}\frac{1}{\binom{k-1}{d_\sigma}}[Y_{\sigma(1)}Y_{\sigma(2)}\cdots Y_{\sigma(k)}]. \tag{3.9}$$

Note that Equation (3.8) is a formal expansion and need not converge in general. However, the series converges when nested commutators of increasing depth do not grow too rapidly. Moreover, since this chapter focuses on symmetric product formulas as base sequences, in particular the second-order Trotter–Suzuki formula, the even-order terms in the BCH expansion vanish. We summarize these two features in the following result, which provides a rigorous convergence guarantee.

Proposition 3.3.1. *Let $X_1, \cdots, X_\Gamma$ be matrices. Suppose there exists a $J \geq 1$ such that for $k \geq J$, we have $\sum_{\gamma_1,\cdots,\gamma_k=1}^{\Gamma} \|[X_{\gamma_1} X_{\gamma_2} \cdots X_{\gamma_k}]\| \leq 1$. Then we have $e^{X_1/2} e^{X_2/2} \cdots e^{X_\Gamma/2} e^{X_\Gamma/2} \cdots e^{X_2/2} e^{X_1/2} = e^Z$ where $Z = X_1 + \cdots + X_\Gamma + \sum_{k \in 1+2\mathbb{Z}^+} \Phi_k$ and*

$$\Phi_k = \sum_{\substack{\sum i_j + i'_j = k, \\ i_j, i'_j \geq 0}} \prod_{i_j, i'_j = 1}^{\Gamma} \frac{1}{i_j! i'_j!} \phi_k(\underbrace{X_1/2, \cdots, X_1/2}_{i_1}, \cdots, \underbrace{X_\Gamma/2, \cdots, X_\Gamma/2}_{i_\Gamma}, \underbrace{X_\Gamma/2, \cdots, X_\Gamma/2}_{i'_\Gamma}, \cdots, \underbrace{X_1/2, \cdots, X_1/2}_{i'_1}).$$

Furthermore, we have $\|\Phi_k\| \leq \frac{1}{k^2} \sum_{\gamma_1,\cdots,\gamma_k=1}^{\Gamma} \|[X_{\gamma_1} X_{\gamma_2} \cdots X_{\gamma_k}]\|$.

Proof. For notational simplicity, define $X_{\Gamma+j} = X_{\Gamma+1-j}$ and $i_{\Gamma+j} = i'_{\Gamma+1-j}$ for $j \geq 1$. We also write $\widetilde{X}_j = X_j/2$. By the standard BCH formula in Equation (3.8), we formally have

$$Z = \widetilde{X_1} + \cdots + \widetilde{X_\Gamma} + \sum_{k \geq 2} \Phi_k, \tag{3.10}$$

where

$$\Phi_k = \sum_{\substack{\sum i_j = k, \\ i_j \geq 0}} \frac{1}{i_1! \cdots i_{2\Gamma}!} \phi_k(\underbrace{\widetilde{X}_1, \cdots, \widetilde{X}_1}_{i_1}, \cdots, \underbrace{\widetilde{X}_{2\Gamma}, \cdots, \widetilde{X}_{2\Gamma}}_{i_{2\Gamma}}). \tag{3.11}$$

All terms in Equation (3.11) are nested commutators of depth k, according to the definition of ϕ_k in Equation (3.9). For fixed k and a tuple $(\gamma_1, \ldots, \gamma_k)$, where $\gamma_i \in \{1, \ldots, 2\Gamma\}$, we count

the number of occurrences of the commutator $[\widetilde{X}_{\gamma_1}\widetilde{X}_{\gamma_2}\cdots\widetilde{X}_{\gamma_k}]$ in Φ_k.[1] First, there is a unique summand in Equation (3.11) with index $(i_1,\ldots,i_{2\Gamma})$ that contains the desired commutator, since the multiplicities of the matrices are uniquely determined by $(\gamma_1,\ldots,\gamma_k)$. In the corresponding term ϕ_k, the number of admissible permutations is $i_1!\cdots i_{2\Gamma}!$. Denote this set of permutations by $\widetilde{S}$, so that $|\widetilde{S}| = i_1!\cdots i_{2\Gamma}!$. Consequently, after combining terms with the same word $(\gamma_1,\ldots,\gamma_k)$, the absolute value of its coefficient is bounded by $\frac{1}{i_1!\cdots i_{2\Gamma}!}\frac{1}{k^2}\sum_{\sigma\in\widetilde{S}}\frac{1}{\binom{k-1}{d_\sigma}} \leq \frac{1}{k^2}$. Therefore, we have

$$\|\Phi_k\| \leq \frac{1}{k^2}\sum_{\gamma_1,\cdots,\gamma_k=1}^{2\Gamma}\|[\widetilde{X}_{\gamma_1}\widetilde{X}_{\gamma_2}\cdots\widetilde{X}_{\gamma_k}]\| = \frac{1}{k^2 2^k}\sum_{\gamma_1,\cdots,\gamma_k=1}^{2\Gamma}\|[X_{\gamma_1}X_{\gamma_2}\cdots X_{\gamma_k}]\|. \tag{3.12}$$

$X_{\Gamma+j} = X_{\Gamma+1-j}$ implies that $\sum_{\gamma_1,\ldots,\gamma_k=1}^{2\Gamma}\|[X_{\gamma_1}X_{\gamma_2}\cdots X_{\gamma_k}]\| = 2^k\sum_{\gamma_1,\ldots,\gamma_k=1}^{\Gamma}\|[X_{\gamma_1}X_{\gamma_2}\cdots X_{\gamma_k}]\|$. Thus, we have

$$\|\Phi_k\| \leq \frac{1}{k^2}\sum_{\gamma_1,\cdots,\gamma_k=1}^{\Gamma}\|[X_{\gamma_1}X_{\gamma_2}\cdots X_{\gamma_k}]\|. \tag{3.13}$$

For sufficiently large k, the assumption $\sum_{\gamma_1,\ldots,\gamma_k=1}^{\Gamma}\|[X_{\gamma_1}X_{\gamma_2}\cdots X_{\gamma_k}]\| \leq 1$ implies that $\|\Phi_k\| \leq 1/k^2$, which establishes absolute convergence. We have shown that Equation (3.10) holds rigorously. Compared with the desired formula $Z = X_1 + \cdots + X_\Gamma + \sum_{k\in 1+2\mathbb{Z}^+}\Phi_k$, it remains to show that $\Phi_k = 0$ whenever k is even. To this end, we use the symmetry of the product formula considered here by introducing a parameter s with $|s| \leq 1$. Let

$$O(s) = e^{sX_1/2}e^{sX_2/2}\cdots e^{sX_\Gamma/2}e^{sX_\Gamma/2}\cdots e^{sX_2/2}e^{sX_1/2}. \tag{3.14}$$

Then, according to Equation (3.10), we have $O(s) = e^{\sum_{k\geq 1}s^k\Phi_k}$ and $O(-s) = e^{\sum_{k\geq 1}(-s)^k\Phi_k}$. By

[1] The counting here is symbolic: we regard $\widetilde{X}_{\Gamma+j}$ and $\widetilde{X}_{\Gamma+1-j}$ as distinct operators, although they coincide in the special case of the second-order Trotter–Suzuki formula. This convention simplifies the counting while preserving technical correctness, since the argument applies to the more general case of 2Γ possibly distinct matrices.

definition, $O(s)O(-s) = O(-s)O(s) = I$. Hence the exponentials $e^{\sum_{k\geq 1} s^k \Phi_k}$ and $e^{\sum_{k\geq 1}(-s)^k \Phi_k}$ commute. It follows that, for sufficiently small $|s|$, the matrices $\sum_{k\geq 1} s^k \Phi_k$ and $\sum_{k\geq 1}(-s)^k \Phi_k$ also commute [Wer97]. Therefore, we have

$$I = O(s)O(-s) = e^{(\sum_{k\geq 1} s^k \Phi_k)+(\sum_{k\geq 1}(-s)^k \Phi_k)} = e^{\sum_{k\in 2\mathbb{Z}^+} 2s^k \Phi_k}, \tag{3.15}$$

which implies that $\Phi_k = 0$ for all even k. This completes the proof. □

3.3.2 Representation of Powers of the Second-Order Product Formula

We now derive a polynomial expansion of $\big(U_2(\Delta/k)\big)^k$ in powers of $1/k$. We first apply the Baker–Campbell–Hausdorff (BCH) formula to express the product formula as the exponential of a single matrix, which facilitates the computation of its powers. We then expand these powers using a high-order variation-of-parameters formula, which can be viewed as the analogue of the truncated Dyson series in the interaction picture. For completeness, we state this formula in Lemma 3.3.2 and provide its proof.

Lemma 3.3.2. *Let A and B be two anti-Hermitian matrices. Then for any positive integer p, we have $e^{A+B} - e^A$ is given by*

$$\sum_{l=1}^{p-1} \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{l-1}} e^{A(1-s_1)} B e^{A(s_1-s_2)} B \cdots e^{A(s_{l-1}-s_l)} B e^{As_l} ds_l \cdots ds_2 ds_1 + R_p(A,B), \tag{3.16}$$

where $R_p(A,B)$ is an operator depending on A, B and p such that $\|R_p(A,B)\| \leq \frac{\|B\|^p}{p!}$.

Proof. The variation-of-parameters formula for two matrices A and B states that

$$e^{(A+B)t} = e^{At} + \int_0^t e^{A(t-s)} B e^{(A+B)s} ds. \tag{3.17}$$

This formula can be derived as follows. Let $V_1(t) = e^{At}$ and $V_2(t) = e^{(A+B)t}$. Then V_1 and V_2 solve the ODEs

$$\frac{dV_1(t)}{dt} = AV_1(t), \quad V_1(0) = I, \tag{3.18}$$

$$\frac{dV_2(t)}{dt} = AV_2(t) + BV_2(t), \quad V_2(0) = I. \tag{3.19}$$

We regard V_2 as a perturbation of V_1, with perturbation term $BV_2(t)$. The variation-of-parameters formula then gives $V_2(t) = V_1(t) + \int_0^t e^{A(t-s)} BV_2(s) ds$, which is precisely Equation (3.17). To derive a high-order formula, we apply Equation (3.17) with $t = 1$, obtaining $e^{A+B} = e^A + \int_0^t e^{A(1-s)} B e^{(A+B)s} ds$. Applying Equation (3.17) again to replace $e^{(A+B)s}$ in the preceding equation yields

$$e^{A+B} = e^A + \int_0^1 e^{A(1-s_1)} B e^{As_1} ds_1 + \int_0^1 \int_0^{s_1} e^{A(1-s_1)} B e^{A(s_1-s_2)} B e^{(A+B)s_2} ds_2 ds_1. \tag{3.20}$$

Repeating this substitution once more gives

$$e^{(A+B)} = e^A + \int_0^1 e^{A(1-s_1)} B e^{As_1} ds_1 + \int_0^1 \int_0^{s_1} e^{A(1-s_1)} B e^{A(s_1-s_2)} B e^{As_2} ds_2 ds_1 \tag{3.21}$$

$$+ \int_0^1 \int_0^{s_1} \int_0^{s_2} e^{A(1-s_1)} B e^{A(s_1-s_2)} B e^{A(s_2-s_3)} B e^{(A+B)s_3} ds_3 ds_2 ds_1. \tag{3.22}$$

Iterating this formula p times yields

$$e^{(A+B)} = e^A + \sum_{l=1}^{p-1} \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{l-1}} e^{A(1-s_1)} B e^{A(s_1-s_2)} B \cdots e^{A(s_{l-1}-s_l)} B e^{As_l} ds_l \cdots ds_2 ds_1 \quad (3.23)$$

$$+ \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{p-1}} e^{A(1-s_1)} B e^{A(s_1-s_2)} B \cdots e^{A(s_{p-1}-s_p)} B e^{(A+B)s_p} ds_p \cdots ds_2 ds_1, \quad (3.24)$$

which is precisely the form of Equation (3.16), with

$$R_p(A,B) = \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{p-1}} e^{A(1-s_1)} B e^{A(s_1-s_2)} B \cdots e^{A(s_{p-1}-s_p)} B e^{(A+B)s_p} ds_p \cdots ds_2 ds_1. \quad (3.25)$$

We can bound R_p as $\|R_p(A,B)\| \leq \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{p-1}} \|B\|^p ds_p \cdots ds_2 ds_1 = \frac{\|B\|^p}{p!}$ because A and B are anti-Hermitian matrices. This completes the proof. □

We now establish an expansion for powers of the second-order Trotter formula.

Lemma 3.3.3. *Let $U_2(s)$ be the second-order Trotter-Suzuki formula, and $U(s) = e^{-iHs}$ be the exact evolution operator. Define $\alpha_{\mathrm{comm},j}$ to be the summation of the norms of all the j-th nested commutators, i.e., $\alpha_{\mathrm{comm},j} = \sum_{\gamma_1,\cdots,\gamma_j=1}^{\Gamma} \|[H_{\gamma_1} H_{\gamma_2} \cdots H_{\gamma_j}]\|$. Suppose that there a integer $J \geq$ such that $\Delta \leq \inf_{j \geq J} \alpha_{\mathrm{comm},j}^{-1/j}$. Then, for any $k, p \in \mathbb{N}$, we have*

$$(U_2(\Delta/k))^k = U(\Delta) + \sum_{j \in 2\mathbb{Z}^+}^{\infty} \widetilde{E}_{j+1,p}(\Delta) \frac{1}{k^j} + \widetilde{F}_p(\Delta, k). \quad (3.26)$$

Here $\widetilde{E}_{j+1,p}$'s are operators independent of k and can be bounded as

$$\|\widetilde{E}_{j+1,p}(\Delta)\| \leq \Delta^{j+1} \left(\sum_{l=1}^{\min\{j/2, p-1\}} \frac{\Delta^{l-1}}{l!} \left(\sum_{\substack{j_1,\cdots,j_l \in 2\mathbb{Z}^+, \\ j_1+\cdots+j_l=j}} \left(\prod_{\kappa=1}^{l} \alpha_{\mathrm{comm},j_\kappa+1} \right) \right) \right), \quad (3.27)$$

and $\widetilde{F}_p(\Delta, k)$ *is an operator such that*

$$\|\widetilde{F}_p(\Delta,k)\| \leq \frac{\Delta^{3p}}{p!} \sum_{j\in 2\mathbb{Z}^+, j\geq 2p} \Delta^{j-2p} \left(\sum_{\substack{j_1,\cdots,j_p\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_p=j}} \left(\prod_{\kappa=1}^{p} \alpha_{\mathrm{comm},j_\kappa+1} \right) \right). \tag{3.28}$$

Proof. Using Proposition 3.3.1, we may express the second-order Trotter formula as $U_2(s) = e^{-iHs+\sum_{j\in 1+2\mathbb{Z}^+} E_j s^j}$, where

$$E_j = \sum_{\substack{\sum(i_\gamma+i'_\gamma)=j,\\ i_\gamma,i'_\gamma\geq 0}} \frac{(-i)^j}{i_1!\cdots i_\Gamma! i'_1!\cdots i'_\Gamma!} \phi_k \left(\underbrace{\frac{H_1}{2},\cdots,\frac{H_1}{2}}_{i_1},\cdots,\underbrace{\frac{H_\Gamma}{2},\cdots,\frac{H_\Gamma}{2}}_{i_\Gamma},\underbrace{\frac{H_\Gamma}{2},\cdots,\frac{H_\Gamma}{2}}_{i'_\Gamma},\cdots,\underbrace{\frac{H_1}{2},\cdots,\frac{H_1}{2}}_{i'_1} \right).$$

By Proposition 3.3.1, this expansion converges if $s^j\alpha_{\mathrm{comm},j} \leq 1$ for sufficiently large j. This condition is guaranteed for all $s \leq \Delta$ under our assumption on Δ. Moreover, Proposition 3.3.1 gives the bound $\|E_j\| \leq \frac{1}{j^2}\alpha_{\mathrm{comm},j} \leq \alpha_{\mathrm{comm},j}$. Then

$$(U_2(\Delta/k))^k = \exp\left(-iH\Delta + \sum_{j\in 1+2\mathbb{Z}^+} E_j \frac{\Delta^j}{k^{j-1}} \right). \tag{3.29}$$

We expand $\big(U_2(\Delta/k)\big)^k$ as a polynomial in $1/k$. Applying Lemma 3.3.2 with $A = -iH\Delta$ and $B = \sum_{j\in 2\mathbb{Z}^+} E_{j+1}\frac{\Delta^{j+1}}{k^j}$, we obtain

$$\begin{aligned}(U_2(\Delta/k))^k = U(\Delta) + \sum_{l=1}^{p-1} \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{l-1}} e^{A(1-s_1)} B e^{A(s_1-s_2)} B \cdots e^{A(s_{l-1}-s_l)} B e^{As_l} ds_l \cdots ds_2 ds_1 \\ + R_p(A,B).\end{aligned} \tag{3.30}$$

First, consider the summation term. Since B is a polynomial in $1/k$, this term can also be

written as a polynomial in $1/k$. Let $s_0 = 1$, and let ds denote $ds_l \cdots ds_1$ whenever the meaning is clear. We have

$$\sum_{l=1}^{p-1} \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{l-1}} e^{A(1-s_1)} B e^{A(s_1-s_2)} B \cdots e^{A(s_{l-1}-s_l)} B e^{As_l} ds_l \cdots ds_2 ds_1 \tag{3.31}$$

$$= \sum_{l=1}^{p-1} \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{l-1}} ds \left(\prod_{\kappa=l}^{1} \left(\sum_{j \in 2\mathbb{Z}^+} e^{A(s_{\kappa-1}-s_\kappa)} E_{j+1} \frac{\Delta^{j+1}}{k^j} \right) \right) e^{As_l} \tag{3.32}$$

$$= \sum_{l=1}^{p-1} \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{l-1}} ds \left(\sum_{j_1, \cdots, j_l \in 2\mathbb{Z}^+} \left(\prod_{\kappa=l}^{1} e^{A(s_{\kappa-1}-s_\kappa)} E_{j_\kappa+1} \right) \frac{\Delta^{l+\sum_\kappa j_{\kappa'}}}{k^{\sum_\kappa j_{\kappa'}}} \right) e^{As_l} \tag{3.33}$$

$$= \sum_{l=1}^{p-1} \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{l-1}} ds \left(\sum_{j \in 2\mathbb{Z}^+, j \geq 2l} \sum_{\substack{j_1, \cdots, j_l \in 2\mathbb{Z}^+, \\ j_1 + \cdots + j_l = j}} \left(\prod_{\kappa=l}^{1} e^{A(s_{\kappa-1}-s_\kappa)} E_{j_\kappa+1} \right) \frac{\Delta^{l+j}}{k^j} \right) e^{As_l} \tag{3.34}$$

$$= \sum_{j \in 2\mathbb{Z}^+} \widetilde{E}_{j+1,p}(\Delta) \frac{1}{k^j}, \tag{3.35}$$

where the operator $\widetilde{E}_{j+1,p}(\Delta)$ is independent of k and can be written explicitly as

$$\widetilde{E}_{j+1,p}(\Delta) = \sum_{l=1}^{\min\{j/2, p-1\}} \sum_{\substack{j_1, \cdots, j_l \in 2\mathbb{Z}^+, \\ j_1 + \cdots + j_l = j}} \Delta^{l+j} \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{l-1}} ds \left(\prod_{\kappa=l}^{1} e^{A(s_{\kappa-1}-s_\kappa)} E_{j_\kappa+1} \right) e^{As_l}. \tag{3.36}$$

Moreover, the norm of $\widetilde{E}_{j+1,p}(\Delta)$ can be upper bounded by

$$\|\widetilde{E}_{j+1,p}(\Delta)\| \leq \sum_{l=1}^{\min\{j/2,p-1\}} \sum_{\substack{j_1,\cdots,j_l\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_l=j}} \Delta^{l+j} \int_0^1 \int_0^{s_1} \cdots \int_0^{s_{l-1}} ds \left(\prod_{\kappa=l}^{1} \|E_{j_\kappa+1}\|\right) \tag{3.37}$$

$$\leq \Delta^{j+1} \left(\sum_{l=1}^{\min\{j/2,p-1\}} \frac{\Delta^{l-1}}{l!} \left(\sum_{\substack{j_1,\cdots,j_l\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_l=j}} \prod_{\kappa=1}^{l} \|E_{j_\kappa+1}\| \right) \right) \tag{3.38}$$

$$\leq \Delta^{j+1} \left(\sum_{l=1}^{\min\{j/2,p-1\}} \frac{\Delta^{l-1}}{l!} \left(\sum_{\substack{j_1,\cdots,j_l\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_l=j}} \prod_{\kappa=1}^{l} \alpha_{\mathrm{comm},j_\kappa+1} \right) \right). \tag{3.39}$$

Next, we analyze the remainder term $R_p(A,B)$. By Lemma 3.3.2, we can bound it as

$$\|R_p(A,B)\| \leq \frac{\|B\|^p}{p!} \tag{3.40}$$

$$\leq \frac{1}{p!} \left(\sum_{j\in 2\mathbb{Z}^+} \|E_{j+1}\| \frac{\Delta^{j+1}}{k^j} \right)^p \tag{3.41}$$

$$= \frac{\Delta^p}{p!} \sum_{j_1,\cdots,j_p\in 2\mathbb{Z}^+} \left(\prod_{\kappa=1}^{p} \|E_{j_\kappa+1}\| \right) \frac{\Delta^{j_1+\cdots+j_p}}{k^{j_1+\cdots+j_p}} \tag{3.42}$$

$$\leq \frac{\Delta^{3p}}{p!} \sum_{j\in 2\mathbb{Z}^+, j\geq 2p} \Delta^{j-2p} \left(\sum_{\substack{j_1,\cdots,j_p\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_p=j}} \left(\prod_{\kappa=1}^{p} \alpha_{\mathrm{comm},j_\kappa+1} \right) \right). \tag{3.43}$$

We use the notation $\widetilde{F}_p(\Delta,k)$ in the statement to emphasize its dependence on Δ and k. This completes the proof. □

3.3.3 Error Bound for Short-Time MPF

Combining Lemma 3.3.3 with the order conditions in Equation (3.4), we obtain the following single-step error bound for the MPF.

Theorem 3.3.4. *Let $U_{\mathrm{MP}}(\Delta)$ denote the 2nd-based MPF as $U_{\mathrm{MP}}(\Delta) = \sum_{l=1}^{M} a_l U_2(\Delta/k_l)^{k_l}$. Suppose that there a $J \geq 1$ such that $\Delta \leq \inf_{j\geq J} \alpha_{\mathrm{comm},j}^{-1/j}$. Then we have*

$$\|U_{\mathrm{MP}}(\Delta) - U(\Delta)\| \leq \|\vec{a}\|_1 \sum_{j\in 2\mathbb{Z}^+, j\geq 2m} \sum_{l=1}^{m} \frac{\Delta^{j+l}}{l!} \left(\sum_{\substack{j_1,\cdots,j_l\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_l=j}} \left(\prod_{\kappa=1}^{l} \alpha_{\mathrm{comm},j_\kappa+1} \right) \right). \tag{3.44}$$

Proof. For any $p \geq 1$, Lemma 3.3.3 together with the MPF order conditions in Equation (3.4) gives

$$U_{\mathrm{MP}}(\Delta) = U(\Delta) + \sum_{j\in 2\mathbb{Z}^+, j\geq 2m} \sum_{l=1}^{M} a_l \widetilde{E}_{j+1,p}(\Delta) \frac{1}{k_l^j} + \sum_{l=1}^{M} a_l \widetilde{F}_p(\Delta, k_l). \tag{3.45}$$

Using the bounds on $\widetilde{E}$ and $\widetilde{F}$, and using the crude bound $k_l \geq 1$, we obtain

$$\|U_{\mathrm{MP}}(\Delta) - U(\Delta)\| \leq \|\vec{a}\|_1 \sum_{j\in 2\mathbb{Z}^+, j\geq 2m} \Delta^{j+1} \left(\sum_{l=1}^{\min\{j/2, p-1\}} \frac{\Delta^{l-1}}{l!} \left(\sum_{\substack{j_1,\cdots,j_l\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_l=j}} \left(\prod_{\kappa=1}^{l} \alpha_{\mathrm{comm},j_\kappa+1} \right) \right) \right) \tag{3.46}$$

$$+ \|\vec{a}\|_1 \frac{\Delta^{3p}}{p!} \sum_{j\in 2\mathbb{Z}^+, j\geq 2p} \Delta^{j-2p} \left(\sum_{\substack{j_1,\cdots,j_p\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_p=j}} \left(\prod_{\kappa=1}^{p} \alpha_{\mathrm{comm},j_\kappa+1} \right) \right). \tag{3.47}$$

Choosing $p = m$, the summation over l in Equation (3.46) ranges from 1 to $m-1$. Moreover, Equation (3.47) has the same form as the summand in Equation (3.46) with $l = m$. Hence,

$$\|U_{\mathrm{MP}}(\Delta) - U(\Delta)\| \leq \|\vec{a}\|_1 \sum_{j\in 2\mathbb{Z}^+, j\geq 2m} \sum_{l=1}^{m} \frac{\Delta^{j+l}}{l!} \left(\sum_{\substack{j_1,\cdots,j_l\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_l=j}} \left(\prod_{\kappa=1}^{l} \alpha_{\mathrm{comm},j_\kappa+1} \right) \right). \tag{3.48}$$

This completes the proof. □

3.3.4 Complexity Analysis

We are now ready to estimate the overall complexity of the second-order-based MPF for long-time simulation, following the implementation outlined in Section 3.2. The complexity estimate is stated in Theorem 3.3.5. Its proof follows the general strategy of [LKW19, Theorem 2], adapted to incorporate the commutator-scaling representation for powers of the second-order Trotter formula.

Theorem 3.3.5. *Consider the Hamiltonian simulation problem up to time T using 2nd-based MPF of convergence order $2m$. Then, in order to obtain an ϵ-approximation with probability at least $1-\epsilon$, it suffices to use $\mathcal{O}(m^2(\log m)^2\mu_m T\left(\frac{\mu_m T}{\epsilon}\right)^{1/(2m)})$ queries to the controlled version of U_2. Here*

$$\mu_m = \sup_{\substack{j\in 2\mathbb{Z}^+, j\geq 2m,\\ 1\leq l\leq m}} \lambda_{j,l}, \qquad \lambda_{j,l} = \left(\sum_{\substack{j_1,\cdots,j_l\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_l=j}} \left(\prod_{\kappa=1}^{l} \alpha_{\mathrm{comm},j_\kappa+1}\right)\right)^{\frac{1}{j+l}}, \tag{3.49}$$

where $\alpha_{\mathrm{comm},j} = \sum_{\gamma_1,\cdots,\gamma_j=1}^{\Gamma} \|[H_{\gamma_1}H_{\gamma_2}\cdots H_{\gamma_j}]\|$.

Proof. Suppose that the time interval $[0,T]$ is divided into r equal-length segments, and let $\Delta = T/r$. Then Theorem 3.3.4 gives

$$\|U_{\mathrm{MP}}(\Delta) - U(\Delta)\| \leq \|\vec{a}\|_1 \sum_{j\in 2\mathbb{Z}^+, j\geq 2m} \sum_{l=1}^{m} \frac{(\mu_m\Delta)^{j+l}}{l!} := \epsilon_\Delta. \tag{3.50}$$

Choose $r = 2\mu_m T\left(\frac{2\mu_m T\|\vec{a}\|_1}{\epsilon}\right)^{1/(2m)}$. For any $j \geq 2m$ and $1 \leq l \leq m$, we have $\frac{r(\mu_m\Delta)^{j+l}}{l!} =$

$\frac{(\mu_m T)^{j+l}}{r^{j+l-1}} \leq \frac{\epsilon}{2^{j+l}\|\vec{a}\|_1}$. Therefore, we have

$$r\epsilon_\Delta = \|\vec{a}\|_1 \sum_{\substack{j\in 2\mathbb{Z}^+ \\ j\geq 2m}} \sum_{l=1}^{m} \frac{(\mu_m T)^{j+l}}{l! r^{j+l-1}} \leq \|\vec{a}\|_1 \sum_{\substack{j\in 2\mathbb{Z}^+ \\ j\geq 2m}} \sum_{l=1}^{m} \frac{\epsilon}{2^{j+l} l! \|\vec{a}\|_1} < \sum_{j\geq 2m} \frac{\epsilon(e^{1/2}-1)}{2^j} \leq \frac{1}{2}\epsilon. \tag{3.51}$$

The global error can be bounded by the triangle inequality. Specifically, as Equation (3.50) implies, $\|U_{\mathrm{MP}}(\Delta)\| \leq \|U(\Delta)\| + \epsilon_\Delta = 1 + \epsilon_\Delta$. Hence,

$$\|U_{\mathrm{MP}}(T/r)^r - U(T)\| = \left\| \sum_{j=0}^{r-1} U_{\mathrm{MP}}(\Delta)^{r-j-1} \left(U_{\mathrm{MP}}(\Delta) - U(\Delta)\right) U(\Delta)^j \right\| \tag{3.52}$$

$$\leq \sum_{j=0}^{r-1} \|U_{\mathrm{MP}}\|^{r-j-1} \|U_{\mathrm{MP}}(\Delta) - U(\Delta)\| \tag{3.53}$$

$$\leq r\epsilon_\Delta (1+\epsilon_\Delta)^{r-1} < \frac{\epsilon}{2}\left(1 + \frac{\epsilon}{2r}\right)^r < \frac{\epsilon}{2} e^{\epsilon/2} < \epsilon. \tag{3.54}$$

Since $\|\vec{a}\|_1 = \mathcal{O}(\log m)$, we have $\|\vec{a}\|_1^{1/(2m)} = \mathcal{O}(1)$. Therefore, the choice of r becomes $r = \mathcal{O}\left(\mu_m T \left(\frac{\mu_m T}{\epsilon}\right)^{1/(2m)}\right)$. We now count the overall query complexity. In each short-time MPF $U_{\mathrm{MP}}(\Delta)$, the LCU step uses $\|\vec{k}\|_1$ queries to U_2. For long-time simulation, the success probability in each step must be boosted to at least $1 - \mathcal{O}(\epsilon/r)$ using robust oblivious amplitude amplification [Ber+15b], which requires $\mathcal{O}(\|\vec{a}\|_1)$ queries to U_{MP}. Therefore, the number of queries to U_2 per time step is $\mathcal{O}(\|\vec{a}\|_1 \|\vec{k}\|_1)$, and the total query complexity for long-time simulation is $\mathcal{O}(r\|\vec{a}\|_1\|\vec{k}\|_1)$. For the well-conditioned 2nd-based MPF, we have $\|\vec{a}\|_1 = \mathcal{O}(\log m)$ and $\|\vec{k}\|_1 = \mathcal{O}(m^2 \log m)$. Hence, the total number of queries to U_2 is $\mathcal{O}(r\|\vec{a}\|_1\|\vec{k}\|_1) = \mathcal{O}\left(m^2 (\log m)^2 \mu_m T \left(\frac{\mu_m T}{\epsilon}\right)^{1/(2m)}\right)$. This completes the proof. □

Theorem 3.3.5 gives an m-specific complexity estimate. In practice, one may choose m to minimize the query complexity, provided that the growth rate of high-order nested commutators

remains bounded independently of m. Specifically, we obtain the following result.

Corollary 3.3.6. *Consider the Hamiltonian simulation problem up to time T using 2nd-based MPF. Let μ_m be a commutator-related factor as defined in Theorem 3.3.5, and suppose that there exists an integer M such that $\sup_{m\geq M}\mu_m \leq \mu$. Then, in order to obtain an ϵ-approximation with probability at least $1-\epsilon$, it suffices to use $\mathcal{O}\left(\mu T \operatorname{poly}\log\left(\frac{\mu T}{\epsilon}\right)\right)$ queries to the controlled version of U_2.*

Proof. By Theorem 3.3.5, the query complexity for $m \geq M$ is given by $\mathcal{O}(m^2(\log m)^2\mu T\left(\frac{\mu T}{\epsilon}\right)^{1/(2m)})$. Choose $m = \left\lceil \frac{1}{2}\log\left(\frac{\mu T}{\epsilon}\right)\right\rceil$. For sufficiently small ϵ or sufficiently large T, this choice satisfies $m \geq M$. Then $\left(\frac{\mu T}{\epsilon}\right)^{1/(2m)} \leq \left(\frac{\mu T}{\epsilon}\right)^{1/\log\left(\frac{\mu T}{\epsilon}\right)} = e$. Therefore, the overall complexity becomes

$$\mathcal{O}\left(m^2(\log m)^2\mu T\right) = \mathcal{O}\left(\mu T\left(\log\left(\frac{\mu T}{\epsilon}\right)\right)^2\left(\log\log\left(\frac{\mu T}{\epsilon}\right)\right)^2\right). \tag{3.55}$$

This completes the proof. □

By Theorem 3.3.5 and Corollary 3.3.6, the query complexity of the 2nd-based MPF depends crucially on the commutator parameters μ_m and μ, which involve sums over all indices $j_1,\ldots,j_l$ satisfying $j_1+\cdots+j_l=j$. This summation can be further simplified by noting that the number of positive integer solutions to $j_1+\cdots+j_l=j$ is $\binom{j-1}{l-1} < 2^{j-1}$. Hence,

$$\mu_m = \sup_{\substack{j\in 2\mathbb{Z}^+, j\geq 2m\\ 1\leq l\leq m}}\left(\sum_{\substack{j_1,\cdots,j_l\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_l=j}}\left(\prod_{\kappa=1}^{l}\alpha_{\mathrm{comm},j_\kappa+1}\right)\right)^{\frac{1}{j+l}} \leq 2\sup_{\substack{j\in 2\mathbb{Z}^+, j\geq 2m,\\ 1\leq l\leq m}}\max_{\substack{j_1,\cdots,j_l\in 2\mathbb{Z}^+,\\ j_1+\cdots+j_l=j}}\left(\prod_{\kappa=1}^{l}\alpha_{\mathrm{comm},j_\kappa+1}\right)^{\frac{1}{j+l}}. \tag{3.56}$$

Equation (3.56) is useful for computing commutator scaling in specific applications.

3.4 Applications

In this section, we analyze the computational complexity of the second-order-based multi-product formula (MPF) in several applications, including the simulation of second-quantized electronic structure Hamiltonians and the solution of the Schrödinger equation for one-dimensional continuum systems.

3.4.1 Second-Quantized Electronic Structure

Consider a second-quantized plane-wave electronic structure Hamiltonian with n spin orbitals. The Hamiltonian takes the form $H = T + U + V$. Here T is the kinetic term, expressed as a sum of creation and annihilation operators, while U and V denote the one- and two-body potential operators, respectively. This application has been studied in detail in the context of commutator scaling for product formulas [Chi+21]. As shown in [Chi+21], the commutator factor $\alpha_{\mathrm{comm},p+1}$ can be bounded as $\alpha_{\mathrm{comm},j} \leq Cn^j$, where $C \geq 1$ is independent of n and j. Hence, for any fixed (j, l) and any positive integers j_κ satisfying $j_1 + \cdots + j_l = j$, we have

$$\left(\prod_{\kappa=1}^{l} \alpha_{\mathrm{comm},j_\kappa+1}\right)^{\frac{1}{j+l}} \leq \left(C^l n^{\sum j_\kappa + l}\right)^{\frac{1}{j+l}} = \left(C^l n^{j+l}\right)^{\frac{1}{j+l}} \leq Cn. \tag{3.57}$$

By Equation (3.56), we have $\mu_m = \mathcal{O}(n)$ and $\mu = \mathcal{O}(n)$. Then Corollary 3.3.6 implies that U_2 must be used $\mathcal{O}(nT \operatorname{polylog}(nT/\epsilon))$ times. Since each implementation of U_2 has gate complexity linear in n, the overall gate complexity is $\mathcal{O}\left(n^2 T \operatorname{poly}\log\left(\frac{nT}{\epsilon}\right)\right)$.

3.4.2 Unbounded Hamiltonian Simulation

In continuous-variable quantum systems, the Hamiltonian H is often unbounded because of the kinetic energy operator $-\Delta$, where Δ denotes the Laplacian. Together with a smooth potential $V(\mathbf{x})$ for $\mathbf{x} \in \Omega \subseteq \mathbb{R}^d$, this gives the Hamiltonian $H = -\Delta + V(\mathbf{x})$, leading to the *unbounded Hamiltonian simulation problem* [AFL21a; BFZ25]. We analyze this problem using the MPF in the one-dimensional setting $d = 1$, with domain $\Omega = [0, 1]$ and a smooth periodic potential. Discretizing the Schrödinger equation by a finite-difference scheme with N uniformly spaced nodes $x_j = j/N$, $0 \le j \le N-1$, and periodic boundary conditions yields

$$i\frac{\partial}{\partial t}\Psi(t) = (L + V)\,\Psi(t), \tag{3.58}$$

The matrices L and V are given by

$$L = N^2 \begin{pmatrix} 2 & -1 & & & & -1 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & \ddots & & \\ & & \ddots & \ddots & -1 & \\ & & & -1 & 2 & -1 \\ -1 & & & & -1 & 2 \end{pmatrix}, \qquad V = \begin{pmatrix} V_0 & & & \\ & V_1 & & \\ & & \ddots & \\ & & & V_{N-1} \end{pmatrix}, \tag{3.59}$$

where $V_i = V(i/N)$. Clearly, $\|L\| = \mathcal{O}(N^2)$ and $\|V\| = \mathcal{O}(1)$. To perform a complete MPF error analysis, we must estimate higher-order nested commutators of L and V. We begin by bounding the spectral norm of a special class of nested commutators, which serves as the baseline for the

complexity analysis below. Throughout this subsection, all indices are understood modulo N.

Lemma 3.4.1. *Fix $j, N \in \mathbb{N}$, and let L and V be as in Equation (3.59). Then $\| \operatorname{ad}_L^j(V)\| = \mathcal{O}(jN^j)$.*

Proof. We prove the following statements by induction on j:

1. $\operatorname{ad}_L^j(V)$ is a symmetric (respectively, anti-symmetric) matrix when j is even (respectively, odd).

2. $\operatorname{ad}_L^j(V)$ is a periodic banded matrix with bandwidth j. If B_i^j for $i = -j, \ldots, j$ denote the bands of $\operatorname{ad}_L^j(V)$, then B_i^j is zero if i is even (respectively odd) is if j is odd (respectively even).

3. Each entry of $\operatorname{ad}_L^j(V)/N^j$ is a first order backward (respectively, central) finite difference approximation of the j-th derivative of $V(x)$ if j is odd (respectively, even). In fact, the non-zero entries are determined as follows:

   (a) (odd j). Let $j = 2m-1$ for $m \geq 1$. For $p = 1, \cdots, m$, the q-th entry of the band B_{2p-1}^j is a first order backward finite different at node $m+p+q-2$ modulo N with constant coefficient $\pm\binom{j}{\lceil j/2 \rceil + (p-1)}$ multiplying each entry such that the sign alternates among the bands starting from B_1^j onward with a negative (respectively, positive) sign if m is even (respectively, odd).

   (b) (even j) Let $j = 2m$ for $m \geq 1$. For $p = 0, \cdots, m$, the q-th entry of the band B_{2p}^j is a first order backward finite different at node $p - 1 + q$ modulo N with constant coefficient $\pm\binom{j}{j/2+p}$ multiplying each entry such that the sign alternates among the

bands starting from B_0^j onward with a negative (respectively, positive) sign if m is odd (respectively, even).

We refer to the final condition as the finite difference condition. The case $j = 1$ is proved in [AFL21b, Lemma 8]; however, we reproduce the details below to align it with our proof for the case $j > 1$. We compute the commutator $\mathrm{ad}_L(V) = [L, V]$. A direct computation shows that

$$(\mathrm{ad}_L(V))_{kl} = N^2(V_{k-1} - V_{l-1})\delta_{|k-l|,1} = N\left(\frac{V_{k-1} - V_{l-1}}{1/N}\right)\delta_{|k-l|,1}. \tag{3.60}$$

In matrix notation, we have

$$\mathrm{ad}_L(V) = N\begin{pmatrix} 0 & V_1 - V_0 & & & -(V_{N-1} - V_0) \\ -(V_1 - V_0) & 0 & V_2 - V_1 & & \\ & -(V_2 - V_1) & 0 & \ddots & \\ & & \ddots & \ddots & V_{N-1} - V_{N-2} \\ V_{N-1} - V_0 & & & -(V_{N-1} - V_{N-2}) & 0 \end{pmatrix}. \tag{3.61}$$

The symmetry, periodic bandedness, and finite-difference properties described above are readily verified by inspecting Equation (3.61). Since $V(x)$ is assumed to be smooth, we have $V_m^{(1)} = \mathcal{O}(1)$. Hence, each entry of $\mathrm{ad}_L(V)/N$ is $\mathcal{O}(1)$. Since each row and column has at most $j + 1 = 2 = \mathcal{O}(1)$ nonzero entries, it follows that $\|\mathrm{ad}_L(V)\|_{1,\infty} = \mathcal{O}(N)$. Therefore, we have

$$\|\mathrm{ad}_L(V)\| \le \sqrt{\|\mathrm{ad}_L(V)\|_1 \cdot \|\mathrm{ad}_L(V)\|_\infty} = \sqrt{\mathcal{O}(N^2)} = \mathcal{O}(N). \tag{3.62}$$

Assume now that the claim holds for some $j \ge 1$. We prove the claim for $j + 1$. By definition,

$\mathrm{ad}_L^{j+1}(V) = [L, \mathrm{ad}_L^j(V)]$. Since L is symmetric, $\mathrm{ad}_L^{j+1}(V)$ is antisymmetric, respectively symmetric, according as $\mathrm{ad}_L^j(V)$ is symmetric, respectively antisymmetric, because the commutator of a symmetric matrix with a symmetric, respectively antisymmetric, matrix is antisymmetric, respectively symmetric. Moreover, since L is a periodic banded matrix of bandwidth 1, and $\mathrm{ad}_L^j(V)$ is a periodic banded matrix of bandwidth j, the subadditivity of bandwidth under products, and hence under commutators, implies that $\mathrm{ad}_L^{j+1}(V)$ is periodic banded with bandwidth at most $j+1$. It remains to prove the finite-difference condition. WLOG, assume that j is odd. Note that

$$(\mathrm{ad}_L^{j+1}(V))_{kl} = [L, \mathrm{ad}_L^j(V)]_{kl} = N^2(((\mathrm{ad}_L^j(V))_{k,l-1} + (\mathrm{ad}_L^j(V))_{k,l+1}) \tag{3.63}$$

$$- ((\mathrm{ad}_L^j(V))_{k-1,l} + (\mathrm{ad}_L^j(V))_{k+1,l})), \tag{3.64}$$

We will use this formula to compute $(\mathrm{ad}_L^{j+1}(V))_{kl}$. WLOG, assume that $(\mathrm{ad}_L^{j+1}(V))_{kl}$ is contained within bands $B_2^{j+1}, \cdots, B_{2j-2}^{j+1}$, ensuring that only bands $B_1^j, B_3^j, \ldots, B_{2j-3}^j, B_{2j-1}^j$ from $\mathrm{ad}_L^j(V)$ contribute to the determination of $(\mathrm{ad}_L^{j+1}(V))_{kl}$. Assume that $(\mathrm{ad}_L^{j+1}(V))_{kl}$ is the q-th entry in band B_{2p}^j for $p = 1, \cdots, j-1$. Hence, $(\mathrm{ad}_L^{j+1}(V))_{kl}$ is proportional to central finite difference at node $(j+1)/2 + p + q - 1$ modulo N. Using the induction hypothesis, we have

$$(\mathrm{ad}_L^j(V))_{k,l-1} = q\text{-th entry in } B_{2p-1}^j \propto \text{BFD at node } \frac{j+1}{2} + p + q - 2, \tag{3.65}$$

$$(\mathrm{ad}_L^j(V))_{k,l+1} = (q-1)\text{-th entry in } B_{2p+1}^j \propto \text{BFD at node } \frac{j+1}{2} + p + q - 2, \tag{3.66}$$

$$(\mathrm{ad}_L^j(V))_{k-1,l} = (q-1)\text{-th entry in } B_{2p-1}^j \propto \text{BFD at node } \frac{j+1}{2} + p + q - 3, \tag{3.67}$$

$$(\mathrm{ad}_L^j(V))_{k+1,l} = q\text{-th entry in } B_{2p+1}^j \propto \text{BFD at node } \frac{j+1}{2} + p + q - 1. \tag{3.68}$$

Here, BFD represents the phrase backward finite difference. All indices above are considered modulo N. Assuming that $j = 2m - 1$ such that m is odd and p is even, we have

$$(\mathrm{ad}_L^j(V))_{k,l-1} = +\frac{1}{1/N^j}\binom{j}{\lceil j/2\rceil + (p-1)}\sum_{i=0}^{j}(-1)^{j-i}\binom{j}{i}V\left(\frac{j+1}{2}+p+q-2-i\right), \quad (3.69)$$

$$(\mathrm{ad}_L^j(V))_{k,l+1} = -\frac{1}{1/N^j}\binom{j}{\lceil j/2\rceil + p}\sum_{i=0}^{j}(-1)^{j-i}\binom{j}{i}V\left(\frac{j+1}{2}+p+q-2-i\right), \quad (3.70)$$

$$(\mathrm{ad}_L^j(V))_{k-1,l} = +\frac{1}{1/N^j}\binom{j}{\lceil j/2\rceil + (p-1)}\sum_{i=0}^{j}(-1)^{j-i}\binom{j}{i}V\left(\frac{j+1}{2}+p+q-3-i\right), \quad (3.71)$$

$$(\mathrm{ad}_L^j(V))_{k+1,l} = -\frac{1}{1/N^j}\binom{j}{\lceil j/2\rceil + p}\sum_{i=0}^{j}(-1)^{j-i}\binom{j}{i}V\left(\frac{j+1}{2}+p+q-1-i\right). \quad (3.72)$$

A straightforward computation now shows that $(\mathrm{ad}_L^{j+1}(V))_{kl}/N$ is given by

$$\frac{(\mathrm{ad}_L^{j+1}(V))_{kl}}{N} = \frac{1}{N}(((\mathrm{ad}_L^j(V))_{k,l-1} + (\mathrm{ad}_L^j(V))_{k,l+1}) - ((\mathrm{ad}_L^j(V))_{k-1,l} + (\mathrm{ad}_L^j(V))_{k+1,l})), \quad (3.73)$$

$$= \frac{1}{N^{j+1}}\binom{j+1}{(j+1)/2+p}\sum_{i=0}^{j+1}(-1)^{i}\binom{j+1}{i}V\left(p+q-1+\frac{j+1}{2}-i\right), \quad (3.74)$$

$$= \frac{1}{N^{j+1}}\binom{j+1}{(j+1)/2+p}\sum_{i=0}^{j+1}(-1)^{i}\binom{j+1}{i}V\left(p+q-1+\frac{j+1}{2}-i\right). \quad (3.75)$$

This completes the inductive argument. Combining the preceding estimates, each entry of $\mathrm{ad}_L^{j+1}(V)/N^{j+1}$ is $\mathcal{O}(1)$, since $V(x)$ is assumed to be smooth. Moreover, each row and column of $\mathrm{ad}_L^{j+1}(V)/N^{j+1}$ contains at most $\mathcal{O}(j+1)$ nonzero entries. Therefore, $\|\mathrm{ad}_L^{j+1}(V)\|_{1,\infty} = \mathcal{O}((j+1)N^{j+1})$. Hence, we have

$$\|\mathrm{ad}_L^{j+1}(V)\| \leq \sqrt{\|\mathrm{ad}_L^{j+1}(V)\|_1 \cdot \|\mathrm{ad}_L^{j+1}(V)\|_\infty} = \mathcal{O}((j+1)N^{j+1}). \quad (3.76)$$

This completes the proof. □

Since the straightforward but tedious computation in Equation (3.73) was omitted, we illustrate the difference between the finite-difference and central-difference cases through an explicit example with $j = 3$. The corresponding expression for $j = 2$ can be found in [AFL21b, Lemma 8].

Example 3.4.2. Let $N = 8$. For $j = 3$, let $V_j^{(3)} \approx \frac{1}{(1/N)^3}\left(V_j - 3V_{j-1} + 3V_{j-2} - V_{j-3}\right)$ denote the backward finite difference approximation to $V^{(3)}(x_j)$. The matrix $N^3 \operatorname{ad}_L^3(V)$ is given as follows:

$$\begin{pmatrix}
0 & -3\left(3V_0 - 3V_1 + V_2 - V_7\right) & 0 & -V_0 + 3V_1 - 3V_2 + V_3 & 0 & -V_0 + V_5 - 3V_6 + 3V_7 & 0 & -3\left(3V_0 - V_1 + V_6 - 3V_7\right) \\
3\left(3V_0 - 3V_1 + V_2 - V_7\right) & 0 & 3\left(V_0 - 3V_1 + 3V_2 - V_3\right) & 0 & -V_1 + 3V_2 - 3V_3 + V_4 & 0 & 3V_0 - V_1 + V_6 - 3V_7 & 0 \\
0 & -3\left(V_0 - 3V_1 + 3V_2 - V_3\right) & 0 & 3\left(V_1 - 3V_2 + 3V_3 - V_4\right) & 0 & -V_2 + 3V_3 - 3V_4 + V_5 & 0 & -3V_0 + 3V_1 - V_2 + V_7 \\
V_0 - 3V_1 + 3V_2 - V_3 & 0 & -3\left(V_1 - 3V_2 + 3V_3 - V_4\right) & 0 & 3\left(V_2 - 3V_3 + 3V_4 - V_5\right) & 0 & -V_3 + 3V_4 - 3V_5 + V_6 & 0 \\
0 & V_1 - 3V_2 + 3V_3 - V_4 & 0 & -3\left(V_2 - 3V_3 + 3V_4 - V_5\right) & 0 & 3\left(V_3 - 3V_4 + 3V_5 - V_6\right) & 0 & -V_4 + 3V_5 - 3V_6 + V_7 \\
V_0 - V_5 + 3V_6 - 3V_7 & 0 & V_2 - 3V_3 + 3V_4 - V_5 & 0 & -3\left(V_3 - 3V_4 + 3V_5 - V_6\right) & 0 & 3\left(V_4 - 3V_5 + 3V_6 - V_7\right) & 0 \\
0 & -3V_0 + V_1 - V_6 + 3V_7 & 0 & V_3 - 3V_4 + 3V_5 - V_6 & 0 & -3\left(V_4 - 3V_5 + 3V_6 - V_7\right) & 0 & -3\left(V_0 - V_5 + 3V_6 - 3V_7\right) \\
3\left(3V_0 - V_1 + V_6 - 3V_7\right) & 0 & 3V_0 - 3V_1 + V_2 - V_7 & 0 & V_4 - 3V_5 + 3V_6 - V_7 & 0 & 3\left(V_0 - V_5 + 3V_6 - 3V_7\right) & 0
\end{pmatrix}$$

For larger N and $j > 3$, the corresponding matrices can be generated using the code available in [Aft25]. In all cases, the factor N^j can be absorbed so that every matrix entry remains $\mathcal{O}(1)$. This argument follows the spirit of Lemma 3.4.1 and extends to arbitrary j.

We now turn to the general case. The key idea is that, for an arbitrary nested commutator, the entries can be rearranged by a suitable permutation without changing the spectral norm asymptotically. This reduction allows us to apply the argument used in Lemma 3.4.1. We will invoke the Lemma 3.4.3 which is proved in Section A.1.

Lemma 3.4.3. *Let $N \in \mathbb{N}$ and $X_N, Y_N \in \mathbb{C}^{N\times N}$ such that $\|X_N\|_F = \Theta(\|Y_N\|_F)$ (as a function of N), where $\|\cdot\|_F$ denotes the Frobenius norm. Let $A_N \in \mathbb{C}^{N\times N}$ be a unitarily diagonalizable matrix such that the eigenvalues $\lambda_1, \cdots, \lambda_N$ of A_N are such that $|\lambda_i - \lambda_j| = \Theta\left(\frac{1}{N^\alpha}\right)$ for some*

$\alpha \geq 0$ *and* $i \neq j$ *is satisfied. We have* $\|[A_N, X_N]\|_F = \Theta(\|[A_N, Y_N]\|_F)$. *Moreover, if* B_N *is a diagonal matrix in the standard basis whose eigenvalues satisfy the condition above, possibly with* $\beta \geq 0$ *different from* α, *then* $\|[B_N, [A_N, X_N]]\|_F = \Theta(\|[A_N, [B_N, Y_N]]\|_F)$.

Before proceeding, observe that, in order to apply Lemma 3.4.3, we must assume that $V(x)$ satisfies $|V_i - V_j| = \Theta(1/N^\alpha)$ for some $\alpha \geq 0$. The upper bound with $\alpha \leq 1$ follows from the smoothness of $V(x)$ on $[0, 1]$, which implies Lipschitz continuity. In what follows, we restrict attention to smooth periodic potentials, for which a matching lower bound holds when $0 < \alpha \leq 1$.

Proposition 3.4.4. *Fix* $j, N \in \mathbb{N}$. *Let* L *and* V *be as in Equation* (3.59), *and let* $B_{j+1,k} \subseteq \{0, 1\}^{j+1}$ *denote the set of binary sequences of length* $j + 1$ *such that* k *of the terms are equal to* 1. *For* $s \in B_{j+1,k}$, *consider the nested commutator* $C^s_{j+1,k}(L, V) = [H_{s_1}, [H_{s_2}, \ldots, [H_{s_j}, H_{s_{j+1}}]]]$, *where each* $H_{s_i} \in \{L, V\}$ *such that* $H_{s_i} = L$ *if* $s_i = 1$ *and* $H_{s_i} = V$ *if* $s_i = 0$. *We have* $\|C^s_{j+1,k}(L, V)\| = \mathcal{O}(kN^{k+1/2})$.

Proof. The case $j = 1$ is straightforward, as the only nonzero commutators are of the form $\pm \operatorname{ad}_L(V)$, which are already handled by Lemma 3.4.1. For $j = 2$, the nonzero commutators are $\pm[L, [L, V]]$ and $\pm[V, [L, V]]$. The first of these is again covered by Lemma 3.4.1. For the second, we observe that

$$\|[V, [L, V]]\| \leq 2\|V\|\|[L, V]\| = \mathcal{O}(N), \tag{3.77}$$

which establishes the claim for $j = 2$ as well. We may therefore assume $j \geq 3$ in the remainder of the argument. Let $s = (s_1, s_2, \cdots, s_j, s_{j+1})$. We can assume that $H_{s_j} \neq H_{s_{j+1}}$ since otherwise $C^s_{j+1,k}$ would be zero. In particular, we can assume that $H_{s_j} = L$ $(k \geq 1)$ and $H_{s_{j+1}} = V$. Write $s = (s_1, \cdots, s_{j-1}, 1, 0) = (s', 1, 0)$. If $\sigma \in S_{j-1}$, we write $\sigma(s')$ for the permutation

$\sigma(s') = (s'_{\sigma(1)}, \cdots, s'_{\sigma(j-1)})$. We show that $\|C^{(s',1,0)}_{j+1,k}\|_F = \Theta(\|C^{(\sigma(s'),1,0)}_{j+1,k}\|_F)$. That is,

$$\left\|[H_{s_1}, [H_{s_2}, \ldots, [H_{s_{j-1}}, [L, V]]]]\right\|_F = \Theta(\|[H_{s_{\sigma(1)}}, [H_{s_{\sigma(2)}}, \ldots, [H_{s_{\sigma(j-1)}}, [L, V]]]]\|_F). \tag{3.78}$$

Since any $\sigma \in S_{j-1}$ can be written as a product of adjacent transpositions, it suffices to consider the case where σ is a single adjacent transposition. Let $\sigma = (i, i+1)$ for $i = 1, \cdots, j-2$. It suffices to assume that $s_i \neq s_{i+1}$; otherwise, the claim is trivial. WLOG, we may assume that $s_i = 1$ and $s_{i+1} = 0$. We prove Equation (3.78) by using Lemma 3.4.3. We first justify the application of this result. Let A_N be as in Lemma 3.4.3. $A_N = V$ satisfies the eigenvalue condition in Lemma 3.4.3. It is well-known that the eigenvalues of C matrix in Equation (3.59) are given $\lambda_k = 2 - 2\cos\left(\frac{2\pi k}{N}\right) = 4\sin^2\left(\frac{\pi k}{N}\right)$ for $i = 1, \cdots, N$. Therefore, we have

$$|\lambda_k - \lambda_l| = 4\left|\sin^2\left(\frac{\pi k}{N}\right) - \sin^2\left(\frac{\pi l}{N}\right)\right| = 4\left|\sin\left(\frac{\pi(k+l)}{N}\right)\sin\left(\frac{\pi(k-l)}{N}\right)\right| \tag{3.79}$$

For large enough N, we have

$$|\lambda_k - \lambda_l| = \Theta\left(\frac{(k+l)(k-l)}{N^2}\right) = \Theta\left(\frac{1}{N^2}\right). \tag{3.80}$$

The last estimate follows since, for sufficiently large N, the depth $j = \mathcal{O}(1)$, and we only need to consider k, l such that $|k-l| = \mathcal{O}(j)$ modulo N. We also have $k + l = \mathcal{O}(j)$ modulo N. Hence, $A_N = L$ satisfies eigenvalue condition in Lemma 3.4.3. since L is N^2 times the circulant matrix, C. This justifies an application of Lemma 3.4.3 which we would now like to invoke repeatedly to prove Equation (3.78). Let X_N and Y_N be as defined in Lemma 3.4.3, which for our purposes, are nested commutators of L and V that differ by an adjacent transposition. In

$k\backslash N$	50	100	500	1000
0	593.303	626.442	654.308	657.878
5	0.812	0.032	1.210×10^{-5}	3.869×10^{-7}
10	0.005	6.899×10^{-6}	9.010×10^{-13}	9.076×10^{-16}

Table 3.1: The ratio in Equation (3.82) is computed for various values of k (rows) and N (columns).

order to iteratively apply Lemma 3.4.3 with $A_N = L, V$ and reach the desired conclusion, it suffices to assume that $\sigma = (1, 2)$ and prove that

$$\left\|[L, [V, [H_{s_3}, \ldots, [H_{s_{j-1}}, [L, V]]]]]\right\|_F = \Theta(\|[V, [L, [H_{s_3}, \ldots, [H_{s_{j-1}}, [L, V]]]]]\|_F). \tag{3.81}$$

An application of Lemma 3.4.3 with $B_N = V$ and $A_N = L$ in the same lemma proves this assertion as well. We now prove the stated bound on the spectral norm of $C_{j+1,k}(L, V)$. Since the $\|\cdot\|$ is dominated by $\|\cdot\|_F$, it suffices to prove the bound on $\|C_{j+1,k}(L, V)\|_F$. By our discussion above on permutation invariance, we may, WLOG, assume that $C_{j+1,k}(L, V)$ is of the form $C_{j+1,k}(L, V) = [V, [V, \ldots, [L, [L, [L, V]]]]]$. We now bound $\|C_{j+1,k}(L, V)\|_F$ by observing that $\|C_{j+1,k}(L, V)\|_F \leq \sqrt{N}\|C_{j+1,k}(L, V)\|$. Since $\mathcal{O}(\|[V, \cdot]\|) = \mathcal{O}(\|\cdot\|)$, it suffices to consider the case of the commutator $\mathrm{ad}_L^k(V)$. The result now follows by invoking Lemma 3.4.1. □

We remark that, although the $\mathcal{O}(kN^{k+1/2})$ bound on $C_{j+1,k}$ is sufficient for the analysis of the MPF algorithm, it is rather conservative. For example, the extra factor of $N^{1/2}$ can be avoided if one does not rely on interpolation between the spectral and Frobenius norms, since that interpolation is precisely what introduces this factor. Nevertheless, our numerical investigations suggest that the actual growth is often significantly smaller than $\mathcal{O}(kN^k)$. We illustrate this behavior below with an example.

Example 3.4.5. Consider the nested commutator $\mathrm{ad}^k_{\mathrm{ad}^3_L(V)}(V)$ for $k \geq 1$. For various values of k and N, we report the ratio

$$\frac{\mathrm{ad}^k_{\mathrm{ad}^3_L(V)}(V)}{3N^{3+\frac{1}{2}}} \tag{3.82}$$

in Table 3.1. We observe that for $k = 0$, the ratio is asymptotically non-zero, indicating that our error estimate in Lemma 3.4.1 is tight. However, for $k > 0$, the ratio tends to zero asymptotically, indicating that the spectral norm of the commutator can be much smaller asymptotically than our worst-case estimate in Proposition 3.4.4 when $k > 0$. The data presented in Table 3.1 was generated using the code provided in [Aft25].

Given the commutator bound in Lemma 3.4.1, we now analyze the complexity of the MPF implementation for unbounded Hamiltonian simulation in one dimension for the Schrödinger equation. In what follows, we use the more conservative bound $\|C_{j+1,k}\| = \mathcal{O}(kN^{k+1/2}) = \mathcal{O}(jN^{j+1/2}) = \mathcal{O}(2^j N^{j+1/2})$. Hence, there exists a constant $C \geq 1$ such that $\alpha_{\mathrm{comm},j} \leq C\, 2^j N^{j+1/2}$. For any fixed pair (j, l) and any positive even integers j_κ satisfying $j_1 + \cdots + j_l = j$, we have

$$\left(\prod_{\kappa=1}^{l} \alpha_{\mathrm{comm},j_\kappa+1}\right)^{\frac{1}{j+l}} \leq \left(C^l 2^{\sum j_\kappa + l} N^{\sum j_\kappa + 3l/2}\right)^{\frac{1}{j+l}} = \left(C^l 2^{j+l} N^{j+3l/2}\right)^{\frac{1}{j+l}}. \tag{3.83}$$

Therefore, $\mu_m = \mathcal{O}(N^{7/6})$, since the supremum is attained when $j = 2m$ and $l = m$. This implies that $\mu = \mathcal{O}(N^{7/6})$. Therefore, by Corollary 3.3.6, the number of controlled applications of the second-order product formula U_2 is $\mathcal{O}\left(N^{7/6}T \operatorname{polylog}\left(\frac{N^{7/6}T}{\epsilon}\right)\right)$ Each implementation of U_2 requires exponentials of V and L. The potential term V is diagonal, while L is diagonalized by the discrete Fourier transform. Thus, assuming the required diagonal phase oracles/arithmetic for e^{-itV} and e^{-itL} are implemented with polylogarithmic overhead, the dom-

inant Fourier-transform cost per application of U_2 is $\mathcal{O}(n^2) = \mathcal{O}(\log^2 N)$. Hence the total gate complexity is $\mathcal{O}\left(N^{7/6}\log^2(N)\,T\ \mathrm{polylog}\left(\frac{N^{7/6}T}{\epsilon}\right)\right)$.

3.5 Multi-Product Formula Based on High Order Product Formulas

We have established a complexity estimate for the second-order-based MPF. We now examine the effect of using product formulas of different orders as the base sequence. Our main observation is that the convergence order of the base sequence affects the commutator factor in the overall complexity of the algorithm.

3.5.1 First-Order Trotter

Let us first discuss the case $p = 1$ as an illustrative example. Since the first-order Trotter formula is not symmetric, its BCH expansion contains both odd- and even-order terms:

$$(U_1(\Delta/k))^k = \exp\left(-iH\Delta + \sum_{j\geq 2} E_j^{(1)} \frac{\Delta^j}{k^{j-1}}\right). \tag{3.84}$$

Here we add the superscript (1) to the operators E_j to emphasize that the $E_j^{(1)}$ arise in the expansion of the first-order Trotter formula and may differ from the operators E_j used in Section 3.3. Then Lemma 3.3.2, implies that the order condition for an mth-order MPF is

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ (k_1^{(1)})^{-1} & (k_2^{(1)})^{-1} & \cdots & (k_m^{(1)})^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ (k_1^{(1)})^{-m+1} & (k_2^{(1)})^{-m+1} & \cdots & (k_m^{(1)})^{-m+1} \end{pmatrix} \begin{pmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_m^{(1)} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.85}$$

Notice that a well-conditioned solution of Equation (3.85) can be obtained directly from that of Equation (3.4) by setting $k_j^{(1)} = k_j^2$ and $a_j^{(1)} = a_j$. The corresponding 1-norms become $\|\vec{a}^{(1)}\|_1 = \mathcal{O}(\log m)$ and $\|\vec{k}^{(1)}\|_1 = \mathcal{O}(m^4 (\log m)^2)$. The query complexity of the first-order-based MPF can be established by following the same steps as in Section 3.3. The only difference is that all even-order terms are also present in the BCH expansion. Consequently, in the commutator factor μ, one must also include commutators involving an even number of Hamiltonians, which allows the indices $(j_1, \ldots, j_l)$ in Equation (3.49) to be arbitrary positive integers.

Corollary 3.5.1. *Consider the Hamiltonian simulation problem up to time T using 1st-based MPF. It suffices to use $\mathcal{O}(\mu^{(1)} T \operatorname{poly}\log\left(\mu^{(1)} T/\epsilon\right))$ queries to controlled version of U_1 in order to obtain an ϵ-approximation with probability at least $1-\epsilon$. Here, $\mu^{(1)} = \sup_{m \geq M} \mu_m^{(1)}$ for a sufficiently large M, and*

$$\mu_m^{(1)} = \sup_{(j,l): j \geq m, 1 \leq l \leq m} \lambda_{j,l}^{(1)}, \qquad \lambda_{j,l}^{(1)} \geq \left(\sum_{\substack{j_1, \cdots, j_l \in \mathbb{Z}^+, \\ j_1 + \cdots + j_l = j}} \prod_{\kappa=1}^{l} \alpha_{\mathrm{comm}, j_\kappa + 1} \right)^{\frac{1}{j+l}}. \tag{3.86}$$

3.5.2 High-Order Symmetric Product Formula

We now discuss the $2p$th-order MPF for $p \geq 1$. We assume that the underlying product formula is symmetric; that is, $U_{2p}(s) U_{2p}(-s) = I$. The BCH expansion then takes the form

$$\left(U_{2p}(\Delta/k)\right)^k = \exp\left(-iH\Delta + \sum_{j \in 2\mathbb{Z}^+, j \geq 2p} E_{j+1}^{(2p)} \frac{\Delta^{j+1}}{k^j} \right). \tag{3.87}$$

Notice that, by symmetry, the expansion involves only odd-order terms, and the perturbation Hamiltonian starts at order $2p+1$ due to the convergence order of the product formula. There-

fore, the order condition for the corresponding $2m$th-order MPF becomes

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ (k_1^{(2p)})^{-2p} & (k_2^{(2p)})^{-2p} & \cdots & (k_m^{(2p)})^{-2p} \\ (k_1^{(2p)})^{-2p-2} & (k_2^{(2p)})^{-2p-2} & \cdots & (k_m^{(2p)})^{-2p-2} \\ \vdots & \vdots & \ddots & \vdots \\ (k_1^{(2p)})^{-2m+2} & (k_2^{(2p)})^{-2m+2} & \cdots & (k_m^{(2p)})^{-2m+2} \end{pmatrix} \begin{pmatrix} a_1^{(2p)} \\ a_2^{(2p)} \\ a_3^{(2p)} \\ \vdots \\ a_m^{(2p)} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.88}$$

Unlike the first- and second-order-based MPFs, it is not clear whether Equation (3.88) admits a well-conditioned solution, in the sense that both $\|\vec{a}^{(2p)}\|$ and $\|\vec{k}^{(2p)}\|$ are at most polynomial in m. Nevertheless, the numerical results in [LKW19] suggest the existence of well-conditioned solutions at least for $p = 2$, corresponding to the fourth-order-based MPF. If one assumes the existence of well-conditioned coefficients and powers, then the following complexity estimate can be established by the same approach used in Section 3.3.

Corollary 3.5.2. *Consider the Hamiltonian simulation problem up to time T using MPF based on $2p$-th order symmetric product formula. Suppose that there exist $\vec{k}^{(2p)}$ and $\vec{a}^{(2p)}$ satisfying Equation (3.88) and $\|\vec{k}^{(2p)}\|, \|\vec{a}^{(2p)}\| = \mathcal{O}(\mathrm{poly}(m))$. In order to obtain an ϵ-approximation with probability at least $1-\epsilon$, it suffices to use $\mathcal{O}(\mu^{(2p)} T \,\mathrm{poly}\log\left(\frac{\mu^{(2p)} T}{\epsilon}\right))$ queries to controlled version of U_{2p}. Here, $\mu^{(2p)} = \sup_{m \geq M} \mu_m^{(2p)}$ for a sufficiently large M, and*

$$\mu_m^{(2p)} = \sup_{(j,l): j \geq 2m, 1 \leq l \leq m} \lambda_{j,l}^{(2p)}, \qquad \lambda_{j,l}^{(2p)} \geq \left(\sum_{\substack{j_1, \cdots, j_l \in 2\mathbb{Z}^+, \\ j_1, \cdots, j_l \geq 2p, \\ j_1 + \cdots + j_l = j}} \prod_{\kappa=1}^{l} \alpha_{\mathrm{comm}, j_\kappa + 1} \right)^{\frac{1}{j+l}}. \tag{3.89}$$

3.5.3 Effects and Comparison

We now compare MPFs based on different base sequences. A caveat is that we assume the existence of well-conditioned MPFs, which remains to be further explored. From Corollary 3.3.6, Corollary 3.5.1, and Corollary 3.5.2, we observe that the scalings in the evolution time T and precision parameter ϵ are essentially the same, up to logarithmic factors: they are nearly linear in T and polylogarithmic in $1/\epsilon$, achieving near-optimal time and precision dependence. The only difference lies in the commutator factor $\mu^{(p)}$, which is the supremum of another commutator factor $\lambda_{j,l}^{(p)}$. We observe that MPFs based on higher-order product formulas depend on nested commutators of greater depth. Specifically, for a pth-order-based MPF, the commutator factor $\mu^{(p)}$ depends on $\alpha_{\mathrm{comm},k}$ for all $k \geq p+1$. Therefore, MPFs based on higher-order product formulas may be asymptotically advantageous in applications where the homogenized quantity $\alpha_{\mathrm{comm},k}^{1/k}$ decreases as k increases.

3.6 Conclusion

We have established a complexity analysis for the second-order-based MPF that incorporates both explicit commutator scaling and near-optimal dependence on evolution time and target precision. Our results demonstrate advantages over standard product formulas and leading post-Trotter methods across a range of applications. We discuss several natural directions for future work below.

Our commutator-scaling parameter depends on all nested commutators of depth at least 3. While we demonstrate the effectiveness of this scaling in several applications, analyzing nested commutators of arbitrarily high depth may remain challenging for certain systems, such

as nuclear effective field theories [Wat+23]. It is therefore of practical interest to establish commutator scaling for the 2nd-based MPF involving only low-depth nested commutators. Since the announcement of the first draft of our work, such a result has been obtained by Mizuta [Miz26] for the special class of k-local, g-extensive Hamiltonians. To the best of our knowledge, however, the general case remains open and is still worth pursuing. We expect that such a result could be obtained by refining our analysis and using a truncated BCH formula with an appropriate remainder term, since the high-depth nested commutators in our current analysis arise solely from the use of the infinite BCH series.

Beyond the 2nd-based MPF, we also provide a complexity estimate for the $2p$-th-based MPF incorporating commutator scaling. Similar open questions arise for the $2p$-th-based MPF as in the 2nd-based case. More importantly, our general result for the $2p$-th-based MPF assumes the existence of a well-conditioned $2p$-th-based MPF. It is therefore both interesting and important to construct well-conditioned MPF schemes with base sequences of order higher than two.

Our analysis proceeds by proving an upper bound on the operator-norm distance between the MPF operator U_{MP} and the ideal time-evolution operator U. This operator-norm bound provides a worst-case performance guarantee for the MPF over all initial states. However, in specific simulation instances, the relevant error metric may instead be the state-dependent error, namely $\|U_{\mathrm{MP}} |\psi_0\rangle - U |\psi_0\rangle \|$ for a given input state $|\psi_0\rangle$. Such a vector-norm error can be substantially smaller than the worst-case operator-norm bound when restricted to a subspace of states of interest. Therefore, deriving state-dependent vector-norm error bounds may lead to improved complexity estimates in practical simulations. This approach has recently been applied to standard product formulas and shown to yield certain advantages [JL00; ŞS21; SHC21;

AFL21b; YC22; Zha+22; CB24; Bur+23; Bur+24]. It would be interesting to investigate whether the MPF can similarly benefit from state-dependent error bounds.

Our analysis shows that the MPF can simultaneously achieve commutator scaling and near-optimal dependence on time and precision. Nevertheless, further complexity improvements may be possible by refining the MPF analysis or designing improved algorithms. First, although our analysis yields near-optimal dependence on the evolution time T and precision ϵ, it still contains additional polylogarithmic factors in T, and the dependence on $\log(1/\epsilon)$ has degree greater than one. By comparison, qubitization and quantum singular value transformation algorithms for quantum simulation achieve query complexity $\mathcal{O}(\|H\|T + \log(1/\epsilon))$, where $\|H\|$ is the spectral norm of the Hamiltonian. This scaling is exactly linear in both T and $\log(1/\epsilon)$ and is optimal, matching known lower bounds. However, current analyses indicate that qubitization and quantum singular value transformation methods do not exploit commutator scaling. Thus, it remains open whether one can design quantum simulation algorithms that combine commutator scaling with exactly optimal time and precision dependence.

Another notable feature of optimal time-independent quantum simulation algorithms is additive scaling between precision and other problem parameters. In contrast, our current MPF analysis achieves only multiplicative scaling among all parameters. It is therefore an interesting open problem to develop time-independent quantum simulation algorithms that simultaneously exhibit all three desirable features: commutator scaling, optimal time and precision dependence, and additive scaling.

Chapter 4: Linear Combination of Hamiltonian Simulation with Commutator Scaling

The results presented in this chapter are based on joint work with Dong An and Konstantina Trivisa [AAT26] and are included here with their permission. All aspects of the work were carried out by Junaid Aftab. Dong An and Konstantina Trivisa acted as mentors for this project.

4.1 Introduction

Recall from Section 1.3.2 that the objective of time-dependent quantum simulation problem for linear differential equations is to prepare a quantum state $|u(T)\rangle$ evolving according to the equation

$$\frac{d}{dt}u(t) = -A(t)u(t) + b(t), \quad t \in [0, T]. \tag{4.1}$$

Several quantum algorithms have been developed for linear differential equations of the form Equation (4.1) (see, e.g., [Ber14; Ber+17; CL20; Kro23; FLT23; BC24; JLY24; LS26]). Existing approaches mostly based on the Quantum Linear System Algorithm (QLSA) [HHL09; CKS17; Cos+22; Dal26], typically embedding $A(t)$ into a larger dilated matrix and solving an associated enlarged linear system with additional implementation and complexity overhead.

Recently, An et al. [ALL23; ACL26] and Low and Somma [LS25] introduced and analyzed the Linear Combination of Hamiltonian Simulation (LCHS) framework, which expresses the solution of Equation (4.1) as an integral over unitary evolutions. This approach simulates *non-unitary dynamics* by combining Hamiltonian simulation algorithms developed for *unitary dynamics*. The resulting algorithms achieve optimal state-preparation cost and optimal scaling in both the evolution time and the target precision.

From a broader perspective, LCHS is a special case of the Linear Combination of Unitaries (LCU) framework of Childs and Wiebe [CW12a], as it represents non-unitary dynamics through linear combinations of unitary simulation primitives. A related LCU-based construction is the Multi-Product Formula (MPF) [LKW19; AAT24; MIF24; Miz26], combining Lie–Trotter–Suzuki product formulas [Llo96] to obtain high-order Hamiltonian simulation algorithms. Specifically, MPFs construct these high-order approximations by linearly combining sequences of low-order Trotter steps with carefully chosen weights. This systematic cancellation of algorithmic errors circumvents the deep exponential nesting required by standard high-order product formulas, achieving near-optimal time and precision dependence while maintaining favorable commutator scaling from product formulas [Chi+21]. However, for general non-unitary dynamics, existing approaches, including LCHS and other advanced algorithms, cannot currently achieve both of these desirable properties simultaneously. This limitation naturally motivates the central question of our work: can we design an efficient quantum algorithm for general non-unitary systems with both near-optimal scaling (in time and precision) and commutator scaling?

In this chapter, we answer this question affirmatively in a quadrature-dependent sense. We develop an LCHS–MPF algorithm for non-unitary dynamics that applies to both time-independent and time-dependent settings, and illustrate its use in applications arising from linear

differential equations and open quantum systems. In this algorithm, LCHS reduces non-unitary simulation to a family of Hamiltonian simulations, while MPFs implement these simulations with commutator-sensitive complexity. The key observation is that the outer LCHS quadrature does not merely control discretization error. Rather, the quadrature nodes also determine the Hamiltonians passed to the inner simulation routine, and therefore affect the commutator structure, step-size restrictions, LCU normalization, weighted error accumulation, and final query complexity. Thus, in our setting, quadrature selection and commutator-sensitive Hamiltonian simulation cannot be analyzed independently. By tracking this dependence explicitly, we obtain a quantum algorithm for non-unitary simulation whose complexity reflects both the quadrature structure of LCHS and the commutator scaling of MPFs.

4.1.1 Related Works

Several recent works study when non-unitary quantum simulation can inherit structural advantages from unitary Hamiltonian simulation. Wang et al. [Wan+26b] recently derived commutator-based Trotter error bounds for Lindbladian simulation. Their work is close in motivation to ours. Both analyses ask whether the favorable commutator scaling of Hamiltonian product formulas persists in dissipative dynamics. The technical settings, however, are distinct. Their bounds apply directly to product-formula approximations of Lindblad generators and use Richardson extrapolation for observable estimation, whereas our approach first uses the LCHS representation to reduce dissipative linear evolution to a quadrature-indexed family of Hamiltonian simulations. Thus, the commutators appearing in our analysis are those of the Hamiltonians generated by the LCHS quadrature, and the main issue is to track how this

quadrature dependence enters the MPF error and complexity estimates.

A second related direction concerns quantum algorithms for non-unitary dynamics based on matrix-function and eigenvalue-transformation representations. An et al. [An+24] extended the LCHS approach from matrix exponentials to broader matrix eigenvalue transformations represented through Laplace transforms. Low and Su [LS24] introduced quantum eigenvalue processing, an eigenvalue transformation framework for non-normal matrices that addresses related non-Hermitian phenomena from a different algorithmic perspective. These works complement the present paper by broadening the class of matrix functions that can be implemented on quantum computers. By contrast, the focus of the present work is not to enlarge the class of admissible matrix functions, but to analyze how the post-quadrature LCHS representation interacts with a commutator-sensitive Hamiltonian simulation primitive.

Contour-integral methods provide another point of comparison. Takahira et al. [Tak+21] used Cauchy's integral formula and block encoding to implement matrix functions as linear combinations of shifted matrix inverses. More recently, Jiang and An [JA26] analyzed the complexity of contour-integral-based quantum eigenvalue transformations, with applications to Hamiltonian simulation, matrix polynomials, and linear differential equations. Wang et al. [Wan+25] developed a contour-based matrix-decomposition framework for non-unitary dynamics using a finite decomposition derived from Cauchy's residue theorem. These approaches are related to LCHS through analytic matrix-function representations and quadrature or discretization arguments. However, their elementary operations are typically shifted resolvents or Hermitian transformations, rather than Hamiltonian evolutions of the form $e^{-i(H+kL)t}$ whose commutator structure is tracked in the present work.

Other transform-based approaches have also been developed in the literature. The tech-

nique of *Schrödingerization* [JLY23; JLY24; Jin+25; JLW25] converts non-unitary linear dynamics into unitary dynamics in one higher dimension, achieving optimal or near-optimal matrix-query dependence after suitable smooth initialization. Jin et al. [JMZ26] proposed a transmutation-based method for dissipative diffusion generated by positive semi-definite operators, expressing the diffusion semigroup via the Kannai transform as a Gaussian-weighted superposition of unitary wave propagators. A recent Poisson-summation framework by Wang et al. [Wan+26a] further emphasizes the role of spectral aliasing and discretization in non-unitary matrix transformations. These works illustrate the principle that non-unitary simulation can often be reduced to superpositions of unitary or resolvent-type primitives.

4.1.2 Organization

The remainder of the chapter is organized as follows. Section 4.2 reviews recent developments in Multi-Product Formulas not covered in Chapter 3. Similarly, reviews recent developments in Linear Combination Of Hamiltonian Simulation algorithm not covered in Section 1.3.2. Section 4.4 analyzes the pre-quadrature approximation, truncation, and inner-simulation errors for the time-independent homogeneous case of our algorithm. Section 4.5 develops the post-quadrature LCHS–MPF error and complexity analysis, including the abstract theorem, commutator-scaling specializations, and comparison of quadrature rules. Section 4.6 extends the framework to non-homogeneous and time-dependent settings. Section 4.7 discusses applications to the fractional diffusion equation, the advection–diffusion equation, and no-jump dynamics regime for a dissipative Ising model model.

4.2 Multi-Product Formulas

Chapter 3 provided the first complete analysis of the error scaling of multi-product formulas (MPFs). However, the results in Chapter 3 have certain limitations that have recently been addressed by other authors. The first limitation is that the infimum in Theorem 3.3.4 is taken over commutators of arbitrarily large depth. If $\alpha_{\mathrm{comm},j} = \Omega(N^{j^\gamma})$ for some $\gamma > 1$, then this infimum vanishes, and no positive admissible Δ exists. Mizuta [Miz26] addressed this issue for the special class of q-local, g-extensive Hamiltonians by showing that only commutators up to finite depth need to be considered in the short-time MPF bound.

Proposition 4.2.1 (Theorem 4 in [Miz26]). *Let $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$ be a q-local, g-extensive Hamiltonian*[1]*. Fix $m \geq 1$. For a truncation parameter $p_0 \in \mathbb{N}$, define*

$$\mu_{m,p_0} := \sup_{\substack{j,l\in\mathbb{N} \\ j\geq 2m,\ 1\leq l\leq \lfloor j/2\rfloor}} \left(\sum_{\substack{2\leq j_1,\ldots,j_l\leq p_0-1 \\ j_1+\cdots+j_l=j}} \prod_{\kappa=1}^{l} \alpha_{\mathrm{comm},\, j_\kappa+1}(H) \right)^{\frac{1}{j+l}}. \tag{4.2}$$

Let $\epsilon \in (0,1)$, $N \geq 1$ and let $p_0 := p_0(N,\epsilon) = \lceil \log(3N/\epsilon) \rceil$. Choose a $\Delta > 0$ such that $0 < \Delta \leq \min\left\{ \frac{1}{16e^3 p_0 q g},\ \frac{1}{4\mu_{m,p_0}} \right\}$. Then we have

$$\|U_{\mathrm{MP}}(\Delta) - U(\Delta)\| \leq 2e^{1/2} \|\vec{a}\|_1 (\mu_{m,p_0}\Delta)^{2m+1} + \|\vec{a}\|_1 \|\vec{b}\|_1 \epsilon. \tag{4.3}$$

The second limitation is that the results in Chapter 3 apply only to the time-independent case. This issue was recently resolved for the special class of q-local, g-extensive Hamiltonians

[1] Let $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$. For each γ, let X_γ denote the support of H_γ. We say that H is q-local if $|X_\gamma| \leq q$ for all γ and $[H_\gamma, H_{\gamma'}] = 0$ whenever $X_\gamma \cap X_{\gamma'} = \varnothing$. Moreover, we say that H is g-extensive if $\sum_{\gamma:\, j\in X_\gamma} \|H_\gamma\| \leq g$ for all j.

by Mizuta et al. [MIF24], who provided an analogue of Proposition 4.2.1 for the time-dependent case.

Proposition 4.2.2 (Specialization of Theorem 17 in [MIF24]). *Let* $H(t) = \sum_{\gamma=1}^{\Gamma} H_\gamma(t)$ *be a smooth time-dependent* q*-local,* g*-extensive Hamiltonian*[2]*. Moreover, assume there exists an* $f > 0$ *such that* $\sup_{\tau\in[0,\Delta]} \sum_{\gamma:\, j\in X_\gamma} \left\| \frac{d^n}{d\tau^n} H_\gamma(\tau) \right\| \leq f^n g$ *for all* j *and all* $n \geq 1$*. Fix* $m \geq 1$*. For* $1 \leq \gamma \leq \Gamma + 1$*, define*

$$D_\gamma(\tau) := \begin{cases} \mathrm{ad}_{H_\gamma(\tau)} + i\dfrac{d}{d\tau}, & 1 \leq \gamma \leq \Gamma, \\ \Gamma \dfrac{d}{d\tau}, & \gamma = \Gamma + 1. \end{cases} \tag{4.4}$$

For $p \geq 1$*, define* $\alpha_{\mathrm{comm},p}(\tau) := \sum_{\gamma_0=1}^{\Gamma} \sum_{\gamma_1,\ldots,\gamma_p=1}^{\Gamma+1} \left\| \left(D_{\gamma_1}(\tau) \cdots D_{\gamma_p}(\tau) H_{\gamma_0}(\tau)\right) H_{\gamma_0}(\tau) \right\|$*. For a truncation parameter* $p_0 \in \mathbb{N}$*, define* $\mu_{p_0} := \sup_{\tau\in[0,\Delta]} \max_{\substack{p\in\mathbb{N} \\ 2\leq p\leq p_0}} (\alpha_{\mathrm{comm},p}(\tau))^{\frac{1}{p+1}}$*. Let* $\epsilon \in (0,1)$*,* $N \geq 1$*, and let* $p_0 := p_0(N,\epsilon) = \lceil \log(2N/\epsilon) \rceil$*. Assume* $p_0 \geq 2$ *and choose a* $\Delta > 0$ *such that* $0 < \Delta \leq \min\left\{ \frac{1}{8e^3 p_0 (2qg + 2\Gamma f)},\ \frac{1}{8\mu_{p_0}} \right\}$*. Then we have*

$$\|U_{\mathrm{MP}}(\Delta) - U(\Delta)\| \ \leq\ \|\vec{a}\|_1 (4\mu_{p_0}\Delta)^{2m+1} + \|\vec{a}\|_1 \|\vec{b}\|_1 \epsilon. \tag{4.5}$$

4.3 Linear Combination Of Hamiltonian Simulation

We summarized the Linear Combination of Hamiltonian Simulation (LCHS) framework in Section 1.3.2. Here we discuss recent developments in LCHS that are used in this chapter.

[2] In the time-dependent case, we say that H is g-extensive if $\sup_{\tau\in[0,\Delta]} \sum_{\gamma:\, j\in X_\gamma} \|H_\gamma(\tau)\| \leq g$ for all j. The definition of q-local remains the same.

4.3.1 An et al.'s Formalism

A limitation of the LCHS formalism discussed in Section 1.3.2 is that, since the kernel $(1+k^2)^{-1} = \Omega(k^{-2})$ decays only quadratically, the resulting quantum algorithm incurs a computational overhead of $\mathcal{O}(1/\epsilon)$. Subsequently, An et al. [ACL26] proposed a more general LCHS framework. Their key observation was that Equation (1.12) need only hold for $x \geq 0$. That is, one requires a kernel function $\hat{f}(k)$ such that

$$e^{-x} = \int_{\mathbb{R}} \frac{\hat{f}(k)}{\pi(1-ik)} e^{-ikx}\, dk, \quad x \geq 0. \tag{4.6}$$

They used this observation and introduced a family of identities demonstrating that Equation (4.6) holds in the matrix setting. The identities take the form

$$\mathcal{T} e^{-\int_0^t A(s)\, ds} = \int_{\mathbb{R}} \frac{\hat{f}(k)}{\pi(1-ik)} U_k(t) dk, \quad t \in [0,T] \tag{4.7}$$

in the case $b(t) \equiv 0$. Once again, the case $b(t) \neq 0$ is then derived via an application of Duhamel's principle. An et al. proposed using $\hat{f}_\beta(z) = \frac{1}{2\pi e^{-2^\beta} e^{(1+iz)^\beta}}$ for $\beta \in (0,1)$ and derived that choosing $K = \mathcal{O}((\log(1/\epsilon))^{1/\beta})$ suffices in the integral truncation step to achieve ϵ-accuracy. The resulting quantum algorithm can solve Equation (4.1) for time T with error at most ϵ using

$$\mathcal{O}\left(\frac{\|u(0)\| + \|b\|_{L^1[0,T]}}{\|u(T)\|} \alpha_A T \left(\log \frac{1}{\epsilon} \right)^{1+1/\beta} \right), \tag{4.8}$$

where $\alpha_A \geq \sup_{t\in[0,T]} \|A(t)\|$, queries to the input oracle for $A(t)$, invoked as part of a time-independent Hamiltonian simulation algorithm, and using $\mathcal{O}\left(\frac{\|u(0)\|+\|b\|_{L^1[0,T]}}{\|u(T)\|}\right)$ queries to the

state preparation oracles for $u(0)$.

4.3.2 Low and Somma's Formalism

Recently, Low and Somma [LS25] introduced an approximate LCHS framework that improves upon the formalism of An et al. Their key observation is that the requirement that Equation (4.6) hold exactly can be relaxed so that it need only hold approximately. That is, it suffices to find a function f such that, for $\epsilon > 0$, we have $|e^{-x} - f(x)| \leq \epsilon$ for $x \geq 0$, and such that Equation (4.6) holds only approximately. This relaxed requirement allows them to work with a broader family of kernel functions. Their main result is summarized below.

Proposition 4.3.1 (Theorem 1 in [LS25]). *For $T \geq 0$ and $n \geq 1$, let $A(t) = L(t) + iH(t) \in \mathbb{C}^{2^n \times 2^n}$ be a time-dependent matrix for $t \in [0, T]$, where $L(t)$ and $H(t)$ are Hermitian operators such that $\|L\|_{L^1[0,T]}$ and $\|H\|_{L^1[0,T]}$ are finite. Consider the uniform strip $S_{[-y_0,0]} := \{z \in \mathbb{C} : \mathrm{Im}(z) \in [-y_0, 0]\}$ for some $y_0 > 1$. On $S_{[-y_0,0]}$, assume that the complex-valued kernel function $\hat{f}(z)$ is such that $\lim_{|z|\to\infty} \hat{f}(z) = 0$, $(z + i)\hat{f}(z)$ is analytic and the residue at $z = -i$ is $\mathrm{Res}(\hat{f}, -i) = i/\sqrt{2\pi}$. For $t \in [0, T]$, we have*

$$\left\| \frac{1}{\sqrt{2\pi}} \int_{-R}^{R} \hat{f}(k) U_k(t) dk - \mathcal{T} e^{-\int_0^t A(s)\, ds} \right\| \leq \frac{1}{\sqrt{2\pi}} \int_{[-R,R]^c} |\hat{f}(k)| dk + \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} |\hat{f}(k - iy_0)| dk. \quad (4.9)$$

We briefly compare the two formalisms. The analysis of An et al. assumes that $\hat{f}(z) = \mathcal{O}(|z|^{-\alpha})$ for some $\alpha > 0$ as $|z| \to \infty$ in the lower half-plane. By contrast, Low and Somma require only that the kernel function satisfy $\lim_{|z|\to\infty} \hat{f}(z) = 0$ within an infinite strip of finite width in the lower half-plane. This weaker condition admits a broader class of kernels and leads to a more flexible approximate LCHS algorithm. We will use Low and Somma's formalism in

this chapter.

4.4 Pre-Quadrature Error Analysis

In this section, we give a pre-quadrature error analysis of the algorithm in the time-independent homogeneous case. Throughout, let $A = L + iH$, where L and H are Hermitian and $L \succeq 0$. In the pre-quadrature regime, the algorithm has three main sources of error:

1. (Approximation) The approximation error, $E_{\text{approx}}(y_0)$, arising by replacing e^{-AT} with the LCHS representation. That is, $E_{\text{approx}}(y_0) := \|\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(k) U_k(T)\, dk - e^{-AT}\|$. By Equation (4.9), after taking the limit $R \to \infty$, it is bounded by $\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} |\hat{f}(k - iy_0)|\, dk$.

2. (Truncation) The truncation error, $E_{\text{trunc}}(R)$, arising from restricting $\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(k)\, U_k(t)\, dk$ to $[-R, R]$. That is, $E_{\text{trunc}}(R) := \|\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(k) U_k(T)\, dk - \frac{1}{\sqrt{2\pi}} \int_{-R}^{R} \hat{f}(k) U_k(T)\, dk\|$. By Equation (4.9), since $U_k(T)$ is unitary, it is bounded by $\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}\setminus[-R,R]} |\hat{f}(k)|\, dk$.

3. (Inner Simulation) The simulation error $E_{\text{MPF}}(T)$, arising from replacing each $U_k(T)$ by its MPF approximation $\widetilde{U}_k(T)$.

   In what follows, we use the four-parameter kernel family introduced by Low and Somma [LS25]:

$$\hat{f}_{a,b}(k; c, d) = \frac{(b+1)^{a-1}}{\sqrt{2\pi}} \frac{e^{d(1-ik)} e^{-\frac{k^2+1}{4c^2}}}{(1-ik)(b+ik)^{a-1}}, \quad a \in \mathbb{N},\ b, c > 0,\ d \in \mathbb{R}. \tag{4.10}$$

We call $\vec{\theta} = (a, b, c, d)$ the kernel profile and write $\hat{f}(k)$ for $\hat{f}_{a,b}(k; c, d)$ when clear. Section 4.4.1 discusses the approximation error, Section 4.4.2 the truncation error, Section 4.4.3 the inner simulation error, and Section 4.4.4 discusses the coupling between the truncation and inner simulation errors.

4.4.1 Approximation Error

We first analyze the approximation error $E_{\text{approx}}(y_0)$.

Lemma 4.4.1. *Fix a kernel profile $\vec{\theta} = (a, b, c, d)$ and $y_0 > 1$. We have*

$$E_{\text{approx}}(y_0) \le \frac{(b+1)^{a-1}}{2\pi} e^{\,d(1-y_0)+\frac{y_0^2-1}{4c^2}} \int_{\mathbb{R}} \frac{e^{-k^2/(4c^2)}\,dk}{\sqrt{(y_0-1)^2+k^2}\,((b+y_0)^2+k^2)^{(a-1)/2}}. \tag{4.11}$$

Proof. Set $z = k - iy_0$. Since $1 - iz = (1 - y_0) - ik$ and $b + iz = (b + y_0) + ik$, we have $|1 - iz| = \sqrt{(y_0 - 1)^2 + k^2}$ and $\sqrt{(b + y_0)^2 + k^2}$. We also have $|e^{d(1-iz)}| = e^{d(1-y_0)}$. Moreover, since $z^2 = (k-iy_0)^2 = k^2-2iky_0-y_0^2$, we have $|e^{-(z^2+1)/(4c^2)}| = e^{-(k^2-y_0^2+1)/(4c^2)} = e^{(y_0^2-1)/(4c^2)}e^{-k^2/(4c^2)}$. Therefore, we have

$$|\hat{f}_{a,b}(k - iy_0; c, d)| = \frac{(b+1)^{a-1}}{\sqrt{2\pi}} e^{\,d(1-y_0)+\frac{y_0^2-1}{4c^2}} \frac{e^{-k^2/(4c^2)}}{\sqrt{(y_0-1)^2+k^2}\,((b+y_0)^2+k^2)^{(a-1)/2}}. \tag{4.12}$$

Integrating over $\mathbb{R}$ and multiplying by $1/\sqrt{2\pi}$ gives Equation (4.11). □

The integral in Lemma 4.4.1 has an exact representation in terms of the confluent hypergeometric function, but that form is less transparent. We instead use the bound $((b + y_0)^2 + k^2)^{(a-1)/2} \ge (b + y_0)^{a-1}$ to obtain the simpler upper bound

$$E_{\text{approx}}(y_0) \le \frac{(b+1)^{a-1}}{(b+y_0)^{a-1}} \frac{e^{\,d(1-y_0)+\frac{y_0^2-1}{4c^2}}}{2\pi} \int_{\mathbb{R}} \frac{e^{-k^2/(4c^2)}}{\sqrt{(y_0-1)^2+k^2}}\,dk. \tag{4.13}$$

This bound is explicit and retains the essential dependence on y_0 and a, b, c, d. Lemma 4.4.2, which is proved in Section B.1, evaluates this integral form.

Lemma 4.4.2. *For $s, \sigma > 0$, we have*

$$\int_{\mathbb{R}} \frac{e^{-k^2/(4\sigma^2)}}{\sqrt{k^2+s^2}}\, dk = e^{s^2/(8\sigma^2)} K_0\left(\frac{s^2}{8\sigma^2}\right), \tag{4.14}$$

where K_0 denotes the modified Bessel function of the second kind.

Using Lemma 4.4.2 with $s = y_0 - 1$, $\sigma = c$ and $K_0(x) \le \sqrt{\frac{\pi}{2x}} e^{-x}$ for $x > 0$, we obtain

$$E_{\text{approx}}(y_0) \le \frac{c}{\sqrt{\pi}} \frac{(b+1)^{a-1}}{(y_0-1)(b+y_0)^{a-1}} e^{d(1-y_0)+\frac{y_0^2-1}{4c^2}}. \tag{4.15}$$

Equation (4.15) motivates minimizing the right-hand side and identifying kernel profiles for which this bound is at most ϵ_{approx}. In Lemma 4.4.3, which is proved in Section B.2 , we first study existence and uniqueness of minimizers with respect to y_0.

Lemma 4.4.3. *Fix a kernel profile $\vec{\theta} = (a, b, c, d)$. Define the surrogate cost functional*

$$B(y_0) := \frac{(b+1)^{a-1}}{(y_0-1)(b+y_0)^{a-1}} e^{d(1-y_0)+\frac{y_0^2-1}{4c^2}}, \quad y_0 > 1. \tag{4.16}$$

Then $\frac{d}{dy_0} \log B(y_0) = -d + \frac{y_0}{2c^2} - \frac{1}{y_0-1} - \frac{a-1}{b+y_0}$. Consequently, any stationary point y_0^* of $\log B$ satisfies

$$\frac{y_0^*}{2c^2} - d = \frac{1}{y_0^*-1} + \frac{a-1}{b+y_0^*}. \tag{4.17}$$

Moreover, we have $\frac{d^2}{dy_0^2} \log B(y_0) = \frac{1}{2c^2} + \frac{1}{(y_0-1)^2} + \frac{a-1}{(b+y_0)^2} > 0$. Therefore, $\log B$ is strictly convex on $(1, \infty)$ and has at most one stationary point. Any such stationary point is the unique global minimizer of B. Moreover, B admits a global minimizer.

Lemma 4.4.3 implies that, for a fixed kernel profile $\vec{\theta} = (a, b, c, d)$, the surrogate cost $B(y_0)$

admits a unique minimizer y_0^*. If $\epsilon_{\text{approx}} > 0$ denotes the approximation error budget, then

$$\text{there exists } y_0 \in (1,\infty) \text{ such that } \frac{c}{\sqrt{\pi}} B(y_0) \le \epsilon_{\text{approx}} \iff \frac{c}{\sqrt{\pi}} B(y_0^*) \le \epsilon_{\text{approx}}. \tag{4.18}$$

Equation (4.18) gives a sufficient condition for $E_{\text{approx}}(y_0) \le \epsilon_{\text{approx}}$. It remains to determine when y_0^* satisfies $\frac{c}{\sqrt{\pi}} B(y_0^*) \le \epsilon_{\text{approx}}$. Proposition 4.4.4 gives a condition in terms of the parameter d.

Proposition 4.4.4. *Let $\epsilon_{\text{approx}} > 0$. For $s > 1$, define*

$$\Phi(s) := \frac{s}{2c^2} - \frac{1}{s-1} - \frac{a-1}{b+s}, \tag{4.19}$$

$$\Psi(s) := \frac{c}{\sqrt{\pi}(s-1)} \left(\frac{b+1}{b+s}\right)^{a-1} \exp\left(1 - \frac{(s-1)^2}{4c^2} + \frac{(a-1)(s-1)}{b+s}\right). \tag{4.20}$$

Φ is strictly increasing on $(1,\infty)$, Ψ is strictly decreasing on $(1,\infty)$, and there exists a unique $s_\epsilon \in (1,\infty)$ such that $\Psi(s_\epsilon) = \epsilon_{\text{approx}}$. Moreover, for a fixed kernel profile $\vec{\theta} = (a,b,c,d)$, let y_0^* be the unique minimizer of B from Lemma 4.4.3. Then $\frac{c}{\sqrt{\pi}} B(y_0^*) \le \epsilon_{\text{approx}}$ if and only if $d \ge \Phi(s_\epsilon)$.

Proof. By Lemma 4.4.3, B admits a unique minimizer $y_0^* \in (1,\infty)$ satisfying $d = \Phi(y_0^*)$. We have $\Phi'(s) > 0$, so Φ is strictly increasing on $(1,\infty)$. Moreover, $\Phi(s) \to -\infty$ as $s \to 1^+$ and $\Phi(s) \to +\infty$ as $s \to \infty$. Hence, Φ is a bijection from $(1,\infty)$ onto $\mathbb{R}$. Using $d = \Phi(y_0^*)$ in $\frac{c}{\sqrt{\pi}} B(y_0^*)$ gives $\frac{c}{\sqrt{\pi}} B(y_0^*) = \Psi(y_0^*)$. We now show that Ψ is strictly decreasing. Differentiating $\log \Psi$ gives

$$\frac{d}{ds} \log \Psi(s) = -\frac{1}{s-1} - \frac{s-1}{2c^2} - \frac{(a-1)(s-1)}{(b+s)^2} < 0. \tag{4.21}$$

Thus Ψ is strictly decreasing on $(1,\infty)$. Moreover, we have $\Psi(s) \to +\infty$ as $s \to 1^+$ and $\Psi(s) \to 0$

as $s \to \infty$. Hence, there exists a unique $s_\epsilon \in (1, \infty)$ such that $\Psi(s_\epsilon) = \epsilon_{\text{approx}}$. Finally, since Ψ is strictly decreasing and $d = \Phi(y_0^*)$, with Φ strictly increasing, we obtain

$$\frac{c}{\sqrt{\pi}} B(y_0^*) \le \epsilon_{\text{approx}} \iff \Psi(y_0^*) \le \epsilon_{\text{approx}} \iff y_0^* \ge s_\epsilon \iff d = \Phi(y_0^*) \ge \Phi(s_\epsilon). \tag{4.22}$$

This completes the proof. □

Proposition 4.4.4 shows that taking d above an explicit lower bound ensures $E_{\text{approx}}(y_0) \le \epsilon_{\text{approx}}$. The threshold is determined by the unique solution of $\Psi(s) = \epsilon_{\text{approx}}$, which can be computed numerically. Corollary 4.4.5, which is proved in Section B.3, gives explicit sufficient conditions on c and d and determines a corresponding choice of y_0.

Corollary 4.4.5. *Let $\epsilon_{\text{approx}} \in (0, 1)$ and let $\vec{\theta} = (a, b, c, d)$ be a kernel profile. Then the following statements hold.*

1. *Define $\omega_\epsilon := W(e^{2a}/(2\pi\epsilon_{\text{approx}}^2))$ and $\widetilde{x}_0 := 1 + c\sqrt{2\omega_\epsilon}$, where W denotes the Lambert W-function. If*

$$d \ge \frac{\widetilde{x}_0}{2c^2} - \frac{1}{\widetilde{x}_0 - 1} - \frac{a-1}{b + \widetilde{x}_0}, \tag{4.23}$$

   then there exists $y_0 > 1$ such that $\frac{c}{\sqrt{\pi}} B(y_0) \le \epsilon_{\text{approx}}$. Consequently, $E_{\text{approx}}(y_0) \le \epsilon_{\text{approx}}$.

2. *Define $\overline{x}_0 := 1 + 2c\sqrt{a + \log(1/\epsilon_{\text{approx}})}$. If*

$$d \ge \frac{\overline{x}_0}{2c^2} - \frac{1}{\overline{x}_0 - 1} - \frac{a-1}{b + \overline{x}_0}, \tag{4.24}$$

   then there exists $y_0 > 1$ such that $\frac{c}{\sqrt{\pi}} B(y_0) \le \epsilon_{\text{approx}}$. Consequently, $E_{\text{approx}}(y_0) \le \epsilon_{\text{approx}}$.

3. *The sufficient lower bound on d in Equation (4.24) is obtained by evaluating $\Phi(s)$ at $\overline{x}_0$. In particular, $\Phi(\overline{x}_0) = \frac{1}{2c^2} + \frac{1}{c}\sqrt{a + \log(1/\epsilon_{\text{approx}})} - \frac{1}{\overline{x}_0 - 1} - \frac{a-1}{b+\overline{x}_0}$. For fixed b and c, as either $a \to \infty$ or $\epsilon_{\text{approx}} \to 0$, or both, we have $\Phi(\overline{x}_0) = \frac{1}{2c^2} + \mathcal{O}\left(\frac{1}{c}\sqrt{a + \log(1/\epsilon_{\text{approx}})}\right)$. Therefore, a sufficient lower bound on d grows at most on the order of $\frac{1}{2c^2} + \frac{1}{c}\sqrt{a + \log(1/\epsilon_{\text{approx}})}$.*

4.4.2 Truncation Error

We now compute the truncation error in Lemma 4.4.6.

Lemma 4.4.6. *Fix a kernel profile $\vec{\theta} = (a, b, c, d)$ and $R > 0$. We have*

$$E_{\text{trunc}}(R) \le \frac{c}{\sqrt{\pi}} \left(\frac{b+1}{b}\right)^{a-1} e^{d - \frac{1}{4c^2}} \operatorname{erfc}\left(\frac{R}{2c}\right). \tag{4.25}$$

Proof. It is clear that $|\hat{f}_{a,b}(k; c, d)| = \frac{(b+1)^{a-1} e^{d-\frac{1}{4c^2}}}{\sqrt{2\pi}} \frac{e^{-k^2/(4c^2)}}{\sqrt{1+k^2}(b^2+k^2)^{(a-1)/2}}$. Hence, we have

$$\begin{aligned} E_{\text{trunc}}(R) &\le \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}\setminus[-R,R]} |\hat{f}_{a,b}(k; c, d)|\, dk && (4.26)\\ &= \frac{(b+1)^{a-1} e^{d-\frac{1}{4c^2}}}{2\pi} \int_{|k|>R} \frac{e^{-k^2/(4c^2)}}{\sqrt{1+k^2}(b^2+k^2)^{(a-1)/2}}\, dk && (4.27)\\ &= \frac{(b+1)^{a-1} e^{d-\frac{1}{4c^2}}}{\pi} \int_{R}^{\infty} \frac{e^{-k^2/(4c^2)}}{\sqrt{1+k^2}(b^2+k^2)^{(a-1)/2}}\, dk. && (4.28) \end{aligned}$$

For $k \ge 0$, the elementary inequalities $\sqrt{1+k^2} \ge 1$ and $(b^2+k^2)^{(a-1)/2} \ge b^{a-1}$ imply that

$$E_{\text{trunc}}(R) \le \frac{(b+1)^{a-1} e^{d-\frac{1}{4c^2}}}{\pi b^{a-1}} \int_R^\infty e^{-k^2/(4c^2)}\, dk = \frac{(b+1)^{a-1} e^{d-\frac{1}{4c^2}}}{\pi b^{a-1}} c\sqrt{\pi} \operatorname{erfc}\left(\frac{R}{2c}\right). \tag{4.29}$$

This gives Equation (4.25). □

The bound in Lemma 4.4.6 shows that $E_{\text{trunc}}(R) \to 0$ as $R \to \infty$. This motivates consid-

ering the set of radii for which the truncation error is below a fixed tolerance $\epsilon' > 0$

$$\mathcal{H}_{\epsilon'} := \{R > 0 : E_{\text{trunc}}(R) < \epsilon'\}. \tag{4.30}$$

Corollary 4.4.7 gives a computable lower bound ensuring $R \in \mathcal{H}_{\epsilon'}$.

Corollary 4.4.7. *Let $\epsilon' > 0$. The set $\mathcal{H}_{\epsilon'}$ is non-empty for every $\epsilon' > 0$. For a kernel profile $\vec{\theta} = (a, b, c, d)$, define*

$$\vartheta(\vec{\theta}) := \frac{\sqrt{\pi}}{c} \left(\frac{b}{b+1}\right)^{a-1} \frac{\epsilon'}{e^{d-\frac{1}{4c^2}}}. \tag{4.31}$$

If $\vartheta(\vec{\theta}) < 1$, then every $R > 2c\ \mathrm{erfc}^{-1}(\vartheta(\vec{\theta}))$ is in $\mathcal{H}_{\epsilon'}$. If $\vartheta(\vec{\theta}) \geq 1$, then every $R > 0$ is in $\mathcal{H}_{\epsilon'}$.

Proof. $\mathcal{H}_{\epsilon'} \neq \emptyset$ since $\lim_{R\to\infty} E_{\text{trunc}}(R) = 0$ by Lemma 4.4.6. If $\vartheta(\vec{\theta}) < 1$, then Lemma 4.4.6 shows that $E_{\text{trunc}}(R) < \epsilon'$ is guaranteed whenever $\mathrm{erfc}(R/(2c)) < \vartheta(\vec{\theta})$. Since erfc is strictly decreasing on $[0, \infty)$, this is equivalent to $R > 2c\ \mathrm{erfc}^{-1}(\vartheta(\vec{\theta}))$. If $\vartheta(\vec{\theta}) \geq 1$, then $\mathrm{erfc}(R/(2c)) < 1 \leq \vartheta(\vec{\theta})$, and the same estimate gives $E_{\text{trunc}}(R) < \epsilon'$ for all $R > 0$. □

4.4.3 Inner Simulation Error

We next compute the inner simulation error. For $k \in \mathbb{R}$, define $G_k := H + kL$. Since G_k consists of H and L, each nested commutator is formed by choosing either H or L at each position. This motivates the following definition.

Definition 4.4.8. Let $\mathcal{C}_{j,\ell}(H, L) := \sum_{\substack{M\in\{H,L\}^j \\ N_L(M)=\ell}} \|[M_1, M_2, \ldots, M_j]\|$ for integers $j \geq 2$ and $0 \leq \ell \leq j$, where $N_L(M)$ denotes the number of occurrences of L in the word $M = (M_1, \ldots, M_j) \in \{H, L\}^j$.

Lemma 4.4.9 formalizes the fact that the nested commutators of G_k depend polynomially on k.

Lemma 4.4.9. *Fix $j \geq 2$. The j-fold nested commutator profile associated with the decomposition $G_k = H + kL$ satisfies $\alpha_{\mathrm{comm},j}(G_k) := \sum_{M_1,\ldots,M_j \in \{H,kL\}} \|[M_1, \ldots, M_j]\| = \sum_{\ell=0}^{j} |k|^\ell \mathcal{C}_{j,\ell}(H, L)$. In particular, $\alpha_{\mathrm{comm},j}(G_k)$ is a polynomial in $|k|$ of degree at most j, with coefficients determined by nested commutators of H and L.*

Proof. By multilinearity of nested commutators, each word in the decomposition $G_k = H + kL$ containing exactly ℓ copies of L contributes a factor k^ℓ. Hence its norm contributes $|k|^\ell$. Grouping the resulting commutator terms by the number ℓ of occurrences of L proves the claim. □

We approximate $U_k(t) = e^{-iG_k t}$ by a $2m$-th order MPF built from the second-order product formula, denoted $U_k^{(2m)}$. We state the error bound abstractly, which is instantiated later using either Theorem 3.3.4 or Proposition 4.2.1.

Assumption 4.4.10. For $R > 0$, consider the time-independent family $\{G_k\}_{|k|\leq R}$. Fix $m, N \geq 1$ and $\delta > 0$, and let $U_k^{(2m)}(t)$ denote the order-$2m$ MPF approximation to $U_k(t)$. Assume there exists a non-negative function $\Phi_m : [-R, R] \times \mathbb{N} \to \mathbb{R}_+$ depending on nested commutators of G_k and the system size N and satisfying $\Phi^*_{m,R}(N) := \sup_{|k|\leq R} \Phi_m(k, N) < \infty$, a step size $\Delta(R, N, \delta) > 0$ and a remainder term $R_m(\delta) \geq 0$ such that, for all $|k| \leq R$ and $0 < \Delta \leq \Delta(R, N, \delta)$, we have

$$\|U_k^{(2m)}(\Delta) - U_k(\Delta)\| \leq C_m \Delta^{2m+1} \Phi_m(k, N) + R_m(\delta) =: \eta_k(\Delta, k, N, \delta, m), \tag{4.32}$$

where $C_m > 0$ is constant. For simplicity, we write $\eta_k(\Delta)$ for $\eta_k(\Delta, k, N, \delta, m)$.

Remark 4.4.11. Assumption 4.4.10 captures different MPF error scalings through $\Phi_m(k, N)$. Under Theorem 3.3.4, expanding G_k gives a convergent series in $|k|$. In contrast, Proposition 4.2.1 involves commutator products followed by a $(j+l)$-th root. This yields non-polynomial dependence on k. Thus, Assumption 4.4.10 covers both polynomial and root-type commutator scalings. Also, $R_m(\delta) = 0$ for Theorem 3.3.4, while $R_m(\delta) = \|\vec{a}\|_1 \|\vec{b}\|_1 \delta$ for Proposition 4.2.1.

We now quantify how the abstract error bound in Assumption 4.4.10 propagates through the LCHS representation. We first define the quantities needed for the short-time analysis in Definition 4.4.12, prove a one-step MPF error bound for step size Δ in Lemma 4.4.13, and extend it to $[0, T]$ in Proposition 4.4.14. Since the argument does not use specific properties of $\hat{f}_{a,b}(k; c, d)$, we write $\hat{f}(k)$ unless the parameter dependence is needed.

Definition 4.4.12. Let $\hat{f}$ be a kernel (as in Equation (4.10)). For $R > 0$, define the truncated kernel norm by $\alpha_{\hat{f},R} := \|\hat{f}\|_{L^1[-R,R]}/\sqrt{2\pi}$. Under Assumption 4.4.10, define the truncated profile average

$$\Lambda_m(R, N) := \frac{1}{\sqrt{2\pi}} \int_{-R}^{R} |\hat{f}(k)| \, \Phi_m(k, N) dk. \tag{4.33}$$

We first establish the error bound for a short time step Δ in Lemma 4.4.13.

Lemma 4.4.13. *Fix $R, \delta > 0$, $m, N \geq 1$. For all $0 < \Delta \leq \Delta(R, N, \delta)$, we have*

$$E_{\mathrm{MPF}}(\Delta) = \left\| \frac{1}{\sqrt{2\pi}} \int_{-R}^{R} \hat{f}(k) \big(U_k^{(2m)}(\Delta) - U_k(\Delta) \big) dk \right\| \leq C_m \Delta^{2m+1} \Lambda_m(R, N) + \alpha_{\hat{f},R} R_m(\delta). \tag{4.34}$$

Proof. Applying Assumption 4.4.10 and the triangle inequality, we have

$$\left\| \frac{1}{\sqrt{2\pi}} \int_{-R}^{R} \hat{f}(k) \big(U_k^{(2m)}(\Delta) - U_k(\Delta) \big) dk \right\| \leq \frac{1}{\sqrt{2\pi}} \int_{-R}^{R} |\hat{f}(k)| \| U_k^{(2m)}(\Delta) - U_k(\Delta) \| dk \tag{4.35}$$

$$\leq \frac{1}{\sqrt{2\pi}} \int_{-R}^{R} |\hat{f}(k)| (C_m \Delta^{2m+1} \Phi_m(k, N) + R_m(\delta)) dk. \tag{4.36}$$

The first term is $C_m \Delta^{2m+1} \Lambda_m(R, N)$ by Equation (4.33) and the second term is $\alpha_{\hat{f},R} R_m(\delta)$. □

We now extend this short-time error bound to the full simulation over $[0, T]$ in Proposition 4.4.14.

Proposition 4.4.14. *Fix $R, \delta > 0$, $m, N \geq 1$. Let $T > 0$, $r \in \mathbb{N}$. Set $\Delta := T/r$. Assume that $0 < \Delta \leq \Delta(R, N, \delta)$. Then*

$$E_{\mathrm{MPF}}(T) = \left\| \frac{1}{\sqrt{2\pi}} \int_{-R}^{R} \hat{f}(k) \Big(\big(U_k^{(2m)}(\Delta) \big)^r - U_k(T) \Big) dk \right\| \leq \frac{r}{\sqrt{2\pi}} \int_{-R}^{R} |\hat{f}(k)| \eta_k(\Delta) (1 + \eta_k(\Delta))^{r-1} dk, \tag{4.37}$$

where $\eta_k(\Delta) := C_m \Delta^{2m+1} \Phi_m(k, N) + R_m(\delta)$. In particular, if $\eta_R(\Delta) = \sup_{|k| \leq R} \eta_k(\Delta)$, then

$$\left\| \frac{1}{\sqrt{2\pi}} \int_{-R}^{R} \hat{f}(k) \left((U_k^{(2m)}(\Delta))^r - U_k(T) \right) dk \right\| \leq \frac{C_m \frac{T^{2m+1}}{r^{2m}} \Lambda_m(R, N) + r \alpha_{\hat{f},R} R_m(\delta)}{(1 + \eta_R(T/r))^{1-r}}. \tag{4.38}$$

Proof. Fix $k \in [-R, R]$ and write $A_k := U_k^{(2m)}(\Delta)$ and $B_k := U_k(\Delta)$. Since $U_k(T) = U_k(\Delta)^r = B_k^r$, the telescoping identity gives $A_k^r - B_k^r = \sum_{j=0}^{r-1} A_k^{r-1-j} (A_k - B_k) B_k^j$. Taking norms and using that B_k is unitary, so $\|B_k^j\| = 1$, we obtain $\|A_k^r - B_k^r\| \leq \sum_{j=0}^{r-1} \|A_k\|^{r-1-j} \|A_k - B_k\|$. By Assumption 4.4.10, $\|A_k - B_k\| \leq \eta_k(\Delta)$. Hence, $\|A_k\| \leq \|B_k\| + \|A_k - B_k\| \leq 1 + \eta_k(\Delta)$, and

therefore we have

$$\|A_k^r - B_k^r\| \leq \sum_{j=0}^{r-1} (1+\eta_k(\Delta))^{r-1-j}\eta_k(\Delta) \leq r\eta_k(\Delta)(1+\eta_k(\Delta))^{r-1}. \tag{4.39}$$

Multiplying by $|\hat{f}(k)|/\sqrt{2\pi}$ and integrating over $[-R, R]$ yields

$$\left\| \frac{1}{\sqrt{2\pi}} \int_{-R}^{R} \hat{f}(k) \Big(\big(U_k^{(2m)}(\Delta)\big)^r - U_k(T) \Big) dk \right\| \leq \frac{r}{\sqrt{2\pi}} \int_{-R}^{R} |\hat{f}(k)|\, \eta_k(\Delta)(1+\eta_k(\Delta))^{r-1} dk, \tag{4.40}$$

which proves Equation (4.37). Noting that for $|k| \leq R$, $\eta_k(\Delta) \leq \eta_R(T/r)$ implies that

$$\frac{r}{\sqrt{2\pi}} \int_{-R}^{R} |\hat{f}(k)|\, \eta_k(\Delta)(1+\eta_k(\Delta))^{r-1} dk \leq \frac{r}{\sqrt{2\pi}} (1+\eta_R(T/r))^{r-1} \int_{-R}^{R} |\hat{f}(k)|\, \eta_k(\Delta) dk. \tag{4.41}$$

Using the definition of $\eta_k(\Delta)$ and the quantities $\Lambda_m(R, N)$ and $\alpha_{\hat{f},R}$, we have $\frac{1}{\sqrt{2\pi}} \int_{-R}^{R} |\hat{f}(k)|\, \eta_k(\Delta) dk = C_m \Delta^{2m+1} \Lambda_m(R, N) + \alpha_{\hat{f},R} R_m(\delta)$. Substituting $\Delta = T/r$ gives $r(C_m \Delta^{2m+1} \Lambda_m(R, N) + \alpha_{\hat{f},R} R_m(\delta)) = C_m \frac{T^{2m+1}}{r^{2m}} \Lambda_m(R, N) + r\, \alpha_{\hat{f},R} R_m(\delta)$. This completes the proof. □

4.4.4 Coupling of Truncation and Inner Simulation Errors

The key point of Section 4.4.3 is that, after integrating over $k \in [-R, R]$, the inner simulation error depends on $\Lambda_m(R, N)$. Hence, R and the inner simulation error are coupled. Increasing R reduces truncation error but, when $\Phi_m(k, N)$ grows with $|k|$, also increases $\Lambda_m(R, N)$ and amplifies simulation error. We capture this trade-off through a feasibility problem. Define

$$\mathcal{F}_{\epsilon_{\text{comb}}} := \left\{ (R, \delta, r) \in (0, \infty)^2 \times \mathbb{N} : E_{\text{trunc}}(R) + E_{\text{MPF}}(T) \leq \epsilon_{\text{comb}}, \frac{T}{r} \leq \Delta(R, N, \delta) \right\}. \quad (4.42)$$

Here ϵ_{comb} is the total error budget for the truncation and inner simulation errors. If $E_{\text{trunc}}(R) \geq \epsilon_{\text{comb}}$, then $(R, \delta, r) \notin \mathcal{F}_{\epsilon_{\text{comb}}}$ for every $r \in \mathbb{N}$ and $\delta > 0$. Hence a positive error budget for the inner simulation is available only when $R \in \mathcal{H}_{\epsilon_{\text{comb}}}$. For such R, define $\epsilon_R := \epsilon_{\text{comb}} - E_{\text{trunc}}(R) > 0$. After fixing $R \in \mathcal{H}_{\epsilon_{\text{comb}}}$, the problem therefore reduces to determining pairs $(r, \delta) \in \mathbb{N} \times (0, \infty)$ such that $E_{\text{MPF}}(T) \leq \epsilon_R$ and $\frac{T}{r} \leq \Delta(R, N, \delta)$. The choice of r depends on whether Theorem 3.3.4 or Proposition 4.2.1 instantiates Assumption 4.4.10. We defer this discussion to Section 4.5.

4.5 Post-Quadrature Complexity Analysis

We now analyze the overall query complexity of our algorithm. We fix an admissible kernel from Equation (4.10). Section 4.5.1 gives the post-quadrature error bound, Section 4.5.2 describes the implementation, Section 4.5.3 derives complexity under the abstract MPF estimate in Assumption 4.4.10, and Sections 4.5.4 and 4.5.5 specialize the analysis to Theorem 3.3.4 and Proposition 4.2.1, respectively. Section 4.5.6 discusses various quadrature rules.

4.5.1 Post-Quadrature Error Bound

The analysis in Section 4.4.4 was carried out before discretizing the LCHS integral. We now show that the same coupling persists after applying a quadrature rule. Let $Q = \{(k_i, w_i)\}_{i \in \mathcal{I}_Q}$ be a quadrature rule, with $k_i \in \mathbb{R}$ and $w_i \in \mathbb{C}$. Define its radius by $R_Q := \max_{i \in \mathcal{I}_Q} |k_i|$. The corresponding ideal quadrature operator is $W_Q^{\text{ideal}} := \sum_{i \in \mathcal{I}_Q} v_i U_{k_i}(T)$, where $v_i := \frac{w_i}{\sqrt{2\pi}} \hat{f}_{a,b}(k_i; c, d)$, and

the associated quadrature mass is $\alpha_Q := \sum_{i\in\mathcal{I}_Q} |v_i|$. Under Assumption 4.4.10, we also define the discrete weighted MPF profile by $\Lambda_{m,Q} := \sum_{i\in\mathcal{I}_Q} |v_i|\Phi_m(k_i, N)$. This is the post-quadrature analogue of $\Lambda_m(R, N)$. We also define $\Phi^*_{m,Q} := \max_{i\in\mathcal{I}_Q} \Phi_m(k_i, N)$. Let $U^{(2m)}_{k_i}(T)$ denote the $2m$-th order MPF approximation to $U_{k_i}(T)$ over $[0, T]$, and define $W^{\mathrm{MPF}}_Q := \sum_{i\in\mathcal{I}_Q} v_i U^{(2m)}_{k_i}(T)$. Specific quadrature rules are discussed in Section 4.5.6. We have the following discrete analog of Proposition 4.4.14.

Corollary 4.5.1. *Let $Q = \{(k_i, w_i)\}_{i\in\mathcal{I}_Q}$ be a quadrature rule and set $R_Q := \max_{i\in\mathcal{I}_Q} |k_i|$. Fix $m, N \geq 1$, $T > 0$, $r \in \mathbb{N}$, and $\delta > 0$. Set $\Delta := T/r$. Assume that $0 < \Delta \leq \Delta(R_Q, N, \delta)$. Then*

$$\|W^{\mathrm{ideal}}_Q - W^{\mathrm{MPF}}_Q\| \leq \sum_{i\in\mathcal{I}_Q} |v_i| \|U_{k_i}(T) - U^{(2m)}_{k_i}(T)\| \leq r \sum_{i\in\mathcal{I}_Q} |v_i| \eta_{k_i}(\Delta)(1 + \eta_{k_i}(\Delta))^{r-1}. \tag{4.43}$$

In particular, if $\eta_Q(\Delta) := \max_{i\in\mathcal{I}_Q} \eta_{k_i}(\Delta)$, then

$$\|W^{\mathrm{ideal}}_Q - W^{\mathrm{MPF}}_Q\| \leq (1 + \eta_Q(T/r))^{r-1} \left(C_m \frac{T^{2m+1}}{r^{2m}} \Lambda_{m,Q} + r\alpha_Q R_m(\delta) \right). \tag{4.44}$$

Proof. The result follows by the same telescoping argument as in Proposition 4.4.14, followed by the definitions of α_Q and $\Lambda_{m,Q}$. □

Corollary 4.5.1 shows that, after fixing the quadrature rule, the inner simulation error couples to the quadrature error through R_Q, α_Q, and $\Lambda_{m,Q}$. This is the post-quadrature analogue of Section 4.4.4. We end with a total error bound, which is the starting point for the complexity analysis in Section 4.5.3.

Lemma 4.5.2. *Fix a kernel profile $\vec{\theta} = (a, b, c, d)$ such that d and y_0 are chosen as in Proposition 4.4.4 or Corollary 4.4.5. Let Q be a quadrature rule, $T, \delta > 0$, $r \in \mathbb{N}$ and set $\Delta = T/r$. If*

$0 < \Delta \leq \Delta(R_Q, N, \delta)$, *then*

$$\|e^{-AT} - W_Q^{\mathrm{MPF}}\| \leq E_{\mathrm{approx}}(y_0) + E_{\mathrm{quad}}(Q) \tag{4.45}$$

$$+ (1 + \eta_Q(T/r))^{r-1} \left(C_m \frac{T^{2m+1}}{r^{2m}} \Lambda_{m,Q} + r\alpha_Q R_m(\delta) \right). \tag{4.46}$$

Proof. Let $W_\infty := \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}_{a,b}(k; c, d) U_k(T)\, dk$, and define the quadrature error associated with Q by $E_{\mathrm{quad}}(Q) := \|W_\infty - W_Q^{\mathrm{ideal}}\|$. By the triangle inequality, we have

$$\|e^{-AT} - W_Q^{\mathrm{MPF}}\| \leq \|e^{-AT} - W_\infty\| + \|W_\infty - W_Q^{\mathrm{ideal}}\| + \|W_Q^{\mathrm{ideal}} - W_Q^{\mathrm{MPF}}\|. \tag{4.47}$$

The first term is bounded by $E_{\mathrm{approx}}(y_0)$, the second term is $E_{\mathrm{quad}}(Q)$, and the third term follows from Corollary 4.5.1. Since the nodes of Q lie in $[-R_Q, R_Q]$, the MPF step-size condition is imposed with R_Q, yielding a bound involving the quadrature-induced quantities α_Q, $\Lambda_{m,Q}$, and η_Q. □

Lemma 4.5.2 shows that the quadrature rule affects the final error through $E_{\mathrm{quad}}(Q)$ and through R_Q, α_Q, and $\Lambda_{m,Q}$. Thus, quadrature couples discretization error to quantum implementation error. This yields the post-quadrature problem. Choose an admissible rule Q with $E_{\mathrm{quad}}(Q) < \epsilon_{\mathrm{comb}}$, leaving residual budget $\epsilon_Q := \epsilon_{\mathrm{comb}} - E_{\mathrm{quad}}(Q) > 0$. Then choose $(r, \delta) \in \mathbb{N} \times (0, \infty)$ such that $(1 + \eta_Q(T/r))^{r-1}(C_m \frac{T^{2m+1}}{r^{2m}} \Lambda_{m,Q} + r\alpha_Q R_m(\delta)) \leq \epsilon_Q$ and $\frac{T}{r} \leq \Delta(R_Q, N, \delta)$. This is the post-quadrature analogue of the discussion in Section 4.4.4.

4.5.2 Quantum Implementation

The complexity estimates below address two tasks in a fully quantum implementation. The first task is to construct a block-encoding of the post-quadrature approximation $W_Q^{\mathrm{MPF}} = \sum_{i\in\mathcal{I}_Q} v_i U_{k_i}^{(2m)}(T) = \sum_{i\in\mathcal{I}_Q} v_i \left(\sum_{j=1}^m a_j \left[U_{2,k_i}\left(\frac{T}{rb_j}\right)\right]^{b_j}\right)^r$. Here v_i denotes the quadrature coefficients, while a_j, b_j denote the coefficients from Chapter 3. Note that here we use k_j instead of b_j. Given Q, define the quadrature preparation oracles $\mathrm{PREP}_{\mathrm{R}}^Q |0\rangle = \frac{1}{\sqrt{\alpha_Q}} \sum_{i\in\mathcal{I}_Q} \sqrt{v_i} |i\rangle$ and $\mathrm{PREP}_{\mathrm{L}}^Q |0\rangle = \frac{1}{\sqrt{\alpha_Q}} \sum_{i\in\mathcal{I}_Q} \overline{\sqrt{v_i}} |i\rangle$, where $\alpha_Q := \sum_{i\in\mathcal{I}_Q} |v_i|$. For each $i \in \mathcal{I}_Q$, let $\mathcal{U}_Q$ be a unitary block-encoding of $U_{k_i}^{(2m)}(T)$, normalized to one after amplitude amplification at each of the r time steps. Hence, we have $(\langle 0| \otimes I)\mathcal{U}_Q(|0\rangle \otimes I) = U_{k_i}^{(2m)}(T)$. Define the controlled unitary $\mathrm{SELECT}_Q := \sum_{i\in\mathcal{I}_Q} |i\rangle\langle i| \otimes \mathcal{U}_Q$. By the outer LCU construction, we obtain a block-encoding of W_Q^{MPF}

$$(\langle 0| \langle 0| \otimes I)(\mathrm{PREP}_{\mathrm{L}}^{Q\dagger} \otimes I)\, \mathrm{SELECT}_Q (\mathrm{PREP}_{\mathrm{R}}^Q \otimes I)(|0\rangle |0\rangle \otimes I) = \frac{W_Q^{\mathrm{MPF}}}{\alpha_Q}. \tag{4.48}$$

Writing $K_m := \|\vec{a}\|_1 \|\vec{b}\|_1$, one controlled implementation of $U_{k_i}^{(2m)}(T)$ with r time steps costs $\mathcal{O}(rK_m)$ controlled second-order product-formula queries. Hence, one use of SELECT_Q has the same cost. One block-encoding of $W_Q^{\mathrm{MPF}}/\alpha_Q$ uses SELECT_Q, the quadrature preparation oracles $\mathrm{PREP}_{\mathrm{L}}^Q$, $\mathrm{PREP}_{\mathrm{R}}^Q$, and their required inverses in the outer LCU construction.

Remark 4.5.3. The implementation of W_Q^{MPF} is a nested LCU construction. The outer LCU is the quadrature sum over q, with normalization α_Q, while the inner LCU implements each selected $U_{k_i}^{(2m)}(T)$ by an MPF. We do not flatten this into one LCU over $(q, j_1, \ldots, j_r)$, since that would introduce normalization $\alpha_Q \|\vec{a}\|_1^r$, which is not the implementation model used here.

The second task is to prepare the normalized output state $W_Q^{\mathrm{MPF}}u(0)/\|W_Q^{\mathrm{MPF}}u(0)\|$. Write $|\widetilde{u}(0)\rangle := u(0)/\|u(0)\|$. Applying the block-encoding of $W_Q^{\mathrm{MPF}}/\alpha_Q$ to $|\widetilde{u}(0)\rangle$ and post-selecting the ancillas on $|0\rangle$ gives the un-normalized state $W_Q^{\mathrm{MPF}}|\widetilde{u}(0)\rangle/\alpha_Q$, whose norm is $\frac{\|W_Q^{\mathrm{MPF}}|\widetilde{u}(0)\rangle\|}{\alpha_Q} = \frac{\|W_Q^{\mathrm{MPF}}u(0)\|}{\alpha_Q\|u(0)\|}$. Hence, amplitude amplification requires $\chi_Q(u(0)) := \frac{\alpha_Q\|u(0)\|}{\|W_Q^{\mathrm{MPF}}u(0)\|}$ uses of the block-encoding, up to absolute constants. Therefore, normalized state preparation costs $\mathcal{O}(\chi_Q(u(0))rK_m)$ controlled second-order product-formula queries, $\mathcal{O}(\chi_Q(u(0)))$ uses of the quadrature preparation oracles and their inverses, and $\mathcal{O}(\chi_Q(u(0)))$ uses of the state-preparation oracle for $|\widetilde{u}(0)\rangle$ and its inverse.

Remark 4.5.4. Let $u(T) := e^{-AT}u(0)$. If $\|W_Q^{\mathrm{MPF}} - e^{-AT}\| \le \epsilon$ and $\epsilon\|u(0)\| \le \|u(T)\|/2$, then $\|W_Q^{\mathrm{MPF}}u(0)\| \ge \|u(T)\|\epsilon\|u(0)\| \ge \frac{\|u(T)\|}{2}$. Consequently, $\chi_Q(u(0)) \le 2\alpha_Q\|u(0)\|/\|u(T)\|$, so the state query complexity is $\mathcal{O}\Big(\frac{\|u(0)\|}{\|u(T)\|}\alpha_Q rK_m\Big)$, which becomes $\mathcal{O}(\alpha_Q rK_m/\|u(T)\|)$ when $\|u(0)\| = 1$. This is the standard form in which normalized-state query complexity is presented in the literature. See, for example, the discussion in Section 4.3.

4.5.3 Overall Query Complexity

Split the error tolerance as $\epsilon_{\mathrm{approx}} + \epsilon_{\mathrm{comb}} \le \epsilon$, where $\epsilon_{\mathrm{approx}}$ controls $E_{\mathrm{approx}}(y_0)$ and ϵ_{comb} controls the combined quadrature and inner-simulation errors. Proposition 4.4.4 and Corollary 4.4.5 gives sufficient conditions on c, d to ensure the approximation error is bounded blow by $\epsilon_{\mathrm{approx}}$. Choose a quadrature rule Q such that $E_{\mathrm{quad}}(Q) < \epsilon_{\mathrm{comb}}$, and define $\epsilon_Q := \epsilon_{\mathrm{comb}} - E_{\mathrm{quad}}(Q)$. We first prove an abstract fixed-$2m$-order overall complexity estimate, based

on Assumption 4.4.10, for arbitrary Q. For $\delta > 0$, define

$$r_Q(\delta) := \left\lceil \max\left\{ \frac{T}{\Delta(R_Q, N, \delta)}, \left(2C_m T^{2m+1} \Phi^*_{m,Q}\right)^{1/2m}, \left(\frac{2e\, C_m T^{2m+1} \Lambda_{m,Q}}{\epsilon_Q} \right)^{1/2m} \right\} \right\rceil. \tag{4.49}$$

Theorem 4.5.5. *Fix $m, N \in \mathbb{N}$ and $T > 0$. Let $\epsilon, \epsilon_{\text{approx}}, \epsilon_{\text{comb}} > 0$ satisfy $\epsilon_{\text{approx}} + \epsilon_{\text{comb}} \leq \epsilon$. Let $\vec{\theta} = (a, b, c, d)$ be an admissible kernel profile, and let Q be a quadrature rule such that $E_{\text{quad}}(Q) < \epsilon_{\text{comb}}$. Set $\epsilon_Q := \epsilon_{\text{comb}} - E_{\text{quad}}(Q) > 0$. Suppose there exists $\delta > 0$ such that*

$$r_Q(\delta) R_m(\delta) \leq \frac{1}{2}, \qquad r_Q(\delta) \alpha_Q R_m(\delta) \leq \frac{\epsilon_Q}{2e}. \tag{4.50}$$

Choose $r = r_Q(\delta)$. Then $\|e^{-AT} - W_Q^{\text{MPF}}\| \leq \epsilon$. Moreover, a block-encoding of $W_Q^{\text{MPF}}/\alpha_Q$ can be implemented using $\mathcal{Q}_{\text{block}} = \mathcal{O}\left(r_Q(\delta) K_m\right)$ controlled second-order product-formula queries. If $W_Q^{\text{MPF}} u(0) \neq 0$, the normalized output state proportional to $W_Q^{\text{MPF}} |u(0)\rangle$ can be prepared using $\mathcal{Q}_{\text{state}} = \mathcal{O}\left(\chi_Q(u(0)) r_Q(\delta) K_m\right)$ controlled second-order product-formula queries. Equivalently, this uses $\mathcal{O}(\chi_Q(u(0)))$ applications of the block-encoding of $W_Q^{\text{MPF}}/\alpha_Q$, hence $\mathcal{O}(\chi_Q(u(0)))$ calls to SELECT_Q, the quadrature preparation oracles, their inverses, and the state-preparation oracle for $|\widetilde{u}(0)\rangle$.

Proof. Let $r = r_Q(\delta)$. By Equation (4.49), we have $\frac{T}{r} \leq \Delta(R_Q, N, \delta)$, $C_m \frac{T^{2m+1}}{r^{2m}} \Phi^*_{m,Q} \leq \frac{1}{2}$ and $C_m \frac{T^{2m+1}}{r^{2m}} \Lambda_{m,Q} \leq \frac{\epsilon_Q}{2e}$. Together with Equation (4.50), this gives $r\eta_Q(T/r) = C_m \frac{T^{2m+1}}{r^{2m}} \Phi^*_{m,Q} + rR_m(\delta) \leq 1$. Hence, we have $(1 + \eta_Q(T/r))^{r-1} \leq (1 + \eta_Q(T/r))^r \leq e^{r\eta_Q(T/r)} \leq e$. Using this estimate, the third condition in Equation (4.49), and Equation (4.50), we obtain

$$(1 + \eta_Q(T/r))^{r-1} \left(C_m \frac{T^{2m+1}}{r^{2m}} \Lambda_{m,Q} + r\alpha_Q R_m(\delta) \right) \leq e \left(\frac{\epsilon_Q}{2e} + \frac{\epsilon_Q}{2e} \right) = \epsilon_Q. \tag{4.51}$$

By the estimate in Lemma 4.5.2 and the admissibility of the kernel and quadrature rule, we have $\|e^{-AT} - W_Q^{\mathrm{MPF}}\| \leq \epsilon_{\mathrm{approx}} + E_{\mathrm{quad}}(Q) + \epsilon_Q = \epsilon_{\mathrm{approx}} + \epsilon_{\mathrm{comb}} \leq \epsilon$. The block-encoding and normalized-state query estimates follow from Section 4.5.2. □

In what follows, define the m-dependent post-quadrature effective commutator scale

$$\mu_{m,Q}(\delta) := \max\left\{\frac{1}{\Delta(R_Q, N, \delta)}, (C_m \Phi^*_{m,Q})^{1/2m}, (C_m \Lambda_{m,Q})^{1/2m}\right\}. \tag{4.52}$$

It can then be checked that Equation (4.49) implies that

$$r_Q(\delta) = \mathcal{O}\left(1 + \mu_{m,Q}(\delta) \max\left\{T,\, T^{1+\frac{1}{2m}},\, T^{1+\frac{1}{2m}} \epsilon_Q^{-\frac{1}{2m}}\right\}\right), \tag{4.53}$$

for $\epsilon_Q > 0$. Under the additional assumption $0 < \epsilon_Q < 1$, this simplifies to

$$r_Q(\delta) = \mathcal{O}\left(1 + \mu_{m,Q}(\delta) \max\left\{T,\, T^{1+\frac{1}{2m}} \epsilon_Q^{-\frac{1}{2m}}\right\}\right). \tag{4.54}$$

In what follows, assume $0 < \epsilon_Q < 1$. In Corollary 4.5.6, we allow mild MPF-order dependence in the effective commutator scale and optimize over the order to obtain complexity logarithmic in T/ϵ_Q, up to the growth envelope F.

Corollary 4.5.6. *Assume the error budget, kernel profile, and quadrature hypotheses of Theorem 4.5.5. Let $K_m := \|\vec{a}\|_1 \|\vec{b}\|_1$ and $0 < \epsilon_Q < 1$. Suppose there exist $m_0 \in \mathbb{N}$, $\mu_Q \geq 1$, and a non-decreasing function $F : [m_0, \infty) \to [1, \infty)$ such that one can choose $\delta_m > 0$ satisfying*

$$\mu_{m,Q}(\delta_m) \leq \mu_Q F(m). \tag{4.55}$$

for $m \geq m_0$. Assume also that, for every $m \geq m_0$, we have $r_Q(\delta_m)R_m(\delta_m) \leq \frac{1}{2}$ and $r_Q(\delta_m)\alpha_Q R_m(\delta_m) \leq \frac{\epsilon_Q}{2e}$. Choose $m = \max\left\{m_0, \left\lceil \log\left(e + \frac{T}{\epsilon_Q}\right)\right\rceil\right\}$ and set $\delta = \delta_m$ and $r = r_Q(\delta_m)$. Then $\|e^{-AT} - W_Q^{\mathrm{MPF}}\| \leq \epsilon$. Moreover, we have

$$\mathcal{Q}_{\mathrm{block}} = \mathcal{O}\left((1 + \mu_Q F(m) \max\{1, T\}) \left(\log\left(e + \frac{T}{\epsilon_Q}\right)\right)^2 \left(\log\log\left(e^e + \frac{T}{\epsilon_Q}\right)\right)^2 \right). \tag{4.56}$$

If $W_Q^{\mathrm{MPF}} u(0) \neq 0$, we have

$$\mathcal{Q}_{\mathrm{state}}(u(0)) = \mathcal{O}\left(\chi_Q(u(0)) (1 + \mu_Q F(m) \max\{1, T\}) \left(\log\left(e + \frac{T}{\epsilon_Q}\right)\right)^2 \left(\log\log\left(e^e + \frac{T}{\epsilon_Q}\right)\right)^2 \right). \tag{4.57}$$

If $T \geq 1$ and $\mu_Q T F(m) \geq 1$, then Equation (4.56) and Equation (4.57) simplify to

$$\mathcal{Q}_{\mathrm{block}} = \mathcal{O}\left(\mu_Q T F(m) \left(\log\left(e + \frac{T}{\epsilon_Q}\right)\right)^2 \left(\log\log\left(e^e + \frac{T}{\epsilon_Q}\right)\right)^2 \right), \tag{4.58}$$

and

$$\mathcal{Q}_{\mathrm{state}}(u(0)) = \mathcal{O}\left(\chi_Q(u(0)) \, \mu_Q T F(m) \left(\log\left(e + \frac{T}{\epsilon_Q}\right)\right)^2 \left(\log\log\left(e^e + \frac{T}{\epsilon_Q}\right)\right)^2 \right). \tag{4.59}$$

Proof. The admissibility conditions allow us to apply Theorem 4.5.5 with $\delta = \delta_m$, so the error is at most ϵ. By Theorem 4.5.5, we have $\mathcal{Q}_{\mathrm{block}} = \mathcal{O}(r_Q(\delta_m) K_m)$. Using Equation (4.54) and Equation (4.55), we obtain

$$r_Q(\delta_m) = \mathcal{O}\left(1 + \mu_Q F(m) \max\left\{T, \, T^{1+\frac{1}{2m}} \epsilon_Q^{-\frac{1}{2m}}\right\}\right). \tag{4.60}$$

Since $T^{1+\frac{1}{2m}}\epsilon_Q^{-\frac{1}{2m}} = T\left(\frac{T}{\epsilon_Q}\right)^{1/2m}$, the choice $m = \max\left\{m_0, \left\lceil \log\left(e+\frac{T}{\epsilon_Q}\right)\right\rceil\right\}$ implies $\left(\frac{T}{\epsilon_Q}\right)^{1/2m} = \mathcal{O}(1)$. Therefore, we have $r_Q(\delta_m) = \mathcal{O}\left(1+\mu_Q F(m)\max\{1,T\}\right)$. Consequently, we have $\mathcal{Q}_{\text{block}} = \mathcal{O}\left(K_m\left(1+\mu_Q F(m)\max\{1,T\}\right)\right)$. Since $K_m = \mathcal{O}(m^2(\log m)^2)$, the choice of m gives

$$m^2(\log m)^2 = \mathcal{O}\left(\left(\log\left(e+\frac{T}{\epsilon_Q}\right)\right)^2\left(\log\log\left(e^e+\frac{T}{\epsilon_Q}\right)\right)^2\right), \tag{4.61}$$

with constants depending at most on m_0. Hence, we have

$$\mathcal{Q}_{\text{block}} = \mathcal{O}\left(\left(1+\mu_Q F(m)\max\{1,T\}\right)\left(\log\left(e+\frac{T}{\epsilon_Q}\right)\right)^2\left(\log\log\left(e^e+\frac{T}{\epsilon_Q}\right)\right)^2\right). \tag{4.62}$$

This proves Equation (4.56). Multiplying by $\chi_Q(u(0))$ gives Equation (4.57). If $T \geq 1$ and $\mu_Q T F(m) \geq 1$, then $1+\mu_Q F(m)\max\{1,T\} = 1+\mu_Q T F(m) = \mathcal{O}(\mu_Q T F(m))$. The derivation establishes Equation (4.58) and Equation (4.59). □

In most applications considered in this paper, we have $F(m) = m^a\big(\log(e+m)\big)^b$ for some $a, b \geq 0$. Then Corollary 4.5.6 implies the following explicit polylogarithmic form:

$$\begin{aligned}\mathcal{Q}_{\text{block}} = \mathcal{O}\Bigg(&\left(1+\mu_Q\max\{1,T\}\left(\log\left(e+\frac{T}{\epsilon_Q}\right)\right)^a\left(\log\left(e+\log\left(e+\frac{T}{\epsilon_Q}\right)\right)\right)^b\right)\\ &\times\left(\log\left(e+\frac{T}{\epsilon_Q}\right)\right)^2\left(\log\left(e+\log\left(e+\frac{T}{\epsilon_Q}\right)\right)\right)^2\Bigg).\end{aligned} \tag{4.63}$$

Indeed, the choice of m implies $m = \mathcal{O}\left(\log\left(e+\frac{T}{\epsilon_Q}\right)\right)$. Therefore, we have

$$F(m) = \mathcal{O}\left(\left(\log\left(e+\frac{T}{\epsilon_Q}\right)\right)^a\left(\log\left(e+\log\left(e+\frac{T}{\epsilon_Q}\right)\right)\right)^b\right). \tag{4.64}$$

Substituting this into Equation (4.56) gives Equation (4.63). The normalized-state estimate follows by multiplying by $\chi_Q(u(0))$.

4.5.4 General Commutator Scaling Specialization

The analysis in Section 4.5.3 allowed arbitrary quadrature under Assumption 4.4.10. We now specialize to Theorem 3.3.4, still with arbitrary quadrature. We first discuss the applicability of Theorem 3.3.4. For each $G_{k_i} = H + k_i L$, we must compute $\inf_{j\geq J} \alpha_{\mathrm{comm},j}(G_{k_i})^{-1/j}$. Define $\mathcal{C}_j(H,L) := \max_{0\leq \ell\leq j} \mathcal{C}_{j,\ell}(H,L)$. We have

$$\alpha_{\mathrm{comm},j}(G_{k_i}) \leq \sum_{\ell=0}^{j} |k_i|^\ell \mathcal{C}_{j,\ell}(H,L) \leq \mathcal{C}_j(H,L)(j+1)\max\{1,|k_i|\}^j. \tag{4.65}$$

Let $\chi_J(H,L) := \inf_{j\geq J} \mathcal{C}_j(H,L)^{-1/j}$. For $J \geq 1$, we have

$$\inf_{j\geq J} \alpha_{\mathrm{comm},j}(G_{k_i})^{-1/j} \geq \frac{\inf_{j\geq J}((j+1)^{-1/j}\mathcal{C}_j(H,L)^{-1/j})}{\max\{1,|k_i|\}} \tag{4.66}$$

$$\geq \frac{(\inf_{j\geq J}(j+1)^{-1/j})(\inf_{j\geq J}\mathcal{C}_j(H,L)^{-1/j})}{\max\{1,|k_i|\}} \tag{4.67}$$

$$= \frac{\chi_J(H,L)}{(J+1)^{1/J}\max\{1,|k_i|\}} \geq \frac{\chi_J(H,L)}{(J+1)^{1/J}\max\{1,R_Q\}} := \kappa_Q. \tag{4.68}$$

The last inequality follows from $|k_i| \leq R_Q$. Thus Theorem 3.3.4 applies uniformly to all G_{k_i} whenever $\chi_J(H,L) > 0$ and $\Delta \leq \rho_Q$ for some $0 < \rho_Q < \kappa_Q$. The parameter ρ_Q is a uniform admissible step-size and enters the error profile because Theorem 3.3.4 gives an MPF error series

in powers of $\Delta^{j+\ell}$, while the abstract framework factors out Δ^{2m+1}. Set $C_m := \|\vec{a}\|_1$ and define

$$\Phi_{m,\nu}(H,L) := \sum_{\substack{j\in 2\mathbb{Z}_+\\ j\geq 2m}} \sum_{l=1}^{m} \frac{\rho_Q^{j+l-(2m+1)}}{l!} \sum_{\substack{j_1,\dots,j_l\in 2\mathbb{Z}_+\\ j_1+\cdots+j_l=j}} \sum_{\substack{\ell_1,\dots,\ell_l\geq 0\\ 0\leq \ell_\kappa\leq j_\kappa+1\\ \ell_1+\cdots+\ell_l=\nu}} \prod_{\kappa=1}^{l} \mathcal{C}_{j_\kappa+1,\ell_\kappa}(H,L). \tag{4.69}$$

for $\nu \geq 0$. Define the discrete quadrature moment $M_\alpha^Q := \sum_{i\in\mathcal{I}_Q} |v_i|\,|k_i|^\alpha$. The relevant quantities appearing in the complexity estimates are $\Lambda_{m,Q} := \sum_{\nu\geq 0} \Phi_{m,\nu}(H,L) M_\nu^Q$ and $\Phi^*_{m,Q} := \max_{i\in\mathcal{I}_Q} \Phi_{m,\nu}(H,k_iL)$. In Lemma 4.5.7, which is proved in Section B.4, we record the convergence of the profiles before stating the complexity consequence. This is the only point where the strict inequality $\rho_Q < \kappa_Q$ is used.

Lemma 4.5.7. *Assume that there exists $J \geq 1$ such that $\chi_J(H,L) := \inf_{j\geq J} \mathcal{C}_j(H,L)^{-1/j} > 0$. Let $0 < \rho_Q < \kappa_Q$. The series defining $\Phi_{m,\nu}(H,L)$, $\Lambda_{m,Q}$, and $\Phi^*_{m,Q}$ are absolutely convergent.*

It remains to verify that the MPF error has the abstract form required in Assumption 4.4.10. By Theorem 3.3.4, for each $i \in \mathcal{I}_Q$ and $\Delta \leq \rho_Q$, we have

$$\|U_{\mathrm{MPF},k_i}(\Delta) - e^{-iG_{k_i}\Delta}\| \leq \|\vec{a}\|_1 \sum_{\substack{j\in 2\mathbb{Z}_+\\ j\geq 2m}} \sum_{l=1}^{m} \frac{\Delta^{j+l}}{l!} \sum_{\substack{j_1,\dots,j_l\in 2\mathbb{Z}_+\\ j_1+\cdots+j_l=j}} \prod_{\kappa=1}^{l} \left(\sum_{\ell=0}^{j_\kappa+1} |k_i|^\ell \mathcal{C}_{j_\kappa+1,\ell}(H,L) \right). \tag{4.70}$$

Since $j \geq 2m$ and $l \geq 1$, we have $j + l \geq 2m + 1$. Thus, for $\Delta \leq \rho_Q$, we have $\Delta^{j+l} \leq \Delta^{2m+1}\rho_Q^{j+l-(2m+1)}$. Let $S_n(r) := \sum_{\ell=0}^{n} r^\ell \mathcal{C}_{n,\ell}(H,L)$. Hence, Equation (4.70) gives

$$\|U_{\mathrm{MPF},k_i}(\Delta) - e^{-iG_{k_i}\Delta}\| \leq C_m \Delta^{2m+1} \sum_{\substack{j\in 2\mathbb{Z}_+\\ j\geq 2m}} \sum_{l=1}^{m} \frac{\rho_Q^{j+l-(2m+1)}}{l!} \sum_{\substack{j_1,\dots,j_l\in 2\mathbb{Z}_+\\ j_1+\cdots+j_l=j}} \prod_{\kappa=1}^{l} S_{j_\kappa+1}(|k_i|), \tag{4.71}$$

Denote the nested sum above as $\Phi_m(|k_i|)$. Its maximum over $i \in \mathcal{I}_Q$ is precisely $\Phi^*_{m,Q}$. Summing

against the quadrature weights gives

$$\sum_{i\in\mathcal{I}_Q} |v_i| \|U_{\mathrm{MPF},k_i}(\Delta) - e^{-iG_{k_i}\Delta}\| \le C_m \Delta^{2m+1} \sum_{i\in\mathcal{I}_Q} |v_i| \Phi_m(|k_i|) = C_m \Delta^{2m+1} \Lambda_{m,Q}. \tag{4.72}$$

Since $R_m(\delta) = 0$ in Theorem 3.3.4, r_Q and $\mu_{m,Q}$ are defined as in Equations (4.49) and (4.52) by setting $\Delta(R_Q, N) := \rho_Q$ and suppressing δ. The fixed-order and optimized error-complexity bounds then follow from Theorem 4.5.5 and Corollary 4.5.6. We record in Lemma 4.5.8 an easier sufficient condition for computing commutator scaling in practice.

Lemma 4.5.8. *For $j \ge 2$, set $\alpha_{Q,j} := \max_{i\in\mathcal{I}_Q} \alpha_{\mathrm{comm},j}(G_{k_i})$ and $\omega := \sup_{j\ge 2} \alpha_{Q,j}^{1/j}$. Assume $0 < \omega < \infty$. If $\rho_Q \omega < 1$, then Theorem 3.3.4 applies uniformly to all G_{k_i}, $i \in \mathcal{I}_Q$, for every $\Delta \le \rho_Q$. Moreover, we have*

$$\mu_{m,Q} \le \max\left\{\rho_Q^{-1}, \max\{1, \alpha_Q^{1/(2m)}\} (\|\vec{a}\|_1 S_m)^{1/(2m)} \omega^{1+1/(2m)}\right\}, \tag{4.73}$$

where $S_m := \sum_{\substack{j\in 2\mathbb{Z}_+ \\ j\ge 2m}} \sum_{l=1}^{m} \frac{1}{l!} \binom{j-1}{l-1} (\rho_Q \omega)^{j+l-(2m+1)}$.

Proof. By definition, $\alpha_{\mathrm{comm},j}(G_{k_i}) \le \alpha_{Q,j} \le \omega^j$. Hence, $\inf_{i\in\mathcal{I}_Q} \inf_{j\ge 2} \alpha_{\mathrm{comm},j}(G_{k_i})^{-1/j} \ge \omega^{-1}$. If $\Delta \le \rho_Q$ and $\rho_Q \omega < 1$, then $\Delta \le \rho_Q < \omega^{-1} \le \inf_{i\in\mathcal{I}_Q} \inf_{j\ge 2} \alpha_{\mathrm{comm},j}(G_{k_i})^{-1/j}$, so the step-size condition in Theorem 3.3.4 holds uniformly. For each product in the error profile in Theorem 3.3.4, we have $\prod_{\kappa=1}^{l} \alpha_{Q,j_\kappa+1} \le \prod_{\kappa=1}^{l} \omega^{j_\kappa+1} = \omega^{j+l}$ since $j_1 + \cdots + j_l = j$. The number of positive integer solutions of $j_1 + \cdots + j_l = j$ is $\binom{j-1}{l-1}$, and the even restriction can only reduce this count. Therefore, we have $\Phi_{m,Q}^* \le \omega^{2m+1} S_m$. Since $\Lambda_{m,Q} \le \alpha_Q \Phi_{m,Q}^*$, we get

$$(\|\vec{a}\|_1 \Lambda_{m,Q})^{1/(2m)} \le \alpha_Q^{1/(2m)} (\|\vec{a}\|_1 S_m)^{1/(2m)} \omega^{1+1/(2m)}. \tag{4.74}$$

Together with the step-size restriction ρ_Q^{-1}, this gives (4.73). □

The upper bound ω in Lemma 4.5.8 is easier to compute in practice. The condition $\rho_Q\omega < 1$ ensures both uniform applicability and convergence of the simplified error profiles. Moreover, $\mu_{m,Q}$ in Equation (4.73) is independent of m in most applications. Indeed, α_Q is independent of m, ω is independent of m for applications considered in Section 4.7, and Lemma 4.5.9, proved in Section B.5, gives $\sup_{m\geq 1} S_m < \infty$.

Lemma 4.5.9. *Consider the following expression*

$$S_m(x) := \sum_{\substack{j\in 2\mathbb{Z}_+ \\ j\geq 2m}} \sum_{l=1}^{m} \frac{1}{l!}\binom{j-1}{l-1} x^{j+l-(2m+1)} \tag{4.75}$$

for $0 < x < 1$. *We have* $\sup_{m\geq 1} S_m(x)^{1/(2m)} < \infty$.

Since the MPF coefficients satisfy $\|\vec{a}\|_1 = \mathcal{O}(\log m)$, we also have $\sup_{m\geq 1} \|\vec{a}\|_1^{1/(2m)} < \infty$. This shows that $\sup_{m\geq 1}(\|\vec{a}\|_1 S_m)^{1/(2m)} < \infty$. Thus, the sufficient condition $\rho_Q\omega < 1$ gives not only fixed-m convergence of the simplified profile, but also the uniform-in-m boundedness required to invoke the optimized-order estimate.

4.5.5 Local Hamiltonian Specialization

We next use the locality-based MPF estimate from Proposition 4.2.1. Unlike Theorem 3.3.4, it includes the remainder $R_m(\delta) = K_m\delta$, where $K_m = \|\vec{a}\|_1\|\vec{b}\|_1$. Since Proposition 4.2.1 gives $p_0 = \mathcal{O}(\log(3N/\delta))$, set $\delta_{p_0} := 3Ne^{-p_0}$. We choose $p_0 > \log(3N)$, so $\delta_{p_0} \in (0, 1)$. For $p_0 \in \mathbb{N}$, define $\bar{\mu}_{m,Q}(p_0) := \max_{i\in\mathcal{I}_Q} \mu_{m,p_0}(G_{k_i})$, where μ_{m,p_0} is defined in Equation (4.2). Since H is q_H-local and g_H-extensive, and L is q_L-local and g_L-extensive, set $q_Q := \max\{q_H, q_L\}$

and $g_Q := g_H + R_Q g_L$. Then $G_{k_i} = H + k_i L$ is uniformly q_Q-local and g_Q-extensive for all $i \in \mathcal{I}_Q$. The corresponding step-size threshold is $\Delta_{Q,p_0} := \min\left\{\frac{1}{16e^3 p_0 q_Q g_Q}, \frac{1}{4\bar{\mu}_{m,Q}(p_0)}\right\}$. Set $C_m = 2e^{1/2}\|\vec{a}\|_1$. The quantities appearing in the complexity estimate are

$$\Lambda_{m,Q}(p_0) := \sum_{i\in\mathcal{I}_Q} |v_i|\, \mu_{m,p_0}(G_{k_i})^{2m+1}, \quad \Phi^*_{m,Q}(p_0) := \bar{\mu}_{m,Q}(p_0)^{2m+1}. \tag{4.76}$$

For fixed $m \geq 1$ and $p_0 \in \mathbb{N}$, define

$$r_{Q,p_0} := \left\lceil \max\left\{ \frac{T}{\Delta_{Q,p_0}}, \left(2C_m T^{2m+1}\Phi^*_{m,Q}(p_0)\right)^{1/2m}, \left(\frac{2eC_m T^{2m+1}\Lambda_{m,Q}(p_0)}{\epsilon_Q}\right)^{1/2m} \right\} \right\rceil \tag{4.77}$$

In Lemma 4.5.10, proved in Section B.6, we derive a growth condition on μ_{m,p_0} needed to control r_{Q,p_0} before applying the results of Section 4.5.3.

Lemma 4.5.10. *Let H be a q-local, g-extensive Hamiltonian on N sites. For every $p_0 \geq 2$, the quantity $\mu_{m,p_0}(H)$ satisfies $\mu_{m,p_0}(H) \leq C_{q,g,N}(1+p_0)^{4/3}$, where one may take $C_{q,g,N} := \max\{1, 2qg\}\max\{1, Ng\}^{1/3}$. Consequently, for a quadrature family $\{G_{k_i} : i \in \mathcal{I}_Q\}$ which is uniformly q_Q-local and g_Q-extensive, one has $\overline{\mu}_{m,Q}(p_0) := \max_{i\in\mathcal{I}_Q} \mu_{m,p_0}(G_{k_i}) \leq C_{\mu,Q,m}(1+p_0)^{4/3}$, with $C_{\mu,Q,m} := \max\{1, 2q_Q g_Q\}\max\{1, Ng_Q\}^{1/3}$.*

By Lemma 4.5.10, $\bar{\mu}_{m,Q}(p_0)$, and hence Δ^{-1}_{Q,p_0}, grow at most polynomially in p_0. By Equation (4.76), we have $\Phi^*_{m,Q}(p_0) \leq \bar{\mu}_{m,Q}(p_0)^{2m+1}$ and $\Lambda_{m,Q}(p_0) \leq \alpha_Q \bar{\mu}_{m,Q}(p_0)^{2m+1}$. Thus, $\Phi^*_{m,Q}(p_0)$, $\Lambda_{m,Q}(p_0)$, and r_{Q,p_0} grow at most polynomially in p_0. Since $\delta_{p_0} = 3Ne^{-p_0}$, we have $r_{Q,p_0}K_m\delta_{p_0} \to 0$ as $p_0 \to \infty$. Hence, there exists $p_0 > \log(3N)$ such that

$$K_m\delta_{p_0} \leq \min\left\{\frac{1}{2r_{Q,p_0}}, \frac{\epsilon_Q}{2e\,\alpha_Q r_{Q,p_0}}\right\}. \tag{4.78}$$

Under Proposition 4.2.1, $R_m(\delta) = K_m\delta$. By Equation (4.78), we have $r_m R_m(\delta_m) = r_m K_m \delta_m \le \frac{1}{2}$ and $r_m \alpha_Q R_m(\delta_m) = r_m \alpha_Q K_m \delta_m \le \frac{\epsilon_Q}{2e}$. Thus, the accumulated locality-remainder conditions in Theorem 4.5.5 hold. By Equation (4.77), we have $\frac{T}{r_m} \le \Delta_{Q,p_0}$, $\frac{C_m T^{2m+1} \Phi^*_{m,Q}(p_0)}{r_m^{2m}} \le \frac{1}{2}$ and $\frac{C_m T^{2m+1} \Lambda_{m,Q}(p_0)}{r_m^{2m}} \le \frac{\epsilon_Q}{2e}$. Therefore, we have $r_m \eta_Q(T/r_m) = \frac{C_m T^{2m+1} \Phi^*_{m,Q}(p_0)}{r_m^{2m}} + r_m R_m(\delta_m) \le 1$. Combining the accumulated-remainder condition with the third inequality above gives the post-quadrature MPF error budget required in Theorem 4.5.5. Hence, all hypotheses of Theorem 4.5.5 hold, and the fixed-order error and complexity estimates follow. Define

$$\mu_{m,Q} := \max\left\{ \frac{1}{\Delta_{Q,p_0(m)}}, (C_m \Phi^*_{m,Q}(p_0(m)))^{1/2m}, (C_m \Lambda_{m,Q}(p_0(m)))^{1/2m} \right\}. \tag{4.79}$$

For the optimized statement, apply Corollary 4.5.6 with the admissible parameters $\delta_m = \delta_{p_0(m)}$, $r_Q(\delta_m) = r_m$, and $\mu_{m,Q}(\delta_m) = \mu_{m,Q}$. The growth condition $\mu_{m,Q} \le \mu_Q F(m)$ verifies the hypothesis in Equation (4.55) for the chosen m.

4.5.6 Quadrature Rules

We now fix a quadrature rule Q. For a family $\{Q_\eta\}$ of quadrature rules, we choose the quadrature parameters η so that $E_{\text{quad}}(Q_\eta) < \epsilon_{\text{comb}}$, leaving the residual budget $\epsilon_{Q_\eta} := \epsilon_{\text{comb}} E_{\text{quad}}(Q_\eta) > 0$. The final choice of quadrature parameters is then governed by the complexity estimates in Theorem 4.5.5 and Corollary 4.5.6.

4.5.6.1 Uniform Trapezoidal Rule

We first specialize to the uniform trapezoidal rule. Following the Low–Somma discretization strategy [LS25], we compare the infinite LCHS integral with the infinite trapezoidal sum

using strip analyticity, and then truncate the lattice to obtain the quadrature rule used in the LCU implementation. We use the notation $F_T(z) := \frac{1}{\sqrt{2\pi}} \hat{f}_{a,b}(z; c, d) U_z(T)$. Then $W_\infty = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}_{a,b}(k; c, d) U_k(T)\, dk = \int_{\mathbb{R}} F_T(k)\, dk$. We first state Lemma 4.5.11, which is proved in Section B.7, a basic estimate used in the complexity analysis of the uniform trapezoidal quadrature rule.

Lemma 4.5.11. *Fix a kernel profile $\vec{\theta} = (a, b, c, d)$, and define*

$$\rho_* := \begin{cases} 1, & a = 1, \\ \min\{1, b\}, & a > 1. \end{cases} \tag{4.80}$$

For every $\rho \in (0, \rho_*)$, F_T is analytic on an open set containing the closed strip $\overline{S}_\rho := \{z \in \mathbb{C} : |\operatorname{Im} z| \le \rho\}$. Moreover, $F_T(x + i\beta)$ decays uniformly to zero on $\overline{S}_\rho$ as $|x| \to \infty$, and $\sup_{|\beta| \le \rho} \int_{\mathbb{R}} \|F_T(x + i\beta)\|\, dx \le M_{a,b}(\rho)$, where $D_{a,b}(\rho) = (b - \rho)^{a-1}$ and

$$M_{a,b}(\rho) := \frac{c}{\sqrt{\pi}} \frac{(b+1)^{a-1}}{(1-\rho) D_{a,b}(\rho)} \exp\left(d + \frac{\rho^2 - 1}{4c^2} + \rho \max\{-d, d + T\|L\|\} \right). \tag{4.81}$$

We use the uniform trapezoidal rule because it converges exponentially for functions analytic in a strip around the real axis [LS25]. This is formalized below.

Lemma 4.5.12. *Let $\rho > 0$. Suppose that g is analytic on an open set containing $\overline{S}_\rho$ and that g decays uniformly to zero on $\overline{S}_\rho$ as $|z| \to \infty$. If $M > 0$ satisfies $\int_{\mathbb{R}} \|g(x + i\beta)\|\, dx \le M$ for every $\beta \in (-\rho, \rho)$, then, for every $h > 0$, we have*

$$\left\| \int_{\mathbb{R}} g(x)\, dx - h \sum_{q \in \mathbb{Z}} g(qh) \right\| \le \frac{2M}{e^{2\pi\rho/h} - 1}. \tag{4.82}$$

Proof. This is the exponentially convergent trapezoidal estimate of Trefethen–Weideman [TW14, Theorem 5.1], in the matrix-valued form used by Low–Somma [LS25, Lemma 10]. □

We now analyze the uniform trapezoidal rule in our setting. Fix $R > 0$ and $h > 0$, and set $n_h := \lceil \frac{R}{h} \rceil$ and $R_h := n_h h$. Then $R_h \geq R$ and $R_h < R + h$. Define the finite uniform trapezoidal rule $Q_{R,h}^{\mathrm{trap}} := \{(qh, h)\}_{q=-n_h}^{n_h}$. For this rule, $R_{Q_{R,h}^{\mathrm{trap}}} = R_h$. The LCHS quadrature coefficients are $v_q = \frac{h}{\sqrt{2\pi}} \hat{f}_{a,b}(qh; c, d)$. The corresponding ideal quadrature operator is $W_{R,h}^{\mathrm{trap,ideal}} := \frac{h}{\sqrt{2\pi}} \sum_{q=-n_h}^{n_h} \hat{f}_{a,b}(qh; c, d) U_{qh}(T) = h \sum_{q=-n_h}^{n_h} F_T(qh)$. The associated LCU normalization is $\alpha_{R,h}^{\mathrm{trap}} := \frac{h}{\sqrt{2\pi}} \sum_{q=-n_h}^{n_h} |\hat{f}_{a,b}(qh; c, d)|$ and discrete moments are $M_{\nu,R,h}^{\mathrm{trap}} := \frac{h}{\sqrt{2\pi}} \sum_{q=-n_h}^{n_h} |\hat{f}_{a,b}(qh; c, d)|\, |qh|^{\nu}$ for $\nu \geq 0$. Moreover, the omitted lattice tail is $E_{\mathrm{tail},h}(R_h) := \frac{h}{\sqrt{2\pi}} \sum_{|q|>n_h} |\hat{f}_{a,b}(qh; c, d)|$. Proposition 4.5.13 bounds the quadrature errors $E_{\mathrm{tail},h}(R_h)$ and $E_{\mathrm{trap}}(h, \rho)$, the latter arising from Lemma 4.5.12.

Proposition 4.5.13. *Fix a kernel profile $\vec{\theta} = (a, b, c, d)$, and let $\rho \in (0, \rho_*)$, where ρ_* is defined in Equation (4.80). Let $R > 0$, $h > 0$. Then*

$$E_{\mathrm{quad}}(Q_{R,h}^{\mathrm{trap}}) = \|W_\infty - W_{R,h}^{\mathrm{trap,ideal}}\| \leq E_{\mathrm{trap}}(h, \rho) + E_{\mathrm{tail},h}(R_h), \tag{4.83}$$

where $E_{\mathrm{trap}}(h, \rho) := \frac{2M_{a,b}(\rho)}{e^{2\pi\rho/h} - 1}$. Moreover, we have $E_{\mathrm{tail},h}(R_h) \leq E_{\mathrm{trunc}}(R_h) \leq E_{\mathrm{trunc}}(R)$. Consequently, we have $E_{\mathrm{quad}}(Q_{R,h}^{\mathrm{trap}}) \leq E_{\mathrm{trap}}(h, \rho) + E_{\mathrm{trunc}}(R)$.

Proof. Applying Lemma 4.5.12 to $g = F_T$, and using Lemma 4.5.11 with $M = M_{a,b}(\rho)$, gives

$$\left\| \int_{\mathbb{R}} F_T(x)\, dx - h \sum_{q \in \mathbb{Z}} F_T(qh) \right\| \leq E_{\mathrm{trap}}(h, \rho). \tag{4.84}$$

Therefore, we have $\|W_\infty - W_{R,h}^{\mathrm{trap,ideal}}\| \leq \| \int_{\mathbb{R}} F_T(x)\, dx - h \sum_{q \in \mathbb{Z}} F_T(qh) \| + h \sum_{|q|>n_h} \|F_T(qh)\| \leq$

$E_{\mathrm{trap}}(h,\rho) + E_{\mathrm{tail},h}(R_h)$. It remains to compare the lattice tail with the continuous truncation error. For real k, $U_k(T)$ is unitary and

$$|\hat{f}_{a,b}(k;c,d)| = \frac{(b+1)^{a-1}e^{d-\frac{1}{4c^2}}}{\sqrt{2\pi}} \frac{e^{-k^2/(4c^2)}}{\sqrt{1+k^2}(b^2+k^2)^{(a-1)/2}}. \tag{4.85}$$

For $k > 0$, $\frac{d}{dk}\log|\hat{f}_{a,b}(k;c,d)| = -\frac{k}{2c^2} - \frac{k}{1+k^2} - (a-1)\frac{k}{b^2+k^2} < 0$. Hence $k \mapsto |\hat{f}_{a,b}(k;c,d)|$ is decreasing on $[0,\infty)$. Since $R_h = n_h h$, monotonicity gives $h\sum_{q=n_h+1}^{\infty}|\hat{f}_{a,b}(qh;c,d)| \le \int_{R_h}^{\infty}|\hat{f}_{a,b}(k;c,d)|\,dk$. The same argument applies on the negative half-line. Multiplying by $1/\sqrt{2\pi}$ gives $E_{\mathrm{tail},h}(R_h) \le E_{\mathrm{trunc}}(R_h)$. Finally, since $R_h \ge R$, monotonicity of E_{trunc} gives $E_{\mathrm{trunc}}(R_h) \le E_{\mathrm{trunc}}(R)$. □

Proposition 4.5.13 bounds the quadrature error by mesh and tail errors. Then Corollary 4.5.14 gives sufficient conditions for controlling both. In what follows, we use the notation $C_{\mathrm{tail}} := \frac{c}{\sqrt{\pi}}\left(\frac{b+1}{b}\right)^{a-1} e^{d-\frac{1}{4c^2}}$, $C_{\mathrm{ker}} := \frac{1}{2\pi}\left(\frac{b+1}{b}\right)^{a-1} e^{d-\frac{1}{4c^2}}$ and $\widetilde{C}_{\mathrm{ker}} := \frac{(b+1)^{a-1}}{2\pi}e^{d-\frac{1}{4c^2}}$. Corollary 4.5.14 bounds the quantities that determine the post-quadrature parameters for the trapezoidal quadrature rule.

Corollary 4.5.14. *Let $\rho \in (0,\rho_*)$ and $\epsilon_{\mathrm{mesh}}, \epsilon_{\mathrm{tail}} > 0$. Define $\ell_{\mathrm{tail}} := \max\left\{1, \log\left(\frac{C_{\mathrm{tail}}}{\epsilon_{\mathrm{tail}}}\right)\right\}$ and $\ell_{\mathrm{mesh}} := \log\left(1 + \frac{2M_{a,b}(\rho)}{\epsilon_{\mathrm{mesh}}}\right)$. Choose $h = \min\left\{1, 2c, \frac{2\pi\rho}{\ell_{\mathrm{mesh}}}\right\}$ and $R = 2c\sqrt{\ell_{\mathrm{tail}}}$, and set $n_h = \lceil R/h \rceil$, $R_h = n_h h$. Then $E_{\mathrm{quad}}(Q_{R,h}^{\mathrm{trap}}) \le \epsilon_{\mathrm{mesh}} + \epsilon_{\mathrm{tail}}$. Moreover, we have*

$$|\mathcal{I}_{Q_{R,h}^{\mathrm{trap}}}| = 2n_h + 1 = \mathcal{O}\left(1 + (1+c)\sqrt{\ell_{\mathrm{tail}}} + \frac{c}{\rho}\sqrt{\ell_{\mathrm{tail}}}\,\ell_{\mathrm{mesh}}\right), \tag{4.86}$$

$$R_{Q_{R,h}^{\mathrm{trap}}} = R_h = \mathcal{O}\left(c\sqrt{\ell_{\mathrm{tail}}} + \min\left\{1, 2c, \frac{\rho}{\ell_{\mathrm{mesh}}}\right\}\right). \tag{4.87}$$

For a fixed admissible kernel profile and balanced budgets $\epsilon_{\mathrm{mesh}} \asymp \epsilon_{\mathrm{tail}} \asymp \epsilon_q$, we obtain

$$|\mathcal{I}_{Q_{R,h}^{\mathrm{trap}}}| = \mathcal{O}\left(\frac{c}{\rho}\log^{3/2}\left(\frac{1}{\epsilon_q}\right)\right), \quad R_{Q_{R,h}^{\mathrm{trap}}} = \mathcal{O}\left(c\sqrt{\log\left(\frac{1}{\epsilon_q}\right)}\right), \tag{4.88}$$

for fixed ρ and c. For $\nu \geq 0$, define

$$\mathfrak{M}_{\nu,a,b}(c) := \begin{cases} 1, & \nu < a-1, \\ 1+\log(1+c), & \nu = a-1, \\ 1+c^{\nu-a+1}, & \nu > a-1. \end{cases} \tag{4.89}$$

Then there exists a constant $C_{\nu,a,b}^{\mathrm{trap}} > 0$, depending only on ν, a, b, such that

$$M_{\nu,R,h}^{\mathrm{trap}} \leq \widetilde{C}_{\mathrm{ker}} C_{\nu,a,b}^{\mathrm{trap}} \mathfrak{M}_{\nu,a,b}(c). \tag{4.90}$$

In particular, $\alpha_{R,h}^{\mathrm{trap}} = M_{0,R,h}^{\mathrm{trap}} \leq \widetilde{C}_{\mathrm{ker}} C_{0,a,b}^{\mathrm{trap}} \mathfrak{M}_{0,a,b}(c)$.

Proof. The choice of h controls the mesh error. By Lemma 4.5.12, $E_{\mathrm{trap}}(h,\rho) \leq \epsilon_{\mathrm{mesh}}$. It remains to control the truncation error. By Corollary 4.4.7, $E_{\mathrm{trunc}}(R) \leq C_{\mathrm{tail}}\,\mathrm{erfc}(R/(2c))$. Since $\mathrm{erfc}(x) \leq e^{-x^2}$, the condition $R \geq 2c\sqrt{\ell_{\mathrm{tail}}}$ implies $E_{\mathrm{trunc}}(R) \leq \epsilon_{\mathrm{tail}}$. Combining these estimates with Proposition 4.5.13 gives $E_{\mathrm{quad}}(Q_{R,h}^{\mathrm{trap}}) \leq E_{\mathrm{trap}}(h,\rho) + E_{\mathrm{trunc}}(R) \leq \epsilon_{\mathrm{mesh}} + \epsilon_{\mathrm{tail}}$. Since $n_h = \lceil R/h \rceil$, we have $|\mathcal{I}_{Q_{R,h}^{\mathrm{trap}}}| = 2n_h + 1 \leq 2R/h + 3$. Moreover, we have $\frac{1}{h} = \max\left\{1, \frac{1}{2c}, \frac{\ell_{\mathrm{mesh}}}{2\pi\rho}\right\} = \mathcal{O}\left(1 + \frac{1}{c} + \frac{\ell_{\mathrm{mesh}}}{\rho}\right)$. Substituting $R = 2c\sqrt{\ell_{\mathrm{tail}}}$ gives the bound on $|\mathcal{I}_{Q_{R,h}^{\mathrm{trap}}}|$ in Equation (4.86). Moreover, $R_h = n_h h \leq R + h$, giving the bound on $R_{Q_{R,h}^{\mathrm{trap}}}$ in Equation (4.87). The balanced-budget estimates follow from $\ell_{\mathrm{mesh}} = \ell_{\mathrm{tail}} = \mathcal{O}(\log(1/\epsilon_q))$, for fixed ρ and c. It remains to

bound the LCU normalization and discrete moments. For real k, Equation (4.85) implies $\frac{1}{\sqrt{2\pi}}|\hat{f}_{a,b}(k;c,d)| = \widetilde{C}_{\ker}\frac{e^{-k^2/(4c^2)}}{\sqrt{1+k^2}(b^2+k^2)^{(a-1)/2}}$. Therefore, we have

$$M^{\mathrm{trap}}_{\nu,R,h} \le \widetilde{C}_{\ker} h \sum_{q\in\mathbb{Z}} \frac{e^{-(qh)^2/(4c^2)}|qh|^{\nu}}{\sqrt{1+(qh)^2}(b^2+(qh)^2)^{(a-1)/2}}. \tag{4.91}$$

For $k > 0$, we have

$$\frac{1}{\sqrt{1+k^2}(b^2+k^2)^{(a-1)/2}} \le \min\left\{\frac{1}{b^{a-1}}, \frac{1}{k^a}\right\}. \tag{4.92}$$

Indeed, the first bound follows from $\sqrt{1+k^2} \ge 1$ and $(b^2+k^2)^{(a-1)/2} \ge b^{a-1}$, while the second follows from $\sqrt{1+k^2} \ge k$ and $(b^2+k^2)^{(a-1)/2} \ge k^{a-1}$. Let

$$\kappa_b := \begin{cases} 1, & a = 1, \\ b^{(a-1)/a}, & a > 1. \end{cases} \tag{4.93}$$

We split the full-line sum in Equation (4.91) into the regions $|qh| \le \kappa_b$ and $|qh| > \kappa_b$. Since $h \le 1$, the first contribution is bounded by a constant depending only on ν, a, b. More precisely,

$$h \sum_{|qh|\le\kappa_b} \frac{e^{-(qh)^2/(4c^2)}|qh|^{\nu}}{\sqrt{1+(qh)^2}(b^2+(qh)^2)^{(a-1)/2}} \le C_{\nu,a,b}. \tag{4.94}$$

On the complementary region, Equation (4.92) gives

$$h \sum_{|qh|>\kappa_b} \frac{e^{-(qh)^2/(4c^2)}|qh|^{\nu}}{\sqrt{1+(qh)^2}(b^2+(qh)^2)^{(a-1)/2}} \le h \sum_{|qh|>\kappa_b} e^{-(qh)^2/(4c^2)}|qh|^{\nu-a}. \tag{4.95}$$

The latter Riemann sum has three regimes. If $\nu < a-1$, then $|k|^{\nu-a}$ is integrable at infinity, and hence $h\sum_{|qh|>\kappa_b} e^{-(qh)^2/(4c^2)}|qh|^{\nu-a} \le C_{\nu,a,b}$. If $\nu = a-1$, then the summand behaves like

$1/|qh|$ until the Gaussian cutoff scale $|qh| \sim c$, and therefore $h \sum_{|qh|>\kappa_b} e^{-(qh)^2/(4c^2)}|qh|^{-1} \leq C_{a,b}\big(1+\log(1+c)\big)$. If $\nu > a-1$, comparison with the corresponding Gaussian integral gives

$$h \sum_{|qh|>\kappa_b} e^{-(qh)^2/(4c^2)}|qh|^{\nu-a} \leq C_{\nu,a,b} + C_\nu \int_0^\infty e^{-k^2/(4c^2)} k^{\nu-a}\, dk \leq C_{\nu,a,b} + C_\nu c^{\nu-a+1}. \tag{4.96}$$

Combining these three cases with Equation (4.91) proves Equation (4.90). The normalization bound is the case $\nu = 0$, and the asymptotic statement follows directly from the definition of $\mathfrak{M}_{\nu,a,b}(c)$. □

Corollary 4.4.5 shows that, when the kernel parameters satisfy $a, d = \mathcal{O}(1)$, one may take $c = \Omega(\sqrt{\log(1/\epsilon_{\text{approx}})})$. Assuming that all error-tolerance parameters are of the same order as $\epsilon > 0$, Corollary 4.5.14 implies

$$|\mathcal{I}_{Q^{\text{trap}}_{R,h}}| = \mathcal{O}\left(\frac{1}{\rho}\log^2\left(\frac{1}{\epsilon_q}\right)\right), \quad R_{Q^{\text{trap}}_{R,h}} = \mathcal{O}\left(\log\left(\frac{1}{\epsilon_q}\right)\right). \tag{4.97}$$

Moreover, Corollary 4.5.14 yields exponential mesh-error decay in $1/h$, Gaussian tail decay in R, and moment estimates that retain the rational decay of the kernel, improving the crude Gaussian scaling $\mathcal{O}((\log(1/\epsilon_q))^{(\nu+1)/2})$ to an a-dependent scaling.

4.5.6.2 Sinh–Sinh Trapezoidal Rule

We next consider a non-uniform quadrature rule obtained by applying the trapezoidal rule after the change of variables $k = \phi_\eta(x) := \eta \sinh x$, where $0 < \eta \leq 2c$ is a free scale parameter. Set $\lambda_\eta := \eta/(2c)$. Under this change of variables, the Gaussian factor in $\hat{f}_{a,b}$ becomes $e^{-\phi_\eta(x)^2/(4c^2)} = e^{-\lambda_\eta^2 \sinh^2 x}$, so the transformed integrand has double-exponential decay

along the real axis. The choice $\eta = 2c$ gives the Gaussian-normalized case $e^{-\lambda_\eta^2 \sinh^2 x} = e^{-\sinh^2 x}$. See [TW14, Section 15] for more on this quadrature rule. Define

$$G_{T,\eta}(z) := F_T(\phi_\eta(z))\phi_\eta'(z) = \eta \cosh z \, F_T(\eta \sinh z). \tag{4.98}$$

Then, by the change of variables $k = \phi_\eta(x)$, we have $W_\infty = \int_{\mathbb{R}} F_T(k)\, dk = \int_{\mathbb{R}} G_{T,\eta}(x)\, dx$. The exponential convergence of the sinh–sinh rule follows by applying the infinite-line trapezoidal estimate to the transformed integrand $G_{T,\eta}$. The relevant analyticity strip is now an x-strip rather than the original k-strip. Let

$$\beta_{*,\eta} := \sup\{\beta > 0 : G_{T,\eta} \text{ is analytic on an open set containing } \{z \in \mathbb{C} : |\operatorname{Im} z| \le \beta\}\}. \tag{4.99}$$

Equivalently, $\beta_{*,\eta}$ is limited by the nearest preimage of the poles $k = -i$ and $k = ib$ under $z \mapsto \eta \sinh z$, and by the requirement that the transformed Gaussian decay on horizontal translates. In particular, one may restrict to $\beta < \pi/4$, since $\operatorname{Re}(\sinh^2(x + i\beta)) = \sinh^2 x \cos(2\beta) - \sin^2 \beta$ has positive quadratic growth in $|x|$ whenever $|\beta| < \pi/4$. For $\beta \in (0, \min\{\beta_{*,\eta}, \pi/4\})$, define $\delta_{1,\eta}(\beta) := \inf_{x\in\mathbb{R}|\omega|\le\beta} |1 - i\phi_\eta(x + i\omega)|$ and $\delta_{b,\eta}(\beta) := \inf_{\substack{x\in\mathbb{R} \\ |\omega|\le\beta}} |b + i\phi_\eta(x + i\omega)|$. Here $\delta_{b,\eta}(\beta)$ is only needed when $a > 1$. Since $\beta < \beta_{*,\eta}$, these quantities are strictly positive. Set $\rho_\beta := \cos(2\beta) > 0$ and $A_{\eta,\beta} := \eta \sin\beta \left(|d| + T\|L\|\right)$. Define

$$M_{\sinh,\eta}(\beta) := \frac{\eta(b+1)^{a-1}}{2\pi\, \delta_{1,\eta}(\beta)\, \delta_{b,\eta}(\beta)^{a-1}} \exp\left(d - \frac{1}{4c^2} + \lambda_\eta^2 \sin^2\beta + A_{\eta,\beta} + \frac{A_{\eta,\beta}^2}{2\lambda_\eta^2 \rho_\beta}\right) \sqrt{\frac{2\pi}{\lambda_\eta^2 \rho_\beta}}, \tag{4.100}$$

with the convention $\delta_{b,\eta}(\beta)^{a-1} = 1$ when $a = 1$.

Lemma 4.5.15. *Let $0 < \eta \le 2c$ and $\beta \in (0, \min\{\beta_{*,\eta}, \pi/4\})$. Then $G_{T,\eta}$ is analytic on an open set containing the closed strip $\overline{S}_\beta := \{z \in \mathbb{C} : |\operatorname{Im} z| \le \beta\}$. Moreover, $G_{T,\eta}(x + i\omega)$ decays uniformly to zero on $\overline{S}_\beta$ as $|x| \to \infty$, and*

$$\sup_{|\omega| \le \beta} \int_{\mathbb{R}} \|G_{T,\eta}(x + i\omega)\| \, dx \le M_{\sinh,\eta}(\beta). \tag{4.101}$$

Consequently, for every $h > 0$,

$$\left\| \int_{\mathbb{R}} G_{T,\eta}(x) \, dx - h \sum_{q \in \mathbb{Z}} G_{T,\eta}(qh) \right\| \le \frac{2M_{\sinh,\eta}(\beta)}{e^{2\pi\beta/h} - 1}. \tag{4.102}$$

Proof. Let $z = x + i\omega$, with $|\omega| \le \beta$, and write $\phi_\eta(z) = \eta \sinh z = u + iv$, where $u = \eta \sinh x \cos\omega$ and $v = \eta \cosh x \sin\omega$. Since $\beta < \beta_{*,\eta}$, the transformed contour stays a positive distance from the pole $k = -i$, and, when $a > 1$, from the pole $k = ib$. Hence $|1 - i\phi_\eta(z)| \ge \delta_{1,\eta}(\beta)$ and, for $a > 1$, $|b + i\phi_\eta(z)| \ge \delta_{b,\eta}(\beta)$. Moreover, $|e^{d(1 - i\phi_\eta(z))}| = e^{d(1+v)} \le e^d e^{|d||v|}$. Using the time-independent bound for $U_k(T)$ as in Lemma 4.5.11, we have $\|U_{\phi_\eta(z)}(T)\| \le e^{T\|L\| \, |v|}$. Since $|v| \le \eta \sin\beta \, \cosh x$, the combined non-Gaussian exponential growth is bounded by $\exp(A_{\eta,\beta} \cosh x)$. Next, we have $\operatorname{Re}(\sinh^2(x + i\omega)) = \sinh^2 x \cos(2\omega) - \sin^2\omega \ge \rho_\beta \sinh^2 x - \sin^2\beta$. Therefore, the Gaussian factor satisfies

$$|e^{-(\phi_\eta(z)^2+1)/(4c^2)}| = e^{-\lambda_\eta^2 \operatorname{Re}(\sinh^2 z) - 1/(4c^2)} \le \exp\left(-\lambda_\eta^2 \rho_\beta \sinh^2 x + \lambda_\eta^2 \sin^2\beta - \frac{1}{4c^2} \right). \tag{4.103}$$

Also $|\cosh(x+i\omega)| \le \cosh x$. Combining these estimates gives

$$\|G_{T,\eta}(x+i\omega)\| \le \frac{\eta(b+1)^{a-1}}{2\pi\,\delta_{1,\eta}(\beta)\,\delta_{b,\eta}(\beta)^{a-1}} \exp\left(d - \frac{1}{4c^2} + \lambda_\eta^2 \sin^2\beta\right) \tag{4.104}$$

$$\times \cosh x \exp\left(-\lambda_\eta^2 \rho_\beta \sinh^2 x + A_{\eta,\beta}\cosh x\right). \tag{4.105}$$

Using $\cosh x \le 1 + |\sinh x|$ and $A_{\eta,\beta}|\sinh x| \le \frac{\lambda_\eta^2\rho_\beta}{2}\sinh^2 x + \frac{A_{\eta,\beta}^2}{2\lambda_\eta^2\rho_\beta}$, we obtain $-\lambda_\eta^2\rho_\beta\sinh^2 x + A_{\eta,\beta}\cosh x \le -\frac{\lambda_\eta^2\rho_\beta}{2}\sinh^2 x + A_{\eta,\beta} + \frac{A_{\eta,\beta}^2}{2\lambda_\eta^2\rho_\beta}$. Hence, we have

$$\int_{\mathbb{R}} \|G_{T,\eta}(x+i\omega)\|\,dx \le \frac{\eta(b+1)^{a-1}}{2\pi\,\delta_{1,\eta}(\beta)\,\delta_{b,\eta}(\beta)^{a-1}} \exp\left(d - \frac{1}{4c^2} + \lambda_\eta^2\sin^2\beta + A_{\eta,\beta} + \frac{A_{\eta,\beta}^2}{2\lambda_\eta^2\rho_\beta}\right) \tag{4.106}$$

$$\times \int_{\mathbb{R}} \cosh x\, e^{-\frac{\lambda_\eta^2\rho_\beta}{2}\sinh^2 x}\,dx. \tag{4.107}$$

With the substitution $u = \sinh x$, $du = \cosh x\,dx$, we get $\int_{\mathbb{R}} \cosh x\, e^{-\frac{\lambda_\eta^2\rho_\beta}{2}\sinh^2 x}\,dx = \int_{\mathbb{R}} e^{-\frac{\lambda_\eta^2\rho_\beta}{2}u^2}\,du = \sqrt{\frac{2\pi}{\lambda_\eta^2\rho_\beta}}$. This proves Equation (4.101) with $M_{\sinh,\eta}(\beta)$. The same bound also implies uniform decay as $|x| \to \infty$. Finally, applying Lemma 4.5.12 to $g = G_{T,\eta}$ gives Equation (4.102). □

Lemma 4.5.15 gives exponential mesh convergence in $1/h$. The advantage of the sinh–sinh transformation is that the lattice truncation error decays double-exponentially in the x-truncation length. Fix $Y > 0$ and $h > 0$. As in Section 4.5.6.1, set $n_h := \lceil \frac{Y}{h} \rceil$ and $Y_h := n_h h$. Define the finite sinh–sinh quadrature rule $Q_{Y,h}^{\sinh,\eta} := \{(\eta\sinh(qh, h\cosh(qh))\}_{q=-n_h}^{n_h}$. For this rule, $R_{Q_{Y,h}^{\sinh,\eta}} = \eta\sinh(Y_h)$. The corresponding ideal quadrature operator is

$$W_{Y,h}^{\sinh,\eta,\text{ideal}} := h\sum_{q=-n_h}^{n_h} G_{T,\eta}(qh) = \frac{\eta h}{\sqrt{2\pi}} \sum_{q=-n_h}^{n_h} \cosh(qh)\,\hat{f}_{a,b}(\eta\sinh(qh); c, d) U_{\eta\sinh(qh)}(T). \tag{4.108}$$

The LCHS quadrature coefficients are $v_q = \frac{\eta h \cosh(qh)}{\sqrt{2\pi}} \hat{f}_{a,b}(\eta \sinh(qh); c, d)$. The LCU normalization is $\alpha_{Y,h}^{\sinh,\eta} := \sum_{q=-n_h}^{n_h} |v_q|$, and the discrete moments are $M_{\nu,Y,h}^{\sinh,\eta} := \sum_{q=-n_h}^{n_h} |v_q| \, |\eta \sinh(qh)|^\nu$, for $\nu \geq 0$. Moreover, the omitted transformed-lattice tail is $E_{\mathrm{tail},h}^{\sinh,\eta}(Y_h) := h \sum_{|q|>n_h} \|G_{T,\eta}(qh)\|$. Proposition 4.5.16 is the analogue of Proposition 4.5.13.

Proposition 4.5.16. *Let $0 < \eta \leq 2c$ and $\beta \in (0, \min\{\beta_{*,\eta}, \pi/4\})$. Let $Y > 0$, $h > 0$. Then*

$$E_{\mathrm{quad}}(Q_{Y,h}^{\sinh,\eta}) = \left\| W_\infty - W_{Y,h}^{\sinh,\eta,\mathrm{ideal}} \right\| \leq E_{\sinh,\eta}(h, \beta) + E_{\mathrm{tail},h}^{\sinh,\eta}(Y_h), \tag{4.109}$$

where $E_{\sinh,\eta}(h, \beta) := \frac{2M_{\sinh,\eta}(\beta)}{e^{2\pi\beta/h}-1}$. Moreover, if $\lambda_\eta \sinh Y_h \geq 1$, then

$$E_{\mathrm{tail},h}^{\sinh,\eta}(Y_h) \leq 2c\sqrt{\pi}\, C_{\mathrm{ker}} \, \mathrm{erfc}(\lambda_\eta \sinh Y_h) \leq 2c\sqrt{\pi}\, C_{\mathrm{ker}} \, e^{-\lambda_\eta^2 \sinh^2 Y_h}, \tag{4.110}$$

where $C_{\mathrm{ker}} := \frac{1}{2\pi} \left(\frac{b+1}{b}\right)^{a-1} e^{d - \frac{1}{4c^2}}$.

Proof. The mesh contribution follows from Lemma 4.5.15. For the tail, use the real-axis bound $\|G_{T,\eta}(x)\| \leq \eta C_{\mathrm{ker}} \cosh x \, e^{-\lambda_\eta^2 \sinh^2 x}$, which follows from the Gaussian majorant for $|\hat{f}_{a,b}|$ and unitarity of $U_k(T)$ for real k. The function $x \mapsto \cosh x \, e^{-\lambda_\eta^2 \sinh^2 x}$ is decreasing whenever $\lambda_\eta \sinh x \geq 1$. Hence, if $\lambda_\eta \sinh Y_h \geq 1$, then monotonicity gives

$$E_{\mathrm{tail},h}^{\sinh,\eta}(Y_h) \leq 2 \int_{Y_h}^{\infty} \eta C_{\mathrm{ker}} \cosh x \, e^{-\lambda_\eta^2 \sinh^2 x} \, dx \tag{4.111}$$

$$= 2\eta C_{\mathrm{ker}} \int_{\sinh Y_h}^{\infty} e^{-\lambda_\eta^2 u^2} \, du = 2c\sqrt{\pi}\, C_{\mathrm{ker}} \, \mathrm{erfc}(\lambda_\eta \sinh Y_h). \tag{4.112}$$

The final inequality follows from $\mathrm{erfc}(x) \leq e^{-x^2}$. □

Proposition 4.5.16 shows that the sinh–sinh rule has exponential mesh convergence in $1/h$

and double-exponential truncation decay in the transformed cutoff Y. Finally, Corollary 4.5.17 bounds the quantities determining the implementation complexity of the sinh quadrature rule. Since the proof is similar to that of Corollary 4.5.14, it is given in Section B.8.

Corollary 4.5.17. *Let $0 < \eta \le 2c$, $\beta \in (0, \min\{\beta_{*,\eta}, \pi/4\})$, and $\epsilon_{\mathrm{mesh}}, \epsilon_{\mathrm{tail}} > 0$. Let $ell_{\mathrm{mesh}}^{\sinh,\eta} := \log\left(1 + \frac{2M_{\sinh,\eta}(\beta)}{\epsilon_{\mathrm{mesh}}}\right)$ and $\ell_{\mathrm{tail}}^{\sinh,\eta} := \max\left\{1, \log\left(\frac{2c\sqrt{\pi}C_{\mathrm{ker}}}{\epsilon_{\mathrm{tail}}}\right)\right\}$. Choose $h = \min\left\{1, \frac{2\pi\beta}{\ell_{\mathrm{mesh}}^{\sinh,\eta}}\right\}$, $Y = \max\left\{1, \operatorname{arsinh}\left(\frac{\sqrt{\ell_{\mathrm{tail}}^{\sinh,\eta}}}{\lambda_\eta}\right)\right\} = \max\left\{1, \operatorname{arsinh}\left(\frac{2c}{\eta}\sqrt{\ell_{\mathrm{tail}}^{\sinh,\eta}}\right)\right\}$ and set $n_h = \lceil Y/h \rceil$, $Y_h = n_h h$. Then $E_{\mathrm{quad}}(Q_{Y,h}^{\sinh,\eta}) \le \epsilon_{\mathrm{mesh}} + \epsilon_{\mathrm{tail}}$. Moreover, we have*

$$|\mathcal{I}_{Q_{Y,h}^{\sinh,\eta}}| = 2n_h + 1 = \mathcal{O}\left(1 + \log\left(1 + \frac{\sqrt{\ell_{\mathrm{tail}}^{\sinh,\eta}}}{\lambda_\eta}\right)\left(1 + \frac{\ell_{\mathrm{mesh}}^{\sinh,\eta}}{\beta}\right)\right), \tag{4.113}$$

$$R_{Q_{Y,h}^{\sinh,\eta}} = \eta \sinh(Y_h) = \mathcal{O}\left(c\sqrt{\ell_{\mathrm{tail}}^{\sinh,\eta}} + \eta\right). \tag{4.114}$$

For a fixed admissible kernel profile and balanced computational budgets $\epsilon_{\mathrm{mesh}} \asymp \epsilon_{\mathrm{tail}} \asymp \epsilon_q$, we obtain

$$|\mathcal{I}_{Q_{Y,h}^{\sinh,\eta}}| = \mathcal{O}\left(\frac{1}{\beta}\log\left(\frac{1}{\epsilon_q}\right)\log\left(1 + \frac{2c}{\eta}\sqrt{\log\left(\frac{1}{\epsilon_q}\right)}\right)\right), \quad R_{Q_{Y,h}^{\sinh,\eta}} = \mathcal{O}\left(c\sqrt{\log\left(\frac{1}{\epsilon_q}\right)} + \eta\right). \tag{4.115}$$

Define $\mathfrak{M}_{\nu,a,b}(c)$ as in Equation (4.89). Then there exists a constant $C_{\nu,a,b}^{\sinh} > 0$, depending only on ν, a, b, such that $M_{\nu,Y,h}^{\sinh,\eta} \le \widetilde{C}_{\mathrm{ker}} C_{\nu,a,b}^{\sinh} \mathfrak{M}_{\nu,a,b}(c)$. In particular, $\alpha_{Y,h}^{\sinh,\eta} = M_{0,Y,h}^{\sinh,\eta} \le \widetilde{C}_{\mathrm{ker}} C_{0,a,b}^{\sinh} \mathfrak{M}_{0,a,b}(c)$. Equivalently, the sinh–sinh moments satisfy the same scaling as in Corollary 4.5.14.

If one takes $a, d = \mathcal{O}(1)$, $c = \Theta(\sqrt{\log(1/\epsilon_{\mathrm{approx}})})$, and assumes that all error parameters are of order $\epsilon > 0$, then Corollary 4.5.17 shows that, for a fixed free-scale choice $\eta = \Theta(\min(1, b))$,

the transformed strip width satisfies $\beta^{-1} = \mathcal{O}(1)$. Therefore

$$|\mathcal{I}_{Q_{Y,h}^{\sinh,\eta}}| = \mathcal{O}\left(\log\left(\frac{1}{\epsilon_q}\right)\log\log\left(\frac{1}{\epsilon_q}\right)\right), \tag{4.116}$$

which is asymptotically smaller than $\log^{3/2}(1/\epsilon_q)$.

4.6 Extensions to Non-Homogeneous and Time-Dependent Cases

In Section 4.4 and Section 4.5, we analyzed the time-independent homogeneous case in detail. In this section, we explain how the same framework extends to more general cases, namely the non-homogeneous case in Section 4.6.1 and the time-dependent case in Section 4.6.2.

4.6.1 Non-Homogeneous Case

We first consider the time-independent non-homogeneous problem

$$\frac{d}{dt}u(t) = -Au(t) + b(t). \tag{4.117}$$

As before, write $A = L + iH$, with L, H Hermitian and $L \succeq 0$. By Duhamel's principle, the solution of Equation (4.117) is

$$u(T) = e^{-AT}u(0) + \int_0^T e^{-A(T-s)}b(s)\,ds. \tag{4.118}$$

Equation (4.118) shows that the non-homogeneous term is a superposition of homogeneous evolutions over times $T-s$. Applying the LCHS representation to each homogeneous propagator

in the Duhamel term gives the two-variable representation

$$\int_0^T e^{-A(T-s)}b(s)\,ds \approx \frac{1}{\sqrt{2\pi}}\int_0^T\int_{\mathbb{R}} \hat{f}_{a,b}(k;c,d)U_k(T-s)b(s)\,dk\,ds. \tag{4.119}$$

Thus, the non-homogeneous contribution involves two discretizations: the LCHS quadrature Q in the Fourier variable k, and the time quadrature P in the Duhamel variable s. This viewpoint is used below. For $\tau \in [0,T]$, define the ideal and MPF-implemented post-quadrature operators by $W_Q^{\text{ideal}}(\tau) := \sum_{i\in\mathcal{I}_Q} v_i U_{k_i}(\tau)$ and $W_Q^{\text{MPF}}(\tau) := \sum_{i\in\mathcal{I}_Q} v_i U_{k_i}^{(2m)}(\tau)$. The quantities α_Q, $\Lambda_{m,Q}$, and $\Phi_{m,Q}^*$ are defined as before. Let $P = \{(s_p,\omega_p) : p \in \mathcal{I}_P\}$ be a quadrature rule on $[0,T]$ for the integral in Equation (4.119). We write $\tau_p := T - s_p$ and $B_P := \sum_{p\in\mathcal{I}_P} |\omega_p|\,\|b(s_p)\|$. We assume $B_P < \infty$.

The assumption $b \in L^1([0,T])$ suffices for Equation (4.118), but does not define point values $b(s_p)$. The quadrature implementation is therefore stated under the additional source-access assumption that the selected values $b(s_p)$ are well-defined and accessible via a state-preparation oracles. Discretizing Equation (4.119) by Q in k and P in s, and then replacing each $U_{k_i}(\tau_p)$ by its MPF approximation, gives $\frac{1}{\sqrt{2\pi}}\int_0^T\int_{\mathbb{R}} \hat{f}_{a,b}(k;c,d)U_k(T-s)b(s)\,dk\,ds \approx \sum_{p\in\mathcal{I}_P}\sum_{i\in\mathcal{I}_Q} \omega_p v_i U_{k_i}^{(2m)}(\tau_p)b(s_p)$. Equivalently, the non-homogeneous LCHS–MPF approximation is

$$u_{Q,P}^{\text{MPF}}(T) := W_Q^{\text{MPF}}(T)u(0) + \sum_{p\in\mathcal{I}_P} \omega_p W_Q^{\text{MPF}}(\tau_p)b(s_p). \tag{4.120}$$

We state the error estimate in terms of two quadrature errors. First, define the uniform-in-time

LCHS quadrature error by

$$E_{\mathrm{quad}}^{[0,T]}(Q) := \sup_{0\le\tau\le T} \left\| \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}_{a,b}(k;c,d) U_k(\tau)\, dk - W_Q^{\mathrm{ideal}}(\tau) \right\|. \tag{4.121}$$

The quadrature estimates in Section 4.5.6 also bound $E_{\mathrm{quad}}^{[0,T]}(Q)$. Indeed, the strip estimates used there are monotone in the final time. Replacing T by any $\tau \in [0,T]$ only decreases the factor involving $T\|L\|$. Hence the same quadrature rule Q, chosen using the endpoint time T, satisfies $E_{\mathrm{quad}}^{[0,T]}(Q) \le E_{\mathrm{mesh}}(Q;T) + E_{\mathrm{tail}}(Q;T)$, with the right-hand side given by the corresponding estimates in Corollary 4.5.14 or Corollary 4.5.17. Second, define the time-quadrature error by

$$E_{\mathrm{time}}(P) := \left\| \int_0^T e^{-A(T-s)} b(s)\, ds - \sum_{p\in\mathcal{I}_P} \omega_p e^{-A(T-s_p)} b(s_p) \right\|. \tag{4.122}$$

The quantity $E_{\mathrm{time}}(P)$ is kept abstract below. Once P and the regularity of the representative b are specified, it can be bounded by the corresponding quadrature estimate.

Lemma 4.6.1. *Fix an admissible kernel profile $\vec{\theta} = (a,b,c,d)$ and a quadrature rule Q. Let $T,\delta > 0$, $m,r,N \in \mathbb{N}$, and set $\Delta = T/r$. Assume that $0 < \Delta \le \Delta(R_Q,N,\delta)$. Then*

$$\sup_{0\le\tau\le T} \left\| e^{-A\tau} - W_Q^{\mathrm{MPF}}(\tau) \right\| \le E_{\mathrm{approx}}(y_0) + E_{\mathrm{quad}}^{[0,T]}(Q) \tag{4.123}$$

$$+ (1+\eta_Q(T/r))^{r-1} \left(C_m \frac{T^{2m+1}}{r^{2m}} \Lambda_{m,Q} + r\alpha_Q R_m(\delta) \right). \tag{4.124}$$

Proof. The proof is the same as that of Lemma 4.5.2, applied uniformly for $\tau \in [0,T]$. The approximation and quadrature errors are bounded by $E_{\mathrm{approx}}(y_0)$ and $E_{\mathrm{quad}}^{[0,T]}(Q)$, respectively. For the MPF term, the step size over time τ is τ/r, so $\tau/r \le T/r \le \Delta(R_Q,N,\delta)$. Since $\tau \le T$,

the principal MPF error term is bounded by the corresponding expression with T in place of τ. This proves Equations (4.123) and (4.124). □

We now combine the three error sources in the non-homogeneous construction. The homogeneous propagator error is needed uniformly for all $\tau \in [0, T]$, since the source term contains propagators $e^{-A(T-s)}$. Proposition 4.6.2 separates this uniform LCHS–MPF error from the additional time-quadrature error used to discretize the source term.

Proposition 4.6.2. *Let $u(T)$ be the solution of Equation (4.117), and let $u_{Q,P}^{\mathrm{MPF}}(T)$ be defined by Equation (4.120). Then*

$$\left\| u(T) - u_{Q,P}^{\mathrm{MPF}}(T) \right\| \leq E_{\mathrm{time}}(P) + \mathcal{E}_Q^{[0,T]}(r, \delta) \left(\|u(0)\| + B_P \right), \tag{4.125}$$

where

$$\mathcal{E}_Q^{[0,T]}(r, \delta) := E_{\mathrm{approx}}(y_0) + E_{\mathrm{quad}}^{[0,T]}(Q) + (1 + \eta_Q(T/r))^{r-1} \left(C_m \frac{T^{2m+1}}{r^{2m}} \Lambda_{m,Q} + r \alpha_Q R_m(\delta) \right). \tag{4.126}$$

Proof. Using Duhamel's formula, Equation (4.118), and the definition of $u_{Q,P}^{\mathrm{MPF}}(T)$, we write

$$u(T) - u_{Q,P}^{\mathrm{MPF}}(T) = \left(e^{-AT} - W_Q^{\mathrm{MPF}}(T) \right) u(0) \tag{4.127}$$

$$+ \left(\int_0^T e^{-A(T-s)} b(s)\, ds - \sum_{p \in \mathcal{I}_P} \omega_p e^{-A(T-s_p)} b(s_p) \right) \tag{4.128}$$

$$+ \sum_{p \in \mathcal{I}_P} \omega_p \left(e^{-A\tau_p} - W_Q^{\mathrm{MPF}}(\tau_p) \right) b(s_p). \tag{4.129}$$

Taking norms, using Equation (4.122), and applying Lemma 4.6.1 to T and to each $\tau_p \in [0, T]$,

we obtain

$$\left\|u(T) - u_{Q,P}^{\mathrm{MPF}}(T)\right\| \leq \mathcal{E}_Q^{[0,T]}(r,\delta)\|u(0)\| + E_{\mathrm{time}}(P) + \mathcal{E}_Q^{[0,T]}(r,\delta)\sum_{p\in\mathcal{I}_P}|\omega_p|\,\|b(s_p)\|. \tag{4.130}$$

This proves Equation (4.125). □

Proposition 4.6.2 gives an abstract error decomposition. We next choose the time-quadrature, approximation, and post-quadrature error budgets so that their total contribution is at most the target accuracy ϵ. This gives the fixed-order non-homogeneous analogue of Theorem 4.5.5.

Corollary 4.6.3. *Let $\epsilon > 0$. Choose $\epsilon_{\mathrm{time}}, \epsilon_{\mathrm{approx}}$, and $\epsilon_{\mathrm{comb}} > 0$ such that*

$$\epsilon_{\mathrm{time}} + (\|u(0)\| + B_P)\,(\epsilon_{\mathrm{approx}} + \epsilon_{\mathrm{comb}}) \leq \epsilon. \tag{4.131}$$

Fix P and assume that $E_{\mathrm{time}}(P) \leq \epsilon_{\mathrm{time}}$. Choose an admissible kernel satisfying $E_{\mathrm{approx}}(y_0) \leq \epsilon_{\mathrm{approx}}$. Suppose that Q satisfies $E_{\mathrm{quad}}^{[0,T]}(Q) < \epsilon_{\mathrm{comb}}$, and define $\epsilon_Q := \epsilon_{\mathrm{comb}} - E_{\mathrm{quad}}^{[0,T]}(Q) > 0$. If $r = r_Q(\delta)$ is chosen as in Equation (4.49), with this value of ϵ_Q, and if the admissibility conditions $r_Q(\delta)R_m(\delta) \leq \frac{1}{2}$ and $r_Q(\delta)\alpha_Q R_m(\delta) \leq \frac{\epsilon_Q}{2e}$ hold, then $\|u(T) - u_{Q,P}^{\mathrm{MPF}}(T)\| \leq \epsilon$.

Proof. For $r = r_Q(\delta)$, the same argument used in Theorem 4.5.5 gives $\mathcal{E}_Q^{[0,T]}(r_Q(\delta),\delta) \leq \epsilon_{\mathrm{approx}} + \epsilon_{\mathrm{comb}}$. Combining this bound with Proposition 4.6.2 and Equation (4.131) proves the claim. □

We now describe the implementation cost. Assume state-preparation oracles for $|\widetilde{u}(0)\rangle := u(0)/\|u(0)\|$, when $u(0) \neq 0$, and for $\left|\widetilde{b}(s_p)\right\rangle := b(s_p)/\|b(s_p)\|$, for each $p \in \mathcal{I}_P$ with $b(s_p) \neq 0$. Terms with zero vector coefficients are omitted from the outer LCU, and the phases of ω_p are

included in the outer preparation. Let $\beta_P := \|u(0)\| + B_P$. Define the source-preparation oracle

$$\mathrm{PREP}^P |0\rangle = \frac{1}{\sqrt{\beta_P}} \left(\sqrt{\|u(0)\|}\, |\star\rangle + \sum_{p \in \mathcal{I}_P} \sqrt{|\omega_p|\, \|b(s_p)\|}\, |p\rangle \right), \tag{4.132}$$

with the phases of ω_p included in the corresponding left-preparation state. The vector in Equation (4.120) is then implemented by an outer LCU over $\{\star\} \cup \mathcal{I}_P$, with normalization β_P. Since each term applies a block-encoding of $W_Q^{\mathrm{MPF}}(\tau)$ with normalization α_Q, the total normalization is $\alpha_Q \beta_P$. Thus, the amplitude amplification factor is $\chi_{Q,P}(u(0), b) := \frac{\alpha_Q \beta_P}{\|u_{Q,P}^{\mathrm{MPF}}(T)\|}$. Corollary 4.6.4 gives the complexity estimates.

Corollary 4.6.4. *Under the assumptions of Corollary 4.6.3, the normalized state proportional to $u_{Q,P}^{\mathrm{MPF}}(T)$ can be prepared using $\mathcal{Q}_{\mathrm{state}} = \mathcal{O}(\chi_{Q,P}(u(0), b)\, r_Q(\delta) K_m)$ controlled second-order product-formula queries, together with $\mathcal{O}(\chi_{Q,P}(u(0), b))$ calls to the data-preparation oracles for $|\widetilde{u}(0)\rangle$, $\left|\widetilde{b}(s_p)\right\rangle$, and their inverses, as well as to PREP^P, $\mathrm{PREP}_{\mathrm{L}}^Q$, and $\mathrm{PREP}_{\mathrm{R}}^Q$. Moreover, under the optimized-order assumptions of Corollary 4.5.6, we have*

$$\mathcal{Q}_{\mathrm{state}} = \mathcal{O}\left(\chi_{Q,P}(u(0), b)\, (1 + \mu_Q F(m) \max\{1, T\}) \left(\log\left(e + \frac{T}{\epsilon_Q} \right) \right)^2 \left(\log\log\left(e^e + \frac{T}{\epsilon_Q} \right) \right)^2 \right). \tag{4.133}$$

Proof. The implementation uses the nested LCU construction from Section 4.5.2, with an additional LCU combining the initial-condition and discretized Duhamel terms. One block-encoding costs $\mathcal{O}(r_Q(\delta) K_m)$ controlled second-order product-formula queries, and amplitude amplification uses $\mathcal{O}(\chi_{Q,P}(u(0), b))$ such applications. The optimized estimates follow by substituting the optimized bound for $r_Q(\delta_m) K_m$ from Corollary 4.5.6. The factor $F(m)$ is inherited from

Equation (4.55). □

Remark 4.6.5. Let $u(T)$ denote the exact solution of Equation (4.117). If $E_{\text{time}}(P)+\mathcal{E}_Q^{[0,T]}(r,\delta)\beta_P \leq \frac{\|u(T)\|}{2}$, then $\|u_{Q,P}^{\text{MPF}}(T)\| \geq \|u(T)\|-E_{\text{time}}(P)-\mathcal{E}_Q^{[0,T]}(r,\delta)\beta_P \geq \frac{\|u(T)\|}{2}$. Consequently, $\chi_{Q,P}(u(0),b) \leq 2\alpha_Q\beta_P/\|u(T)\|$, and the state query complexity is $\mathcal{O}\left(\frac{\alpha_Q\beta_P}{\|u(T)\|}\, r_Q(\delta)K_m\right)$. If the time-quadrature rule satisfies the source-normalization estimate $B_P = \sum_{p\in\mathcal{I}_P} |\omega_p|\, \|b(s_p)\| \leq C_{\text{src}}\|b\|_{L^1[0,T]}$, with $C_{\text{src}} > 0$ independent of the target precision, then $\beta_P \leq \|u(0)\| + C_{\text{src}}\|b\|_{L^1[0,T]}$. In particular, when $C_{\text{src}} = \mathcal{O}(1)$, the normalized state-preparation cost has the usual non-homogeneous normalization factor $(\|u(0)\| + \|b\|_{L^1[0,T]})/\|u(T)\|$, up to the LCHS normalization α_Q and the MPF simulation cost. This is the standard form in which normalized-state query complexity is presented in the literature. See, for example, the discussion in Section 4.3.

We see that no new commutator quantities arise. The source $b(t)$ affects only source-state preparation, the time-quadrature error $E_{\text{time}}(P)$, and the normalization B_P. The post-quadrature quantities R_Q, α_Q, $\Lambda_{m,Q}$, $\Phi^*_{m,Q}$, r_Q, μ_Q are inherited from the homogeneous analysis.

4.6.2 Time-Dependent Case

We now extend the analysis to the time-dependent case for the special case local and extensive Hamiltonians. For simplicity, we consider the homogeneous problem.

$$\frac{d}{dt}u(t) = -A(t)u(t), \qquad A(t) = L(t) + iH(t), \tag{4.134}$$

where, for every $t \in [0,T]$, the matrices $L(t)$ and $H(t)$ are Hermitian and $L(t) \succeq 0$. The extension to the non-homogeneous case follows as in Section 4.6.1. Let $U_A(s,t) := \mathcal{T}\exp\left(-\int_s^t A(\tau)d\tau\right)$ denote the non-unitary propagator from s to t. As before, for $k \in \mathbb{R}$, define $G_k(t) := H(t)+kL(t)$

and let $U_k(s,t) := \mathcal{T}\exp\Big(-i\int_s^t G_k(\tau)d\tau\Big)$. The approximation and truncation estimates from Section 4.4.1 and Lemma 4.4.6 continue to apply after replacing e^{-AT} by $U_A(0,T)$ and $U_k(T)$ by $U_k(0,T)$. In particular, for the kernel profile $\vec{\theta} = (a,b,c,d)$, the same admissible choices of d and y_0 from Proposition 4.4.4 and Corollary 4.4.5 continue to hold. Motivated by Proposition 4.2.2, define

$$C_m := 4^{2m+1}, \quad R_m(\delta) := \|\vec{a}\|_1\|\vec{b}\|_1\delta, \quad \Phi_m(k,N,\delta) := \mu_{k,p_0}^{2m+1}, \quad \mu_{R,p_0} := \sup_{|k|\le R}\mu_{k,p_0}, \tag{4.135}$$

where μ_{k,p_0} denotes the quantity μ_{p_0} from Proposition 4.2.2 applied to the Hamiltonian $G_k(t)$, p_0 is defined as in Proposition 4.2.2, and $\|\vec{a}\|_1$ and $\|\vec{b}\|_1$ are the 1-norms of the MPF coefficient vectors discussed above. Throughout this subsection, we assume that the family $\{G_k(t) : |k| \le R\}$ admits a decomposition into q_R-local, g_R-extensive terms, and that the derivative bound in Proposition 4.2.2 holds with a constant f_R, uniformly in k and in the initial time of the sub-interval. Define

$$\Delta(R,N,\delta) := \min\left\{\frac{1}{8e^3p_0(2q_Rg_R + 2\Gamma f_R)}, \frac{1}{8\mu_{R,p_0}}\right\}. \tag{4.136}$$

We first record the pointwise short-time estimate obtained from Proposition 4.2.2.

Lemma 4.6.6. *Fix $R > 0$, $m, N \ge 1$, and $\delta \in (0,1)$. Let $0 < \Delta \le \Delta(R,N,\delta)$. For $|k| \le R$ and $s \in [0, T-\Delta]$, let $\widetilde{U}_{k,s}^{(2m)}$ denote the order-$2m$ time-dependent MPF approximation to $U_k(s, s+\Delta)$. Then*

$$\|\widetilde{U}_{k,s}^{(2m)} - U_k(s,s+\Delta)\| \le \|\vec{a}\|_1 C_m \Delta^{2m+1}\Phi_m(k,N,\delta) + R_m(\delta). \tag{4.137}$$

Proof. By assumption, the constants q_R, g_R, f_R, and μ_{R,p_0} control the estimate uniformly for

every $|k| \leq R$ and every initial time s. Now apply Proposition 4.2.2 to the shifted Hamiltonian $\tau \mapsto G_k(s+\tau)$ for $\tau \in [0, \Delta]$. □

The passage from short-time to long-time simulation is not identical to the time-independent case. This is because the exact propagator on $[0, T]$ is an ordered product of generally different interval propagators $U_k(t_j, t_{j+1})$, rather than a power of a single fixed short-time propagator. Let $t_j = j\Delta$, where $\Delta = T/r$. Then

$$U_k(0,T) = \overleftarrow{\prod_{0 \leq j \leq r-1}} U_k(t_j, t_{j+1}). \tag{4.138}$$

The factors in Equation (4.138) are generally distinct, so one cannot use the identity $U_k(0,T) = U_k(0,\Delta)^r$. We instead use the following telescoping estimate for non-stationary products.

Lemma 4.6.7. *Let $r \geq 1$, and set $t_j = j\Delta$ for $j = 0, \ldots, r$, where $\Delta = T/r$. For fixed k, define $U_{k,j} := U_k(t_j, t_{j+1})$ and $\widetilde{U}_{k,j} := \widetilde{U}^{(2m)}_{k,t_j}$. Set $U_k^{(2m)}(0,T) := \overleftarrow{\prod}_{0 \leq j \leq r-1} \widetilde{U}_{k,j}$. If $\|\widetilde{U}_{k,j} - U_{k,j}\| \leq \eta_k(\Delta)$ for some $\eta_k(\Delta) > 0$ for every $j = 0, \ldots, r-1$, then*

$$\|U_k^{(2m)}(0,T) - U_k(0,T)\| \leq r\eta_k(\Delta)\left(1 + \eta_k(\Delta)\right)^{r-1}. \tag{4.139}$$

Proof. We have

$$\overleftarrow{\prod_{0 \leq j \leq r-1}} \widetilde{U}_{k,j} - \overleftarrow{\prod_{0 \leq j \leq r-1}} U_{k,j} = \sum_{\ell=0}^{r-1} \left(\overleftarrow{\prod_{\ell+1 \leq j \leq r-1}} \widetilde{U}_{k,j} \right) (\widetilde{U}_{k,\ell} - U_{k,\ell}) \left(\overleftarrow{\prod_{0 \leq j \leq \ell-1}} U_{k,j} \right). \tag{4.140}$$

Empty products are interpreted as the identity. Since $U_{k,j}$ is unitary, $\|U_{k,j}\| = 1$. Moreover, we have $\|\widetilde{U}_{k,j}\| \leq \|U_{k,j}\| + \|\widetilde{U}_{k,j} - U_{k,j}\| \leq 1 + \eta_k(\Delta)/$ Taking norms gives $\|U_k^{(2m)}(0,T) - U_k(0,T)\| \leq$

$\sum_{\ell=0}^{r-1} \eta_k(\Delta)\,(1+\eta_k(\Delta))^{r-1-\ell}$. The right-hand side is bounded by $r\eta_k(\Delta)(1+\eta_k(\Delta))^{r-1}$. This proves Equation (4.139). □

We now pass to the post-quadrature setting. The construction remains the same as before, but we recall the relevant details for completeness. Let $Q = \{(k_i, w_i)\}_{i\in\mathcal{I}_Q}$ be a quadrature rule, and define v_i, α_Q, and R_Q as before. Similarly, define $\Lambda_{m,Q}$ as before, and set $\Phi^*_{m,Q} := \max_{i\in\mathcal{I}_Q} \Phi_m(k_i, N, \delta)$. The ideal and MPF-implemented quadrature operators, W_Q^{ideal} and W_Q^{MPF}, respectively, are defined as above. As in Corollary 4.5.1, we have

$$\|W_Q^{\text{ideal}} - W_Q^{\text{MPF}}\| \le (1+\eta_Q(T/r))^{r-1}\left(\|\vec{a}\|_1 C_m \frac{T^{2m+1}}{r^{2m}}\Lambda_{m,Q} + r\alpha_Q R_m(\delta)\right), \tag{4.141}$$

where $\eta_Q(\Delta) := \max_{i\in\mathcal{I}_Q}\left(\|\vec{a}\|_1 C_m \Delta^{2m+1}\Phi_m(k_i, N, \delta) + R_m(\delta)\right)$. Define the quadrature error $E_{\text{quad}}(Q)$ as before. Combining the approximation, quadrature, and inner simulation errors gives

$$\left\|U_A(0,T) - W_Q^{\text{MPF}}\right\| \le E_{\text{approx}}(y_0) + E_{\text{quad}}(Q) \tag{4.142}$$

$$+ (1+\eta_Q(T/r))^{r-1}\left(\|\vec{a}\|_1 C_m \frac{T^{2m+1}}{r^{2m}}\Lambda_{m,Q} + r\alpha_Q R_m(\delta)\right). \tag{4.143}$$

The implementation is the same as in Section 4.5.2, except that the controlled operation for a fixed quadrature node k_i implements the ordered product $U_{k_i}^{(2m)}(0,T) = \prod_{0\le j\le r-1}^{\leftarrow} \widetilde{U}_{k_i,j}^{(2m)}$ Thus one controlled implementation of $U_{k_i}^{(2m)}(0,T)$ still uses $\mathcal{O}(rK_m)$ controlled second-order product-formula queries, provided each time-dependent second-order product-formula segment is available with the same oracle cost model as in Proposition 4.2.2. The complexity of the algorithm follows from the preceding analysis in Section 4.5.3 and Section 4.5.5. We do not repeat those

details here.

4.7 Applications

In this section, we present applications of our algorithm and analyze its implementation complexity. Section 4.7.1 discusses the fractional diffusion equation with an imaginary potential, Section 4.7.2 discusses the advection–diffusion equation, and Section 4.7.3 discusses the no-jump dynamics regime in a dissipative Ising model.

4.7.1 Fractional Diffusion Equation

Let $0 < s < 1$. Consider the following partial differential equation:

$$\begin{cases} \partial_t u(t,x) = -(-\Delta)^s u(t,x) + (a(t,x) - iv(t,x))u(t,x) + b(t,x), & (t,x) \in (0,T] \times \mathbb{T}^d, \\ u(0,x) = u_0(x), & x \in \mathbb{T}^d. \end{cases} \tag{4.144}$$

Here $(-\Delta)^s$ is a fractional power of the positive semidefinite Laplacian on $\mathbb{T}^d$, $a, v : [0,T] \times \mathbb{T}^d \to \mathbb{R}$ are real-valued potentials, and $b : [0,T] \times \mathbb{T}^d \to \mathbb{C}$ is a source term. If $v \equiv 0$, then Equation (4.144) reduces to a linear fractional reaction–diffusion equation. The operator $(-\Delta)^s$ models non-local diffusion, $a(t,x)$ gives a linear reaction term, and the imaginary potential $-iv(t,x)$ produces local phase modulation, analogous to a Larmor-frequency-offset term in Bloch–Torrey-type models. Thus, Equation (4.144) is a forced fractional Bloch–Torrey-type equation with an additional real potential. Fractional Bloch–Torrey models have been used for anomalous diffusion in complex media, especially in magnetic resonance applications [Tor56;

Mag+08; Yu+13; BB17; MMG20]. In what follows, we take $U, b \equiv 0$ and assume that v is time-independent.

Remark 4.7.1. See also [ALL23, Equation 10], which considers complex absorbing potentials in open-system dynamics through a similar Schrödinger-type equation with a complex-valued potential.

Discretize Equation (4.144) on an M^d periodic grid with mesh size $h = 1/M$ and $N = M^d$. Let F_M be the one-dimensional discrete Fourier transform, $F_N := F_M^{\otimes d}$, and let $-\Delta_{h,1}$ be the one-dimensional periodic discrete Laplacian, with eigenvalues $4M^2 \sin^2(\pi m/M)$, $m \in \mathbb{Z}_M$. The d-dimensional periodic discrete Laplacian is given by the Kronecker sum

$$-\Delta_{h,d} = \sum_{j=1}^{d} I^{\otimes(j-1)} \otimes (-\Delta_{h,1}) \otimes I^{\otimes(d-j)}, \quad -\Delta_{h,1} = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & & -1 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & \ddots & & \\ & & \ddots & \ddots & -1 & \\ & & & -1 & 2 & -1 \\ -1 & & & & -1 & 2 \end{pmatrix}. \tag{4.145}$$

Consequently, $-\Delta_{h,d}$ is diagonalized by $F_N = F_M^{\otimes d}$, with eigenvalues $\lambda_m = 4M^2 \sum_{\alpha=1}^{d} \sin^2\left(\frac{\pi m_\alpha}{M}\right)$ for $m = (m_1, \ldots, m_d) \in \mathbb{Z}_M^d$. Therefore, the discretized fractional Laplacian is given by

$$L_s := (-\Delta_{h,d})^s = F_N^\dagger \operatorname{diag}\left[\left(4M^2 \sum_{\alpha=1}^{d} \sin^2\left(\frac{\pi m_\alpha}{M}\right)\right)^s\right]_{m \in \{0,\ldots,M-1\}^d} F_N. \tag{4.146}$$

Discretize the potential as $H := \operatorname{diag}(v(x_i/M))_{i=1}^N$, with $x_i \in \mathbb{Z}_M^d$. If $\vec{u}(t)$ approximates $u(t, x)$

on the grid and $\vec{u}_0$ is the initial condition, then the semi-discrete equation is

$$\begin{cases} \dfrac{d\vec{u}}{dt} = -(L_s + iH)\vec{u}(t), \\ \vec{u}(0) = \vec{u}_0. \end{cases} \tag{4.147}$$

The estimate in Proposition 4.2.1 does not apply to Equation (4.147), since L_s is non-local for $0 < s < 1$. On $\mathbb{R}^d$, the fractional Laplacian has the singular integral representation

$$(-\Delta)^s u(x) = \frac{2^{2s}\Gamma(s + \frac{d}{2})}{\pi^{d/2}|\Gamma(-s)|}\, \text{p. v.} \int_{\mathbb{R}^d} \frac{u(x) - u(y)}{|x - y|^{d+2s}}\, dy \qquad 0 < s < 1. \tag{4.148}$$

On $\mathbb{T}^d$, the kernel is obtained by periodization. Thus, $(-\Delta)^s u(x)$ depends on all spatial values $u(y)$, with algebraically decaying interactions. This non-locality is inherited by L_s, so the standard qubit encoding of L_s is non-local. On the other hand, Theorem 3.3.4 does apply. Using $\|[A, B]\| \leq 2\|A\|\|B\|$, we obtain $C_{j,\ell}(H, L_s) \leq \binom{j}{\ell} 2^{j-1}\|H\|^{j-\ell}\|L_s\|^{\ell}$. With $C_j(H, L_s) := \max_{0\leq\ell\leq j} C_{j,\ell}(H, L_s)$, we have $C_j(H, L_s) \leq 2^{j-1}\big(\|H\| + \|L_s\|\big)^j$. It follows that

$$\chi_J(H, L_s) := \inf_{j\geq J} C_j(H, L_s)^{-1/j} \geq \frac{1}{2(\|H\| + \|L_s\|)} > 0. \tag{4.149}$$

Since $L_s \succeq 0$ and $H = H^\dagger$, our algorithm applies. We compute the commutator scaling used in the complexity analysis. Since L_s is diagonalized by F_N, we have $(F_N L_s F_N^\dagger)_{p,q} = \lambda_p^s \delta_{p,q}$, where λ_p is an eigenvalue of $-\Delta_{h,d}$. The Fourier-basis matrix of H is

$$(F_N H F_N^\dagger)_{p,q} = \frac{1}{N} \sum_{x\in\mathbb{Z}_M^d} v(x/M) e^{-2\pi i(p-q)\cdot x/M} =: \widehat{V}_{p-q}. \tag{4.150}$$

Lemma 4.7.2, which is proved in Section B.9, estimates differences of fractional-Laplacian eigenvalues. Define $\beta_s := \min\{2s, 1\}$, $\theta_s := 2s - \beta_s = \max\{0, 2s-1\}$, and let $|\cdot|_{\mathrm{per}}$ denote distance on $\mathbb{Z}_M^d$.

Lemma 4.7.2. *Let $0 < s < 1$. There exists a constant $C_{s,d} > 0$, independent of N, such that $|\lambda_p^s - \lambda_q^s| \le C_{s,d} N^{\theta_s/d} |p-q|_{\mathrm{per}}^{\beta_s}$ for all $p, q \in \mathbb{Z}_M^d$.*

Using Lemma 4.7.2, Lemma 4.7.3 estimates the scaling of $\mathrm{ad}_{L_s}^j(H)$ for $j \ge 1$. This serves as a prototype for bounding general commutator words in Proposition 4.7.4.

Lemma 4.7.3. *For every $\ell \ge 1$, we have $(F_N \,\mathrm{ad}_{L_s}^\ell(H) F_N^\dagger)_{p,q} = (\lambda_p^s - \lambda_q^s)^\ell \widehat{V}_{p-q}$. Consequently, we have*

$$\|\,\mathrm{ad}_{L_s}^\ell(H)\| \le C_{\ell,V,s,d}^{(N)} N^{\ell\theta_s/d}, \tag{4.151}$$

where $C_{\ell,V,s,d}^{(N)} := C_{s,d}^\ell \sum_{r \in \mathbb{Z}_M^d} |r|_{\mathrm{per}}^{\ell\beta_s} |\widehat{V}_r|$. In particular, if $\sup_{M \ge 1} \sum_{r \in \mathbb{Z}_M^d} |r|_{\mathrm{per}}^{\ell\beta_s} |\widehat{V}_r| < \infty$ then $C_{\ell,V,s,d}^{(N)}$ is uniformly bounded in N. In this case we may write $\|\,\mathrm{ad}_{L_s}^\ell(H)\| \le C_{\ell,V,s,d} N^{\ell\theta_s/d}$, where $C_{\ell,V,s,d}$ independent of N.

Proof. Since $F_N L_s F_N^\dagger$ is diagonal, we have

$$(F_N [L_s, A] F_N^\dagger)_{p,q} = ([F_N L_s F_N^\dagger, F_N A F_N^\dagger])_{p,q} = (\lambda_p^s - \lambda_q^s)(F_N A F_N^\dagger)_{p,q}. \tag{4.152}$$

for any matrix A. With $A = H$, Equation (4.150) gives $(F_N [L_s, H] F_N^\dagger)_{p,q} = (\lambda_p^s - \lambda_q^s)\widehat{V}_{p-q}$. Iterating Equation (4.152) gives the matrix elements of $F_N \,\mathrm{ad}_{L_s}^\ell(H) F_N^\dagger$. By Lemma 4.7.2, we have

$$|(F_N \,\mathrm{ad}_{L_s}^\ell(H) F_N^\dagger)_{p,q}| \le C_{s,d}^\ell N^{\ell\theta_s/d} |p-q|_{\mathrm{per}}^{\ell\beta_s} |\widehat{V}_{p-q}|. \tag{4.153}$$

Using Equation (4.153), we have

$$\sum_{q\in\mathbb{Z}_M^d} |(F_N \operatorname{ad}_{L_s}^\ell(H) F_N^\dagger)_{p,q}| \le C_{s,d}^\ell N^{\ell\theta_s/d} \sum_{q\in\mathbb{Z}_M^d} |p-q|_{\mathrm{per}}^{\ell\beta_s} |\widehat{V}_{p-q}| = C_{s,d}^\ell N^{\ell\theta_s/d} \sum_{r\in\mathbb{Z}_M^d} |r|_{\mathrm{per}}^{\ell\beta_s} |\widehat{V}_r|. \quad (4.154)$$

Hence, we have

$$\|F_N \operatorname{ad}_{L_s}^\ell(H) F_N^\dagger\|_\infty \le C_{s,d}^\ell N^{\ell\theta_s/d} \sum_{r\in\mathbb{Z}_M^d} |r|_{\mathrm{per}}^{\ell\beta_s} |\widehat{V}_r|. \quad (4.155)$$

Similarly, we have $\|F_N \operatorname{ad}_{L_s}^\ell(H) F_N^\dagger\|_1 \le C_{s,d}^\ell N^{\ell\theta_s/d} \sum_{r\in\mathbb{Z}_M^d} |r|_{\mathrm{per}}^{\ell\beta_s} |\widehat{V}_r|$. Using the estimate $\|A\| \le \sqrt{\|A\|_1 \|A\|_\infty}$ which valid for any matrix A, we obtain

$$\|\operatorname{ad}_{L_s}^\ell(H)\| = \|F_N \operatorname{ad}_{L_s}^\ell(H) F_N^\dagger\| \le C_{s,d}^\ell N^{\ell\theta_s/d} \sum_{r\in\mathbb{Z}_M^d} |r|_{\mathrm{per}}^{\ell\beta_s} |\widehat{V}_r|. \quad (4.156)$$

This proves Equation (4.151). □

We now extend Lemma 4.7.3 to arbitrary nested commutator words in Proposition 4.7.4, which is proved in Section B.10. In Proposition 4.7.4, we use the weaker consequence of Lemma 4.7.2

$$|\lambda_p^s - \lambda_q^s| \le C_{s,d} N^{\theta_s/d} \big(1 + |p-q|_{\mathrm{per}}\big)^{\beta_s}. \quad (4.157)$$

We remark that the constant in Proposition 4.7.4 is not sharp, since the proof uses the crude bound $\|\operatorname{ad}_H(A)\|_{a,\mathcal{F}} \le 2\mathcal{V}_a^{(N)} \|A\|_{a,\mathcal{F}}$. For special words, direct Fourier-basis formulas give sharper constants. For example, Lemma 4.7.3 uses $(F_N \operatorname{ad}_{L_s}^\ell(H) F_N^\dagger)_{p,q} = (\lambda_p^s - \lambda_q^s)^\ell \widehat{V}_{p-q}$ together with fractional-Laplacian eigenvalue differences. More general nested commutators finite differences of $p \mapsto \lambda_p^s$, which may yield sharper constants. We do not pursue these refinements here.

Proposition 4.7.4. *Let $W_{j,\ell}(H, L_s)$ be any nested commutator word of length $j \geq 2$ containing exactly ℓ occurrences of L_s and $j-\ell$ occurrences of H, with $1 \leq \ell \leq j-1$. Define $\mathcal{V}_a^{(N)} := \sum_{r \in \mathbb{Z}_M^d} (1+|r|_{\mathrm{per}})^a \, |\widehat{V}_r|$. Then $\|W_{j,\ell}(H, L_s)\| \leq 2^{j-\ell} C_{s,d}^{\ell} (\mathcal{V}_{j\beta_s}^{(N)})^{j-\ell} N^{\ell\theta_s/d}$. In particular, if $\mathcal{V}_{j\beta_s}^{(N)}$ is bounded independently of N, then $\|W_{j,\ell}(H, L_s)\| \leq C_{j,V,s,d} N^{\ell\theta_s/d}$, where $C_{j,V,s,d} > 0$ is independent of N.*

We now analyze the implications of Proposition 4.7.4 for the complexity of our algorithm. Since $\mathcal{C}_{j,\ell}(H, L_s)$ collects all commutator words with exactly ℓ occurrences of L_s, Proposition 4.7.4 implies

$$\mathcal{C}_{j,\ell}(H, L_s) \leq 2^j 2^{j-\ell} C_{s,d}^{\ell} (\mathcal{V}_{j\beta_s}^{(N)})^{j-\ell} N^{\ell\theta_s/d} \tag{4.158}$$

for $1 \leq \ell \leq j-1$. Define $A_{V,j}^{(N)} := \max_{1 \leq \ell \leq j-1} \left(2^j 2^{j-\ell} C_{s,d}^{\ell} (\mathcal{V}_{j\beta_s}^{(N)})^{j-\ell} \right)^{1/j}$. Equation (4.158) gives $\mathcal{C}_{j,\ell}(H, L_s) \leq (A_{V,j}^{(N)})^j N^{\ell\theta_s/d}$. Using the definition of $\alpha_{Q,j}$ from Lemma 4.5.8, we obtain

$$\alpha_{Q,j} \leq \sum_{\ell=0}^{j} R_Q^{\ell} \mathcal{C}_{j,\ell}(H, L_s) \leq (A_{V,j}^{(N)})^j \sum_{\ell=1}^{j-1} (R_Q N^{\theta_s/d})^{\ell} \leq \left(2 A_{V,j}^{(N)} (1 + R_Q N^{\theta_s/d}) \right)^j, \tag{4.159}$$

where the last inequality follows from $\sum_{\ell=1}^{j-1} x^{\ell} \leq (2(1+x))^j$ for $x \geq 0$. Hence, if $A_V^{(N)} := \sup_{j \geq 2} A_{V,j}^{(N)} < \infty$, we have $\alpha_{Q,j} \leq (2A_V^{(N)}(1 + R_Q N^{\theta_s/d}))^j$ Therefore, the root-growth scale ω from Lemma 4.5.8 satisfies

$$\omega = \mathcal{O}(A_V^{(N)}(1 + R_Q N^{\theta_s/d})). \tag{4.160}$$

Choose $\rho_Q = c/\omega$ for some fixed $0 < c < 1$. Then $\rho_Q \omega = c < 1$, so Lemma 4.5.8 applies, and $\rho_Q^{-1} = \mathcal{O}(\omega)$. Combining Equation (4.160) with Equation (4.73) gives

$$\mu_Q = \mathcal{O}(A_V^{(N)}(1 + R_Q N^{\theta_s/d})), \tag{4.161}$$

where the $\mathcal{O}$-notation absorbs the fixed finite factor S from Lemma 4.5.8. More explicitly, the optimized MPF overhead contributes the polylogarithmic factor

$$\mathrm{polylog}_Q(T,\epsilon_Q,N) := \mathcal{O}\left(\log\left(\frac{A_V^{(N)}(1+R_Q N^{\theta_s/d})T}{\epsilon_Q}\right)^2\left(\log\log\left(\frac{A_V^{(N)}(1+R_Q N^{\theta_s/d})T}{\epsilon_Q}\right)\right)^2\right), \tag{4.162}$$

$$= \mathcal{O}\left(\log\left(\frac{A_V^{(N)}(1+R_Q N^{\theta_s/d})T}{\epsilon_Q}\right)^4\right), \tag{4.163}$$

where we have assumed that $A_V^{(N)}(1+R_Q N^{\theta_s/d})T/\epsilon_Q \geq e^e$. Substituting Equation (4.161) into Equation (4.58) with $F(m)\equiv 1$, we obtain $\mathcal{Q}_{\mathrm{block}} = \mathcal{O}(A_V^{(N)}(1+R_Q N^{\theta_s/d})T\,\mathrm{polylog}_Q(T,\epsilon_Q,N))$. Similarly, provided $W_Q^{\mathrm{MPF}}u_0 \neq 0$, substituting Equation (4.161) into Equation (4.59), we obtain

$$\mathcal{Q}_{\mathrm{state}}(u_0) = \mathcal{O}(\chi_Q(u_0)A_V^{(N)}(1+R_Q N^{\theta_s/d})T\,\mathrm{polylog}_Q(T,\epsilon_Q,N)). \tag{4.164}$$

4.7.2 Advection-Diffusion Equation

Let $a>0$ and $b\in\mathbb{R}$. Consider the following partial differential equation:

$$\begin{cases} \partial_t u(t,x) = a\Delta u(t,x) + b\nabla\cdot(\vec{1}u(t,x)) - c(t,x)u(t,x) + f(t,x), & (t,x)\in(0,T]\times(0,1)^d, \\ u(0,x) = u_0(x), & x\in[0,1]^d, \\ u(t,x) = 0, & (t,x)\in[0,T]\times\partial[0,1]^d. \end{cases} \tag{4.165}$$

Equation (4.165) is a d-dimensional advection–diffusion equation with homogeneous Dirichlet boundary conditions, where $a\Delta u$ is diffusion, $b\nabla\cdot(\vec{1}u)$ is drift, $c(t,x)\geq 0$ is absorption, and

$f(t,x)$ is the source. Such equations model heat and mass transfer, reactive systems, drug delivery, and related transport phenomena [Par+22; Agu+23; Vil+25; AOY26]. We take $c, f \equiv 0$ for simplicity. Discretize using $M+1$ equidistant grid points per direction, mesh size $h = 1/M$, and $N = (M+1)^d$. The d-dimensional Dirichlet finite-difference Laplacian is the Kronecker sum

$$-\Delta_{h,d} := \sum_{j=1}^{d} I^{\otimes(j-1)} \otimes (-\Delta_{h,1}) \otimes I^{\otimes(d-j)}, \quad -\Delta_{h,1} := \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}. \quad (4.166)$$

By abuse of notation, we use the same notation for the Dirichlet finite-difference Laplacian and the periodic finite-difference Laplacian from Section 4.7.1. Similarly, the d-dimensional Dirichlet finite-difference drift matrix is the Kronecker sum

$$D_{h,d} := \sum_{j=1}^{d} I^{\otimes(j-1)} \otimes D_{h,1} \otimes I^{\otimes(d-j)}, \quad D_{h,1} = \frac{1}{2h} \begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & -1 & 0 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & -1 & 0 \end{pmatrix}. \quad (4.167)$$

Here $\Delta_{h,d}$ discretizes $\Delta = \sum_{i=1}^{d} \partial_{x_i}^2$, while $D_{h,d}$ discretizes $\nabla \cdot (\vec{1}u) = \sum_{i=1}^{d} \partial_{x_i} u$. Note that Since $\Delta_{h,d} = \Delta_{h,d}^\dagger \preceq 0$ $D_{h,d}^\dagger = -D_{h,d}$. Define $L := -a\Delta_{h,d}$ and $H := ibD_{h,d}$. If $\vec{u}(t)$ approximates $u(t,x)$ on the grid and $\vec{u}_0$ is the initial condition, then the semi-discrete equation is in dissipative-

plus-Hamiltonian form is

$$\begin{cases} \dfrac{d\vec{u}}{dt} = -(L + iH)\,\vec{u}(t), \\ \vec{u}(0) = \vec{u}_0. \end{cases} \tag{4.168}$$

Remark 4.7.5. The locality-extensive estimate in Proposition 4.2.1 does not apply directly. Although $\Delta_{h,d}$ and $D_{h,d}$ are sparse in the grid basis, this does not imply q-locality under a many-body tensor-product encoding. For example, in binary encoding, $|m+1\rangle\langle m|$ may involve non-local carries. Nevertheless, Theorem 3.3.4 applies as in Section 4.7.1.

We now derive commutator bounds. For $d = 1$, define $A_1 := -h^2\Delta_{h,1}$ and $B_1 := 2hD_{h,1}$.

Lemma 4.7.6. *Let $\{e_0, \ldots, e_M\}$ denote the standard basis of $\mathbb{C}^{M+1}$. Define the matrices $S := \sum_{j=0}^{M-1} e_j e_{j+1}^T$, $P_0 := e_0 e_0^T$ and $P_M := e_M e_M^T$. We have $A_1 := 2I - S - S^T$ and $B_1 := S - S^T$ and $[S, S^T] = P_0 - P_M$. Consequently, we have*

$$[A_1, B_1] = 2(P_0 - P_M), \quad [-\Delta_{h,1}, D_{h,1}] = \frac{1}{h^3}(P_0 - P_M). \tag{4.169}$$

Proof. The formulas for A_1 and A_2 are immediate from their definitions. By definition, we have $S = \sum_{j=0}^{M-1} e_j e_{j+1}^T$ and $S^T = \sum_{j=0}^{M-1} e_{j+1} e_j^T$. Therefore, we have $SS^T = \sum_{j=0}^{M-1} e_j e_j^T = I - P_M$ and $S^T S = \sum_{j=0}^{M-1} e_{j+1} e_{j+1}^T = I - P_0$. Hence, we have $[S, S^T] = SS^T - S^T S = (I - P_M) - (I - P_0) = P_0 - P_M$. Therefore, we have

$$[A_1, B_1] = [2I - S - S^T, S - S^T] = [-S - S^T, S - S^T] = 2[S, S^T] = 2(P_0 - P_M). \tag{4.170}$$

Hence, we have

$$[-\Delta_{h,1}, D_{h,1}] = \left[\frac{A_1}{h^2}, \frac{B_1}{2h}\right] = \frac{1}{2h^3}[A_1, B_1] = \frac{1}{h^3}(P_0 - P_M). \tag{4.171}$$

This completes the proof. □

Lemma 4.7.6 shows that the non-commutativity of A_1 and B_1 is localized at the boundary. Consequently, repeated commutators remain boundary-supported. This is made precise in Lemma 4.7.7, which is proved in Section B.11.

Lemma 4.7.7. *Let S, P_0, and P_M be defined as in Lemma 4.7.6. For $r \geq 0$, define $\mathcal{B}_r := \operatorname{span}\left\{(S^T)^p P_0 S^q,\ S^p P_M (S^T)^q :\ 0 \leq p, q \leq r\right\}$. For $X \in \mathcal{B}_r$, define the coefficient seminorm*

$$\|X\|_{\mathcal{B}} := \inf\left\{\sum_\eta |c_\eta| :\ X = \sum_\eta c_\eta E_\eta,\ E_\eta \in \left\{(S^T)^p P_0 S^q,\ S^p P_M (S^T)^q :\ 0 \leq p, q \leq r \leq M\right\}\right\}. \tag{4.172}$$

Then $\|X\| \leq \|X\|_{\mathcal{B}}$. Moreover, for every $X \in \mathcal{B}_r$, we have $[A_1, X] \in \mathcal{B}_{r+1}$, $[B_1, X] \in \mathcal{B}_{r+1}$, $\|[A_1, X]\|_{\mathcal{B}} \leq 4\|X\|_{\mathcal{B}}$ and $\|[B_1, X]\|_{\mathcal{B}} \leq 4\|X\|_{\mathcal{B}}$. Consequently, every non-zero right-nested mixed commutator $W_j(A_1, B_1)$ of length $j \geq 2$ satisfies $\|W_j(A_1, B_1)\| \leq 4^{j-1}$.

We lift the $d = 1$ estimate to the d-dimensional tensor-product discretization in Proposition 4.7.8.

Proposition 4.7.8. *Let $L := -a\Delta_{h,d}$ and $H := ibD_{h,d}$. For $1 \leq i \leq d$, define $A_i := I^{\otimes(i-1)} \otimes A_1 \otimes I^{\otimes(d-i)}$ and $B_i := I^{\otimes(i-1)} \otimes B_1 \otimes I^{\otimes(d-i)}$. Then $-\Delta_{h,d} = h^{-2}\sum_{i=1}^d A_i$ and $D_{h,d} = \frac{1}{2h}\sum_{i=1}^d B_i$. Let $W_{j,\ell}(H, L)$ be a non-zero right-nested mixed commutator of length $j \geq 2$, containing exactly*

ℓ occurrences of L and $j-\ell$ occurrences of H, where $1 \le \ell \le j-1$. Then

$$\|W_{j,\ell}(H,L)\| \le d2^{j+\ell-2}a^{\ell}|b|^{j-\ell}h^{-(j+\ell)}. \tag{4.173}$$

Consequently, if $\mathcal{C}_{j,\ell}(H,L)$ denotes the sum over all right-nested mixed commutator words of length j with exactly ℓ occurrences of L, then

$$\mathcal{C}_{j,\ell}(H,L) \le \binom{j}{\ell} d2^{j+\ell-2}a^{\ell}|b|^{j-\ell}h^{-(j+\ell)} \le d2^{2j+\ell-2}a^{\ell}|b|^{j-\ell}h^{-(j+\ell)}. \tag{4.174}$$

Proof. The formulas for $-\Delta_{h,d}$ and $D_{h,d}$ follow directly from their Kronecker-sum definitions. We expand $W_{j,\ell}(H,L)$ multilinearly in A_i and B_i. Clearly, $[A_i,A_k]=[B_i,B_k]=[A_i,B_k]=0$ for $i \ne k$. For each fixed i, Lemma 4.7.7 gives $\|W_j(A_i,B_i)\| \le 4^{j-1}$. Each occurrence of L contributes a factor ah^{-2}, while each occurrence of H contributes a factor $|b|/(2h)$. Summing over the possible coordinate directions contributes only a factor of d. Therefore, we have $\|W_{j,\ell}(H,L)\| \le d4^{j-1}a^{\ell}h^{-2\ell}\left(\frac{|b|}{2h}\right)^{j-\ell} = d2^{j+\ell-2}a^{\ell}|b|^{j-\ell}h^{-(j+\ell)}$. There are at most $\binom{j}{\ell}$ right-nested mixed words of length j with exactly ℓ occurrences of L. Summing Equation (4.173) over these words gives Equation (4.174). □

Since $N = (M+1)^d$ and $h^{-1} = M \le N^{1/d}$ Proposition 4.7.8 implies $\mathcal{C}_{j,\ell}(H,L) \le d2^{2j+\ell-2}a^{\ell}|b|^{j-\ell}N^{(j+\ell)/d}$. We now translate Proposition 4.7.8 into a complexity bound. Let $\alpha_{Q,j} \le \sum_{\ell=1}^{j-1} R_Q^{\ell}\mathcal{C}_{j,\ell}(H,L)$ be as in Lemma 4.5.8. Using Equation (4.174), for every $j \ge 2$ we have

$$\alpha_{Q,j} \leq d2^{2j-2} \sum_{\ell=1}^{j-1} (2aR_Q h^{-2})^{\ell} (|b|h^{-1})^{j-\ell} \leq d2^{2j-2} (|b|h^{-1} + 2aR_Q h^{-2})^{j}. \tag{4.175}$$

Since $h^{-1} = M \leq N^{1/d}$, the root-growth scale ω from Lemma 4.5.8 satisfies

$$\omega = \mathcal{O}(|b|N^{1/d} + aR_Q N^{2/d}). \tag{4.176}$$

Choose $\rho_Q = c/\omega$ for fixed $0 < c < 1$. Then $\rho_Q \omega = c < 1$, so Lemma 4.5.8 applies and $\rho_Q^{-1} = \mathcal{O}(\omega)$. Combining Equation (4.176) with Equation (4.73) gives

$$\mu_Q = \mathcal{O}(|b|N^{1/d} + aR_Q N^{2/d}), \tag{4.177}$$

where the $\mathcal{O}$-notation absorbs the fixed finite factor S from Lemma 4.5.8. For the optimized-order estimate, the MPF overhead contributes the polylogarithmic factor

$$\begin{aligned} \mathrm{polylog}_Q(T, \epsilon_Q, N) &:= \mathcal{O}\left(\log\left(\frac{\left(|b|N^{1/d} + aR_Q N^{2/d}\right) T}{\epsilon_Q} \right)^2 \left(\log\log\left(\frac{\left(|b|N^{1/d} + aR_Q N^{2/d}\right) T}{\epsilon_Q} \right) \right)^2 \right) \\ &= \mathcal{O}\left(\log\left(\frac{\left(|b|N^{1/d} + aR_Q N^{2/d}\right) T}{\epsilon_Q} \right)^4 \right), \end{aligned} \tag{4.178}$$

where in the second line we assume $(|b|N^{1/d} + aR_Q N^{2/d})T/\epsilon_Q \geq e^e$. Substituting Equation (4.177) into Equation (4.58), we obtain $\mathcal{Q}_{\text{block}} = \mathcal{O}((|b|N^{1/d} + aR_Q N^{2/d})T \,\mathrm{polylog}_Q(T, \epsilon_Q, N))$.

Similarly, provided $W_Q^{\mathrm{MPF}}\vec{u}_0 \neq 0$, substituting Equation (4.177) into Equation (4.59) gives

$$\mathcal{Q}_{\mathrm{state}}(\vec{u}_0) = \mathcal{O}(\chi_Q(\vec{u}_0)(|b|N^{1/d} + aR_QN^{2/d})T\,\mathrm{polylog}_Q(T, \epsilon_Q, N)). \tag{4.179}$$

4.7.3 No-Jump Dynamics for a Dissipative Ising Model

We next consider the no-jump regime of a dissipative transverse-field Ising model with local spontaneous emission, formulated as a Lindblad dynamics. The no-jump regime is obtained from the standard quantum-trajectory formalism [DCM92; PK98; LC13; RC23]. Let $\mathcal{G} = (V, E(V))$ be a finite graph. The closed-system transverse-field Ising Hamiltonian is

$$H = \sum_{j\in V} h_j X_j + \sum_{\langle j,\ell\rangle\in E(V)} J_{j\ell} Z_j Z_\ell, \tag{4.180}$$

where $h_j \in \mathbb{R}$ are transverse-field strengths and $J_{j\ell} \in \mathbb{R}$ are Ising couplings. Local spontaneous emission is modeled by Lindblad jump operators $L_j = \sqrt{2\gamma_j}\,\sigma_j^-$, where $\gamma_j \geq 0$ and $\sigma_j^- = |0\rangle\langle 1|$. Thus L_j represents local decay at site j with rate γ_j. The associated Lindblad equation for ρ is

$$\frac{d\rho}{dt} = -i[H, \rho] + \sum_{j\in V}(L_j\rho L_j^\dagger - \frac{1}{2}\{L_j^\dagger L_j, \rho\}). \tag{4.181}$$

Equation (4.181) gives the unconditional evolution averaged over all emission histories. In the quantum-trajectory picture, $L_j\rho L_j^\dagger$ represents a detected jump at site j. Conditioning on records with no detected emissions gives the no-jump evolution, generated by the effective

Hamiltonian

$$H_{\text{eff}} = H - \frac{i}{2}\sum_{j\in V} L_j^\dagger L_j = \sum_{j\in V} h_j X_j + \sum_{\langle j,\ell\rangle\in E(V)} J_{j\ell} Z_j Z_\ell - i\sum_{j\in V}\gamma_j n_j, \tag{4.182}$$

where $n_j = \sigma_j^+\sigma_j^- = |1\rangle\langle 1| = (I - Z_j)/2$. The final term reflects the no-jump conditioning, where configurations more likely to emit are suppressed when no photon is detected. The corresponding dynamics are given by

$$i\frac{d}{dt}|\psi(t)\rangle = H_{\text{eff}}|\psi(t)\rangle\,. \tag{4.183}$$

Let $L = \sum_{j\in V}\gamma_j n_j = \sum_{j\in V}\gamma_j\frac{I-Z_j}{2}$. Then $L \succeq 0$ and $H_{\text{eff}} = H - iL$. Multiplying Equation (4.183) by $-i$, we obtain

$$\frac{d}{dt}|\psi(t)\rangle = -i(H - iL)|\psi(t)\rangle = -(L + iH)|\psi(t)\rangle\,. \tag{4.184}$$

In what follows, take $h_j \equiv h$, $J_{j\ell} \equiv J$, and $\gamma_j \equiv \gamma$, and assume $\mathcal{G}$ has maximum degree D, independent of $|V|$. We first verify locality and extensivity. The Hamiltonian H is 2-local:. Indeed, hX_j is supported on on one site and JZ_jZ_ℓ is supported on two sites. For each site j, at most one onsite term and D interaction terms contain j. Since $\|X_j\| = \|Z_jZ_\ell\| = 1$, we have $\sum_{\alpha:\, j\in\text{supp}(H_\alpha)}\|H_\alpha\| \le |h| + D|J|$. Hence, H is g_H-extensive with $g_H := |h| + D|J|$. Similarly, L is 1-local because each term γn_j is supported only on site j. Hence, $q_L = 1$. Since n_j is a projection, $\|n_j\| = 1$, and we have $\sum_{\alpha:\, j\in\text{supp}(L_\alpha)}\|L_\alpha\| = \|\gamma n_j\| = \gamma$. Thus, L is g_L-extensive with $g_L := \gamma$. Hence, Section 4.5.5 implies that $\{G_{k_i}\}_{i\in\mathcal{I}_Q}$ is q_Q-local and g_Q-extensive, with $q_Q = \max\{q_H, q_L\} = 2$ and $g_Q \le |h| + D|J| + R_Q\gamma$.

We now derive commutator bounds. Our approach uses the Pauli incompatibility graph,

whose vertices are local Pauli summands and whose edges connect anti-commuting Pauli strings. Such commutativity-sensitive counting is in graph-based ordering methods in Trotter-error analyses [Tra+19]. Since $n_j = (I - Z_j)/2$, the identity part commutes with all terms and does not affect nested commutators. Thus, for commutator estimates, $G_k = H + kL$ may be replaced by $\overline{G}_k = h\sum_{j\in V} X_j + J\sum_{\langle j,\ell\rangle\in E(V)} Z_jZ_\ell - \frac{k\gamma}{2}\sum_{j\in V} Z_j$. This replacement leaves every nested commutator of length at least two unchanged. In what follows, we will use the following standard computation for commutators of Pauli strings P and Q

$$\|[xP, yQ]\| = \begin{cases} 0, & PQ = QP, \\ 2|xy|, & PQ = -QP, \end{cases} \tag{4.185}$$

for $x, y \in \mathbb{R}$.

Proposition 4.7.9. *Assume that $\mathcal{G}$ has maximum degree D and that $\gamma \geq 0$. For every $p \geq 2$ and every quadrature node $|k_i| \leq R_Q$, we have*

$$\alpha_{\mathrm{comm},p}(G_{k_i}) \leq |V|\, 4^{p-1}(p-1)!\, 2|h| \left(D|J| + \frac{R_Q\gamma}{2}\right)\left(|h| + D|J| + \frac{R_Q\gamma}{2}\right)^{p-2} \tag{4.186}$$

$$\leq |V|\, 4^{p-1}(p-1)! \left(|h| + D|J| + \frac{R_Q\gamma}{2}\right)^{p}. \tag{4.187}$$

Proof. We work with $\overline{G}_{k_i}$ because $\alpha_{\mathrm{comm},p}(G_{k_i}) = \alpha_{\mathrm{comm},p}(\overline{G}_{k_i})$. Its local Pauli summands are hX_j, JZ_jZ_ℓ, and $-k_i\gamma Z_j/2$. We call hX_j transverse, or X-type, and the remaining summands diagonal, or Z-type. The proof uses graph counting. Each non-zero nested commutator grows a cluster in the Pauli incompatibility graph, and each new summand must overlap the current support. We first record the local counting estimates. If $S \subseteq V$, the total transverse weight

supported on S is $\sum_{j\in S}\|hX_j\| = |S|\,|h|$. Similarly, the total transverse weight supported on S is

$$\sum_{\substack{\langle a,b\rangle\in E(V)\\ \{a,b\}\cap S\neq\emptyset}} \|JZ_aZ_b\| + \sum_{a\in S}\left\|\frac{k_i\gamma}{2}Z_a\right\| \le |S|\left(D|J| + \frac{R_Q\gamma}{2}\right). \tag{4.188}$$

Each vertex belongs to at most D edges, $|k_i| \le R_Q$, and $\gamma \ge 0$. Fix a layer pattern $\tau = (\tau_1,\dots,\tau_p) \in \{X,Z\}^p$, with τ_1 innermost. Since the innermost commutator is nonzero only for opposite types, $\tau_1 \neq \tau_2$. If τ has ν transverse entries, the first summand has total weight at most $|V||h|$ for X-type and $|V|(D|J| + R_Q\gamma/2)$ for Z-type. After a insertions, the cluster support has size at most $2a$, so the next summand has total weight at most $2a|h|$ for X-type and $2a(D|J| + R_Q\gamma/2)$ for Z-type. Each nonzero commutator contributes a factor 2. Thus, the total contribution is bounded by $|V|\,4^{p-1}(p-1)!|h|^\nu\left(D|J| + \frac{R_Q\gamma}{2}\right)^{p-\nu}$. If the pattern has ν transverse entries, then $\tau_1 \neq \tau_2$ forces the first two entries to contain one X and one Z, giving 2 possible orders. The remaining $p-2$ entries contain $\nu-1$ transverse entries, giving $2\binom{p-2}{\nu-1}$ admissible patterns. Summing over $1 \le \nu \le p-1$ gives

$$\alpha_{\mathrm{comm},p}(G_{k_i}) \le |V|\,4^{p-1}(p-1)!\sum_{\nu=1}^{p-1} 2\binom{p-2}{\nu-1}|h|^\nu\left(D|J| + \frac{R_Q\gamma}{2}\right)^{p-\nu} \tag{4.189}$$

$$= |V|\,4^{p-1}(p-1)!\,2|h|\left(D|J| + \frac{R_Q\gamma}{2}\right)\left(|h| + D|J| + \frac{R_Q\gamma}{2}\right)^{p-2}. \tag{4.190}$$

The final inequality follows from $2|h|(D|J| + \frac{R_Q\gamma}{2}) \le (|h| + D|J| + \frac{R_Q\gamma}{2})^2$. □

Proposition 4.7.9 gives estimates for each commutator length p. To apply the estimate in

Section 4.5.5, we need a finite-p_0 growth scale controlling all $2 \le p \le p_0$. Define

$$\Theta_{Q,p_0} := \max_{2\le p\le p_0} \left(4^{p-1}(p-1)!\, 2|h| \left(D|J| + \frac{R_Q\gamma}{2} \right) \left(|h| + D|J| + \frac{R_Q\gamma}{2} \right)^{p-2} \right)^{1/p}. \tag{4.191}$$

Proposition 4.7.9 gives $\alpha_{\mathrm{comm},p}(G_{k_i}) \le |V|\Theta_{Q,p_0}^p$ for $2 \le p \le p_0$. Proposition 4.7.10 converts this finite-order commutator estimate into a bound on $\overline{\mu}_{m,Q}(p_0)$ for the locality-based complexity analysis.

Proposition 4.7.10. *Let $p_0 \ge 3$. We have $\overline{\mu}_{m,Q}(p_0) \le |V|^{1/3}(1+p_0)^{1/3}\Theta_{Q,p_0}$, where $\bar{\mu}_{m,Q}(p_0) := \max_{i\in\mathcal{I}_Q} \mu_{m,p_0}(G_{k_i})$, μ_{m,p_0} is defined in Equation (4.2) and Θ_{Q,p_0} is defined in Equation (4.191).*

Proof. The proof is analogous to that of Lemma 4.5.10, so we omit the details. □

We now instantiate the locality-based complexity estimate. By Proposition 4.7.10 and the definitions of $\Phi^*_{m,Q}(p_0)$ and $\Lambda_{m,Q}(p_0)$ in Equation (4.76), we have

$$\Phi^*_{m,Q}(p_0) \le (|V|^{1/3}(1+p_0)^{1/3}\Theta_{Q,p_0})^{2m+1}, \quad \Lambda_{m,Q}(p_0) \le \alpha_Q(|V|^{1/3}(1+p_0)^{1/3}\Theta_{Q,p_0})^{2m+1}. \tag{4.192}$$

Together with the locality and extensivity bounds, $q_Q = 2$ and $g_Q \le |h| + D|J| + R_Q\gamma$, so Section 4.5.5 applies. Choose p_0, r_m, and δ_m as in Section 4.5.5. By Equation (4.191), we have $\Theta_{Q,p_0} \le Cp_0\left(|h| + D|J| + \frac{R_Q\gamma}{2}\right)$ for some $C > 0$. Hence, we have $\overline{\mu}_{m,Q}(p_0) \le C|V|^{1/3}\left(|h| + D|J| + \frac{R_Q\gamma}{2}\right)(1+p_0)^{4/3}$. Since $\delta_{p_0} = 3|V|e^{-p_0}$ and $K_m = \mathcal{O}(m^2(\log m)^2)$, the admissibility condition in Equation (4.78) holds with $p_0(m) = \mathcal{O}(\log(e+m))$. Thus, the commutator-profile contribution grows as $(\log(e+m))^{4/3}$. For the optimized-order condition, the relevant quantity is $\mu_{m,Q}$, not only $\overline{\mu}_{m,Q}(p_0)$.The rooted $\Phi^*_{m,Q}$- and $\Lambda_{m,Q}$ terms in Equation (4.79) contribute powers of $\overline{\mu}_{m,Q}(p_0)^{1+1/(2m)}$. Hence a safe growth function is $F(m) :=$

$(\log(e+m))^2$. More explicitly, one may take

$$\mu_Q := C_Q \max\left\{1,\ q_Q g_Q,\ |V|^{1/3}\left(|h|+D|J|+\frac{R_Q\gamma}{2}\right),\ \alpha_Q^{1/2}\left[|V|^{1/3}\left(|h|+D|J|+\frac{R_Q\gamma}{2}\right)\right]^{3/2}\right\}, \tag{4.193}$$

where $C_Q > 0$ is independent of m, T, and ϵ_Q, and absorbs the implicit constants. With this choice, the optimized-order growth condition analogous to Equation (4.55) holds as $\mu_{m,Q} \leq \mu_Q(\log(e+m))^2$. Choose $m = \max\left\{m_0, \left\lceil \log\left(e+\frac{T}{\epsilon_Q}\right)\right\rceil\right\}$. Applying the optimized complexity estimate with $F(m) = (\log(e+m))^2$ gives the block encoding complexity to be

$$\mathcal{Q}_{\text{block}} = \mathcal{O}\left(\left(1+\mu_Q \max\{1,T\}(\log(e+m))^2\right)\left(\log\left(e+\frac{T}{\epsilon_Q}\right)\right)^2\left(\log\log\left(e^e+\frac{T}{\epsilon_Q}\right)\right)^2\right). \tag{4.194}$$

Similarly, the normalized output state complexity is

$$\mathcal{O}\left(\chi_Q(u(0))\left(1+\mu_Q \max\{1,T\}(\log(e+m))^2\right)\left(\log\left(e+\frac{T}{\epsilon_Q}\right)\right)^2\left(\log\log\left(e^e+\frac{T}{\epsilon_Q}\right)\right)^2\right). \tag{4.195}$$

4.8 Conclusion

We have established commutator-scaling complexity analysis for the Linear Combination Hamiltonian Simulation algorithm based on Multi-Product Formulas. Several natural directions for future work are discussed below.

A natural question is whether the quadrature dependence can be further optimized beyond the quadrature rules analyzed here. Although the present analysis allows fairly general quadrature rules, the resulting complexity bounds still depend on quantities such as $\Lambda_{m,Q}$, R_Q,

α_Q, and the discrete moments M_ν^Q. It would be interesting to determine whether there exist quadrature constructions that simultaneously improve discretization accuracy, LCU normalization, moment growth, and post-quadrature MPF cost. A related question is how to optimize the LCHS kernel profile jointly with the quadrature rule. The kernel parameters influence the approximation error, truncation error, quadrature radius, LCU normalization, and the discrete moment bounds entering the MPF error profiles. A more systematic kernel–quadrature optimization may therefore improve the overall complexity of the algorithm.

The LCU implementation introduces an overhead governed by the normalization of the linear combination and by the norm of the desired output state. Although oblivious amplitude amplification can boost the success probability efficiently, the resulting state-preparation cost still contains normalization-dependent factors. It would be interesting to determine whether alternative block-encoding constructions, quantum signal processing techniques, or dissipative embeddings can reduce this overhead while preserving the commutator-sensitive advantages of the LCHS–MPF framework.

Several extensions of the time-dependent framework remain open. The analysis in Section 4.6.2 assumes sufficient regularity of the time-dependent generator on each subinterval. It would be interesting to determine whether similar complexity guarantees can be obtained under weaker assumptions, such as piecewise-smooth, discontinuous, or rapidly varying dissipative generators within the LCHS-compatible setting.

Finally, it would be useful to better understand the practical performance of MPF-based dissipative simulation. The present work focuses primarily on asymptotic complexity bounds, but concrete implementations may exhibit additional structure that is not visible at the worst-case level. Numerical studies comparing quadrature rules, truncation strategies, MPF orders,

Trotter numbers, and Hamiltonian-simulation primitives could clarify which parameter regimes are most favorable for near-term or fault-tolerant quantum architectures.

Part II

Chebyshev-Based Quantum Computation

Chapter 5: Chebyshev-Based Function Approximation with Quantum Circuits

The results presented in this chapter are based on joint work with Haizhao Yang and Christoph Schwab [AY25; Aft+25]. The results have been included with their permission. All aspects of the work in [AY25] were carried out by Junaid Aftab, with Haizhao Yang serving as the mentor for the project. The work in [Aft+25] was inspired by discussions with Christoph Schwab, who suggested the problem. All authors contributed equally to this work.

5.1 Introduction

Quantum computing has demonstrated considerable potential for addressing high-dimensional computational problems, including the numerical solution of partial differential equations (PDEs) [CLO21; Liu+21c; JLY23]. However, the majority of existing quantum algorithms are developed under the assumption of fault-tolerant quantum hardware, which remains beyond the reach of current quantum technologies. This gap between theoretical algorithms and practical hardware capabilities has motivated growing interest in approaches that are compatible with near-term quantum devices. Within this context, parameterized quantum circuits (PQCs) have emerged as a promising and flexible paradigm [Ben+19]. As a result, a broad class of PQC-based algorithms has been proposed for a wide range of applications, including quantum machine learning

and variational quantum algorithms [Cer+21; Til+22; Alg+22; Liu+21a].

Parameterized quantum circuits can be viewed as quantum analogues of classical feedforward neural networks. In classical machine learning, substantial progress has been achieved in developing a rigorous theoretical foundation for neural networks. In particular, neural network approximation theory [DHP21] provides a systematic framework for analyzing the expressive power of neural architectures, with an emphasis on deriving approximation error bounds in terms of network width, depth, and the total number of neurons. This body of work seeks to characterize the fundamental capabilities and limitations of neural networks in approximating functions from various function spaces, independent of specific training algorithms or data availability.

In an analogous manner, establishing a rigorous theoretical framework for parameterized quantum circuits is essential for the principled assessment of quantum machine learning models and their potential advantages over classical approaches. Recent work has established universal approximation theorems for PQCs [Pér+21; GTN21; Yu+22; GJ23], which guarantee the existence of quantum circuits capable of approximating target functions to arbitrary accuracy. Despite these theoretical guarantees, the circuit complexity required to realize such approximations on quantum hardware remains poorly understood. Addressing this gap is critical for strengthening universal approximation results by providing explicit circuit constructions and by deriving quantitative bounds on their computational resources.

In this chapter, we build upon the framework introduced in [Yu+24] to further advance the emerging theory of function approximation by quantum neural networks. Specifically, [Yu+24] demonstrates that certain PQCs architectures can efficiently approximate continuous and smooth functions through multivariate polynomial approximation schemes. In contrast,

we construct explicit *unparameterized* quantum circuits that approximate target functions via tensor-product Chebyshev polynomial expansions. This approach enables the approximation of broader classes of functions, including those that do not necessarily satisfy generic continuity or smoothness assumptions. The function spaces considered in this work are summarized below:

We first consider the Korobov function space, which can be viewed as a subspace of certain Sobolev spaces and provides a natural and well-studied framework for high-dimensional approximation problems. Its defining structure is closely aligned with sparse grid decompositions [BG04], which were developed to mitigate the *curse of dimensionality* [BK59] that affects conventional grid-based numerical methods. Sparse grid techniques were originally introduced for high-dimensional numerical integration [Kor59; Smo63] and were later extended to the numerical solution of partial differential equations (PDEs) [GH07; RSY+17; CR19], as well as to approximation schemes for high-dimensional functions. From a functional-analytic perspective, Korobov spaces are subspaces of suitable Sobolev spaces and thus provide a flexible and expressive setting for approximating solutions of high-dimensional PDEs. These properties make the Korobov function space particularly well suited for our study, as they enable the construction of scalable quantum circuits that exploit sparsity and structural regularity in high dimensions.

We next consider a class of target functions motivated by polynomial chaos expansions arising in the approximation of parametric partial differential equations with uncertain or high-dimensional inputs. In this setting, the solution depends on a potentially countable collection of parameters, and its approximability is governed not by the nominal parameter dimension but by the decay and sparsity of the associated expansion coefficients. We focus on countably parametric holomorphic functions defined on infinite-dimensional domains $U = [-1, 1]^{\mathbb{N}}$, which naturally arise in uncertainty quantification and parametric PDE analysis. When suitable summability

conditions are satisfied, accurate approximation can be achieved through finite n-term truncations with convergence rates that are independent of the underlying parameter dimension. This dimension-independent behavior makes this function class a natural complement to the Korobov setting and a meaningful benchmark for assessing the expressive power and circuit complexity of quantum approximation schemes based on Chebyshev expansions.

This chapter thus advances a quantitative approximation theory for quantum neural networks by moving beyond existential universality results toward explicit, resource-aware constructions. By leveraging Chebyshev polynomial expansions and unparameterized quantum circuits, we establish approximation schemes whose complexity reflects the intrinsic structure of the target function spaces rather than their ambient dimension. The Korobov and countably parametric holomorphic settings together demonstrate how sparsity, regularity, and coefficient decay can be systematically exploited to achieve scalable quantum approximations. These results provide a step toward a principled assessment of the practical potential of near-term quantum architectures for high-dimensional numerical problems.

5.1.1 Related Works

We briefly discuss some related works in both neural network and quantum computing literature.

5.1.1.1 Neural Network Literature

Substantial progress has been made toward a rigorous theoretical understanding of deep neural networks (DNNs). It has been shown that DNNs outperform traditional approximation

tools in classical function spaces, such as the space of continuous functions [SYZ20; SYZ22; SYZ21b] and Sobolev spaces [YYX23; Yan+23]. In particular, for d-dimensional continuous functions, DNNs with advanced activation functions can achieve approximation accuracy using a number of parameters that scale polynomially in d [SYZ21a; SYZ21b; ZSY22], thereby avoiding the curse of dimensionality. Moreover, DNNs with simple activation functions can attain dimension-independent approximation rates when the target function belongs to a reduced function space, such as the Barron function class [Bar93; EMW19], certain PDE solution spaces [EW22; LY24; Hut+20; Gro+23; GS23], or sparsity-admitting function spaces as studied in the context of polynomial chaos expansions and Korobov spaces [MD19; YL23; TZ22; LZ25; Fan+25].

5.1.1.2 Quantum Computing Literature

Universal approximation properties PQCs have recently attracted increasing attention. Existing results establish that PQCs possess strong expressive capabilities in a variety of classical function spaces. In particular, Manzano et al. [Man+25] prove the existence of PQCs that can approximate functions in $L^p(\mathbb{T}^d)$ and Sobolev spaces $H^k(\mathbb{T}^d)$. However, these results are primarily existential in nature and do not provide explicit circuit constructions or algorithmic procedures. Beyond universal approximation results, recent work has begun to establish quantitative approximation theory for PQCs. Notably, Chen et al. [Yu+24] derive non-asymptotic error bounds, showing that data re-uploading PQC architectures can efficiently approximate multivariate polynomials and smooth functions. Their analysis further indicates that, under suitable smoothness assumptions, PQCs can achieve approximation efficiency comparable to,

or in some cases exceeding, that of classical deep ReLU networks in terms of circuit depth and parameter count. While approximation theory for PQCs does not directly address optimization and training dynamics, these aspects have been investigated in parallel studies. For example, Liu et al. [Liu+23] analyze the training dynamics of wide quantum neural networks, establishing an analytic framework that yields exponential decay of the training residual. In contrast, You and Wu [YW21] study the loss landscape of quantum neural networks, demonstrating that an exponentially large number of local minima may hinder optimization.

5.1.2 Organization

Section 5.2 explains how to implement linear combinations of Chebyshev polynomials on a quantum computer, which forms the basis of the proposed applications in this chapter. Section 5.3 presents the main results on the approximation of Korobov functions on quantum computers, while Section 5.4 while Section 5.4 establishes corresponding results for the approximation of certain countably parametric holomorphic functions on quantum computers. Section 5.5 provides closing remarks and discusses future research directions arising from this chapter.

5.2 Implementing Linear Combination of Chebyshev Polynomials

We discuss how to implement linear combinations of Chebyshev polynomials on a quantum computer. This follows straightforwardly from the Quantum Signal Processing (QSP) and Linear Combination of Unitaries (LCU) algorithms and forms the basis of the applications proposed in this chapter. The QSP method is fundamental to this construction, as it enables the

implementation of univariate Chebyshev polynomials. This is summarized in the lemma below.

Lemma 5.2.1. *Let $T_r \in \mathbb{R}[x]$ denote the degree-r Chebyshev polynomial of the first kind. There exists a phase angle sequence $\vec{\phi} \in \mathbb{R}^{r+1}$ with $\phi_i = 0$ for all $i = 0, 1, \ldots, r$, and an associated (unparameterized) single-qubit QSP circuit $U_r(x)$ such that $\langle 0| U_r(x) |0\rangle = T_r(x)$ for every $r \in \mathbb{N} \cup \{0\}$ and every $x \in [-1, 1]$. Moreover, $U_r(x)$ has width 1, depth $2r + 1$, and uses $r + 1$ predetermined parameters.*

Remark 5.2.2. Reference [Gil+19a] proves a slightly modified version of Lemma 5.2.1. This discrepancy arises because [Gil+19a] employs a variant of the quantum signal processing circuit that differs from the one adopted here. For completeness, we provide a self-contained proof of the statement below.

Proof. (Lemma 5.2.1) The claim is true for $r = 0$ since in this case $U_r(x) = I$ and the $(1, 1)$ entry of I is $T_0(x) = 1$. We prove by induction that

$$e^{ir \arccos(x)\sigma_X} = \begin{pmatrix} T_r(x) & i\sqrt{1-x^2}S_{r-1}(x) \\ i\sqrt{1-x^2}S_{r-1}(x) & T_r(x) \end{pmatrix}, \tag{5.1}$$

for each $r \geq 1$. For $r = 1$, we have

$$e^{i \arccos(x)\sigma_X} = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix}. \tag{5.2}$$

The claim is true since $T_1(x) = x$ and $S_0(x) = 1$. Assume the claim is true for $r > 1$. We have,

$$e^{i(r+1)\arccos(x)\sigma_X} = e^{ir\arccos(x)\sigma_X} e^{i\arccos(x)\sigma_X} \tag{5.3}$$

$$= \begin{pmatrix} T_r(x) & i\sqrt{1-x^2}S_{r-1}(x) \\ i\sqrt{1-x^2}S_{r-1}(x) & T_r(x) \end{pmatrix} \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} \tag{5.4}$$

$$= \begin{pmatrix} xT_r(x) - (1-x^2)S_{r-1}(x) & i\sqrt{1-x^2}(T_r(x) + xS_{r-1}(x)) \\ i\sqrt{1-x^2}(T_r(x) + xS_{r-1}(x)) & xT_r(x) - (1-x^2)S_{r-1}(x) \end{pmatrix} \tag{5.5}$$

$$= \begin{pmatrix} T_{r+1}(x) & i\sqrt{1-x^2}S_r(x) \\ i\sqrt{1-x^2}S_r(x) & T_{r+1}(x) \end{pmatrix}. \tag{5.6}$$

The last equality follows since the Chebyshev polynomials follows that they also satisfy a pair of mutual recurrence equations:

$$T_{r+1}(x) = xT_r(x) - (1-x^2)S_{r-1}(x), \qquad S_r(x) = (x)T_r(x) + xS_{r-1}(x). \tag{5.7}$$

Since we have

$$U_r(x) = e^{i0\sigma_Z} \prod_{j=1}^{r} \left(e^{i\arccos(x)\sigma_X} e^{i0\sigma_Z} \right) = e^{ir\arccos(x)\sigma_X} = \begin{pmatrix} T_r(x) & i\sqrt{1-x^2}S_{r-1}(x) \\ i\sqrt{1-x^2}S_{r-1}(x) & T_r(x) \end{pmatrix},$$

the claim follows. Clearly, $U_r(x)$ has width 1, depth $2r+1$, and $r+1$ (predetermined) parameters. See Figure 5.1 for the corresponding quantum circuit. □

We can now implement linear combinations of products of Chebyshev polynomials on a quantum computer. To this end, we employ the Linear Combination of Unitaries (LCU)

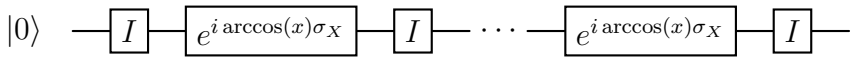


Figure 5.1: The circuit diagram for the quantum circuit $U_r(x)$ in Lemma 5.2.1.

algorithm together with Lemma 5.2.1.

Proposition 5.2.3. *Fix $d, M \in \mathbb{N}$. Let $\vec{x} \in [-1,1]^d$ and consider a linear combination of products of Chebyshev polynomials*

$$f(\vec{x}) = \sum_{i=1}^{M} a_i \prod_{j=1}^{d} T_{n_{ij}}(x_j), \tag{5.8}$$

where $a_1, \cdots, a_M \in \mathbb{R}$ and $n_{i,j} \in \mathbb{N}$ for $i = 1, \cdots, N$ and $j = 1, \cdots, d$. There exists a quantum circuit, $U_{\text{Cheb}}(\vec{x})$, such that if a measurement corresponding to the Pauli σ_Z observable is made only on the first qubit, denoted as $\sigma_Z^{(1)}$ then

$$\langle 0| U_{\text{Cheb}}^{\dagger}(\vec{x}) \sigma_Z^{(1)} U_{\text{Cheb}}(\vec{x}) |0\rangle = f(\vec{x}). \tag{5.9}$$

The quantum circuit $U_{\text{Cheb}}(\vec{x})$ has depth $\mathcal{O}((2\|\vec{n}\|_1 + dM)\log_2 M)$ and width $\mathcal{O}(d + \log_2 M)$.

Proof. For each $n_{i,j} \in \mathbb{N}$, Lemma 5.2.1 implies that there exists a quantum circuit $U_{n_{i,j}}(x_j)$ such that $\langle 0| U_{n_{i,j}}(x_j) |0\rangle = T_{n_{ij}}(x_j)$. If we consider the unitary operator $U_i(\vec{x}) = \bigotimes_{j=1}^{d} U_{n_{i,j}}(x_j)$, then $U_i(\vec{x})$ can be implemented with a width d quantum circuit comprising of $2\|\vec{n}_i\|_1 + d$ quantum gates, where $\vec{n}_i = (n_{i1}, \cdots, n_{id})$. We invoke the LCU algorithm to implement the following operator:

$$U_{\text{LCU}}(\vec{x}) := \sum_{i=1}^{M} a_i U_i(\vec{x}). \tag{5.10}$$

First, consider a state-preparation oracle PREP such that

$$\text{PREP}\,|0\rangle^{\log_2 M} = \frac{1}{\sqrt{\|\vec{a}\|_1}} \sum_{j=1}^{M} \sqrt{a_i}\,|j\rangle\,. \tag{5.11}$$

Since there are M terms in the sum in Equation (5.10), PREP can be implemented using a quantum circuit with $\mathcal{O}(M)$ gates acting on $\mathcal{O}(\log_2 M)$ ancilla qubits. Next, consider the select oracle SEL such that

$$\text{SEL}(\vec{x}) = \sum_{i=1}^{M} U_i(\vec{x}) \otimes |i\rangle\langle i|\,. \tag{5.12}$$

$\text{SEL}(\vec{x})$ acts on a composite quantum system that includes $\mathcal{O}(d)$ computational qubits and $\mathcal{O}(\log_2 M)$ ancilla qubits. Since each $U_i(\vec{x})$ can be implemented using a $2\|\vec{n}_i\|_1 + d$ single-qubit gates, $\text{SEL}(\vec{x})$ can be implemented via

$$\sum_{i=1}^{M}(2\|\vec{n}_i\|_1 + d) = \sum_{i=1}^{M}\sum_{j=1}^{d} 2n_{ij} + dM := 2\|\vec{n}\|_1 + dM \tag{5.13}$$

single qubit gates controlled on $\mathcal{O}(\log_2 M)$ ancilla qubits. A $\mathcal{O}(\log_2 M)$-qubit controlled gate acting can be implemented via a quantum circuit using only CNOT gates and single-qubit gates such that the circuit has depth $\mathcal{O}(\log_2 M)$ [SP22; Yu+24]. Consequently, the overall implementation of $\text{SEL}(\vec{x})$ involves a quantum circuit with depth $\mathcal{O}((2\|\vec{n}\|_1 + dM)\log_2 M)$ and width $\mathcal{O}(d + \log_2 M)$. We can now use the LCU algorithm to implement the operator $U_{\text{LCU}}(\vec{x})$ defined as

$$U_{\text{LCU}}(\vec{x}) = (I_2 \otimes \text{PREP}^\dagger)\,\text{SEL}(\vec{x})(I_2 \otimes \text{PREP}). \tag{5.14}$$

The asymptotic scaling of both the depth and the width for implementing $U_{\text{LCU}}(\vec{x})$ is dominated

by the complexity of $\mathrm{SEL}(\vec{x})$. The desired quantum circuit, $U_{\mathrm{Cheb}}(\vec{x})$, is constructed by applying the Hadamard test to a single ancilla qubit in order to estimate the expected value of U_{LCU}. Note that we have

$$\langle 0| U_{\mathrm{Cheb}}(\vec{x}) |0\rangle = \sum_{i=1}^{M} a_i \prod_{j=1}^{d} \langle 0| U_{n_{ij}}(x_j) |0\rangle = \sum_{i=1}^{M} a_i \prod_{j=1}^{d} T_{n_{ij}}(x_j). \tag{5.15}$$

The additional operations involved in the Hadamard test have $\mathcal{O}(1)$ complexity. Thus, the overall complexity of the algorithm is determined by the implementation of the LCU algorithm. This completes the proof. □

The significance of Proposition 5.2.3 lies in the fact that it provides an explicit quantum circuit, together with rigorous complexity estimates, for implementing an arbitrary linear combination of Chebyshev polynomials on a quantum computer. This construction enables the systematic realization of polynomial transformations of operators within a unified circuit framework. Throughout the remainder of this work, we assume that the corresponding PREP oracle required by this construction can be implemented with the stated complexity. Under this assumption, our results should be interpreted primarily as statements in the approximation theory of quantum circuits. In particular, they characterize the expressive capabilities of quantum circuits by identifying which classes of functions can be efficiently approximated on a quantum computer via linear combinations of Chebyshev polynomials.

5.3 Approximating Korobov Functions

We begin by presenting the hierarchical basis construction that forms the foundation of the sparse grid decomposition. Subsequently, we introduce the Korobov function space and derive an estimate for the quantum circuit complexity required to approximate Korobov functions of low order.

5.3.1 Hierarchical Basis

We first consider the one-dimensional case ($d = 1$). Fix $h_n = 2^{-n} \in \mathbb{R}$ for some $n \in \mathbb{N}$. Many numerical methods are based on uniform discretizations of the computational domain. Let $\mathcal{P}_n$ denote a uniform discretization of the interval $[0, 1]$, consisting of the points $\{ih_n\}_{i=1}^{2^n-1}$. We now define a piecewise linear function, referred to as the *hat function*:

$$\phi(x) = \begin{cases} 1 - |x|, & \text{if } x \in [-1, 1], \\ 0, & \text{otherwise.} \end{cases} \tag{5.16}$$

The approach is to consider a family of finite-dimensional vector spaces spanned by one-dimensional hat functions, which serve as building blocks for approximating more complex functions on the interval $[0, 1]$. Specifically, for each $x_i \in \mathcal{P}_n$, we define the function $\phi_{n,i}(x)$ by

$$\phi_{n,i}(x) = \phi\left(\frac{x - x_i}{h_n}\right). \tag{5.17}$$

The function $\phi_{n,i}(x)$ is a scaled and translated version of the hat function $\phi(x)$, centered at x_i with width h_n. We now consider the finite-dimensional vector space V_n, defined as follows:

$$V_n = \text{span}\{\phi_{n,i} : 1 \leq i \leq 2^n - 1\}. \tag{5.18}$$

The space V_n serves as the approximating space. The corresponding basis functions are referred to as the one-dimensional nodal basis (see Figure 5.2).

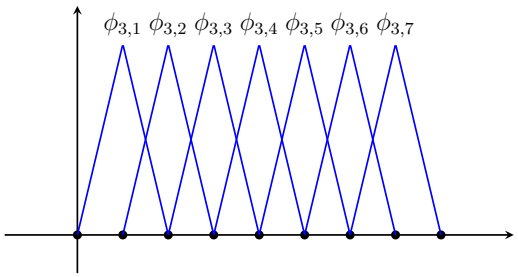


Figure 5.2: The 1-dimensional nodal basis for V_3 consists of functions $\phi_{3,i}$, $1 \leq i \leq 7$.

The sparse grid construction, discussed in the subsequent subsection, exploits the fact that V_n can be represented as a sequence of nested approximation spaces, each associated with a progressively refined grid. The central idea is to select grid points in a hierarchical manner. To this end, consider a sequence of nested grids $\{G_\ell\}_{\ell=1}^n$, where the level-k grid is defined by

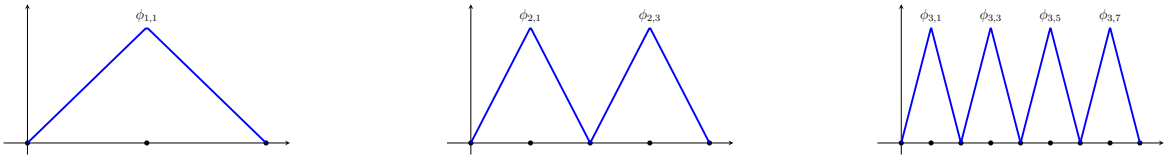
$$G_\ell := \left\{ x_i^{(\ell)} = \frac{i}{2^\ell} : i = 1, 3, 5, \ldots, 2^\ell - 1 \right\}. \tag{5.19}$$

Note that only the odd-indexed points are included. This choice ensures that $G_\ell \cap G_{\ell'} = \emptyset$ for $\ell \neq \ell'$, thereby enabling a hierarchical decomposition. Note that we have $\mathcal{P}_n = \coprod_{\ell=1}^n G_\ell$. This construction avoids redundancy and supports a multi-resolution representation. Observe that if

we consider the finite-dimensional vector space

$$W_\ell = \text{span}\{\phi_{\ell,i} : i = 1, 3, 5 \cdots, 2^\ell - 1\}, \tag{5.20}$$

it follows that the space V_n is the direct sum $V_n = \bigoplus_{1 \le \ell \le n} W_\ell$. The basis functions of W_ℓ are referred to as the one-dimensional hierarchical basis functions (see Figure 5.3). These functions capture the incremental refinements in resolution between successive levels.



(a) W_1 (b) W_2 (c) W_3

Figure 5.3: The 1-dimensional hierarchical basis vectors for W_1, W_2 and W_3 are plotted.

We now extend the construction to higher dimensions ($d > 1$) via a tensor product approach. Consider a uniform discretization of the domain $[0, 1]^d$, and define the d-dimensional *hat functions* as products of the corresponding one-dimensional basis functions.

$$\phi_{\vec{n},\vec{i}}(\vec{x}) = \prod_{j=1}^{d} \phi_{n_j,i_j}(x_j) = \prod_{j=1}^{d} \phi\left(\frac{x_j - x_{i_j}}{h_{n_j}}\right). \tag{5.21}$$

We define the function spaces $V_{\vec{n}}$ and $W_{\vec{\ell}}$ as follows:

$$V_{\vec{n}} = \text{span}\left\{\phi_{\vec{n},\vec{i}} : \vec{1} \le \vec{i} \le 2^{\vec{n}} - \vec{1}\right\}, \tag{5.22}$$

$$W_{\vec{\ell}} = \text{span}\left\{\phi_{\vec{\ell},\vec{i}} : \vec{1} \le \vec{i} \le 2^{\vec{\ell}} - \vec{1},\ i_j \text{ odd for all } j\right\}. \tag{5.23}$$

Here, $V_{\vec{n}}$ denotes the d-dimensional nodal basis, while $W_{\vec{\ell}}$ denotes the d-dimensional hierarchical

basis. The hierarchical basis functions in $W_{\vec{\ell}}$ capture the incremental detail at the resolution level $\vec{\ell}$ and are supported on grids that are coarser or finer depending on $\vec{\ell}$. These spaces satisfy the direct sum decomposition $V_{\vec{n}} = \bigoplus_{\|\vec{\ell}\|_\infty \leq n} W_{\vec{\ell}}$.

5.3.2 Korobov Function Space

We now introduce the Korobov function space, which will be shown to admit approximation by d-dimensional hierarchical basis functions. For $2 \leq p \leq \infty$, the Korobov function space is defined as follows:

$$X^{2,p}([0,1]^d) = \left\{ f \in L^p([0,1]^d) : f|_{\partial[0,1]^d} = 0,\ D^{\vec{k}} f \in L^p([0,1]^d),\ \|\vec{k}\|_\infty \leq 2 \right\}. \tag{5.24}$$

Note that $X^{2,p}([0,1]^d)$ is a subspace of the Sobolev space $H^p([0,1]^d)$. The distinction arises from the multi-index conditions appearing in the definitions of $X^{2,p}([0,1]^d)$ and $H^p([0,1]^d)$. Specifically, the Korobov space is characterized by the condition $\|\vec{k}\|_\infty \leq 2$, whereas the Sobolev space is characterized by the condition $\|\vec{k}\|_1 \leq 2$.

Example 5.3.1. In two dimensions ($d = 2$), a function f belongs to $X^{2,p}([0,1]^2)$ if and only if

$$\frac{\partial f}{\partial x_1},\quad \frac{\partial f}{\partial x_2},\quad \frac{\partial^2 f}{\partial x_1^2},\quad \frac{\partial^2 f}{\partial x_2^2},\quad \frac{\partial^2 f}{\partial x_1 \partial x_2},\quad \frac{\partial^3 f}{\partial x_1^2 \partial x_2},\quad \frac{\partial^3 f}{\partial x_1 \partial x_2^2},\quad \frac{\partial^4 f}{\partial x_1^2 \partial x_2^2} \in L^p([0,1]^2). \tag{5.25}$$

On the other hand, a function f belongs to $H^2([0,1]^2)$ if and only if

$$\frac{\partial f}{\partial x_1},\quad \frac{\partial f}{\partial x_2},\quad \frac{\partial^2 f}{\partial x_1^2},\quad \frac{\partial^2 f}{\partial x_2^2},\quad \frac{\partial^2 f}{\partial x_1 \partial x_2} \in L^p([0,1]^2). \tag{5.26}$$

Hence, $X^{2,p}([0,1]^2) \subseteq H^2([0,1]^2)$.

An important property of the Korobov function space is that any function $f \in X^{2,p}([0,1]^d)$ admits an infinite expansion in terms of hierarchical basis functions. Indeed, it is a standard result that $V = \bigoplus_{\vec{\ell} \in \mathbb{N}^d} W_{\vec{\ell}}$ is dense in $H_0^1([0,1]^d)$; that is, $\overline{V} = H_0^1([0,1]^d)$, where the overline denotes the closure of V in the H^1-norm. The inclusion $X^{2,p}([0,1]^d) \subseteq H_0^1([0,1]^d)$ implies that any function $f \in X^{2,p}([0,1]^d)$ admits an infinite expansion in terms of the hierarchical basis functions. Specifically,

$$f(\vec{x}) = \sum_{\vec{\ell}} \sum_{\vec{i} \in I_{\vec{\ell}}} v_{\vec{\ell},\vec{i}} \phi_{\vec{\ell},\vec{i}}(\vec{x}), \tag{5.27}$$

where the index set $I_{\vec{\ell}}$ is defined as

$$I_{\vec{\ell}} = \left\{ \vec{i} \in \mathbb{N}^d \,\middle|\, \vec{1} \leq \vec{i} \leq 2^{\vec{\ell}} - \vec{1}, \text{ and } i_j \text{ is odd for each } j \right\}. \tag{5.28}$$

In the remainder of this section, we restrict attention to the case $p \in \{2, \infty\}$. This is primarily because the so-called optimal sparse grid decomposition is discussed in [BG04] only for the cases $p = 2$ and $p = \infty$. We have the following result:

Proposition 5.3.2 (Lemma 3.3 in [BG04]). *Let $p \in \{2, \infty\}$ and consider $X^{2,p}([0,1]^d)$ endowed with the seminorm on $X^{2,p}([0,1]^d)$:*

$$|f|_{\vec{2},p} := \left\| \frac{\partial^{2d} f}{\partial x_1^2 \cdots \partial x_d^2} \right\|_{L^p([0,1]^d)}. \tag{5.29}$$

Each $f \in X^{2,p}([0,1]^d)$ can be written as in Equation (5.27) such that

$$v_{\vec{\ell},\vec{i}} = \int_{[0,1]^d} \prod_{j=1}^{d} \left(-2^{-(\ell_j+1)} \phi_{\ell_j,i_j}(x_j)\right) \frac{\partial^{2d} f}{\partial x_1^2 \cdots \partial x_d^2}(\vec{x})\, d\vec{x} \tag{5.30}$$

with the following bound on the coefficients:

1. *If $p = 2$, then*

$$|v_{\vec{\ell},\vec{i}}| \leq 2^{-d} \left(\frac{2}{3}\right)^{d/2} 2^{-(3/2)\cdot\|\vec{\ell}\|_1} |f \mathbb{1}_{\mathrm{supp}(\phi_{\vec{\ell},\vec{i}})}|_{\vec{2},2}, \tag{5.31}$$

   where $\mathrm{supp}(\phi_{\vec{\ell},\vec{i}})$ is the support of $\phi_{\vec{\ell},\vec{i}}$.

2. *If $p = \infty$, then $|v_{\vec{\ell},\vec{i}}| \leq 2^{-d-2|\vec{\ell}|_1} |f|_{\vec{2},\infty}$.*

Note that the coefficients $v_{\vec{\ell},\vec{i}}$ in Proposition 5.3.2 exhibit exponential decay with respect to the level index, satisfying $\|v_{\vec{\ell},\vec{i}}\| = \mathcal{O}(2^{-2|\vec{\ell}|_1})$. This behavior arises from the fact that functions in the Korobov space possess mixed second-order derivatives in all spatial directions. In contrast, the dimension of the subspace $W_{\vec{\ell}}$ at level $\vec{\ell}$ grows exponentially as $\mathcal{O}(2^{|\vec{\ell}|_1})$. An optimization procedure, as described in [BG04], balances the trade-off between the number of degrees of freedom and the resulting approximation error. This leads to the sparse grid decomposition of level $n \in \mathbb{N}$:

$$V_n^s = \bigoplus_{\|\vec{\ell}\|_1 \leq n+d-1} W_{\vec{\ell}}. \tag{5.32}$$

Traditional numerical methods that discretize each coordinate axis using n points result in a total of $\mathcal{O}(n^d)$ grid points. Sparse grid techniques address this challenge by selecting a structured subset of the full tensor-product basis, thereby significantly reducing the number of degrees of freedom while incurring only a modest loss in accuracy, as made precise in Proposition 5.3.3.

Proposition 5.3.3 (Lemma 3.6 & Lemma 3.13 in [BG04]). *Let $d, n \in \mathbb{N}$, $\epsilon > 0$, and $p \in \{2, \infty\}$. Consider the sparse grid decomposition as in Equation (5.32). The number of grid points, N, is $N = \mathcal{O}(2^n n^{d-1})$. Moreover, any $f \in X^{2,p}([0,1]^d)$ can be approximated by*

$$f_n^s(\vec{x}) = \sum_{\substack{\|\vec{\ell}\|_1 \leq n+d-1 \\ \vec{i} \in I_{\vec{\ell}}}} v_{\vec{\ell},\vec{i}} \phi_{\vec{\ell},\vec{i}}(\vec{x}) \ \in \ V_n^s \tag{5.33}$$

such that with respect to the $\|\cdot\|_{L^p([0,1]^d)}$ norm, we have

$$\|f - f_n^s\|_{L^p([0,1]^d)} = \mathcal{O}(N^{-2} \log_2^{3(d-1)} N). \tag{5.34}$$

In particular, for every $\epsilon \in (0,1)$, it suffices to choose $N = \mathcal{O}(\epsilon^{-\frac{1}{2}} \log_2(1/\epsilon)^{\frac{3}{2}(d-1)})$ in order to ensure that $\|f - f_n^s\|_{L^p([0,1]^d)} \leq \epsilon$.

5.3.3 Approximation Rates

We now present our main results on approximating Korobov functions, which provides a rigorous complexity analysis of the proposed quantum circuit for approximating functions in the Korobov space $X^{2,p}([0,1]^d)$. We begin with the cases $p \in \{2, \infty\}$. Recall that each one-dimensional hierarchical basis function is obtained by appropriate scaling and translation of the fundamental *hat function* $\phi(x)$. A key observation, which is both simple and central to our construction, is that the function $1 - |x|$ admits a representation as a linear combination of Chebyshev polynomials of degrees 0 and 1. Specifically, for $x \in [-1, 1]$, we have $1 \pm x = P_0(x) \pm P_1(x)$, where P_0 and P_1 denote the Chebyshev polynomials of the first kind of degrees

0 and 1, respectively. Using this identity, we may express the hat function $\phi(x)$ as

$$\phi(x) = [P_0(x) + P_1(x)]\mathbb{1}_{[-1,0]}(x) + [P_0(x) - P_1(x)]\mathbb{1}_{[0,1]}(x). \tag{5.35}$$

Based on the discussion in the preceding subsection, any function $f \in X^{2,p}([0,1]^d)$, for $p \in \{2, \infty\}$, admits an approximation of the form $f_n^s(\vec{x})$ as defined in Equation (5.33). For each $\vec{x} \in [0,1]^d$, we have

$$f_n^s(\vec{x}) = \sum_{\substack{\|\vec{\ell}\|_1 \leq n+d-1 \\ \vec{i} \in I_{\vec{\ell}}}} \sum_{k_1, \cdots, k_d \in \{0,1\}} (-1)^{\sum_{j=1}^d k_j(\operatorname{sgn}(x_j - x_{i_j})+1)/2} v_{\vec{\ell},\vec{i}} \prod_{j=1}^d P_{k_j}\left(\frac{x_j - x_{i_j}}{h_{\ell_j}}\right) \tag{5.36}$$

$$:= \sum_{\substack{\|\vec{\ell}\|_1 \leq n+d-1 \\ \vec{i} \in I_{\vec{\ell}}}} \sum_{k_1, \cdots, k_d \in \{0,1\}} w_{\vec{\ell},\vec{i}} \prod_{j=1}^d P_{k_j}\left(\frac{x_j - x_{i_j}}{h_{\ell_j}}\right), \tag{5.37}$$

where we have defined

$$w_{\vec{\ell},\vec{i}} = (-1)^{\sum_{j=1}^d k_j(\operatorname{sgn}(x_j - x_{i_j})+1)/2} v_{\vec{\ell},\vec{i}}. \tag{5.38}$$

Remark 5.3.4. We discuss an important point before presenting our main result. The sign of the coefficients $w_{\vec{\ell},\vec{i}}$ in Equation (5.38) is determined by $(-1)^{\sum_{j=1}^d k_j(\operatorname{sgn}(x_j - x_{i_j})+1)/2}$. A priori, this expression depends on the input vector $\vec{x} = (x_1, \ldots, x_d)$. However, this dependence does not present a substantive difficulty. The partition points induce a decomposition of $[0,1]^d$ into hypercubes, with the leftmost corner of each hypercube corresponding to a partition point. Given $\vec{x} \in [0,1]^d$, a quantum circuit can be constructed to identify the hypercube containing $\vec{x}$. If $\delta > 0$ denotes an additional precision parameter, the complexity of this circuit is $\mathcal{O}(\delta \log(1/\delta))$. See [Yu+24, Lemma S14] for a detailed argument and sharper bounds. Once the appropriate

hypercube has been identified, all expressions in Equation (5.38) can be evaluated efficiently in $\mathcal{O}(1)$ time. We do not incorporate this cost into our complexity estimates, as it is treated as a pre-processing step performed after the input $\vec{x} \in [0,1]^d$ is specified and prior to executing the main quantum circuit. The focus of this work is on the approximation-theoretic aspects of quantum circuits and their complexity for function approximation, rather than on practical pre-processing considerations, although such considerations will be relevant for real-world implementations.

We are now ready to present our main results.

Proposition 5.3.5. *Let $d, n \in \mathbb{N}$ and $p \in \{2, \infty\}$. For each $f \in X^{2,p}([0,1]^d)$, let $f_n^s(\vec{x})$ be defined as in Equation (5.36). There exists a quantum circuit $U_f(\vec{x})$ such that, if a measurement corresponding to the Pauli σ_Z observable is performed only on the first qubit, then*

$$\langle 0| U_f^\dagger(\vec{x}) \sigma_Z^{(1)} U_f(\vec{x}) |0\rangle = f_n^s(\vec{x}). \tag{5.39}$$

The quantum circuit $U_f(\vec{x})$ has depth $\mathcal{O}(d2^{n+d}n^{d-1}((n+d)+(d-1)\log_2 n))$ and width $\mathcal{O}(2d+n+(d-1)\log_2 n)$.

Proof. Consider the operator

$$U_f(\vec{x}) = \sum_{\substack{\|\vec{\ell}\|_1 \leq n+d-1 \\ \vec{i} \in I_{\vec{\ell}}}} \sum_{k_1, \cdots, k_d \in \{0,1\}} w_{\vec{\ell},\vec{i}} \underbrace{\bigotimes_{j=1}^{d} U_{k_j}\left(\frac{x_j - x_{i_j}}{h_{\ell_j}}\right)}_{U_{\vec{i},\vec{\ell}}(\vec{x})}. \tag{5.40}$$

From Proposition 5.3.3, the number of terms in the outer summation of Equation (5.40) is $\mathcal{O}(2^n n^{d-1})$. Hence, the total number of terms is $M = \mathcal{O}(2^{n+d}n^{d-1})$. As a result, Proposition 5.2.3

implies that the width of $U_f(\vec{x})$ is

$$\mathcal{O}(d + \log_2 M) = \mathcal{O}(2d + n + (d-1)\log_2 n). \tag{5.41}$$

Furthermore, Lemma 5.2.1 establishes that each unitary operator U_{k_j} can be implemented using a quantum circuit with width and depth $\mathcal{O}(1)$, as it corresponds to Chebyshev polynomials of degree at most one. Therefore, the $\|\cdot\|_1$-norm term appearing in the depth of the quantum circuit in Lemma 5.2.1 scales as $\mathcal{O}(dM)$. As a result, Proposition 5.2.3 implies that the depth of $U_f(\vec{x})$ is

$$\mathcal{O}(3dM\log_2 M) = \mathcal{O}(d2^{n+d}n^{d-1}((n+d) + (d-1)\log_2 n)). \tag{5.42}$$

This completes the proof. □

The significance of Proposition 5.3.5 lies in the provision of an explicit quantum circuit, together with corresponding complexity estimates, for approximating functions in $X^{2,p}([0,1]^d)$ with $p \in \{2, \infty\}$. This result enables us to derive worst-case parameter bounds required to approximate the quantum circuit to arbitrary accuracy.

Proposition 5.3.6. *Let $d \in \mathbb{N}$ and $\epsilon \in (0,1)$. For each $f \in X^{2,p}([0,1]^d)$ such that $p \in \{2,\infty\}$, there exists a quantum circuit $U_{f,\epsilon}(\vec{x})$ such that, if a measurement corresponding to the Pauli σ_Z observable is performed only on the first qubit, then*

$$\langle 0| U^\dagger_{f,\epsilon}(\vec{x})\sigma_Z^{(1)}U_{f,\epsilon}(\vec{x})\,|0\rangle = f_n^s(\vec{x}) \tag{5.43}$$

such that $\|f - f_n^s\|_{L^p([0,1]^d)} \leq \epsilon$. The depth of $U_{f,\epsilon}(\vec{x})$ is $\mathcal{O}\left(\frac{d^2\left(2\log_2^{3/2}(1/\epsilon)\right)^d}{\epsilon^{1/2}\log_2^{3/2}(1/\epsilon)}W\left(\frac{\epsilon^{-1/d}}{d}\log_2^{3/2}\left(\frac{1}{\epsilon}\right)\right)\right)$

and the width of $U_{f,\epsilon}(\vec{x})$ *is* $\mathcal{O}\left(2d + dW\left(\frac{\epsilon^{-\frac{1}{d}}}{d}\log_2^{\frac{3}{2}}\left(\frac{1}{\epsilon}\right)\right)\right)$.

Proof. Let $U_{f,\epsilon}(\vec{x})$ be the unitary operator considered in Proposition 5.3.5, and let $f_n^s(\vec{x})$ be as in Equation (5.36). Using the definition of Lambert's W function, we have the following estimate:

$$N = \mathcal{O}(2^n n^{d-1}) \quad \Longleftrightarrow \quad n = \mathcal{O}\left(\frac{d-1}{\ln 2} W\left(\frac{\ln 2 N^{\frac{1}{d-1}}}{d-1}\right)\right) = \Theta\left(dW\left(\frac{N^{\frac{1}{d}}}{d}\right)\right). \tag{5.44}$$

The last estimate above follows since

$$\lim_{d\to\infty} \frac{(d-1)W\left(\frac{c^{1/(d-1)}}{d-1}\right)}{dW\left(\frac{c^{1/d}}{d}\right)} = \lim_{d\to\infty}\frac{d-1}{d}\cdot\lim_{d\to\infty}\frac{W\left(\frac{c^{1/(d-1)}}{d-1}\right)}{W\left(\frac{c^{1/d}}{d}\right)} = 1. \tag{5.45}$$

for any $c \geq 1$. By Proposition 5.3.3, if we choose N such that $N = \mathcal{O}(\epsilon^{-\frac{1}{2}}\log_2^{\frac{3}{2}(d-1)}(1/\epsilon))$, then the error estimate is satisfied. By Equation (5.44), we have

$$n = \mathcal{O}\left(dW\left(\frac{\epsilon^{-\frac{1}{2d}}}{d}\log_2^{\frac{3(d-1)}{2d}}\left(\frac{1}{\epsilon}\right)\right)\right) = \Theta\left(dW\left(\frac{\epsilon^{-\frac{1}{d}}}{d}\log_2^{\frac{3}{2}}\left(\frac{1}{\epsilon}\right)\right)\right). \tag{5.46}$$

The last estimate above follows since

$$\lim_{d\to\infty}\frac{W\left(\frac{\epsilon^{-\frac{1}{2d}}}{d}\log_2^{\frac{3(d-1)}{2d}}\left(\frac{1}{\epsilon}\right)\right)}{W\left(\frac{\epsilon^{-\frac{1}{d}}}{d}\log_2^{\frac{3}{2}}\left(\frac{1}{\epsilon}\right)\right)} = 1. \tag{5.47}$$

The equation above expresses an upper bound on n as a function of d. To emphasize this, we write n as $n(d)$ in what follows. Having expressed n as a function of d, we now simplify the expression $n(d) + d\log_2 n(d)$ and argue that $n(d) + d\log_2 n(d) = \mathcal{O}(n(d))$. Since $W(x) \leq x$ for

$x \geq 0$, we first note that this yields a slightly looser, but simpler upper bound.

$$n(d) = \mathcal{O}\left(\frac{d}{d}\epsilon^{-\frac{1}{d}} \log_2^{\frac{3}{2}}\left(\frac{1}{\epsilon}\right)\right) = \Theta(\epsilon^{-\frac{1}{d}}). \tag{5.48}$$

The final estimate follows since $\log_2^{3/2}(1/\epsilon) = \mathcal{O}(1)$ for fixed $\epsilon \in (0,1)$. The claim that $n(d) + d\log_2 n(d) = \mathcal{O}(n(d))$ now follows from the following computation:

$$\lim_{d\to\infty} \frac{\epsilon^{-\frac{1}{d}} + d\log_2(\epsilon^{-\frac{1}{d}})}{\epsilon^{-\frac{1}{d}}} = 1 + \lim_{d\to\infty} \frac{d\log_2(\epsilon^{-\frac{1}{d}})}{\epsilon^{-\frac{1}{d}}} = 1 + \frac{\ln(1/\epsilon)}{\ln 2} = \Theta(1). \tag{5.49}$$

By Proposition 5.3.5, the depth of $U_{f,\epsilon}(\vec{x})$ is

$$\mathcal{O}\left(d2^{n+d}n^{d-1}(n + d\log_2 n)\right) = \mathcal{O}\left(2^d N d n(d)\right) \tag{5.50}$$

$$= \Theta\left(\frac{d^2\left(2\log_2^{3/2}(1/\epsilon)\right)^d}{\epsilon^{1/2}\log_2^{3/2}(1/\epsilon)} W\left(\frac{\epsilon^{-1/d}}{d}\log_2^{3/2}\left(\frac{1}{\epsilon}\right)\right)\right). \tag{5.51}$$

Here we have used the estimates that

$$2^d N = \mathcal{O}(2^d\epsilon^{-\frac{1}{2}}\log_2^{\frac{3}{2}(d-1)}(1/\epsilon)) = \Theta\left(\frac{(2\log_2^{\frac{3}{2}}(1/\epsilon))^d}{\epsilon^{\frac{1}{2}}\log_2^{\frac{3}{2}}(1/\epsilon))}\right), \tag{5.52}$$

$$dn(d) = \mathcal{O}\left(d^2 W\left(\frac{\epsilon^{-\frac{1}{d}}}{d}\log_2^{\frac{3}{2}}\left(\frac{1}{\epsilon}\right)\right)\right). \tag{5.53}$$

Since the following limit can be verified by a straightforward computation,

$$\lim_{d\to\infty} \frac{2d + n(d) + (d-1)\log_2 n(d)}{2d + n(d)} = 1,$$

the width of $U_{f,\epsilon}(\vec{x})$ is $\mathcal{O}(2d + n(d) + (d-1)\log_2 n(d)) = \Theta(2d + n(d))$. In particular, the width is $\mathcal{O}\left(2d + dW\left(\frac{\epsilon^{-\frac{1}{d}}}{d}\log_2^{\frac{3}{2}}\left(\frac{1}{\epsilon}\right)\right)\right)$. This completes the proof. □

Remark 5.3.7. All limits stated without proof in the argument for Proposition 5.3.6 can be readily verified using the power series expansion of the Lambert W function. These computations were performed using *Mathematica*.

We now treat the case $p \notin \{2, \infty\}$ separately. This distinction is necessary because the optimal sparse grid decomposition, discussed in the previous section, is obtained by solving the associated optimization problem with respect to the $L^p([0,1]^d)$ norms only for $p \in \{2, \infty\}$. Consequently, the sparse grid decomposition given in Equation (5.32) need not be optimal when $p \notin \{2, \infty\}$. Nevertheless, any function in $X^{2,p}([0,1]^d)$ for $p \notin \{2, \infty\}$ can still be expressed as

$$f(\vec{x}) = \sum_{\vec{\ell}} \sum_{\vec{i} \in \vec{I}_{\vec{\ell}}} v_{\vec{\ell},\vec{i}} \phi_{\vec{\ell},\vec{i}}(\vec{x}), \tag{5.54}$$

and, in practice, we may still use the sparse grid decomposition discussed to approximate any $f \in X^{2,p}([0,1]^d)$ with $p \notin \{2, \infty\}$ by $f_n^s(\vec{x})$.

Corollary 5.3.8. *Let $d \in \mathbb{N}$ and $\epsilon \in (0,1)$. For each $f \in X^{2,p}([0,1]^d)$ such that $2 < p < \infty$, there a exists a quantum circuit $U_{f,\epsilon}(\vec{x})$ (as in Proposition 5.3.6) such that, if a measurement corresponding to the Pauli σ_Z observable is performed only on the first qubit, then*

$$\langle 0 | U_{f,\epsilon}^\dagger(\vec{x}) \sigma_Z^{(1)} U_{f,\epsilon}(\vec{x}) | 0 \rangle = f_n^s(\vec{x}) \tag{5.55}$$

such that we have $\|f - f_n^s\|_{L^p([0,1]^d)} \leq \epsilon$. *The depth of* $U_{f,\epsilon}(\vec{x})$ *is*

$$\mathcal{O}\left(d^2(12\beta\log_2\beta)^\beta\alpha^\beta\epsilon^{-\frac{p}{2p-1}}\log_2^\beta\left(\frac{1}{\epsilon}\right)W\left(\frac{(6\beta\log_2\beta)^\alpha\alpha^\alpha\epsilon^{-\frac{p}{d(2p-1)}}\log_2^\alpha\left(\frac{1}{\epsilon}\right)}{d}\right)\right). \tag{5.56}$$

The width of $U_{f,\epsilon}(\vec{x})$ *is*

$$\mathcal{O}\left(2d + dW\left(\frac{(6\beta\log_2\beta)^\alpha\alpha^\alpha\epsilon^{-\frac{p}{d(2p-1)}}\log_2^\alpha\left(\frac{1}{\epsilon}\right)}{d}\right)\right), \tag{5.57}$$

where we have defined $\alpha = (3p-1)/(2p-1)$ *and* $\beta = \alpha(d-1)$.

Proof. A computation in [TZ22, Equation 6.3] shows that

$$\|f - f_n^s\|_{L^p([0,1]^d)} \leq (\log_2 N)^{\left(3-\frac{1}{p}\right)(d-1)}N^{-\left(2-\frac{1}{p}\right)}. \tag{5.58}$$

Following the derivation presented in [Fan+25, Page 8, below (3.2)], to guarantee an accuracy ϵ we need to choose N such that $N = \left\lceil(6\beta\log_2\beta)^\beta\alpha^\beta\epsilon^{-\frac{p}{2p-1}}\log_2^\beta\left(\frac{1}{\epsilon}\right)\right\rceil$, where $\alpha = (3p-1)/(2p-1)$ and $\beta = \alpha(d-1)$. The unitary operator is constructed as in Proposition 5.3.6. We now derive the complexity of the quantum circuit in this case. Using the definition of Lambert's W function and Equation (5.44), it suffices to choose

$$n = \mathcal{O}\left(dW\left(\frac{N^{\frac{1}{d-1}}}{d}\right)\right) = \mathcal{O}\left(dW\left(\frac{(6\beta\log_2\beta)^\alpha\alpha^\alpha\epsilon^{-\frac{p}{(d-1)(2p-1)}}\log_2^\alpha\left(\frac{1}{\epsilon}\right)}{d}\right)\right), \tag{5.59}$$

$$= \Theta\left(dW\left(\frac{(6\beta\log_2\beta)^\alpha\alpha^\alpha\epsilon^{-\frac{p}{d(2p-1)}}\log_2^\alpha\left(\frac{1}{\epsilon}\right)}{d}\right)\right). \tag{5.60}$$

The equation above provides an upper bound on n as a function of d. To highlight this rela-

tionship, we denote it as $n(d)$ as before. Once again, we have $n(d) + d\log_2 n(d) = \mathcal{O}(n(d))$. To see this, note that since $W(x) \leq x$ for $x \geq 0$, we obtain a slightly looser bound on n given by

$$n = \mathcal{O}\left(\frac{d}{d}(6\beta\log_2\beta)^\alpha\alpha^\alpha\epsilon^{-\frac{p}{d(2p-1)}}\log_2^\alpha\left(\frac{1}{\epsilon}\right)\right) = \Theta((d\log_2(\alpha d))^\alpha\epsilon^{-\frac{p}{d(2p-1)}}). \tag{5.61}$$

The final estimate follows since $\alpha^\alpha = \mathcal{O}(1)$ and $\log_2^\alpha(1/\epsilon) = \mathcal{O}(1)$ for fixed $\epsilon \in (0,1)$ and $2 \leq p \leq \infty$. The claim that $n(d) + d\log_2 n(d) = \mathcal{O}(n(d))$ now follows from the following computation:

$$\lim_{d\to\infty}\frac{n(d)+d\log_2 n(d)}{n(d)} = 1 + \lim_{d\to\infty}\frac{d\log_2((d\log_2(\alpha d))^\alpha\epsilon^{-\frac{p}{d(2p-1)}})}{(d\log_2(\alpha d))^\alpha\epsilon^{-\frac{p}{d(2p-1)}}} = \Theta(1). \tag{5.62}$$

Since the depth of $U_{f,\epsilon}(\vec{x})$ is $\mathcal{O}\left(2^d N d n(d)\right)$, an upper bound on the depth of $U_{f,\epsilon}(\vec{x})$ is therefore given by

$$\Theta\left(2^d d^2(6\beta\log_2\beta)^\beta\alpha^\beta\epsilon^{-\frac{p}{2p-1}}\log_2^\beta\left(\frac{1}{\epsilon}\right)W\left(\frac{(6\beta\log_2\beta)^\alpha\alpha^\alpha\epsilon^{-\frac{p}{d(2p-1)}}\log_2^\alpha\left(\frac{1}{\epsilon}\right)}{d}\right)\right). \tag{5.63}$$

Since $2^d = \mathcal{O}(2^{\alpha(d-1)}) = \mathcal{O}(2^\beta)$, we can absorb the factor 2^d into $(6\beta\log_2\beta)^\beta$ by replacing 2^d by 2^β, and instead use the following estimate:

$$\Theta\left(d^2(12\beta\log_2\beta)^\beta\alpha^\beta\epsilon^{-\frac{p}{2p-1}}\log_2^\beta\left(\frac{1}{\epsilon}\right)W\left(\frac{(6\beta\log_2\beta)^\alpha\alpha^\alpha\epsilon^{-\frac{p}{d(2p-1)}}\log_2^\alpha\left(\frac{1}{\epsilon}\right)}{d}\right)\right). \tag{5.64}$$

Since the width of $U_{f,\varepsilon}$ is $\mathcal{O}(2d + n(d) + (d-1)\log_2 n(d)) = \Theta(2d + n(d))$, an upper bound on

the width of $U_{f,\epsilon}(\vec{x})$ is therefore given by

$$\mathcal{O}\left(2d + dW\left(\frac{(6\beta\log_2\beta)^\alpha\alpha^\alpha\epsilon^{-\frac{p}{d(2p-1)}}\log_2^\alpha\left(\frac{1}{\epsilon}\right)}{d}\right)\right). \tag{5.65}$$

This completes the proof. □

5.4 Approximating Countably Parametric Holomorphic Functions

We begin by presenting background material on generalized polynomial chaos expansions. Subsequently, we introduce approximate rates of convergence for polynomial chaos expansions obtained via best n-term truncations of Taylor and Chebyshev polynomial expansions.

5.4.1 Domains of Holomorphy

Let $\mathscr{U} = [-1,1]^{\mathbb{N}}$. Approximation rate bounds for parametric maps $f : \mathscr{U} \to \mathbb{C}$ will rely on the holomorphic dependence of f on its argument $z \in \mathscr{U}$, as specified in the following standard definition.

Definition 5.4.1. Let $d \in \mathbb{N}\cup\{\infty\}$. For an open set $U \subseteq \mathbb{C}^d$, a map $f : U \to \mathbb{C}$ is holomorphic on U iff it is continuous and if it is separately holomorphic with respect to each argument z_j, i.e. for any fixed $z \in U$, and any $j \in \{1,...,d\}$, the limit $\lim_{h\to 0}\frac{f(x+he_j)-f(z)}{h}$ exists in $\mathbb{C}$. Let $\Omega \subseteq \mathbb{R}^d$ be an open set and let $U \subseteq \mathbb{C}^d$ be open with $\Omega \subseteq U$. A real analytic map $f : \Omega \to \mathbb{R}$ is said to admit a holomorphic extension to U if there exists a holomorphic function $\widetilde{f} : U \to \mathbb{C}$ such that $\widetilde{f}|_\Omega = f$.

Parametric solution families of linear elliptic or parabolic partial differential equations

admit holomorphic dependence on the parameters. The parametric holomorphic maps f considered here arise as so-called *quantities of interest*, for example, linear functionals of parametric solutions [SZ19b]. Approximation rate bounds that are free from the curse of dimensionality (in the case $d = \infty$) are derived from p-summability properties of sequences of generalized polynomial chaos expansion coefficients via Stechkin's lemma [Ste55]. Such summability properties were shown in [SZ19b; CDS11] to follow from order-explicit bounds on the generalized polynomial chaos coefficients. These bounds, in turn, are obtained via *quantified holomorphy* of the parametric maps on suitable closed Bernstein ellipses $\mathscr{E}_\rho \subseteq \mathbb{C}$ with foci at $z = \pm 1$ and semi-axis sum $\rho > 1$, i.e.,

$$\mathscr{E}_\rho = \left\{(z + z^{-1})/2 \mid z \in \mathbb{C},\ 1 \leq |z| \leq \rho\right\}. \tag{5.66}$$

Bounds on the Chebyshev–gPC coefficients in the expansion of $f(z)$ require multivariate extensions of the Bernstein ellipse $\mathscr{E}_\rho$. These are defined as Cartesian products of one-dimensional Bernstein ellipses $\mathscr{E}_{\rho_j} \subseteq \mathbb{C}$ associated with each coordinate $z_j \in \mathbb{C}$. For any $d > 1$, we define $\boldsymbol{\rho} := \prod_{1 \leq j \leq d} \rho_j$, where the product is understood as a countable product in the case $d = \infty$. We then define the closed polyellipse

$$\mathscr{E}_{\boldsymbol{\rho}} := \mathscr{E}_{\rho_1} \times \mathscr{E}_{\rho_2} \times \ldots \times \mathscr{E}_{\rho_d} \ \subseteq \mathbb{C}^d. \tag{5.67}$$

The set of all $f : \mathscr{E}_{\boldsymbol{\rho}} \to \mathbb{C}$ which are holomorphic in $\mathscr{E}_{\boldsymbol{\rho}}$ will be denoted by $\mathscr{H}(\boldsymbol{\rho})$, i.e.

$$\mathscr{H}(\boldsymbol{\rho}) := \{f : \mathscr{E}_{\boldsymbol{\rho}} \to \mathbb{C} \mid f \text{ holomorphic in } \mathscr{E}_{\boldsymbol{\rho}}, \|f\|_{L^\infty(\mathscr{E}_{\boldsymbol{\rho}})} < \infty\}. \tag{5.68}$$

For $d = \infty$, bounds on the gPC coefficient sequences of f require holomorphy on subsets of $\mathbb{C}^{\mathbb{N}}$

that are given by unions of all polyellipses $\mathscr{E}_{\boldsymbol{\rho}}$ for which $\boldsymbol{\rho}$ is $(\vec{b}, \varepsilon)$-admissible. Let $0 < p < 1$ and let $\vec{b} = (b_j)_{j \geq 1} \in (0, \infty)^{\mathbb{N}} \cap \ell^p(\mathbb{N})$. We then define the cylindric *ellipsoidal holomorphy domain*

$$\mathscr{E}(\vec{b}, \epsilon) := \bigcup \left\{ \mathscr{E}_{\boldsymbol{\rho}} \mid \rho_j \geq 1, \sum_{j \geq 1} b_j \left(\frac{\rho_j + 1/\rho_j}{2} - 1 \right) \leq \epsilon \right\}. \tag{5.69}$$

Similarly, given polyradius $\boldsymbol{\rho} \in [1, \infty)^{\mathbb{N}}$, and polydiscs $B_{\boldsymbol{\rho}} = B_{\rho_1} \times B_{\rho_2} \times ... \subseteq \mathbb{C}^{\mathbb{N}}$, define *polydisc holomorphy domains*

$$\mathscr{D}(\vec{b}, \epsilon) := \bigcup \left\{ B_{\boldsymbol{\rho}} | \ \rho_j \geq 1, \ \sum_{j \geq 1} b_j (\rho_j - 1) \leq \epsilon \right\}. \tag{5.70}$$

The conditions imposed on the polyradii $\boldsymbol{\rho}$ in Equation (5.69) and Equation (5.70) are referred to below, and in the references cited here, as $(\vec{b}, \varepsilon)$-*admissibility*. In Equation (5.69) and Equation (5.70), the dependence on $\varepsilon > 0$ could be absorbed into the sequence $\vec{b}$. However, for applications it is convenient to keep track of $\vec{b}$ and ε separately. We also consider the approximation of infinite-dimensional holomorphic functions in the following sense.

Definition 5.4.2. Let $p \in (0, 1)$ and let $\vec{b} \in \ell^p(\mathbb{N})$ be a non-negative sequence. A function $u : \mathscr{U} \to \mathbb{R}$ is called $(\vec{b}, p)$-*holomorphic* if it can be expressed as

$$u(\vec{y}) = U \left(\sum_{j \in \mathbb{N}} y_j \psi_j \right), \quad y \in \mathscr{U}, \tag{5.71}$$

where for some Banach space X, the map $U : X \to \mathbb{R}$ and the sequence $(\psi_j)_{j \in \mathbb{N}} \subseteq X$ satisfy the following conditions:

1. $\|\psi_j\|_X \leq b_j$ for all $j \in \mathbb{N}$,

2. There exists an open subset $O \subseteq X_{\mathbb{C}}$ of the complexified Banach space $X_{\mathbb{C}}$ such that

$$\left\{ \sum_{j \in \mathbb{N}} y_j \psi_j : \vec{y} \in \mathscr{U} \right\} \subseteq O \tag{5.72}$$

and such that $U : O \to \mathbb{C}$ is holomorphic.

Countably-parametric expressions such as Equation (5.71) which arise as functionals G of a parametric family of PDE solutions as in Equation (1.20), Equation (1.21) are known to allow *generalized polynomial chaos* (gPC) expansions (see, e.g., [Bac+17; Zec18] and references there).

5.4.2 Generalized Polynomial Chaos Bases

For gPC expansions, a range of univariate orthogonal polynomial systems in $[-1, 1]$ can be considered (e.g. [Zec18] and the references there). We discuss two of these which are particular amenable to representations by PQCs: first, the so-called Taylor gPC comprising multivariate expansions into monomials $\{\vec{y}^{\vec{\nu}} : \vec{\nu} \in \mathscr{F}\}$ (see [SZ19b; CDS11]), and second, the so-called Chebyshev-gPC expansions which are based on the system $\{T_j\}_{j \geq 0}$ of Chebyshev polynomials of the first kind.

Taylor-gPC Expansions

Our main application is to Chebyshev–gPC expansions; however, for completeness, we also discuss Taylor–gPC expansions. Taylor–gPC expansions for parametric, holomorphic solution families have been employed, for example, in [CCS15; Chk+13] and [SZ19b], where neural network approximations of such expansions were constructed. In this chapter, we build quan-

tum circuits implementing Taylor–gPC expansions, using the results of [Yu+24]. Taylor-gPC expansions are formal expressions of the type

$$u(\vec{y}) = \sum_{\vec{\nu}\in\mathscr{F}} t_{\vec{\nu}} \vec{y}^{\vec{\nu}} \ , \quad \vec{y} \in \mathscr{U} \ , \tag{5.73}$$

where the Taylor gPC coefficients $t_{\vec{\nu}}$ of $u : \vec{y} \to \mathbb{R}$ are given by $t_{\vec{\nu}} := \frac{1}{\vec{\nu}!}(\partial_{\vec{y}}^{\vec{\nu}} u)(\vec{y}) \ |_{\vec{y}=0} \in \mathbb{R}$. References [SZ19b, Proposition 2.3] and [CDS11; CCS13] state that for parametric maps $u : \mathscr{U} \to \mathbb{R}$ which admit a holomorphic extension to the polydisc $\mathscr{D}(\boldsymbol{\rho}, \epsilon)$ defined in Equation (5.70) can be represented in the form Equation (5.73) with the Taylor-gPC expansion Equation (5.73) converging unconditionally for all $\vec{y} \in \mathscr{U}$. The following result on the convergence rate of n-term truncations of Taylor–gPC expansions for $(\vec{b}, \varepsilon)$-holomorphic parametric functions is proved in [SZ19b, Theorem 2.7].

Proposition 5.4.3 (Theorem 2.7 in [SZ19b])**.** *Let $u : \mathscr{U} \to \mathbb{R}$ be $(\vec{b}, \epsilon)$-holomorphic for some sequence $\vec{b} \in (0,\infty)^{\mathbb{N}} \cap \ell^p(\mathbb{N})$ for some $0 < p \leq 1$. Then the sequence $(t_{\vec{\nu}})_{\vec{\nu}\in\mathscr{F}} \in \ell^p(\mathscr{F})$ and there exists a constant $C > 0$ and a sequence $\{\Lambda_n\}_{n\geq 1}$ of nested, d.c. index sets $\Lambda_n \subseteq \mathscr{F}$ such that for all $n \in \mathbb{N}$ holds $\#(\Lambda_n) \leq n$, and the parametric function u in Equation (5.73) satisfies*

$$\sup_{\vec{y}\in\mathscr{U}} \left| u(\vec{y}) - \sum_{\vec{\nu}\in\Lambda_n} t_{\vec{\nu}} \vec{y}^{\vec{\nu}} \right| \leq \sum_{\vec{\nu}\in\mathscr{F}\setminus\Lambda_n} |t_{\vec{\nu}}| \leq C n^{-(1/p-1)} \ , \tag{5.74}$$

and $\max_{\vec{\nu}\in\Lambda_n} \|\vec{\nu}\|_1 \leq C(1+\log(n))$. Furthermore, if $\vec{e}_j \in \Lambda_n$ for some $j \in \mathbb{N}$, then also $\vec{e}_i \in \Lambda_n$ for all $i = 1, 2, ..., j$.

Chebyshev-gPC Expansions

In the remainder of this chapter, we denote by $\{T_j\}_{j\geq 0}$ the sequence of Chebyshev polynomials on $[-1,1]$, assumed to be orthonormal with respect to the Chebyshev probability measure $\varrho^{(1)}$. This definition differs from that of the classical Chebyshev polynomials, $\tilde{T}_j$, which are normalized according to $\tilde{T}_k(1) = 1$ for $k \geq 0$. We have $\sup_{|x|\leq 1} |\tilde{T}_k(x)| = 1$, $T_0 = \pi^{-1/2}\tilde{T}_0$ and $T_k = (2/\pi)^{1/2}\tilde{T}_k$ for $k \geq 1$. For $d \geq 1$ and for $\vec{\nu} \in \mathscr{F}$, we define the tensor-product Chebyshev polynomials

$$T_{\vec{\nu}}(\vec{y}) := \prod_{j\geq 1} T_{\nu_j}(y_j) \ , \quad \vec{y} = (y_j)_{j\geq 1} \in \mathscr{U} \ . \tag{5.75}$$

Observe that, since $\vec{\nu} \in \mathscr{F}$, the products in Equation (5.75) contain only a finite number $|\vec{\nu}|_0$ of nontrivial terms, and for all $\vec{\nu} \in \mathscr{F}$ we have $\sup_{\vec{y}\in\mathscr{U}} |T_{\vec{\nu}}(\vec{y})| \leq (2/\pi)^{\|\vec{\nu}\|_0/2} < 1$. It is clear that $\{T_{\vec{\nu}}\}_{\vec{\nu}\in\mathscr{F}}$ forms an orthonormal basis of $L^2_\varrho(\mathscr{U};\mathbb{R})$, and that there is an isometric isomorphism between $L^2_\varrho(\mathscr{U})$ and $\ell^2(\mathscr{F})$. For a finite index set $\emptyset \neq S \subseteq \mathscr{F}$, define the finite-dimensional polynomial space

$$\mathscr{P}_S = \operatorname{span}\{T_{\vec{\nu}} \mid \vec{\nu} \in S\} \subseteq L^2_\varrho(\mathscr{U}). \tag{5.76}$$

A best n-term L^2_ϱ Chebyshev approximation of $f \in L^2_\varrho(\mathscr{U};\mathbb{R})$ is given by

$$f_n := \operatorname{argmin}\left\{\|f-g\|_{L^2_\varrho(\mathscr{U})} : g \in \mathscr{P}_S, S \subseteq \mathscr{F}, \#(S) \leq n\right\}. \tag{5.77}$$

An explicit best n-term approximation f_n is given by $f_n = \sum_{\vec{\nu}\in S^*} c_{\vec{\nu}} T_{\vec{\nu}}$, where $S^* \subseteq \mathscr{F}$ is a subset with $\#(S^*) \leq n$ that contains multi-indices of n largest coefficients $c_{\vec{\nu}}$ of the Chebyshev-gPC

expansion of f. We have

$$\|f - f_n\|^2_{L^2_\varrho(\mathscr{U})} = \sum_{\mathscr{F}\setminus S^*} |c_{\vec{\nu}}|^2 = \|\vec{c}\|^2_{\ell^2(\mathscr{F}\setminus S^*)}. \tag{5.78}$$

Remark 5.4.4. Apparently, the polynomial space $\mathscr{P}_S$ in Equation (5.76) depends on the chosen basis used to define the span. When S is a downward closed set, however, $\mathscr{P}_S$ is independent of the choice of univariate basis. We omit the details.

We are primarily interested in convergence rates of finite-parametric truncations of Chebyshev–gPC expansions whose coefficient sequences $\vec{c} \in \ell^p(\mathscr{F})$ for some $0 < p < 2$. The following approximation rate bounds for best n-term Chebyshev polynomial approximations of certain *holomorphic* functions $f \in L^2_\varrho(\mathscr{U})$ hold true. Each of these results will be shown to admit a quantum analogue via suitable quantum circuits. The next theorem is a special case of [Zec18, Theorem 2.2.10]. Similar and related results can be found in earlier publications [CDS11; CCS13; CDS11; CCS15].

Theorem 5.4.5 (Infinite-Dimensional Case)**.** *Assume $d = \infty$ and that $f \in \mathscr{H}(\vec{b}, p, \epsilon)$ for some $0 < p \leq 1$, $\epsilon > 0$ and $\vec{b} \in (0,\infty)^{\mathbb{N}} \cap \ell^p(\mathbb{N})$. Then, for every $n \in \mathbb{N}$ there exists a d.c index set $\Lambda_n \subseteq \mathscr{F}$ of cardinality n such that*

$$\max_{\vec{\nu}\in\Lambda_n} \|\vec{\nu}\|_1 \leq C(1 + \log(n)) \tag{5.79}$$

and with the truncated Chebyshev expansion $f^{\text{Cheb}}_{\Lambda_n} := \sum_{\vec{\nu}\in\Lambda_n} c_{\vec{\nu}} T_{\vec{\nu}}$ it holds

$$\|f - f^{\text{Cheb}}_{\Lambda_n}\|_{L^2_\varrho(\mathscr{U})} \leq C(f) n^{-1/p+1/2}, \quad \|f - f^{\text{Cheb}}_{\Lambda_n}\|_{L^\infty(\mathscr{U})} \leq C(f) n^{-1/p+1}, \tag{5.80}$$

for some constant $C(f)$.

Proof. Throughout this proof fix $\tau > 0$, such that for all $\vec{\nu} \in \mathscr{F}$ holds

$$\|T_{\vec{\nu}}\|_{L^\infty(\mathscr{U})} \leq \prod_{i \in \mathbb{N}} (1 + \nu_i)^\tau =: w_{\vec{\nu}}. \tag{5.81}$$

for all $\vec{\nu} \in \mathscr{F}$. See for instance [Zec18, Appendix B.2.1]. According to [Zec18, Theorem 2.2.10 (i)-(iii)] (in the Chebyshev polynomial case, i.e., with $\alpha = \beta = -1/2$), there exists a sequence $(a_{\vec{\nu}})_{\vec{\nu} \in \mathscr{F}}$ of non-negative real numbers such that

$$w_{\vec{\nu}} |c_{\vec{\nu}}| \leq a_{\vec{\nu}} \tag{5.82}$$

for all $\vec{\nu} \in \mathscr{F}$. Note that $(a_{\vec{\nu}})_{\vec{\nu} \in \mathscr{F}} \in \ell^p(\mathscr{F})$, and $(a_{\vec{\nu}})_{\vec{\nu} \in \mathscr{F}}$ is monotonically decreasing in the sense that $a_{\vec{\nu}} \geq a_{\vec{\mu}}$ whenever $\vec{\nu} \leq \vec{\mu}$. Let $(\vec{\nu}_i)_{i \in \mathbb{N}}$ be an arbitrary enumeration of $\mathscr{F}$ such that $(a_{\vec{\nu}_i})_{i \in \mathbb{N}}$ is monotonically decreasing and $\vec{\nu}_i \leq \vec{\mu}_j$ implies $i \leq j$. Such an enumeration exists due to the above stated properties. For each $n \in \mathbb{N}$, $\Lambda_n := \{\vec{\nu}_i : i \leq n\} \subseteq \mathscr{F}$ defines a d.c. set of cardinality n such that $(a_{\vec{\nu}_i})_{i=1}^n$ corresponds to n largest numbers of the sequence $(a_{\vec{\nu}})_{\vec{\nu} \in \mathscr{F}}$. It remains to verify the error bounds Equation (5.80), which are a consequence of Stechkin's lemma [Ste55]. We start with $L^2_\varrho(\mathscr{U})$. Using Parseval's identity

$$\|f - f^{\mathrm{Cheb}}_{\Lambda_n}\|^2_{L^2_\varrho(\mathscr{U})} = \left\| \sum_{\vec{\nu} \in \mathscr{F}} c_{\vec{\nu}} T_{\vec{\nu}} - \sum_{\vec{\nu} \in \Lambda_n} c_{\vec{\nu}} T_{\vec{\nu}} \right\|^2_{L^2_\varrho(\mathscr{U})} = \sum_{\vec{\nu} \in \mathscr{F} \setminus \Lambda_n} c^2_{\vec{\nu}} \leq \sum_{i > n} a^2_{\vec{\nu}_i}. \tag{5.83}$$

As the sequence $(a_{\vec{\nu}_i}) \in \ell^p(\mathbb{N})$ is monotonically decreasing, for all $j \in \mathbb{N}$ we have

$$a_{\vec{\nu}_j}^p \leq \frac{1}{j} \sum_{i=1}^{j} a_{\vec{\nu}_i}^p \leq \frac{1}{j} \sum_{i \in \mathbb{N}} a_{\vec{\nu}_i}^p \quad \text{so that} \quad a_{\vec{\nu}_j} \leq C j^{-1/p}. \tag{5.84}$$

Hence, $a_{\vec{\nu}_j} \leq C j^{-1/p}$, where $C := \|(a_{\vec{\nu}})_{\vec{\nu} \in \mathscr{F}}\|_{\ell^p} < \infty$. Therefore, $\sum_{i>n} a_{\vec{\nu}_i}^2 \leq C^2 \sum_{i>n} i^{-2/p} \lesssim n^{-2/p+1}$, which, together with Equation (5.83), gives the desired bound in $L^2_\varrho(\mathscr{U})$. For the $L^\infty(\mathscr{U})$ bound we proceed in a similar fashion. Using Equation (5.81) and Equation (5.82), we have

$$\|f - f_{\Lambda_n}^{\mathrm{Cheb}}\|_{L^\infty(\mathscr{U})} = \left\| \sum_{\vec{\nu} \in \mathscr{F}} c_{\vec{\nu}} T_{\vec{\nu}} - \sum_{\vec{\nu} \in \Lambda_n} c_{\vec{\nu}} T_{\vec{\nu}} \right\|_{L^\infty(\mathscr{U})} \leq \sum_{\vec{\nu} \in \mathscr{F} \setminus \Lambda_n} |c_{\vec{\nu}}| \|T_{\vec{\nu}}\|_{L^\infty(\mathscr{U})} \leq \sum_{i>n} a_{\vec{\nu}_i}. \tag{5.85}$$

Together with Equation (5.84) this gives

$$\|f - f_{\Lambda_n}^{\mathrm{Cheb}}\|_{L^\infty(\mathscr{U})} \leq C \sum_{i>n} i^{-1/p} \lesssim n^{-1/p+1}. \tag{5.86}$$

This completes the proof. □

The point of Theorem 5.4.5 is that, formally, even though the function f may depend on a countable number of coordinates $\vec{y} = (y_j)_{j \geq 1}$, the approximation rate bounds Equation (5.80) are independent of the number of "active" coordinates contributing to the n-term approximants f_{Λ_n}. The approximation rate bounds Equation (5.80) depend only on the sparsity of the coefficient sequence $\vec{c}$, as quantified by the summability exponent $p \in (0, 1)$. In this sense, the convergence rate bounds Equation (5.80) do not suffer from the *curse of dimensionality*. Theorem Theorem 5.4.5 is formulated for $d = \infty$. It is classical that, for $1 \leq d < \infty$ and $f \in \mathscr{H}(\boldsymbol{\rho})$,

suitably truncated n-term Chebyshev expansions converge exponentially; see [TWZ17; OSZ22]. We state the result below without proof.

Theorem 5.4.6 (Finite-Dimensional Case)**.** *Let $1 \leq d < \infty$. Let $f \in \mathscr{H}(\boldsymbol{\rho})$ for some $\boldsymbol{\rho} \in (1,\infty)^d$ and let $\vec{c} = (c_{\vec{\nu}})_{\vec{\nu}\in\mathbb{N}_0^d} \in \ell^2(\mathbb{N}_0^d)$ denote the sequence of Chebyshev coefficients of f. Then, there exists a constant $C > 0$ (depending on d and on f) and a constant $\gamma > 0$ (depending on $\boldsymbol{\rho}$) such that, for every $n \in \mathbb{N}$, there exists a d.c. set $\Lambda_n \subseteq \mathbb{N}_0^d$ with $\#(\Lambda_n) \leq n$ such that*

$$\|f - f_{\Lambda_n}^{\mathrm{Cheb}}\|_{L^2_\varrho(\mathscr{U})} \leq \|f - f_{\Lambda_n}^{\mathrm{Cheb}}\|_{L^\infty_\varrho(\mathscr{U})} \leq Ce^{-\gamma n^{1/d}}. \tag{5.87}$$

The index sets $\Lambda_n \subseteq \mathbb{N}_0^d$ in Equation (5.87) can be chosen as product sets

$$\Lambda_n := \{\vec{\nu} \in \mathbb{N}_0^d : \|\vec{\nu}\|_\infty \leq k\} = \{0,1,2,...,k\}^d \tag{5.88}$$

with $n = \#(\Lambda_n) = \mathcal{O}(k^d)$. The index sets Λ_n are nested, and are d.c. The approximation $f_\Lambda^{\mathrm{Cheb}}$ is the tensor product projection of f onto $\mathbb{Q}_k^d = \mathbb{P}_k^{\otimes d}$.

5.4.3 Approximation Rates

Using Theorem 5.4.5 and Theorem 5.4.6, we derive quantum circuit representations for n-term truncated Chebyshev polynomial expansions. With n-term generalized polynomial chaos (gPC) approximations available for a narrow class of *multivariate, holomorphic functions* of Chebyshev type, we construct quantum circuits by leveraging the QSP and LCU algorithms. The core of the ensuing existence and approximation rate results is the following multivariate version of Lemma 5.2.1, which follows along the lines of [Yu+24, Appendix B].

Lemma 5.4.7. *Let $\vec{\nu} \in \mathscr{F}$ such that $d = \|\vec{\nu}\|_0 < \infty$. WLOG, let* $\mathrm{supp}(\vec{\nu}) = \{1, \cdots, d\}$. *Consider the d-variate product Chebyshev polynomial $T_{\vec{\nu}}(\vec{y}) = \prod_{j=1}^{d} T_{\nu_j}(y_j)$. There exists a quantum circuit $W_{\vec{\nu}}$ such that*

$$T_{\vec{\nu}}(\vec{y}) = \langle 0| W_{\vec{\nu}}^{\dagger}(\vec{y}) \sigma_Z^{(1)} W_{\vec{\nu}}(\vec{y}) |0\rangle \quad \text{for all} \quad \vec{y} \in [-1,1]^d \,. \tag{5.89}$$

for all $y \in \mathscr{U}$. The width of $W_{\vec{\nu}}$ is the number of variables $d = \|\vec{\nu}\|_0$. The depth of $W_{\vec{\nu}}$ is $\max\{2\nu_1 + 1, \ldots, 2\nu_d + 1\} = 2\|\vec{\nu}\|_\infty + 1$. *The size of $W_{\vec{\nu}}$ is $\|\vec{\nu}\|_1 + d$.*

Proof. Lemma 5.2.1 implies that for each $r \in \mathbb{N}_0$, there exists a quantum U_r such that $\langle 0| U_r(y) |0\rangle = T_r(y)$ for all $y \in [-1,1]$. The width of each $U_r(y)$ is 1, the depth is $2r+1$, and the number of pre-determined parameters is $r+1$. For $\vec{\nu} \in \mathscr{F}$, consider the unitary operator $U_{\vec{\nu}}(\vec{y}) = \bigotimes_{j \geq 1} U_{\nu_j}(y_j)$. Note that, since $U_0 \equiv 1$, with $\mathrm{supp}(\vec{\nu}) = \{1, \cdots, d\}$, we have

$$\langle 0|^{\otimes d} U_{\vec{\nu}}(\vec{y}) |0\rangle^{\otimes d} = \prod_{j=1}^{d} \langle 0| U_{\nu_j}(y_j) |0\rangle = \prod_{j=1}^{d} T_{\nu_j}(y_j) = T_{\vec{\nu}}(\vec{y}). \tag{5.90}$$

The width of $U_{\vec{\nu}}(\vec{y})$ is $\|\vec{\nu}\|_0 \leq d$. Since U_{ν_j} has depth $2\nu_j + 1$, the PQC $U_{\vec{\nu}}$ has depth $\max\{2\nu_1 + 1, \cdots, 2\nu_d + 1\} = 2\|\vec{\nu}\|_\infty + 1$. Since each U_{ν_j} has a total of $\nu_j + 1$ pre-determined parameters, the PQC $U_{\vec{\nu}}(\vec{y})$ has a total of $\nu_1 + \cdots \nu_d + d = \|\vec{\nu}\|_1 + d$ pre-determined parameters. □

We can now establish approximation rates for finite-parametric, holomorphic maps in the setting of Theorem 5.4.6; that is, for a parameter domain $\mathscr{U}$ of finite dimension d.

Theorem 5.4.8. *Assume $1 \leq d < \infty$ and $\mathscr{U} = [-1,1]^d$. Let $f \in \mathscr{H}(\boldsymbol{\rho})$ for some $\boldsymbol{\rho} \in (1,\infty)^d$. Then, there are constants $C, \gamma > 0$ (depending on f and d) such that, for every $n \in \mathbb{N}$, exists a*

quantum circuit, U_n, such that

$$\|f - \langle 0| U_n^\dagger \sigma_Z^{(1)} U_n |0\rangle\|_{L^\infty(\mathscr{U})} \leq e^{-\gamma n^{1/d}}. \tag{5.91}$$

Moreover, the quantum circuit is such that has width $\mathrm{width}(U_n) = \mathcal{O}(d + \log_2(n))$ *and depth* $\mathrm{depth}(U_n) = \mathcal{O}((n^{1/d} + n) d \log_2(n))$.

Proof. The assumption $d < \infty$ and $f \in \mathscr{H}(\boldsymbol{\rho})$ allows to invoke Theorem 5.4.6. Let $f_{\Lambda_n}^{\mathrm{Cheb}} = \sum_{\vec{\nu} \in \Lambda_n} c_{\vec{\nu}} T_{\vec{\nu}}$. U_n is constructed as in Proposition 5.2.3. Given that in Theorem Theorem 5.4.6 we have $\Lambda_n = \{\vec{\nu} \in \mathbb{N}_0^d : |\vec{\nu}|_\infty \leq k\}$ (tensor products of degree k univariate Chebyshev projections) with $n = \mathcal{O}(k^d)$, there are $M = n$ terms in $f_{\Lambda_n}^{\mathrm{Cheb}}$ with respect to some enumeration $j : \{1, \cdots, n\} \to \Lambda_n$ of the multi-index set Λ_n. The estimate of $\mathrm{width}(U_n)$ follows at once from Proposition 5.2.3. Similarly, the estimate of $\mathrm{depth}(U_n)$ from Proposition 5.2.3 is $\mathcal{O}((2\|\vec{v}\|_1 + dn) \log_2(n))$. We observe that the tensor-product index sets $\Lambda_n \subseteq \mathbb{N}_0^d$ have, being d.c.,

$$\max_{\vec{\nu} \in \Lambda_n} \|\vec{\nu}\|_1 = \max_{\vec{\nu} \in \partial\Lambda_n} \|\vec{\nu}\|_1 = \|(k, k, ..., k)\|_1 = dk = \mathcal{O}(dn^{1/d}). \tag{5.92}$$

Hence, we have $\mathcal{O}((2\|\vec{v}\|_1 + dn) \log_2(n)) = \mathcal{O}((n^{1/d} + n) d \log_2(n))$. This completes the proof. □

We now consider the case $d = \infty$. The proof strategy used to establish approximation rate bounds is similar to that employed in the proof of Theorem 5.4.8. We show that only circuit depths logarithmic in n are required, due to Equation (5.79). Since parametric–holomorphic maps arise in a wide range of PDE applications with high-dimensional parametric input, this result is relevant for many applications.

Theorem 5.4.9. *Assume given $f : \mathscr{U} \to \mathbb{R}$ in $\mathscr{H}(\vec{b}, p, \epsilon)$ with some $0 < p \leq 1$ and such that*

$\sup_{\vec{y}\in[-1,1]^{\mathbb{N}}} |f(\vec{y})| \leq 1$. *Then, for every* $n \in \mathbb{N}$ *there exists a quantum circuit,* U_n*, such that the error bounds*

$$\|f - \langle 0| U_n^\dagger \sigma_Z^{(0)} U_n |0\rangle\|_{L^2_\varrho(\mathscr{U})} \leq C(f) n^{-1/p+1/2}, \tag{5.93}$$

$$\|f - \langle 0| U_n^\dagger \sigma_Z^{(1)} U_n |0\rangle\|_{L^\infty(\mathscr{U})} \leq C(f) n^{-1/p+1}. \tag{5.94}$$

hold. Moreover, the quantum circuit has width $\mathcal{O}(\log_2 n)$ *and depth* $\mathcal{O}(n \log_2^2(n))$.

Proof. The assumptions imply that Theorem 5.4.5 and the error bounds Equation (5.80) hold. Let $f_{\Lambda_n}^{\text{Cheb}} = \sum_{\vec{\nu}\in\Lambda_n} c_{\vec{\nu}} T_{\vec{\nu}}$. Using Equation (5.79), there exists a constant $C > 0$ such that $\max_{\vec{\nu}\in\Lambda_n} \|\vec{\nu}\|_1 \leq C(1 + \log_2(n))$. Let $d := \|\vec{v}\|_1 = C(1 + \log_2(n))$. U_n is constructed as in Proposition 5.2.3. The estimate of width(U_n) follows at once from Proposition 5.2.3. Similarly, the estimate of depth(U_n) from Proposition 5.2.3 is $\mathcal{O}((2+n)d \log_2(n)) = \mathcal{O}(n \log_2^2(n))$. This completes the proof. □

Approximation rates based on Taylor–gPC expansions are derived in a similar manner. We omit the details and refer the reader to [Aft+25] for relevant details.

5.5 Conclusion

In this chapter, we have used the QSP and LCU algorithms to construct quantum circuits that approximate Korobov functions and countably parameterized holomorphic functions, thereby establishing a rigorous theoretical foundation for the quantum approximation of a broad class of multivariate functions. These results advance the emerging field of quantum neural network approximation theory and contribute to research at the interface of quantum computing

and scientific computing. We conclude by outlining several natural directions for future work.

An important direction for future research is to investigate whether parameterized quantum circuits are capable of approximating functions belonging to other function spaces, including Besov spaces [Liu+21b], Bochner spaces [AG22], and Fréchet spaces [BDG23].

For $p \notin \{2, \infty\}$, an optimal sparse-grid decomposition of Korobov functions with respect to the L^p norm has not yet been established. A characterization of such a decomposition may yield tighter complexity estimates than those presented in Corollary 5.3.8. Moreover, the sparse grid framework is not limited to the hierarchical basis employed in this work. The adoption of alternative multiscale bases, such as wavelets, may facilitate quantum circuit constructions for approximating functions defined on a broader class of function spaces.

The present analysis is restricted to the space $X^{r,p}([0,1]^d)$ with $r = 2$, although extensions to the case $r \geq 3$ appear feasible. For example, [BG04, Section 4.2] considers higher-degree polynomial bases, including Lagrange polynomials, in the regime $r \geq 4$. Reference [LZ25] applies this approach to two-dimensional convolutional neural networks for the approximation of functions in $X^{r,p}([0,1]^d)$. When Chebyshev nodes are employed, Lagrange polynomials admit an expansion in terms of Chebyshev polynomials, allowing the results of Section 5.2 to be leveraged. This approach, however, requires the use of a non-uniform sparse grid, and a rigorous error analysis of the resulting approximation scheme remains an open problem in approximation theory.

The analysis has been confined to PQC emulations of real-valued, parametric holomorphic maps. It is plausible that regularity assumptions weaker than holomorphy may suffice to obtain analogous results (see, e.g., [HSS24]). Another natural direction for further investigation is the emulation, via PQCs, of parametric maps whose codomains are subsets of separable Hilbert or

Banach spaces. In this setting, PQCs corresponding to appropriate space discretizations must be developed. The resulting quantum circuit representations of PDE solutions constitute an active area of current research, and progress in this direction may be integrated with the results developed in the present work.

All results in this chapter are based on parameterized quantum circuit emulations of finite tensor products of univariate polynomials. In the univariate setting, recent progress on so-called "Infinite QSP" [Ale+24; Don+24; AMT24] has established a close connection between the summability properties of Chebyshev coefficient sequences and the corresponding sequences of (reduced) phase angles in parameterized quantum circuits. A natural direction for future research is to examine the sparsity of (arrays of) phase angles arising in the approximations considered here as a function of the summability of the Chebyshev–gPC coefficient sequences. Such summability properties govern both sparsity and n-term approximation rates of finitely truncated gPC expansions.

Several alternative formulations of the LCU algorithm have been proposed in the literature. For example, [Cha24] presents approaches tailored to intermediate-term quantum computing architectures. A systematic investigation of these formulations may lead to the development of novel quantum circuits for the approximation of d-dimensional functions across a range of function spaces.

We conclude by remarking that the objective of quantum neural network approximation theory is to characterize the fundamental expressive power and inherent limitations of parameterized quantum circuits for function approximation, independently of training procedures or data-related considerations. While this chapter focuses on approximation-theoretic aspects, a comprehensive evaluation of practical performance must also account for factors such as quan-

tum state preparation, data requirements in learning-based settings, parameter optimization, and measurement fidelity. Addressing these complementary issues constitutes an important direction for future research.

Chapter 6: Non-Uniform Quantum Fourier Transform

The results presented in this chapter are based on joint work with Yuehaw Khoo and Haizhao Yang [AKY26]. The results have been included with their permission. All aspects of the work were carried out by Junaid Aftab. Haizhao Yang, together with his collaborator Yuehaw Khoo, acted as mentors for this project. The code generated for parts of this chapter is available in [Aft26].

6.1 Introduction

The Discrete Fourier Transform (DFT), introduced by Gauss and later popularized by Cooley and Tukey [CT65], provides a framework for computing Fourier coefficients from uniformly sampled data and underpins applications in signal processing, data compression, and numerical solutions of differential equations (see, e.g., [TJ18; STW11; BH95]). In practice, however, uniform sampling is often infeasible due to physical constraints, adaptive sampling, or irregular geometries. The Non-Uniform Discrete Fourier Transform (NUDFT) [BM12] generalizes the DFT to arbitrary sampling grids, enabling spectral analysis on non-uniform data and irregular computational domains, with important applications in imaging and numerical methods (see, e.g., [QCT13; Liu+00]). Consequently, a broad class of efficient classical algorithms for computing the NUDFT has been developed (see, e.g., [DR93; War98; PST01; FS03; GL04;

LG05; RT18; WEB25; LL25]).

The Quantum Fourier Transform (QFT) (see, e.g., [Cop02; NC02; Lin22]) is the quantum analogue of the DFT and a central primitive in quantum algorithm design. It appears as a core subroutine in many quantum algorithms, including Shor's factoring algorithm [Sho99], Kitaev's quantum phase estimation [Kit95], quantum linear systems algorithms [HHL09], ground-state energy estimation [AL99], and quantum arithmetic [Dra00; RG17]. Recent work has explored several extensions of the QFT, such as multi-dimensional extensions, approximate implementations with reduced gate complexity, topology-aware variants adapted to specific hardware architectures, and fault-tolerant constructions for error-corrected quantum computation (see, e.g., [Pfe23; NSM20; BSW25; Jin+24; Kla+24; Got14]). In contrast, quantum algorithms for Fourier transforms on non-uniform grids, particularly in a fully quantum setting, remain relatively underexplored.

In this work, we propose a quantum algorithm for the NUDFT, termed the Non-Uniform Quantum Fourier Transform (NUQFT). The construction builds on the low-rank matrix factorization framework of Antolín and Townsend [RT18] and translates this decomposition into an explicit quantum algorithm. The factorization is implemented using Quantum Signal Processing (QSP) [LC17a; Gil+19a] together with the Linear Combination of Unitaries (LCU) [CW12b] algorithms. This yields an explicit block encoding [Gil+19a] of the NUDFT matrix to arbitrary precision, with controlled approximation errors stemming from both the classical low-rank truncation and its quantum realization. Finally, we provide explicit gate-level resource estimates for the resulting circuits under standard oracle access assumptions for non-uniform sampling points.

6.1.1 Related Works

In contrast to the extensive literature on the QFT, little work has addressed quantum implementations of Fourier transforms on non-uniform grids. To the best of our knowledge, the closest related proposal is due to Gyongyosi and Imre [GI12]. That work introduces a method termed "Quantum-SVD," in which a standard QFT on a uniform grid is followed by a classical singular value decomposition–based interpolation step to approximate Fourier coefficients at non-uniform sampling points. As a result, the procedure is hybrid: the Fourier transform is quantum, while the handling of non-uniformity is performed classically. By contrast, the approach developed here provides a fully quantum implementation based on a block-encoding framework. Our work can also be placed in the broader context of efforts to develop circuit-level quantum analogues of classical integral transforms, including the quantum wavelet transform [FW98; BA24], quantum Laplace transform [Zyl24; Sin+25], quantum Hilbert transform [JP25; ZL26], quantum cosine transform [KR01; GGA25], quantum Chebyshev transform [Wil+23], and quantum Hermite transform [Jai+25].

6.1.2 Organization

The remainder of this chapter is organized as follows. Section 6.2 presents a classical algorithm for the non-uniform discrete Fourier transform. Section 6.3 introduces our quantum algorithm, and Section 6.4 provides the corresponding error analysis. Section 6.5 presents some numerical results. Finally, Section 6.6 offers concluding remarks and outlines directions for future research.

6.2 Classical Algorithm for Non-Uniform Discrete Fourier Transform

In this section, we review a classical algorithm for the nonuniform discrete Fourier transform based on a low-rank matrix factorization approach, as presented in [RT18].

6.2.1 Type–II NUDFT

We first present the algorithm for the Type–II NUDFT, which assumes uniform frequency points given by $\omega_k = k$ for $k = 0, \ldots, N-1$.

6.2.1.1 Perturbed Equispaced Grid Setting

We begin by describing the algorithm in the special case where the nonuniform grid points are obtained by perturbing an equispaced grid. WLOG, let $\{t_j\}_{j=0}^{N-1} \subseteq [0,1)$ such that[1]

$$\left| t_j - \frac{j}{N} \right| \leq \frac{\gamma}{N}, \tag{6.1}$$

for some $\gamma \in [0, 1/2]$. The simple but important observation is that the entries of F_{II} can be written as

$$(F_{\mathrm{II}})_{jk} = e^{-2\pi i t_j k} = e^{-2\pi i (t_j - \frac{j}{N})k} e^{-2\pi i \frac{jk}{N}}, \tag{6.2}$$

for $0 \leq j, k \leq N-1$. This observation leads to the matrix decomposition $F_{\mathrm{II}} = A \circ F$, where F is the DFT matrix, and the entries of A are given by $A_{jk} = e^{-2\pi i (t_j - j/N)k}$. The matrix A is defined by sampling the bivariate function $(x, y) \mapsto e^{-2\pi i xy}$ over the domain $[-\gamma/N, \gamma/N] \times [0, N]$. As shown in [RT18], the construction of a low-rank factorization of A is based on

[1] We can compensate for this by introducing a factor of 2π in the complex exponential.

approximating this function by a bivariate polynomial of degree $(K-1)$, which can then be extended to obtain a rank-K approximation of A. This polynomial approximation is derived from a truncated Chebyshev expansion of e^{-ixy} [RT18, Appendix A]. As a result, the resulting rank-K approximation of A takes the form

$$A_K = \sum_{r=0}^{K-1}{}' \underbrace{\left[\sum_{q=0}^{K-1}{}' \alpha_{qr} \left(\exp\!\big(-i\pi N(\vec{t}-\vec{e})\big) \circ T_p\left(\frac{N(\vec{t}-\vec{e})}{\gamma}\right)\right)\right]}_{\vec{u}_r} \underbrace{T_r\left(\frac{2\vec{\omega}^\top}{N} - \vec{1}^\top\right)}_{\begin{cases} 2\vec{v}_r^\top & r=0 \\ \vec{v}_r^\top & r \geq 1. \end{cases}} \tag{6.3}$$

Here $\vec{e} = (0, \frac{1}{N}, \ldots, \frac{N-1}{N})^\top$, $\vec{\omega} = (0, 1, \ldots, N-1)^\top$, and the primes on the summands indicate that the first term should be halved. The coefficients α_{qr} are given by

$$\alpha_{qr} = \begin{cases} 4i^r J_{\frac{q+r}{2}}\left(-\gamma\pi/2\right) J_{\frac{r-q}{2}}\left(-\gamma\pi/2\right), & \text{if } \mod(|q-r|,2) = 0, \\ 0, & \text{otherwise} \end{cases} \tag{6.4}$$

where $J_\nu(\cdot)$ denotes the Bessel function of the first kind of order ν. In order to eliminate the primes in the summand, we define a new quantity α'_{qr}, which will replace α_{qr} in subsequent expressions.

$$\alpha'_{qr} = \begin{cases} \frac{1}{4}\alpha_{qr} & \text{if } q = r = 0, \\ \frac{1}{2}\alpha_{qr} & \text{if } q = 0, r \neq 0 \text{ or } r = 0, q \neq 0, \\ \alpha_{qr} & \text{otherwise} \end{cases} \tag{6.5}$$

The analysis in [RT18] shows that it suffices to choose $K = \mathcal{O}\left(\frac{\log(1/\epsilon'_{\text{trunc}})}{\log\log(1/\epsilon'_{\text{trunc}})}\right)$ to ensure that the approximation satisfies $\|A - A_K\|_{\max} \leq \epsilon'_{\text{trunc}}$ for some $\epsilon'_{\text{trunc}} > 0$. Consequently, the matrix F_{II} can then be approximated as

$$F_{\text{II}} = (A \circ F) \approx (A_K \circ F) = \sum_{r=0}^{K-1} D_{\vec{u}_r} F D_{\vec{v}_r}, \tag{6.6}$$

where $D_{\vec{u}_r} = \text{diag}(\vec{u}_r)$ and $D_{\vec{v}_r} = \text{diag}(\vec{v}_r)$. Since the complexity of applying the DFT matrix, F, is $\mathcal{O}(N \log N)$, the overall complexity of implementing the Type–II NUDFT in the special case where the non-uniform grid points $\{t_j\}_{j=0}^{N-1} \subseteq [0, 1)$ are generated by perturbing an equispaced grid is

$$\mathcal{O}\left(\frac{N \log N \log(1/\epsilon'_{\text{trunc}})}{\log\log(1/\epsilon'_{\text{trunc}})}\right). \tag{6.7}$$

6.2.1.2 General Nonuniform Node Distribution

We now discuss how to implement the Type–II NUDFT when the nodes $\{t_j\}_{j=0}^{N-1} \subseteq [0, 1)$ are distributed arbitrarily. The key idea is to reduce the problem to the equispaced case considered previously by identifying a sequence $\{s_j\}_{j=0}^{N-1} \subseteq \{0, \ldots, N-1\}$ such that

$$s_j = \begin{cases} m & \text{if } |t_j - \frac{m}{N}| \leq 1/2N \text{ and } m \neq N, \\ 0 & \text{if } |t_j - 1| \leq 1/2N. \end{cases} \tag{6.8}$$

Note that s_n/N is the closest point to t_j on the equidistant grid $\{0, 1/N, \ldots, (N-1)/N\} \subseteq [0, 1)$ We have

$$(F_{\text{II}})_{j,k} = e^{-2\pi i t_j k} = e^{-2\pi i (t_j - \frac{s_j}{N})k} e^{-2\pi i \frac{s_j k}{N}}. \tag{6.9}$$

We obtain the matrix decomposition $F_{\text{II}} = B \circ F_{\vec{s}}$, where $(F_{\vec{s}})_{j,k} = e^{-2\pi i s_j k/N}$, and $B_{jk} = e^{-2\pi i(t_j - s_j/N)k}$. Note that $(F_{\vec{s}})_{j,k}$ is the (k, s_j)-th entry of the DFT matrix, F. As before, the matrix B admits a rank-K approximation via a truncated bivariate Chebyshev approximation applied to each entry. The resulting rank-K approximation of B is then given by

$$B_K = \sum_{r=0}^{K-1} \underbrace{\left[\sum_{q=0}^{K-1} \alpha'_{qr} \left(\exp\left(-i\pi N(\vec{t} - \vec{s}/N)\right) \circ T_q\left(2N(\vec{t} - \vec{s}/N)\right)\right)\right]}_{\vec{u}_r} \underbrace{T_r\left(\frac{2\vec{\omega}^\top}{N} - \vec{1}^\top\right)}_{\begin{cases} 2\vec{v}_r^\top & r = 0 \\ \vec{v}_r^\top & r \geq 1 \end{cases}}. \tag{6.10}$$

Here $\vec{s} = (s_0, s_1, \ldots, s_{N-1})$ and $\vec{\omega} = (0, 1, \ldots, N-1)$. The coefficient $\alpha'_{qr} \propto \alpha_{qr}$ in Equation (6.4) is defined with the choice $\gamma = \frac{1}{2}$. Note that $F_{\vec{s}} = FM_{\vec{s}}$, where

$$(M_{\vec{s}})_{ij} = \begin{cases} 1, & \text{if } i = s_j, \\ 0, & \text{otherwise,} \end{cases} \tag{6.11}$$

for $0 \leq i, j \leq N-1$. Consequently, F_{II} in the general case can then be approximated as

$$F_{\text{II}} = (B \circ F) \approx (B_K \circ F) = \sum_{r=0}^{K-1} D_{\vec{u}_r} F_{\vec{s}} D_{\vec{v}_r} = \sum_{r=0}^{K-1} D_{\vec{u}_r} F M_{\vec{s}} D_{\vec{v}_r}. \tag{6.12}$$

Once again, it suffices to choose $K = \mathcal{O}\left(\frac{\log(1/\epsilon'_{\text{trunc}})}{\log\log(1/\epsilon'_{\text{trunc}})}\right)$ to guarantee that $\|B - B_K\|_{\max} \leq \epsilon'_{\text{trunc}}$. The overall computational complexity of the algorithm remains as stated in Equation (6.7).

6.2.2 Reduction to Type–II NUDFT

Algorithms for Type–I and Type–III NUDFTs can now be derived from those of the Type–II NUDFT.

(I) **Type–I NUDFT**. The Type-I NUDFT can be efficiently implemented using the Type-II NUDFT. In particular, the entries of the matrix F_{I} satisfy

$$(F_{\mathrm{I}})_{jk} = e^{-2\pi i \frac{j}{N}\omega_k} = e^{-2\pi i \frac{\omega_k}{N} j} = (F_{\mathrm{II}})_{kj}, \tag{6.13}$$

which shows that F_{I} is the transpose of F_{II}, where now $\frac{\omega_0}{N}, \ldots, \frac{\omega_{N-1}}{N}$ assume the role of non-equispaced sample points in a Type-I NUDFT. Hence, F_{I} can be approximated as

$$F_{\mathrm{I}} = F_{\mathrm{II}}^T \approx \sum_{r=0}^{K-1} (D_{\vec{u}_r} F M_{\vec{s}} D_{\vec{v}_r})^T = \sum_{r=0}^{K-1} D_{\vec{v}_r} M_{\vec{s}}^T F^T D_{\vec{u}_r}. \tag{6.14}$$

Note that F^T can be efficiently computed using the inverse FFT. Therefore, the Type-I NUDFT may be realized via the Type-II NUDFT, inheriting the same computational complexity.

(III) **Type–III NUDFT**. The Type-III NUDFT can be also efficiently implemented using the Type-II NUDFT. It is shown in [RT18] the entries of the matrix F_{III} satisfy

$$(F_{\mathrm{III}}) = A \circ B \circ (F M_{\vec{s}}) := A \circ B \circ (F_{\mathrm{I}})_{\vec{s}}, \tag{6.15}$$

where $A_{jk} = e^{-2\pi i (t_j - s_j/N)\omega_k}$, B is a matrix of rank at most 2 and $(F_{\mathrm{I}})_{\vec{s}}$ is obtained ex-

tracting the columns indexed by $(s_0, \cdots, s_{N-1})$ from F_{I}. Therefore, the Type-III NUDFT may be realized via the Type-II NUDFT, inheriting the same computational complexity.

6.3 Quantum Algorithm for Non-Uniform Discrete Fourier Transform

We now present a quantum algorithm for the Non-Uniform Quantum Fourier Transform (NUQFT) based on the low-rank factorization framework reviewed in Section 6.2. Throughout, we assume that $N = 2^n$ for some $n \geq 1$ and restrict attention to the Type–II NUQFT. The overall approach may be summarized as follows:

1. **Construction of $U_{\vec{v}_r}$.** We first construct quantum circuits that encode the coefficient vectors $\vec{v}_r$, defined as follows:

$$(\vec{v}_r)_j = \begin{cases} 2^{-1} T_0(2j/N - 1), & r = 0, \\ T_r(2j/N - 1), & r \geq 1. \end{cases} \tag{6.16}$$

   For each r, we construct a unitary operator $U_{\vec{v}_r}$ that encodes the entries of $\vec{v}_r$ as amplitudes of a quantum state.

2. **Construction of $U_{\vec{u}_r}$.** We next construct quantum circuits that encode the coefficient vectors $\vec{u}_r$, defined as follows:

$$(\vec{u}_r)_j = \sum_{q=0}^{K-1} \alpha'_{qr}\, e^{-i\pi N(t_j - s_j/N)}\, T_q(2N(t_j - s_j/N))\,. \tag{6.17}$$

   For each r, we construct a unitary operator $U_{\vec{u}_r}$ that encodes the entries of $\vec{u}_r$ as amplitudes of a quantum state.

3. **Block-encoding Operators.** Using $U_{\vec{v}_r}$ and $U_{\vec{u}_r}$, we implement block encodings of the diagonal matrices $D_{\vec{v}_r}$ and $D_{\vec{u}_r}$. These are combined with a standard block encoding of the Fourier transform and the matrix $M_{\vec{s}}$ defined in Equation (6.11).

4. **Assembly via LCU.** Finally, we employ the Linear Combination of Unitaries (LCU) algorithm to combine the block-encoded matrix products, thereby implementing a block encoding of the Type–II non-uniform discrete Fourier transform.

Remark 6.3.1. In what follows, a quantum register denotes a collection of qubits prepared in superposition over computational basis states. Ancilla qubits serve as auxiliary qubits for intermediate operations within a circuit, while a computational qubit is the qubit in which the final result is stored. Any deviation from this convention will be stated explicitly.This convention is illustrated in the quantum circuit diagram shown in Figure 6.1.

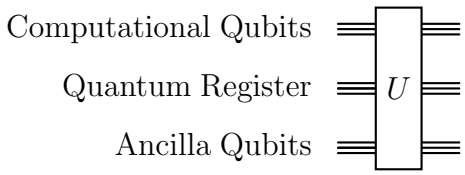


Figure 6.1: Qubit ordering convention used throughout this work: computational qubits (top), quantum register (middle), and ancilla qubits (bottom).

6.3.1 Construction of U_{v_r}

We first construct $U_{\vec{v}_r}$. The case $r = 0$ is particularly simple, since $(\vec{v}_0)_{j,j} = \frac{1}{2}$ because $T_0(x) \equiv 1$. We prepare a quantum register encoding j in uniform superposition, together with an ancilla qubit whose $|0\rangle$ amplitude represents the constant value $\frac{1}{2}$ for all j in the superposition. The corresponding quantum circuit, denoted $U_{\vec{v}_0}$, can be implemented directly.

Proposition 6.3.2. *Let $n \in \mathbb{N}$. The unitary operator $U_{\vec{v}_0}$ can be implemented using $\mathcal{O}(n)$ Hadamard gates and $\mathcal{O}(1)$ single-qubit rotation gates.*

Proof. Prepare an n-qubit register in uniform superposition over all computational basis states using n Hadamard gates. Next, prepare a single computational qubit in the state $\frac{1}{2}|0\rangle + \frac{i\sqrt{3}}{2}|1\rangle$ using a single-qubit rotation of the form $e^{i\pi/3\sigma_X}$. This defines the unitary $U_{\vec{v}_0}$. The corresponding pseudocode is given in Algorithm 2. □

Algorithm 2: Preparation of $U_{\vec{v}_0}$

Input: Integer $n \geq 1$
Output: Quantum state with $U_{\vec{v}_0}$ applied in superposition
Initialize $(n+1)$ qubits $|0\rangle |0\rangle^n$
for $i \leftarrow 2$ **to** $n+1$ **do**
  Apply Hadamard gate H to qubit i in $|0\rangle^n$
Apply single-qubit rotation $e^{i\pi/3\sigma_X}$ to the computational qubit
return Quantum state $U_{\vec{v}_0}|0\rangle|0\rangle^n = \frac{1}{\sqrt{2^n}}\sum_j(\frac{1}{2}|0\rangle + \frac{i\sqrt{3}}{2}|1\rangle)|j\rangle$

Before proceeding, we first prove a lemma on how to compute $1-x$ on a quantum computer, where $x \in (0,1)$.

Lemma 6.3.3. *Let $k \in \mathbb{N}$ and let $x \in (0,1)$ be stored in binary in a k-qubit computational register. There exists a quantum circuit U_{neg} acting on the k-qubit register such that $U_{\text{neg}}|x\rangle = |1-x\rangle$. Moreover, U_{neg} can be implemented using $\mathcal{O}(k)$ σ_X gates, $\mathcal{O}(k)$ CNOT gates, and $\mathcal{O}(k)$ Toffoli gates.*

Proof. Let $x = 0.x_1x_2\ldots x_k$ be a k-bit fractional number expressed in binary, so that $x = \sum_{i=1}^k x_i 2^{-i}$. We define the bitwise complement of x as $\bar{x} = 0.\bar{x}_1\bar{x}_2\ldots\bar{x}_k$, where $\bar{x}_i = 1 - x_i$ for each $i = 1,\ldots,k$. Then

$$\bar{x} = \sum_{i=1}^k \bar{x}_i \cdot 2^{-i} = \sum_{i=1}^k (1-x_i)2^{-i} = \sum_{i=1}^k 2^{-i} - \sum_{i=1}^k x_i 2^{-i} = 1 - 2^{-k} - \sum_{i=1}^k x_i 2^{-i} = 1 - x - 2^{-k}. \quad (6.18)$$

$\bar{x}$ can be computed by applying the σ_X gate to each of the k qubits. It remains to add 2^{-k}. This is a standard application of binary arithmetic and can be implemented by invoking the oracle U_{AM} using $\mathcal{O}(k)$ CNOT and Toffoli gates. The resulting quantum circuit is denoted by U_{neg}. □

We now construct $U_{\vec{v}_r}$ for $r \geq 1$. In Proposition 6.3.4, a parameter $p \in \mathbb{N}$ is selected to facilitate the approximation of $\arccos(\cdot)$ on a quantum computer to arbitrary precision. This parameter is chosen freely at this stage, and will be fixed to a specific value later when required.

Proposition 6.3.4. *Let $p, r, n \in \mathbb{N}$. $U_{\vec{v}_r}$ can be implemented using $\mathcal{O}(n)$ Hadamard gates, $\mathcal{O}(n+p^2)$ CNOT gates, $\mathcal{O}(n)$ σ_X gates and $\mathcal{O}(p)$ single-qubit controlled rotation gates generated by σ_X. The corresponding quantum circuit has depth $\mathcal{O}(p \log p)$.*

Proof. Initialize two quantum registers of n qubits, $\mathcal{O}(p)$ ancilla qubits, and one computational qubit, all initialized to $|0\rangle$. We index the first quantum register of n qubits from 0 to $n-1$. Apply Hadamard gates to the first quantum register to construct the state $\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |0\rangle |j\rangle |0\rangle^n |0\rangle^p$. Let

$$x_j := \frac{2j}{N} - 1 = \frac{\sum_{k=0}^{n-1} j_k 2^k}{2^{n-1}} - 1 = \sum_{k=0}^{n-1} j_k 2^{k-n+1} - 1 = \sum_{k=0}^{n-2} j_k 2^{k-(n-1)} + j_{n-1} - 1 \in [-1, 1). \quad (6.19)$$

We first represent x_j in a fixed-point binary format. In what follows, we index the qubits in the second quantum register from 1 to n. The construction proceeds as follows:

1. For $k = 0$ to $n-2$, apply a CNOT gate with the control on the k-th qubit of the first quantum register and the target on the m-th qubit of the second quantum register, where $m+k = n-1$. This step ensures that the first $n-1$ qubits in the second quantum register

encode the fraction $\sum_{k=0}^{n-2} j_k 2^{k-n+1}$.

2. If $j_{n-1} = 1$, the construction is complete. If $j_{n-1} = 0$, then we must compute $\sum_{k=0}^{n-2} j_k 2^{k-n+1} - 1 \in [-1, 0)$. This is achieved by applying a controlled unitary operator, with the control on the last qubit of the first quantum register and the target on the second quantum register. The operator acts only if the control qubit is in the state $|0\rangle$. The sign is stored in the last qubit of the second quantum register, with $|1\rangle$ denoting a negative sign, while the first $n-1$ qubits encode the absolute value

$$\left| \sum_{k=0}^{n-2} j_k 2^{k-(n-1)} - 1 \right| = 1 - \sum_{k=0}^{n-2} j_k 2^{k-(n-1)}. \tag{6.20}$$

This can be implemented using Lemma 6.3.3.

We now construct the entry $(\vec{v}_r)_j$ in superposition using the quantum signal processing (QSP) algorithm. We implement $e^{ir \arccos(x_j)\sigma_X}$. This requires computing $\theta_j = \arccos(x_j)$, which can be obtained by first computing $\arcsin(x_j)$ using the algorithm of Burge et al. [BBG24] as in Section 2.6.1, and then applying $\arccos(x_j) = \pi/2 - \arcsin(x_j)$. Using the p ancilla qubits to store the approximation $\widehat{\theta}_j = \widehat{\arccos}(x_j)$[2], we have $|\widehat{\theta}_j - \theta_j| \leq 2^{-p+1}$. We label this unitary operator $U_{\arccos}$, which can be implemented using $\mathcal{O}(p^2)$ CNOT gates and a circuit of depth $\mathcal{O}(p \log p)$. Note that we can write

$$\widehat{\theta}_j = b_1^{(\theta_j)} 2^1 + b_0^{(\theta_j)} 2^0 + \sum_{k=-1}^{-(p-2)} b_k^{(\theta_j)} 2^k := \sum_{k=1}^{-(p-2)} b_k^{(\theta_j)} 2^k, \quad b_k^{(\theta_j)} \in \{0, 1\}, \tag{6.21}$$

[2]The p-bit binary representation of $\pi/2$ can be computed classically and used to obtain a p-bit approximation of $\pi/2 - \arcsin(x_j)$ in the p-qubit register containing $\widehat{\theta}_j = \widehat{\arcsin}(x_j)$. Since this operation does not change the overall circuit complexity, we assume that it is incorporated within the quantum circuit, $U_{\arccos}$.

since $\theta \in [0, \pi]$. We can implement $e^{ir\widehat{\theta}_j \sigma_X} := R_X(-r\widehat{\theta}_j)$ via the following formula:

$$R_X(-r\widehat{\theta}_j) = R_X\left(-r \sum_{k=1}^{-(p-2)} b_k^{(\theta_j)} 2^k\right) = \prod_{k=1}^{-(p-2)} R_X(-r2^k)^{b_k^{(\theta_j)}}. \tag{6.22}$$

If we adopt the convention of numbering the qubits in the p-qubit register from 1 down to $-(p-2)$, we have

$$R_X(-r\widehat{\theta}_j) = \prod_{k=1}^{-(p-2)} \left(|0\rangle\langle 0|_k \otimes I + |1\rangle\langle 1|_k \otimes R_X(-r2^k)\right). \tag{6.23}$$

This step requires $\mathcal{O}(p)$ applications of single-qubit controlled rotation gates generated by σ_X. Using [AY25, Equation (45)], which provides the explicit expression for $e^{ir \arccos(x)\sigma_X}$, the resulting quantum state is given by

$$U_{\vec{v}_r} |0\rangle |0\rangle^{2n} |0\rangle^p = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \left(T_r(x_j)|0\rangle + i\sqrt{1-x_j^2} S_{r-1}(x_j)|1\rangle\right) |j\rangle |x_j\rangle^n \left|\widehat{\arccos}(x_j)\right\rangle^p. \tag{6.24}$$

See Algorithm 3 and Figure 6.2 for the corresponding pseudocode and quantum circuit. □

The second quantum register of n qubits, together with the p ancilla qubits, can be uncomputed, yielding a unitary $U'_{\vec{v}_r}$ such that

$$U'_{\vec{v}_r} |0\rangle |0\rangle^{2n} |0\rangle^p = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \left(T_r(x_j)|0\rangle + i\sqrt{1-x_j^2} S_{r-1}(x_j)|1\rangle\right) |j\rangle |0\rangle^n |0\rangle^p. \tag{6.25}$$

This step does not alter the computational complexity of the quantum circuit. For notational simplicity, we continue to denote $U'_{\vec{v}_r}$ by $U_{\vec{v}_r}$.

Algorithm 3: Construction of $U_{\vec{v}_r}$ for $r \geq 1$

Input: $p, n, r \in \mathbb{N}$ and U_{neg} from Lemma 6.3.3
Output: Quantum state with $U_{\vec{v}_r}$ applied in superposition
Initialize $1 + 2n + p$ qubits to $|0\rangle$
Apply Hadamard gates to first register of n qubits
for $k \leftarrow 0$ **to** $n - 2$ **do**
    Apply CNOT with control k in 1st register and target $m = n - k - 1$ in 2nd register
Apply C-U_{neg} with control last qubit of 1st register and target first $n - 1$ qubits on 2nd register
Apply CNOT with control last qubit of 1st register and target last qubit on 2nd register
Compute (an approximation to) $\theta_j = \arccos(x_j)$ using $U_{\arccos}$ on p-qubit ancilla
for $k \leftarrow 1$ down to $-(p-2)$ **do**
    if $b_k^{(\theta_j)} = 1$ in θ_j register **then**
        Apply controlled rotation $e^{ir2^k\sigma_X}$ with control on ancilla qubit k
return Quantum state $U_{\vec{v}_r} |0\rangle |0\rangle^{2n} |0\rangle^p$

6.3.2 Construction of U_{u_r}

We now construct $U_{\vec{u}_r}$. The expression for $(\vec{u}_r)_j$ is determined by $\vec{t} \in [0,1)^{2^n}$. Since $\vec{t}$ is arbitrary, we assume access to the following oracle as part of the algorithm.

Assumption 6.3.5. For $m \geq n$, we assume access to the oracle $O_{\vec{t}}^{(m)}$ such that

$$O_{\vec{t}}^{(m)} : |i\rangle |0\rangle^m \mapsto |i\rangle \left|t_i^{(m)}\right\rangle^m = \bigotimes_{j=1}^{m} |i\rangle \left|t_i^{(j)}\right\rangle, \tag{6.26}$$

where $t_i^{(m)} = \sum_{j=1}^{m} t_i^{(j)} 2^{-j} = 0.t_i^{(1)} \cdots t_i^{(m)}$ is the m-bit binary representation of t_i. The parameter m is left arbitrary for now and will be fixed to a specific value as required.

The first step is to compute $\vec{s} = (s_0, s_1, \ldots, s_{2^n-1})$, as defined in Equation (6.8). The

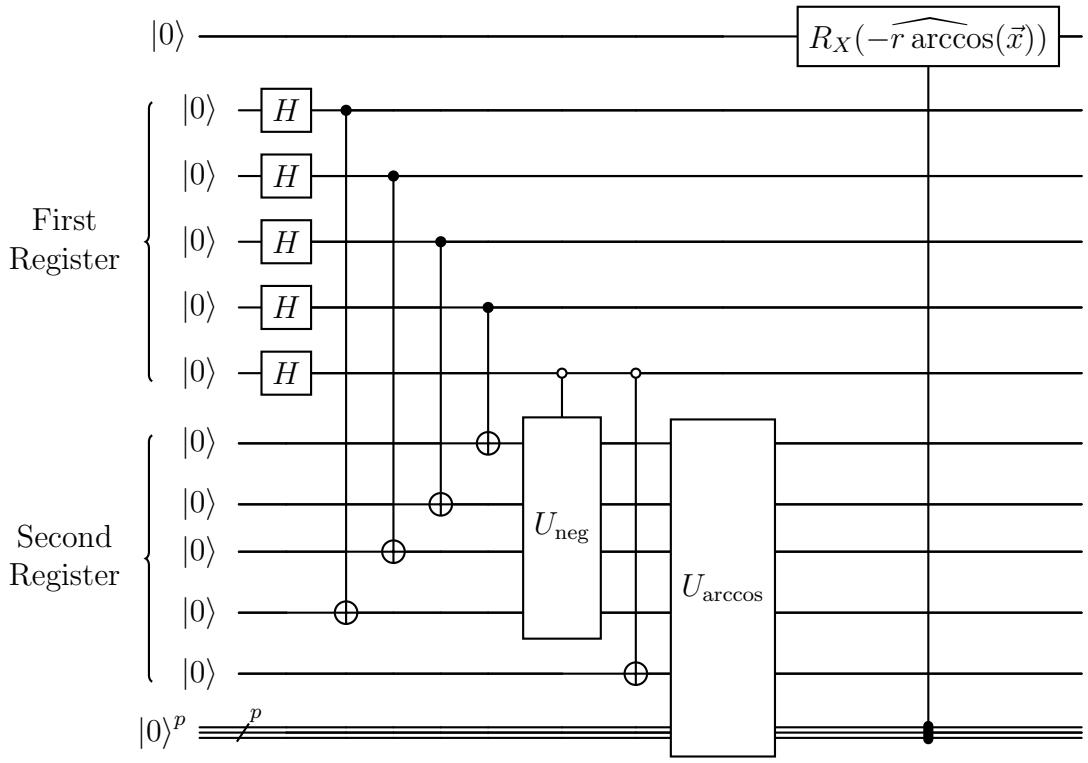


Figure 6.2: Quantum circuit implementing $U_{\vec{v}_r}$ for $r \geq 1$ in the special case $n = 5$.

vector $\vec{s}$ can be computed as

$$s_i = \begin{cases} \text{round}(Nt_i) & \text{if round}(Nt_i) < N \\ 0 & \text{if round}(Nt_i) = N, \end{cases} \tag{6.27}$$

where $N = 2^n$ and round$(\cdot)$ denotes rounding to the nearest integer. In fact, we now explicitly construct a quantum circuit that implements the oracle

$$O_{\vec{s}}^{(m)} : |i\rangle |0\rangle^m \mapsto |i\rangle \left|s_i^{(m)}\right\rangle^m = \bigotimes_{j=1}^{m} |i\rangle \left|s_i^{(j)}\right\rangle, \tag{6.28}$$

where $s_i^{(m)}$ is the m-bit representation of s_i/N. We first claim that the operation $N\vec{t}$ can be implemented efficiently.

Lemma 6.3.6. *Let $n \in \mathbb{N}, m \geq n$ and $\vec{t} \in [0,1)^{2^n}$. $N\vec{t}$ can be computed using $\mathcal{O}(1)$ queries to $O_{\vec{t}}^{(m)}$ and $\mathcal{O}(n)$ Hadamard gates on an $\mathcal{O}(n+m)$ qubit circuit.*

Proof. Initialize an $(n+m)$-qubit quantum register in $|0\rangle$. Apply Hadamard gates to the first n qubits to prepare the uniform index superposition $\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle |0\rangle^m$. Querying $O_{\vec{t}}^{(m)}$ yields

$$O_{\vec{t}}^{(m)} |0\rangle^{n+m} = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \bigotimes_{j=1}^{m} |i\rangle \left| t_i^{(j)} \right\rangle . \tag{6.29}$$

Multiplication by N yields $Nt_i^{(m)} = t_i^{(1)} \cdots t_i^{(n)} . t_i^{(n+1)} \cdots t_i^{(m)}$, where the first n bits correspond to $\lfloor Nt_i \rfloor$, and the remaining $m-n$ bits correspond to the fractional part. No additional quantum gates are required. □

The next step is to round each Nt_i to the nearest integer, a nontrivial operation only when $\left| t_i^{(n+1)} \right\rangle = 1$. This rounding can be implemented as a controlled unitary, denoted U_{round}. In this setting, rounding the binary string $t_i^{(1)} \cdots t_i^{(n)}$ amounts to computing the binary sum $t_i^{(1)} \cdots t_i^{(n)} \oplus 1$. Equivalently, this corresponds to the map $|x\rangle \mapsto |x \oplus y\rangle$, where $|x\rangle = \left| t_i^{(1)} \cdots t_i^{(n)} \right\rangle$ and $|y\rangle = \left| t_i^{(n+1)} \right\rangle$. This operation can be realized using the oracle U_{AM}.

Proposition 6.3.7. *Let $n \in \mathbb{N}$ and $m \geq n$. The oracle $O_{\vec{s}}^{(m)}$ can be implemented using a $\mathcal{O}(n+m)$-qubit quantum circuit, $\mathcal{O}(1)$ queries to the oracle $O_{\vec{t}}^{(m)}$, $\mathcal{O}(n)$ Hadamard gates and $\mathcal{O}(m)$ Toffoli,* CNOT *and σ_X gates.*

Proof. This follows directly from Lemma 6.3.6 together with the use of U_{AM}, noting that the m-bit representation of s_i/N is stored in the m-qubit register upon application of circuit in Figure 6.3. □

Having constructed the oracle $O_{\vec{s}}^{(m)}$, the next step is to construct $U_{\vec{u}_r}$ for $r \geq 0$, which

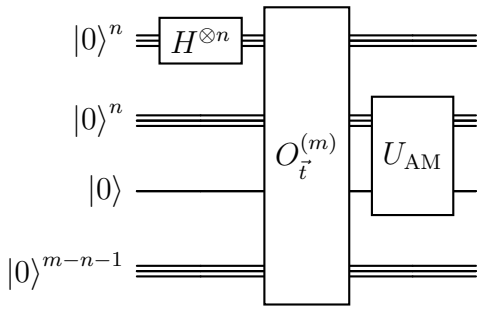


Figure 6.3: Quantum circuit implementing $O_{\vec{s}}^{(m)}$.

implements $(\vec{u}_r)_j$. We first establish a lemma analogous to Lemma 6.3.3, which enables the computation of $x - y$ for $x, y \in [0, 1)$ using the quantum circuit U_{subtract}.

Remark 6.3.8. In what follows, when we state that a quantum register is "multiplied by 2^k" for some $k \geq 1$, we mean a reinterpretation of the same qubit register under a fixed-point encoding in which the binary point is shifted k positions to the right. No quantum gates are applied; this is purely a change in the numerical interpretation of the register contents. Conversely, "division by 2^k" corresponds to shifting the binary point k positions to the left.

Lemma 6.3.9. *Let $m \in \mathbb{N}$ and let $x, y \in [0, 1)$ be stored in k-qubit registers. There exists a unitary operator U_{subtract} that computes the signed difference $x - y \in (-1, 1)$ using $\mathcal{O}(k)$ CNOT, Toffoli, and σ_X gates.*

Proof. Let $x = 0.x_1x_2 \ldots x_k$ and $y = 0.y_1y_2 \ldots y_k$ be the k-bit fractional binary representations of x and y, so that $x = \sum_{i=1}^{k} x_i 2^{-i}$ and $y = \sum_{i=1}^{k} y_i 2^{-i}$. Multiply x and y by 2^k to obtain integers

$$X = \sum_{i=1}^{k} x_i 2^{k-i} = x \cdot 2^k, \quad Y = \sum_{i=1}^{k} y_i 2^{k-i} = y \cdot 2^k. \tag{6.30}$$

Note that $X, Y \in \{0, 1, \ldots, 2^k - 1\}$. Embed X and Y into $(k+1)$-qubit registers by adding a

leading sign bit set to 0, thereby interpreting them as signed integers. We have

$$X = 0 \cdot 2^k + \sum_{i=1}^{k} x_i 2^{k-i}, \quad Y = 0 \cdot 2^k + \sum_{i=1}^{k} y_i 2^{k-i}. \tag{6.31}$$

Compute the bitwise complement of Y:

$$\overline{Y} = 1{\cdot}2^k + \sum_{i=1}^{k} \bar{y}_i 2^{k-i} = 1{\cdot}2^k + \sum_{i=1}^{k} (1-y_i) 2^{k-i} = 1{\cdot}2^k + \sum_{i=1}^{k} 2^{k-i} - \sum_{i=1}^{k} y_i 2^{k-i} = 2^{k+1} - Y - 1. \tag{6.32}$$

$\overline{Y}$ is obtained by applying a σ_X gate to each of the $k+1$ qubits encoding Y. Adding 1 yields $-Y := \overline{Y} + 1 = 2^{k+1} - Y$, ignoring overflow. In binary arithmetic, $\overline{Y} + 1$ is the two's complement of Y and represents its negative when the $(k+1)$-st qubit serves as the sign bit.[3] This justifies the notation $-Y$. Applying U_{AM} to compute $X + (-Y)$ produces the signed integer $Z = X - Y$. Interpreting Z as a fixed-point number by dividing by 2^k gives $\frac{Z}{2^k} = x - y \in (-1, 1)$. Since $x, y \in [0, 1)$, no overflow occurs. The circuit requires $\mathcal{O}(k)$ CNOT, Toffoli, and σ_X gates. □

Algorithm 4: Computation of $x - y$ using signed fixed-point arithmetic (U_{subtract})

Input: k-qubit registers $|x\rangle = |x_1 \cdots x_k\rangle$ and $|y\rangle = |y_1 \cdots y_k\rangle$ encoding $x, y \in [0, 1)$
Output: $(k+1)$-qubit register encoding the signed fixed-point value $x - y$
Scale $|x\rangle$ and $|y\rangle$ by 2^k to obtain integer registers $|X\rangle$ and $|Y\rangle$
Append a leading qubit $|0\rangle$ to each register to form signed $(k+1)$-qubit registers
for $i \leftarrow 1$ **to** $k+1$ **do**
  Apply σ_X to qubit i of $|Y\rangle$
Add 1 to $|Y\rangle$ to obtain the two's-complement encoding of $-Y$
Compute $|X - Y\rangle$ using U_{AM}
Interpret the result as a signed fixed-point number by dividing by 2^k
return Computational register representing $|x - y\rangle$

We construct $U_{\vec{u}_r}$ for $r \geq 0$ by first computing $\vec{t} - \vec{s}/N$ using the preceding routines, and then implementing $T_q\big(2N(\vec{t} - \vec{s}/N)\big)$ via quantum signal processing (QSP). Complex exponential

[3] The identity $-Y := \overline{Y} + 1$ is the standard two's complement representation.

factors are realized through σ_Z-generated rotations, and the weighted Chebyshev terms are combined using the linear combination of unitaries (LCU) technique. We further assume access to a suitable state-preparation oracle.

Assumption 6.3.10. We assume access to a state-preparation oracle

$$\mathrm{PREP}_r \left|0\right\rangle = \frac{1}{\sqrt{\|\vec{\alpha}'_r\|_1}} \sum_{q=0}^{K-1} \sqrt{\alpha'_{qr}} \left|q\right\rangle, \tag{6.33}$$

where α'_{qr} are defined in Equation (6.4) and Equation (6.5) with $\gamma = 1/2$.

Proposition 6.3.11. *Let $m, p, K \in \mathbb{N}$ and $r \in \mathbb{N} \cup \{0\}$. $U_{\vec{u}_r}$ can be implemented using $\mathcal{O}(m + n + p + \log K)$ qubits, $\mathcal{O}(1)$ queries to $O_{\vec{t}}^{(m)}$, $\mathcal{O}(n)$ Hadamard gates, $\mathcal{O}(m + p^2)$ CNOT gates, $\mathcal{O}(m)$ Toffoli gates, $\mathcal{O}(m)$ σ_X gates, and $\mathcal{O}(p + m)$ single-qubit controlled rotation gates. The corresponding quantum circuit has depth $\mathcal{O}(m + p \log p + \log K)$.*

Proof. Consider a $(1 + n + 2m + \mathcal{O}(p))$-qubit system, with all qubits initialized to $|0\rangle$. First, apply Hadamard gates to the quantum register of n qubits, and then apply the oracles $O_{\vec{t}}^{(m)}$ and $O_{\vec{s}}^{(m)}$ on the two m-qubit ancilla registers to construct the state

$$\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |0\rangle|i\rangle|t_i^{(m)}\rangle^m|s_i^{(m)}/N\rangle^m|0\rangle^p. \tag{6.34}$$

Invoking Lemma 6.3.9, we compute $\vec{t} - \vec{s}/N$ in superposition within the quantum register containing $\vec{s}/N$.[4] We then compute an approximation of $\arccos\big(2N(\vec{t} - \vec{s}/N)\big)$. The multiplication by $2N$ is implemented reversibly as a constant-factor scaling on the fixed-point representation of $\vec{t} - \vec{s}/N$. Since $2N$ is a power of two, this operation reduces to a bit shift and therefore does not

[4] A single ancilla qubit can be appended to encode the sign of $t_j - s_j/N$ without affecting the overall complexity. This ancilla qubit is used in Lemma 6.3.9.

affect the overall complexity of the quantum circuit. By invoking $U_{\arccos}$ as in Proposition 6.3.4, we obtain an approximation of $\arccos\big(2N(\vec{t}-\vec{s}/N)\big)$ in superposition within the p-qubit register. We denote the quantum circuit implementing this scaling step followed by $U_{\arccos}$ by $U'_{\arccos}$. As in Proposition 6.3.4, we implement a controlled rotation $R_X\big(-q\arccos\big(2N(\vec{t}-\vec{s}/N)\big)\big)$ to realize the Chebyshev polynomial $T_q(x)$. Similarly, we implement a controlled rotation $R_Z\big(\pi N(\vec{t}-\vec{s}/N)\big)$ to realize the phase factor $e^{-i\pi N(\vec{t}-\vec{s}/N)}$. Denote the unitary operator constructed above by U_{qr}. Finally, using the state-preparation oracle in Equation (6.33), we apply the LCU algorithm to implement the desired expression via the operator $U_{\vec{u}_r} = \sum_{q=0}^{K-1} \alpha'_{qr} U_{qr}$. We need to append an $\mathcal{O}(\log K)$ ancilla qubits to invoke the LCU algorithm. The complexity estimates follow from previous complexity estimates and the observation that a $\mathcal{O}(\log K)$-qubit controlled gate acting can be implemented via a quantum circuit using only CNOT gates and single-qubit gates such that the circuit has depth $\mathcal{O}(\log K)$ [SP22]. If $y_i^{(m)} = t_i^{(m)} - s_i^{(m)}/N$, the resulting quantum state $U_{\vec{u}_r} |0\rangle^{(n+2m+p+1+\mathcal{O}(\log K))}$ is given by

$$\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \left(\frac{1}{\|\vec{\alpha}'_r\|_1} \left(\sum_{q=0}^{K-1} \alpha'_{qr} U_{qr} |0\rangle \right) |0\rangle^{\mathcal{O}(\log K)} + |\perp\rangle \right) |i\rangle \left| t_i^{(m)} \right\rangle^m \left| y_i^{(m)} \right\rangle^m \left| \widehat{\arccos}(y_i^{(m)}) \right\rangle^p , \tag{6.35}$$

where $|\perp\rangle$ is a state that is orthogonal to $\left(\sum_{q=0}^{K-1} \alpha'_{qr} U_{qr} |0\rangle\right) |0\rangle^{\mathcal{O}(\log K)}$. See Figure 6.4 for the corresponding quantum circuit and Algorithm 5 for the corresponding pseudocode. □

Once again, the ancilla qubits can be uncomputed, yielding a unitary $U'_{\vec{u}_r}$ such that the output quantum state is

$$\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \left(\frac{1}{\|\vec{\alpha}'_r\|_1} \left(\sum_{q=0}^{K-1} \alpha'_{qr} U_{qr} |0\rangle \right) |0\rangle^{\mathcal{O}(\log K)} + |\perp\rangle \right) |i\rangle |0\rangle^{2m} |0\rangle^p . \tag{6.36}$$

Algorithm 5: Implementation of $U_{\vec{u}_r}$

Input: Integers $n, m, p \geq 1$, system size $n + 2m + p + 1$ qubits
Output: Operator $U_{\vec{u}_r} = \sum_{q=0}^{K-1} \alpha'_{qr} U_{qr}$ applied in superposition
Initialize all $n + 2m + p + 1$ qubits to $|0\rangle$
Apply Hadamard gates $H^{\otimes n}$ to the n-qubit register
Apply oracles $O_{\vec{t}}^{(m)}$ and $O_{\vec{s}}^{(m)}$ to the two m-qubit registers
Compute $\vec{t} - \vec{s}/N$ in the $\vec{t}$-register using U_{subtract} (see Algorithm 4)
Apply $U_{\arccos}$ to approximate $\arccos\big(\vec{t} - \vec{s}/N\big)$ in the p-qubit register
Apply controlled rotation $R_X(-q \arccos\big(2N(\vec{t} - \vec{s}/N)\big))$ to realize $T_q(x)$
Apply controlled rotation $R_Z(\pi N(\vec{t} - \vec{s}/N))$ to realize $e^{-i\pi N(\vec{t} - \vec{s}/N)}$
Append $\mathcal{O}(\log K)$ ancilla qubits
Denote the resulting unitary as U_{qr}
Use LCU method to implement $U_{\vec{u}_r} = \sum_{q=0}^{K-1} \alpha'_{qr} U_{qr}$ via U_{LCU}
return $U_{\vec{u}_r} |0\rangle^{(n+2m+p+1+\mathcal{O}(\log K))}$

This step does not change the computational complexity of the quantum circuit. For notational simplicity, we continue to denote $U'_{\vec{v}_r}$ by $U_{\vec{v}_r}$. Note that, in Figure 6.4, we have departed from the usual convention of placing the ancilla qubits $|0\rangle^{\mathcal{O}(\log K)}$ below the n-qubit register. This deviation was necessary to avoid awkward placement of the U_{LCU} gate in the circuit diagram.

6.3.3 Block Encoding Operators

We now use $U_{\vec{u}_r}$ and $U_{\vec{v}_r}$ within a block-encoding framework to implement the diagonal matrices $D_{\vec{u}_r}$ and $D_{\vec{v}_r}$. We assume that the ancilla qubits used to implement $U_{\vec{v}_r}$ and $U_{\vec{u}_r}$ have been uncomputed, as discussed above. These qubits will not be referenced in any equations or quantum circuits below; however, their usage is accounted for in the overall complexity analysis of the algorithm in Theorem 6.3.16. Recall that $U_{\vec{v}_r}$ acts on $|j\rangle |0\rangle$ as

$$U_{\vec{v}_r} |j\rangle |0\rangle = |j\rangle \left(v_r(j) |0\rangle + \sqrt{1 - v_r(j)^2} |1\rangle\right). \tag{6.37}$$

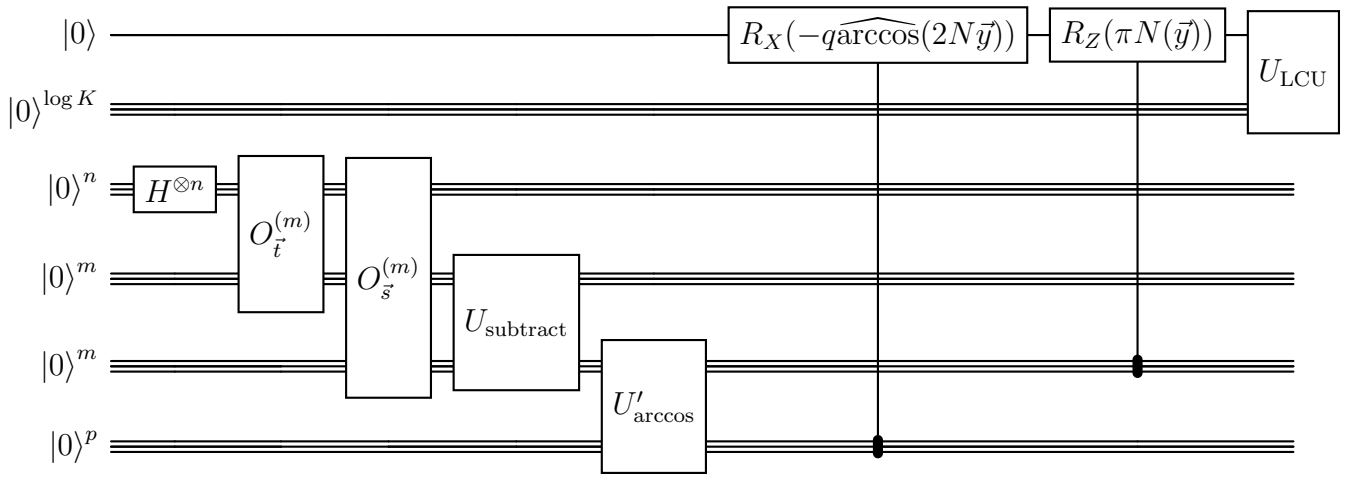


Figure 6.4: Quantum circuit implementing the operator $U_{\vec{u}_r} = \sum_{q=0}^{K-1} \alpha'_{qr} U_{qr}$. We have abbreviated $\vec{t} - \vec{s}/N$ by $\vec{y}$.

The top-left block of $U_{\vec{v}_r}$ corresponds exactly to the desired diagonal matrix: $(\langle 0| \otimes I) U_{\vec{v}_r} (|0\rangle \otimes I) = D_{\vec{v}_r}$. This yields a $(1, 1, 0)$ block encoding of $D_{\vec{v}_r}$.

Example 6.3.12. Let $N = 4$, so that $D_{\vec{v}_r} = \operatorname{diag}\big(v_r(0), v_r(1), v_r(2), v_r(3)\big)$. We abbreviate $i\sqrt{1 - v_r(0)^2}$ as $v'_r(0)$. The matrix representation of $U_{\vec{v}_r}$ block encodes $D_{\vec{v}_r}$ as follows:

$$U_{\vec{v}_r} = \left(\begin{array}{cccc|cccc} v_r(0) & & & & * & & & \\ & v_r(1) & & & & * & & \\ & & v_r(2) & & & & * & \\ & & & v_r(3) & & & & * \\ \hline v'_r(0) & & & & * & & & \\ & v'_r(1) & & & & * & & \\ & & v'_r(2) & & & & * & \\ & & & v'_r(3) & & & & * \end{array}\right) \tag{6.38}$$

We have adopted the basis convention $\{|000\rangle, |010\rangle, |100\rangle, |110\rangle, |001\rangle, |011\rangle, |101\rangle, |111\rangle\}$ in the matrix representation of $U_{\vec{v}_r}$ above, where the first qubit corresponds to the computational qubit. Projecting onto the computational state $|0\rangle$ extracts the top-left submatrix:

$$(\langle 0| \otimes I_4) U_{\vec{v}_r} (|0\rangle \otimes I_4) = \begin{pmatrix} v_r(0) & & & \\ & v_r(1) & & \\ & & v_r(2) & \\ & & & v_r(3) \end{pmatrix} = D_{\vec{v}_r}. \tag{6.39}$$

A similar argument yields a block encoding of $D_{\vec{u}_r}$. The only difference is that $D_{\vec{u}_r}/\|\vec{\alpha}'_r\|_1$ is block encoded, since $\vec{u}_r$ is implemented via the LCU method. Consequently, we obtain a $(\|\vec{\alpha}'_r\|_1, 1, 0)$ block encoding of $D_{\vec{u}_r}$. Furthermore, recall that $F_{\vec{s}} = FM_{\vec{s}}$, where $M_{\vec{s}}$ is defined in Equation (6.11). Since F is unitary, a block encoding of F is immediate. Since $M_{\vec{s}}$ is a binary (0–1) matrix, we can apply Lemma 2.6.5 to obtain a block encoding of $M_{\vec{s}}$. Observe that $M_{\vec{s}}$ is a 1-column-sparse matrix and that the oracle $O_{\vec{s}}^{(m)}$ effectively realizes the column-access oracle O_c. In this setting, it suffices to choose $m = n$. We further note that the row-access oracle O_r corresponds to access to the inverse mapping of $\vec{t} \mapsto \vec{s}$. For generic $\vec{t}$ and $\vec{s}$, constructing O_r from $O_{\vec{t}}$ and $O_{\vec{s}}$ may require additional pre-processing or data-structure assumptions; accordingly, we treat O_r as an independent primitive.

Assumption 6.3.13. We assume black-box access to the row-access oracle O_r required by Lemma 2.6.5.

The oracle O_A can be implemented efficiently. Specifically, we construct O_A by appending

an additional n-qubit register and applying $O_{\vec{s}}^{(n)}$ to obtain

$$|0\rangle^n |i\rangle |j\rangle |0\rangle \mapsto |s_j\rangle |i\rangle |j\rangle |0\rangle . \tag{6.40}$$

Noting that $A_{i,j} = \langle s_j|i\rangle$, we compute the inner product of the n-qubit registers $|s_j\rangle$ and $|i\rangle$, which are computational basis states, and store the resulting binary outcome in the last qubit, thereby encoding $A_{i,j}$. More formally, the entry $A_{i,j}$ is computed as follows.

Lemma 6.3.14. *Let $|i\rangle$ and $|s_j\rangle$ be n-qubit computational basis states, and define $A_{i,j} = \langle s_j|i\rangle \in \{0,1\}$. Then there exists a unitary $U_{i,j}$ acting on an $2n+1$-qubit register such that*

$$U_{i,j}(|i\rangle |s_j\rangle |0\rangle) = |i\rangle |s_j\rangle |A_{i,j}\rangle .$$

Moreover, $U_{i,j}$ can be implemented using $\mathcal{O}(n)$ CNOT gates, $\mathcal{O}(n)$ σ_X gates, $\mathcal{O}(n)$ Toffoli gates and possibly $\mathcal{O}(n)$ ancilla qubits.

Proof. For $r = 1, \ldots, n$, apply a CNOT gate with the r-th qubit of $|i\rangle$ as the control and the r-th qubit of $|s_j\rangle$ as the target. After this operation, the r-th qubit of $|s_j\rangle$ stores $i_r \oplus s_{j,r}$. Next, apply a σ_X gate to each qubit of $|s_j\rangle$ to obtain $i_r \,\mathrm{XNOR}\, s_{j,r}$. Each qubit of $|s_j\rangle$ then stores 1 if $i_r = s_{j,r}$ and 0 otherwise. Finally, apply a C^n-AND gate with the n qubits of $|s_j\rangle$ as controls and the last qubit as the target. This sets the last qubit to 1 if and only if all transformed qubits of $|s_j\rangle$ are equal to 1. Hence, the last qubit correctly encodes $A_{i,j}$. The procedure uses $\mathcal{O}(n)$ CNOT gates, $\mathcal{O}(n)$ σ_X gates, and a single C^n-AND gate, which can be implemented using $\mathcal{O}(n)$ Toffoli gates and, if necessary, $\mathcal{O}(n)$ ancilla qubits. □

Consider the vector

$$\vec{c}_s = (c_0^{s_0}, c_1^{s_1}, \ldots, c_{2^n-1}^{s_{2^n-1}}) \tag{6.41}$$

where $c_i^{s_i}$ denotes the number of occurrences of s_i in $\vec{s}$. Noting that $M_{\vec{s}}$ is a 1-column sparse matrix and a $\|\vec{c}_s\|_\infty$-row sparse matrix, we can construct a $\mathcal{O}(\sqrt{\|\vec{c}_s\|_\infty}, n+3, 0)$ block-encoding of $M_{\vec{s}}$ by invoking Lemma 2.6.5.

Remark 6.3.15. The computational cost of the oracle for $M_{\vec{s}}$ discussed above is evaluated under the assumption that $m = n$. In what follows, m is treated as a free parameter with $m \geq n$. Under this assumption, the computational complexity of implementing the block encoding described in Lemma 6.3.14 does not affect the overall computational complexity of the algorithm derived in Theorem 6.3.16. Accordingly, it is omitted from the subsequent complexity analysis.

6.3.4 Assembly via LCU

The final step is to implement $\sum_{r=0}^{K-1} D_{\vec{u}_r} F_{\vec{s}} D_{\vec{v}_r}$ via a quantum circuit. This is achieved by combining the block encodings constructed above with the LCU algorithm. In Theorem 6.3.16, we present this procedure and derive the corresponding non-asymptotic complexity of the algorithm.

Theorem 6.3.16. *Let $n, m, p, K \in \mathbb{N}$ with $m \geq n$. There exists a quantum circuit, V_{II}, implementing a block encoding of $\sum_{r=0}^{K-1} D_{\vec{u}_r} F M_{\vec{s}} D_{\vec{v}_r}$ with normalization factor*

$$\alpha := K\sqrt{\|\vec{c}_s\|_\infty} \sum_{r=0}^{K-1} \|\vec{\alpha}'_r\|_1. \tag{6.42}$$

and $\mathcal{O}(n + \log K)$ *ancilla qubits. In other words, we have*

$$(\langle 0|^{\mathcal{O}(n+\log K)} \otimes I_{2^n}) V_{\mathrm{II}} (|0\rangle^{\mathcal{O}(n+\log K)} \otimes I_{2^n}) = \frac{1}{\alpha} \sum_{r=0}^{K-1} D_{\vec{u}_r} F M_{\vec{s}} D_{\vec{v}_r}. \tag{6.43}$$

The circuit uses $\mathcal{O}(m+n+p+\log K)$ *qubits,* $\mathcal{O}(n+\log K)$ *Hadamard gates,* $\mathcal{O}(m+p^2)$ CNOT *gates,* $\mathcal{O}(m)$ *Toffoli gates,* $\mathcal{O}(m)$ σ_X *gates, and* $\mathcal{O}(p+m+n^2)$ *single-qubit controlled rotation gates. The circuit depth is* $\mathcal{O}(m + p \log p + \log K)$.

Proof. Fix $r \in \{0, \ldots, K-1\}$. Prepare two single-qubit ancilla registers a_r and b_r together with an n-qubit system register. Observe that $I_{a_r} \otimes I_{b_r} \otimes U_{\mathrm{QFT}}$ is a $(1, 2, 0)$ block encoding of U_{QFT}. By repeated application of Lemma 2.6.4, the unitary

$$U_r := (I_{b_r} \otimes U_{\vec{u}_r})(I_{a_r} \otimes I_{b_r} \otimes U_{\mathrm{QFT}})(I_{a_r} \otimes U_{\vec{v}_r}) \tag{6.44}$$

is a $(\|\vec{\alpha}'_r\|_1, 2, 0)$ block encoding of $D_{\vec{u}_r} F D_{\vec{v}_r}$. Using the construction described above for $M_{\vec{s}}$ and again invoking Lemma 2.6.4, we obtain a $(\|\vec{\alpha}'_r\|_1 \sqrt{\|\vec{c}_s\|_\infty},\ n+5,\ 0)$ block encoding of $D_{\vec{u}_r} F M_{\vec{s}} D_{\vec{v}_r}$, which we continue to denote by U_r. To implement the sum $\sum_{r=0}^{K-1} D_{\vec{u}_r} F M_{\vec{s}} D_{\vec{v}_r}$, we apply the LCU algorithm. Introduce $\mathcal{O}(\log K)$ ancilla qubits and define the select oracle $\mathrm{SEL} = \sum_{r=0}^{K-1} U_r \otimes |r\rangle\langle r|$. To implement the linear combination of unitaries (LCU), we assume access to a separate state-preparation oracle $\mathrm{PREP} := H^{\otimes \mathcal{O}(\log K)}$ Conjugating SEL with PREP yields the standard LCU unitary. The resulting construction is therefore a

$$\left(\sum_{r=0}^{K-1} K \sqrt{\|\vec{c}_s\|_\infty} \|\vec{\alpha}'_r\|_1,\ n + 5 + \log K, 0 \right) \tag{6.45}$$

block encoding of $\sum_{r=0}^{K-1} D_{\vec{u}_r} F M_{\vec{s}} D_{\vec{v}_r}$ [Gil+19a, Lemma 52]. The stated gate counts, qubit complexity, and circuit depth follow from the previous complexity estimates, taking into account the costs of implementing each U_r as well as the overhead by the LCU construction. □

Note that the $\mathcal{O}(n^2)$ term in the expression $\mathcal{O}(p + n^2)$ in Theorem 6.3.16 arises from the requirement to apply $\mathcal{O}(n^2)$ controlled-rotation gates in the standard quantum circuit implementation of the Quantum Fourier Transform. We therefore observe that this quadratic dependence constitutes the dominant contribution to the overall gate-complexity estimates.

6.4 Error Analysis

We now present an error analysis of the quantum algorithm implementing the Type-II NUDFT.

6.4.1 Error Sources

We first identify the sources of error arising in the quantum implementation of the algorithm.

1. **Truncation Error**. Antolín and Townsend [RT18] approximate the NUDFT matrix by a rank-K matrix from a Chebyshev expansion. To achieve $\|\cdot\|_{\max} \leq \epsilon_{\text{trunc}}$, choose $K = \mathcal{O}\left(\frac{\log(1/\epsilon'_{\text{trunc}})}{\log\log(1/\epsilon'_{\text{trunc}})}\right)$, for some $\epsilon'_{\text{trunc}} > 0$.

2. **Finite-Precision Oracle Error**. The sampling points t_j and s_j are given with m-bit precision, introducing errors in $t_j - s_j/N$ that propagate to the complex exponentials and Chebyshev evaluations.

3. **Arccosine Evaluation Error**. Implementing QSP requires rotations about the X axis with angles of the form $\arccos(x)$. Computing $\arccos(x)$ to p-bit precision introduces an approximation error of order $\mathcal{O}(2^{-p})$.

4. **Block-Encoding Error**. These errors accumulate in the block encoding of the constructed operators, reducing the overall accuracy of the resulting unitaries.

6.4.2 Error Bounds

We now bound the total error of the quantum algorithm and specify the parameters (K, m, p) so that the total error remains below the prescribed tolerance ϵ. Let $\widetilde{V}_{\mathrm{II}}$ denote the approximately constructed version of the quantum circuit introduced in Theorem 6.3.16. Moreover, recall that we have approximated $F_{\mathrm{II}} = A \circ (FM_{\vec{s}})$ on a quantum computer by implementing

$$\widetilde{F}_{\mathrm{II}} := A_K \circ (FM_{\vec{s}}) = \sum_{r=0}^{K-1} D_{\vec{u}_r} FM_{\vec{s}} D_{\vec{v}_r}. \tag{6.46}$$

To bound the total error, we apply the triangle inequality

$$\|\widetilde{V}_{\mathrm{II}} - F_{\mathrm{II}}\| \leq \|\widetilde{F}_{\mathrm{II}} - F_{\mathrm{II}}\| + \|\widetilde{V}_{\mathrm{II}} - \widetilde{F}_{\mathrm{II}}\| \tag{6.47}$$

The first term is determined by the truncation error. It has already been analyzed in [RT18] in the $\|\cdot\|_{\max}$ norm. We now derive an estimate for it in the $\|\cdot\|$ norm based on the following lemma.

Lemma 6.4.1. *Let $B, C \in \mathbb{C}^{N\times N}$. We have $\|B \circ C\| \leq \sqrt{N}\|B\|_{\max}\|C\|$.*

Proof. Let $e_1, \ldots, e_N$ be the standard basis vectors in $\mathbb{C}^N$, and let $v = \sum_{i=1}^N v_i e_i$ be any vector.

For each i, we have

$$\|(B \circ C)e_i\|_2 \leq \|B\|_{\max}\|Ce_i\|_2 \leq \|B\|_{\max}\|C\| \tag{6.48}$$

because $(B \circ C)e_i$ is obtained from Ce_i by multiplying each entry by a scalar of magnitude at most $\|B\|_{\max}$. By linearity and the triangle inequality, we have

$$\|(B \circ C)v\|_2 = \left\|\sum_{i=1}^{N} v_i(B \circ C)e_i\right\|_2 \leq \sum_{i=1}^{N} |v_i|\|(B \circ C)e_i\|_2 \leq \|B\|_{\max}\|C\|\|v\|_1. \tag{6.49}$$

Since $\|v\|_1 \leq \sqrt{N}\|v\|_2$, we obtain $\|(B \circ C)v\|_2 \leq \sqrt{N}\|B\|_{\max}\|C\|\|v\|_2$ for all $v \in \mathbb{C}^N$. The claim follows. □

We can use Lemma 6.4.1 to compute an upper bound on K. This is carried out in Theorem 6.4.4 below. Next, we bound the second term in Equation (6.47), which will be used to determine the values of m and p in Theorem 6.4.4. We assume that the block-encoding of $M_{\vec{s}}$ is exact. This assumption is justified because we have black-box access to the oracle O_r, and the oracles O_c and O_A (Lemma 6.3.14) are constructed exactly. Furthermore, we assume that the quantum Fourier transform is implemented exactly, as we analyze the algorithm within the fault-tolerant regime. efine the normalization factors as $\alpha_{\vec{u}_r} = \|\vec{\alpha}'_r\|_1$ and $\alpha_{\vec{v}_r} = 1$, where $\vec{\alpha}'_r$ are the coefficients appearing in Equation (6.17). The analysis in the previous section implies that $U_{\vec{u}_r}$ and $U_{\vec{v}_r}$ are exact block-encodings.

$$U_{\vec{u}_r} = \begin{pmatrix} D_{\vec{u}_r}/\alpha_{\vec{u}_r} & * \\ * & * \end{pmatrix}, \quad U_{\vec{v}_r} = \begin{pmatrix} D_{\vec{v}_r}/\alpha_{\vec{v}_r} & * \\ * & * \end{pmatrix}. \tag{6.50}$$

Since the errors introduced above render the block-encodings inexact in the construction of $\widetilde{V}_{\mathrm{II}}$,

let $\widetilde{U}_{u_r}$ and $\widetilde{U}_{v_r}$ be the implemented block-encodings such that

$$\widetilde{U}_{\vec{u}_r} = \begin{pmatrix} \widetilde{D}_{\vec{u}_r}/\alpha_{\vec{u}_r} & * \\ * & * \end{pmatrix}, \quad \widetilde{U}_{\vec{v}_r} = \begin{pmatrix} \widetilde{D}_{\vec{v}_r}/\alpha_{\vec{v}_r} & * \\ * & * \end{pmatrix}, \tag{6.51}$$

with $\widetilde{D}_{\vec{u}_r}$ and $\widetilde{D}_{\vec{v}_r}$ obtained via finite-precision oracle access and approximate function evaluation. Let $\widetilde{F}$ and $\widetilde{M_{\vec{s}}}$ denote the corresponding exact block-encodings of F and $M_{\vec{s}}$, respectively. Note that $\widetilde{V}_{\mathrm{II}}$ can be written as

$$\widetilde{V}_{\mathrm{II}} = \sum_{r=0}^{K-1} \widetilde{D}_{\vec{u}_r} \widetilde{F} \widetilde{M_{\vec{s}}} \widetilde{D}_{\vec{v}_r}. \tag{6.52}$$

The normalization factor $1/\alpha$, defined in Equation (6.42), is omitted in the following analysis, as it affects only the post-selection success probability. We bound the unitary circuit synthesis error of the unnormalized top-left block in order to determine the required parameters (K, m, p) as functions of the target precision. We have

$$\|\widetilde{V}_{\mathrm{II}} - \widetilde{F}_{\mathrm{II}}\| = \left\| \sum_{r=0}^{K-1} \widetilde{D}_{\vec{u}_r} \widetilde{F} \widetilde{M_{\vec{s}}} \widetilde{D}_{\vec{v}_r} - \sum_{r=0}^{K-1} D_{\vec{u}_r} F M_{\vec{s}} D_{\vec{v}_r} \right\|, \tag{6.53}$$

$$\leq \sum_{r=0}^{K-1} \|(\widetilde{D}_{\vec{u}_r} - D_{\vec{u}_r}) \widetilde{F} \widetilde{M_{\vec{s}}} \widetilde{D}_{\vec{v}_r}\| + \sum_{r=0}^{K-1} \|D_{\vec{u}_r} F M_{\vec{s}} (\widetilde{D}_{\vec{v}_r} - D_{\vec{v}_r})\|. \tag{6.54}$$

Here, the terms involving $\widetilde{F} - F$ and $\widetilde{M_{\vec{s}}} - M_{\vec{s}}$ vanish because we assume that $\|\widetilde{M_{\vec{s}}} - M_{\vec{s}}\| = 0 = \|\widetilde{F} - F\|$. Note that $\|F\| = \|\widetilde{F}\| = 1, \|M_{\vec{s}}\| = \|\widetilde{M_{\vec{s}}}\| = \sqrt{\|\vec{c}_s\|_\infty}$, $\|D_{\vec{u}_r}\| \leq \|\vec{\alpha}'_r\|_1$ and $\|D_{\vec{v}_r}\| \leq 1$. Hence, we have

$$\|\widetilde{V}_{\mathrm{II}} - \widetilde{F}_{\mathrm{II}}\| \leq \sqrt{\|\vec{c}_s\|_\infty} \sum_{r=0}^{K-1} \left(\|\widetilde{D}_{\vec{u}_r} - D_{\vec{u}_r}\| + \|\vec{\alpha}'_r\|_1 \|\widetilde{D}_{\vec{v}_r} - D_{\vec{v}_r}\| \right). \tag{6.55}$$

Since we have

$$\|D_{\vec{v}_r} - \widetilde{D}_{\vec{v}_r}\| = \max_j |(\vec{v}_r)_j - (\widetilde{\vec{v}}_r)_j|, \quad \|D_{\vec{u}_r} - \widetilde{D}_{\vec{u}_r}\| = \max_j |(\vec{u}_r)_j - (\widetilde{\vec{u}}_r)_j|, \tag{6.56}$$

it suffices to bound the worst-case scalar error in a single diagonal entry. We first compute the error in the evaluation of the complex exponential and the Chebyshev polynomials, which will allow us to determine the worst-case scalar error in a single diagonal entry. We first bound the complex exponential term. In what follows, we assume that $m \geq n+1$. In this case, the oracle constructing s_j is exact; hence, the only source of error is in t_j.

Lemma 6.4.2. *Let $m, n \in \mathbb{N}$ such that $m \geq n+1$ and $N = 2^n$. Let $y_j = t_j - s_j/N$ for $j = 0, \ldots, N-1$, where t_j and s_j are provided by m-bit precision oracles. Let $\widetilde{y}_j$ denote the m-bit approximation of y_j. The errors in the complex exponential satisfy*

$$|e^{-i\pi N y_j} - e^{-i\pi N \widetilde{y}_j}| \leq \pi N 2^{-m+1} \tag{6.57}$$

uniformly in j.

Proof. We have $|t_j - \widetilde{t}_j| \leq 2^{-m}$. Hence, we have

$$|e^{-i\pi N y_j} - e^{-i\pi N \widetilde{y}_j}| = \left|e^{-i\pi N(t_j - s_j/N)} - e^{-i\pi N(\widetilde{t}_j - s_j/N)}\right| \tag{6.58}$$

$$\leq \pi N \left|(t_j - s_j/N) - (\widetilde{t}_j - s_j/N)\right| \leq \pi N\, 2^{-m} \tag{6.59}$$

uniformly in j. □

We now bound the error in the Chebyshev polynomials. Note that the Chebyshev polyno-

mials are evaluated at two inputs, $x_j := 2j/N - 1$ and $y_j := (t_j - s_j/N)$ Since $j = 0, \ldots, N-1$, the values $2j/N - 1$ can be computed exactly (see Proposition 6.3.4), as we use at least n qubits. On the other hand, there is an approximation error in computing $t_j - s_j/N$ via the oracles. Hence, we treat these cases separately below. In Lemma 6.4.3, we continue to denote $y_j = 2N\left(t_j - \frac{s_j}{N}\right)$ for simplicity.

Lemma 6.4.3. *Let $m, n \in \mathbb{N}$ such that $m \geq n+1$. Let $q \leq K$ and $j = 0, \ldots, N-1$. Let $T_q(y_j) = \cos(q \arccos(y_j))$ denote the degree-q Chebyshev polynomial of the first kind. The errors in computing the Chebyshev polynomials can be bounded as follows:*

1. *For $x_j = 2j/N - 1$, the error in the QSP implementation due to p-bit finite-precision evaluation of $\arccos(x)$ satisfies*

$$|T_q(x_j) - \widetilde{T}_q(x_j)| \leq q 2^{-p+1} \leq K 2^{-p+1}, \tag{6.60}$$

   uniformly in j.

2. *For $y_j = 2N(t_j - s_j/N)$, where t_j and s_j are provided by m-bit precision oracles, let $\widetilde{y}_j$ denote the m-bit approximation. Let $y_j^* \in (\min(y_j, \widetilde{y}_j), \max(y_j, \widetilde{y}_j))$. The total Chebyshev polynomial error satisfies*

$$|T_q(y_j) - \widetilde{T}_q(y_j)| \leq q \left(2^{-p+1} + \frac{N 2^{-m+1}}{\sqrt{1-(y_j^*)^2}}\right) \leq K \left(2^{-p+1} + \frac{N 2^{-m+1}}{\sqrt{1-(y_j^*)^2}}\right). \tag{6.61}$$

Proof. Let $\theta_j = \arccos(\cdot)$, where the argument is either x_j or y_j, and define $\delta_{\theta_j} := |\theta_j - \widetilde{\theta}_j|$.

1. Let $x_j = 2j/N - 1$. The only source of error is the finite-precision evaluation of $\arccos(x_j)$

to p bits, giving a rotation-angle error $\delta_{\theta_j} \leq 2^{-p+1}$. By the mean-value theorem for the cosine function, the Chebyshev polynomial error is bounded by

$$|T_q(x_j) - \widetilde{T}_q(x_j)| = |\cos(q\theta_j) - \cos\big(q(\theta_j + \delta_{\theta_j})\big)| \leq q\delta_{\theta_j} \leq q2^{-p+1} \leq K2^{-p+1}. \tag{6.62}$$

2. Let $y_j = 2N(t_j - s_j/N)$, where t_j and s_j/N are provided by an m-bit precision oracle. We have $|y_j - \widetilde{y}_j| \leq 2N2^{-m} = N2^{-m+1}$. The rotation angle is $\theta_j = \arccos(y_j)$, while the computed rotation is $\widetilde{\theta}_j$ with finite-precision p-bit evaluation. Applying the mean-value theorem to arccos, there exists $y_j^* \in (\min(y_j, \widetilde{y}_j), \max(y_j, \widetilde{y}_j))$ such that

$$|\theta_j - \arccos(\widetilde{y}_j)| \leq \frac{|y_j - \widetilde{y}_j|}{\sqrt{1 - (y_j^*)^2}} \leq \frac{N2^{-m+1}}{\sqrt{1 - (y_j^*)^2}}. \tag{6.63}$$

Let $\widetilde{\theta}_j = \arccos(\widetilde{y}_j)$. By the triangle inequality, we have

$$\delta_{\theta_j} = |\theta_j - \widetilde{\theta}_j| \leq |\theta_j - \arccos(\widetilde{y}_j)| + |\arccos(\widetilde{y}_j) - \widetilde{\theta}_j| \leq \frac{N2^{-m+1}}{\sqrt{1 - (y_j^*)^2}} + 2^{-p+1}, \tag{6.64}$$

Applying the mean-value theorem to the cosine function,

$$|T_q(y_j) - \widetilde{T}_q(y_j)| = |\cos(q\theta_j) - \cos\Big(q\widetilde{\theta}_j\Big)| \leq q\delta_{\theta_j} = q\left(2^{-p+1} + \frac{N2^{-m+1}}{\sqrt{1 - (y_j^*)^2}}\right). \tag{6.65}$$

Since the maximum Chebyshev degree in the truncated expansion is $q \leq K$, the error is bounded by

$$|T_q(y_j) - \widetilde{T}_q(y_j)| \leq K\left(2^{-p+1} + \frac{N2^{-m+1}}{\sqrt{1 - (y_j^*)^2}}\right). \tag{6.66}$$

This completes the proof. □

We can now compute $\|D_{\vec{v}_r} - \widetilde{D}_{\vec{v}_r}\|$ and $\|D_{\vec{u}_r} - \widetilde{D}_{\vec{u}_r}\|$. Using Lemma 6.4.3, we have

$$\|D_{\vec{v}_r} - \widetilde{D}_{\vec{v}_r}\| \leq \begin{cases} 0, & r = 0, \\ & \\ r2^{-p+1}, & r \geq 1, \end{cases} . \tag{6.67}$$

Similarly, using Lemma 6.4.2, and Lemma 6.4.3, we have

$$\|D_{\vec{u}_r} - \widetilde{D}_{\vec{u}_r}\| \leq \|\vec{\alpha}'_r\|_1 \left(\pi N 2^{-m} + K \left(2^{-p+1} + \max_j \frac{N 2^{-m+1}}{\sqrt{1-(y_j^*)^2}} \right) \right), \tag{6.68}$$

Let $\alpha' = \sum_{r=0}^{K-1} \|\vec{\alpha}'_r\|_1$. Substituting the bounds in Equation (6.55), we have

$$\|\widetilde{V}_{\mathrm{II}} - \widetilde{F}_{\mathrm{II}}\| \leq \sqrt{\|\vec{c}_s\|_\infty} \sum_{r=0}^{K-1} \|\vec{\alpha}'_r\|_1 \left(\pi N 2^{-m} + K 2^{-p+1} + \max_j \frac{N K 2^{-m+1}}{\sqrt{1-(y_j^*)^2}} + K 2^{-p+1} \right), \tag{6.69}$$

$$= \alpha' \sqrt{\|\vec{c}_s\|_\infty} \left(\pi N 2^{-m} + K 2^{-p+2} + \max_j \frac{N K 2^{-m+1}}{\sqrt{1-(y_j^*)^2}} \right). \tag{6.70}$$

We can now complete our error analysis by specifying the parameters (K, m, p) so that the total error remains below the prescribed tolerance ϵ.

Theorem 6.4.4. *Let $\epsilon > 0$ and let $n, m, p, K \in \mathbb{N}$ with $m \geq n+1$ and $N = 2^n$. Let $\widetilde{V}_{\mathrm{II}} = \sum_{r=0}^{K-1} \widetilde{D}_{\vec{u}_r} \widetilde{F} \widetilde{M}_{\vec{s}} \widetilde{D}_{\vec{v}_r}$ denote the quantum implementation of the Type-II non-uniform quantum Fourier transform with Chebyshev truncation degree K, m-bit oracle precision, and p-bit finite-precision evaluation of* $\arccos(x)$ *in the QSP implementation. Let $\|\vec{\alpha}'_r\|_1 = \sum_{q=0}^{K-1} |\alpha'_{qr}|$, where*

α'_{qr} is defined in Equation (6.5). Define

$$\alpha' = \sum_{r=0}^{K-1} \|\vec{\alpha}'_r\|_1, \quad y_j = 2N(t_j - s_j/N), \quad y_j^* \in (\min(y_j, \widetilde{y}_j), \max(y_j, \widetilde{y}_j)), \tag{6.71}$$

where $\widetilde{y}_j$ is a m-bit approximation of y_j. If(K, m, p) are chosen such that

$$K = \mathcal{O}\left(\frac{\log\left(\sqrt{N\|\vec{c}_s\|_\infty}/\epsilon\right)}{\log\log\left(\sqrt{N\|\vec{c}_s\|_\infty}/\epsilon\right)}\right), \tag{6.72}$$

$$m = \left\lceil \log_2 \frac{8\alpha'\sqrt{\|\vec{c}_s\|_\infty}(\pi N + K\max_j(1-(y_j^*))^{-1/2}}{\epsilon} \right\rceil, \tag{6.73}$$

$$p = \left\lceil \log_2 \frac{16\alpha'\sqrt{\|\vec{c}_s\|_\infty}K}{\epsilon} \right\rceil, \tag{6.74}$$

then the total error of the quantum algorithm satisfies $\|\widetilde{V}_{\mathrm{II}} - F_{\mathrm{II}}\| \leq \epsilon$. In particular, it suffices to choose m and p such that

$$m = \mathcal{O}\left(\log\left(\frac{K\sqrt{\|\vec{c}_s\|_\infty}(N + K\max_j(1-(y_j^*))^{-1/2}}{\epsilon}\right)\right), \tag{6.75}$$

$$p = \mathcal{O}\left(\log\left(\frac{\sqrt{\|\vec{c}_s\|_\infty}K^2}{\epsilon}\right)\right). \tag{6.76}$$

Proof. We have

$$\|\widetilde{V}_{\mathrm{II}} - F_{\mathrm{II}}\| \leq \underbrace{\|\widetilde{F}_{\mathrm{II}} - F_{\mathrm{II}}\|}_{\text{truncation error}} + \underbrace{\|\widetilde{V}_{\mathrm{II}} - \widetilde{F}_{\mathrm{II}}\|}_{\text{block-encoding error}}, \tag{6.77}$$

where the first term is the Chebyshev truncation error and the second term is the block-encoding

error. Using Lemma 6.4.1, the first term in Equation (6.47) can be expressed as

$$\left\|\widetilde{F}_{\mathrm{II}} - F_{\mathrm{II}}\right\| = \left\|(A - A_K) \circ (FM_{\vec{s}})\right\| \tag{6.78}$$

$$\leq \sqrt{N}\|A - A_K\|_{\max}\|FM_{\vec{s}}\| \tag{6.79}$$

$$\leq \|\vec{c}_s\|_\infty \sqrt{N}\|A - A_K\|_{\max}. \tag{6.80}$$

The last equality follows since F is unitary, and we have $\|FM_{\vec{s}}\| = \|M_{\vec{s}}\| = \sqrt{\|\vec{c}_s\|_\infty}$. Hence, in order to ensure that $\|\widetilde{F}_{\mathrm{II}} - F_{\mathrm{II}}\| \leq \epsilon/2$ for some $\epsilon > 0$, it suffices to choose $\epsilon'_{\mathrm{trun}} = \epsilon/2\sqrt{N\|\vec{c}_s\|_\infty}$. Hence, it suffices to choose K such that

$$K = \mathcal{O}\left(\frac{\log\left(\frac{\sqrt{N\|\vec{c}_s\|_\infty}}{\epsilon}\right)}{\mathrm{loglog}\left(\frac{\sqrt{N\|\vec{c}_s\|_\infty}}{\epsilon}\right)}\right). \tag{6.81}$$

From (6.69), the block-encoding error satisfies

$$\|\widetilde{V}_{\mathrm{II}} - \widetilde{F}_{\mathrm{II}}\| \leq \alpha'\sqrt{\|\vec{c}_s\|_\infty}\left(\pi N 2^{-m} + K2^{-p+2} + \max_j \frac{NK2^{-m+1}}{\sqrt{1-(y_j^*)^2}}\right). \tag{6.82}$$

Let $C := \alpha'\sqrt{\|\vec{c}_s\|_\infty}$ and $\kappa := \max_j \frac{1}{\sqrt{1-(y_j^*)^2}}$. Using this notation, Equation (6.82) can be succinctly expressed as

$$\|\widetilde{V}_{\mathrm{II}} - \widetilde{F}_{\mathrm{II}}\| \leq C(2^{-m}(\pi N + 2NK\kappa) + K2^{-p+2}). \tag{6.83}$$

To ensure that the right-hand side is at most $\epsilon/2$, we can allocate the error budget equally

between the oracle and QSP contributions, i.e.,

$$C2^{-m}(\pi N + 2NK\kappa) \leq \frac{\epsilon}{4}, \qquad CK2^{-p+2} \leq \frac{\epsilon}{4}. \tag{6.84}$$

Solving for m, we obtain

$$C2^{-m}(\pi N + 2NK\kappa) \leq \frac{\epsilon}{4} \iff m \geq \log_2\left(\frac{4C(\pi N + 2NK\kappa)}{\epsilon}\right). \tag{6.85}$$

Solving for p, we obtain

$$CK2^{-p+2} \leq \frac{\epsilon}{4} \iff 2^{-p} \leq \frac{\epsilon}{16CK} \iff p \geq \log_2\left(\frac{16CK}{\epsilon}\right). \tag{6.86}$$

Substituting the definitions of C and κ yields

$$m = \left\lceil \log_2\left(\frac{4\alpha'\sqrt{\|\vec{c}_s\|_\infty}(\pi N + 2NK\max_j(1-(y_j^*))^{-1/2})}{\epsilon}\right)\right\rceil, \tag{6.87}$$

$$p = \left\lceil \log_2\left(\frac{16\alpha'\sqrt{\|\vec{c}_s\|_\infty}K}{\epsilon}\right)\right\rceil. \tag{6.88}$$

With these choices of m and p, the error satisfies $\|\widetilde{V}_{\mathrm{II}} - \widetilde{F}_{\mathrm{II}}\| \leq \epsilon$. Combined with the choice of K, we therefore have $\|\widetilde{V}_{\mathrm{II}} - F_{\mathrm{II}}\| \leq \epsilon$. We now further simplify the choices of m and p. It is well known [Wat22; AS64] that for $\nu \geq 0$ and $0 \leq x \ll \nu$, we have

$$J_\nu(x) \sim \frac{(x/2)^\nu}{\Gamma(\nu+1)}, \tag{6.89}$$

where Γ is the Gamma function. Furthermore, recall that $\alpha'_{qr} = \mathcal{O}(\alpha_{qr})$ (see Equation (6.5)),

where α_{qr} is defined in Equation (6.4) with $\gamma = \frac{1}{2}$ by

$$\alpha_{qr} = \begin{cases} 4i^r J_{\frac{q+r}{2}}\left(-\pi/4\right) J_{\frac{r-q}{2}}\left(-\pi/4\right), & \text{if } \mathrm{mod}(|q-r|,2) = 0, \\ 0, & \text{otherwise} \end{cases} \tag{6.90}$$

The condition $\mathrm{mod}(|q-r|,2) = 0$ implies that q and r have the same parity. Hence, the order of the Bessel function J is an integer in Equation (6.90). Since we are interested in $|\alpha_{qr}|$, it suffices to evaluate the Bessel functions at $\pi/4$ and to assume $(r-q)/2 \geq 0$, as Bessel functions of integer order are either even or odd. Equation (6.89) now implies that $|\alpha_{qr}| = \mathcal{O}(1)$, since for the fixed value $x_0 = \pi/4$, the magnitude of $J_\nu(x_0)$ decreases rapidly as ν increases. Hence, $\alpha' = \mathcal{O}(K)$, since each of the K summands $|\alpha'_{qr}|$ is $\mathcal{O}(1)$. Based on Equation (6.87) and Equation (6.88), it suffices to choose m and p such that

$$m = \mathcal{O}\left(\log\left(\frac{NK\sqrt{\|\vec{c}_s\|_\infty}(1 + K\max_j(1-(y_j^*))^{-1/2}}{\epsilon}\right)\right), \quad p = \mathcal{O}\left(\log\left(\frac{\sqrt{\|\vec{c}_s\|_\infty}K^2}{\epsilon}\right)\right). \tag{6.91}$$

This completes the proof. □

The expressions in Theorem 6.4.4 are rather involved. Noting that $N = 2^n$ and $\|\vec{c}_s\|_\infty = \mathcal{O}(2^n)$, we can further simplify the $\mathcal{O}(\cdot)$ estimates in Theorem 6.4.4 in terms of n, ϵ, and $\kappa := \max_j \frac{1}{\sqrt{1-(y_j^*)^2}}$. Noting that $\sqrt{N\|\vec{c}_s\|_\infty} = \mathcal{O}(2^n)$, we have

$$\log\left(\frac{\sqrt{N\|\vec{c}_s\|_\infty}}{\epsilon}\right) = \Theta(n) + \log\left(\frac{1}{\epsilon}\right) = \Theta\left(n + \log\left(\frac{1}{\epsilon}\right)\right). \tag{6.92}$$

Hence, we can simplify the complexity estimate for K in Equation (6.72) to

$$K = \mathcal{O}\left(\frac{n + \log(1/\epsilon)}{\log(n + \log(1/\epsilon))}\right) = \mathcal{O}\left(n + \log\left(\frac{1}{\epsilon}\right)\right). \tag{6.93}$$

Similarly, we obtain

$$p = \mathcal{O}\left(n + \log(1/\epsilon) + \log K\right), = \mathcal{O}\left(n + \log\left(\frac{1}{\epsilon}\right) + \log\left(n + \log\left(\frac{1}{\epsilon}\right)\right)\right). \tag{6.94}$$

The last equality follows since $\log K = \mathcal{O}(n + \log(1/\epsilon))$. Clearly, $p = \mathcal{O} = (n + \log(1/\epsilon))$. In a similar manner, we obtain

$$m = \mathcal{O}\left(\log\left(\frac{NK\sqrt{\|\vec{c}_s\|_\infty}(1 + K\kappa)}{\epsilon}\right)\right), \tag{6.95}$$

$$= \mathcal{O}\left(\log\left(\frac{K2^{3n/2}(1 + K\kappa)}{\epsilon}\right)\right), \tag{6.96}$$

$$= \mathcal{O}\left(\log K + n + \log(1 + K\kappa) + \log\left(\frac{1}{\epsilon}\right)\right) = \mathcal{O}\left(n + \log\left(\frac{1}{\epsilon}\right) + \log(1 + K\kappa)\right). \tag{6.97}$$

Corollary 6.4.5. *Let $\epsilon > 0$ and let $n, m, p, K \in \mathbb{N}$ with $m \geq n+1$. Let $\widetilde{V}_{\mathrm{II}}$, y_j and y_j^* be defined as in Theorem 6.4.4. If the parameters (K, m, p) are chosen such that*

$$K = p = \mathcal{O}\left(n + \log\left(\frac{1}{\epsilon}\right)\right), \quad m = \mathcal{O}\left(n + \log\left(\frac{1}{\epsilon}\right) + \log(1 + K\kappa)\right), \tag{6.98}$$

where $\kappa := \max_j \frac{1}{\sqrt{1-(y_j^*)^2}}$, then $\|\widetilde{V}_{\mathrm{II}} - F_{\mathrm{II}}\| \leq \epsilon$.

Proof. This follows from Theorem 6.4.4 and the discussion above. □

6.5 Numerical Results

We present numerical results obtained from both classical simulations and hybrid classical–quantum implementations for small-scale instances. The objective is to validate the new components introduced in the quantum algorithm rather than to re-establish the accuracy of the underlying classical method. The experiments therefore isolate the elements specific to our construction: the low-rank truncation used in the LCU representation, finite-precision discretization of node data, the reversible implementation of the arccos subroutine, the state-preparation routines $U_{v_{\vec{r}}}$ and $U_{u_{\vec{r}}}$, the small-K LCU combination, and the block encoding of the matrix $M_{\vec{s}}$. Accordingly, we run numerical tests for problem sizes accessible to classical simulation. The code used to generate the reported results is available in [Aft26].

6.5.1 Validation of Low-Rank Truncation Error Scaling

The low-rank Chebyshev approximation underlying our construction was extensively analyzed and numerically validated in Antolín and Townsend [RT18]. As our work focuses on translating this decomposition into a quantum block-encoding framework, we do not repeat those experiments and instead only validate the truncation behavior of the low-rank approximation for representative node configurations. The low-rank truncation error is characterized by

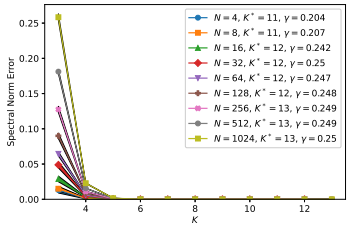
$$\left|\widetilde{F}_{\mathrm{II}}^{(K)} - F_{\mathrm{II}}\right| \leq \|\vec{c}_s\|_\infty \sqrt{N} \|A - A_K\|_{\max}. \tag{6.99}$$

Here, the superscript indicates that $\widetilde{F}_{\mathrm{II}}^{(K)}$ is defined with respect to the truncation parameter K. While $\|A - A_K\|_{\max} = \mathcal{O}(\log(1/\epsilon))$, a more precise estimate is $5\gamma e^{W\left(\frac{\log(140/\epsilon)}{5\gamma}\right)}$, as given in

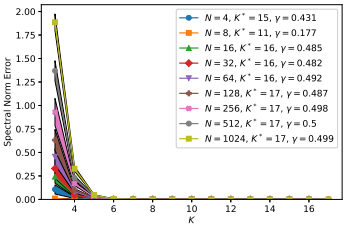
[RG17], where W is Lambert's W function. Hence, we obtain the estimate

$$K = 5\gamma e^{\left(\frac{\log\left(\|\vec{c}_s\|_\infty \sqrt{N}\, 140/\epsilon\right)}{5\gamma}\right)}. \tag{6.100}$$

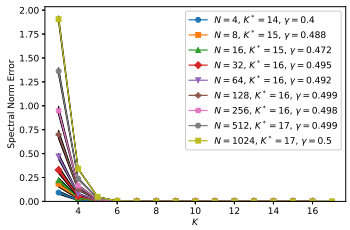
For a range of transform sizes N, we numerically construct $\widetilde{F}_{\mathrm{II}}^{(K)}$ for increasing K and measure the spectral-norm error $\|F_{\mathrm{II}} - \widetilde{F}_{\mathrm{II}}^{(K)}\|$. The experiments cover perturbed equispaced grids, clustered nodes, and randomly distributed nodes in $[0, 1)$. Figure 6.5 confirms the predicted error baselines and shows that moderate K suffices to achieve $\epsilon = 10^{-10}$. This test also demonstrates that, for quantum implementations, the LCU qubit cost is modest. For instance, when $\epsilon = 10^{-5}$, approximately 4 qubits are required.



(a) Perturbed equispaced nodes



(b) Clustered nodes



(c) Uniformly random nodes

Figure 6.5: $\|(F_{\mathrm{II}} - \widetilde{F}_{\mathrm{II}}^{(K)})\|_2$ as a function of K for $N = 2^n$, $n = 2, \ldots, 10$, with accuracy $\epsilon = 10^{-10}$. For each N and node geometry, the error is averaged over 20 independent complex Gaussian test vectors. K^* denotes the minimal K required to achieve the prescribed tolerance.

The low-rank Chebyshev approximation underlying our construction was extensively analyzed and numerically validated in Antolín and Townsend [RT18]. As our work focuses on translating this decomposition into a quantum block-encoding framework, we do not repeat those experiments and instead validate the newly introduced quantum components.

6.5.2 Finite-Precision Effects

We next investigate the effect of finite-precision node representation on the accuracy of the transform. The parameter m governs the precision with which the non-uniform nodes t_j and derived quantities such as s_j are represented. In the theoretical analysis presented in Proposition 6.3.7, the parameter m scales logarithmically with the truncation parameter K and the condition parameter κ. The purpose of this experiment is to demonstrate that the dependence of m on both κ and K is sufficiently mild for practical purposes. This is verified through classical emulation. For a fixed transform size N, we generate node sets using a function parameterized by γ such that

$$\left| t_j - \frac{s_j}{N} \right| = \gamma N, \qquad \gamma \in \left(0, \tfrac{1}{2}\right), \tag{6.101}$$

thereby producing a range of geometric configurations with varying κ. Given a node set $\{x_j\} \subseteq [0, 1)$, we can form the truncated representations

$$t_j^{(m)} = \frac{\mathrm{round}(2^m t_j)}{2^m}, \tag{6.102}$$

and construct the corresponding approximate transform $\widetilde{F}_{\mathrm{II}}^{(m)}$ using these discretized nodes. For each node set, we compute the parameter κ and the minimal precision m^* required to achieve spectral-norm error ϵ. The minimal precision m^* is determined via a bracketing and binary search procedure over m. As shown in Table 6.1, κ increases monotonically with γ, with particularly sharp growth once $\gamma \geq 0.49$. Despite this amplification, the required precision

parameter m^* remains essentially stable across the entire range. This stability arises from compensating effects. Although κ grows rapidly near the boundary, its overall scaling remains moderate (effectively sublinear), and the admissible K also stays moderate. Most importantly, the precision depends on $\log(1 + K\kappa)$, whose logarithmic structure mitigates the impact of large κ. Consequently, even when κ becomes large, $\log(1 + K\kappa)$ grows slowly and m^* changes very little. Thus, the $\log(1 + K\kappa)$ factor in Proposition 6.3.7 is mild, and the parameters m, n, p all scale as $\mathcal{O}(n + \log(1/\epsilon))$ morally.

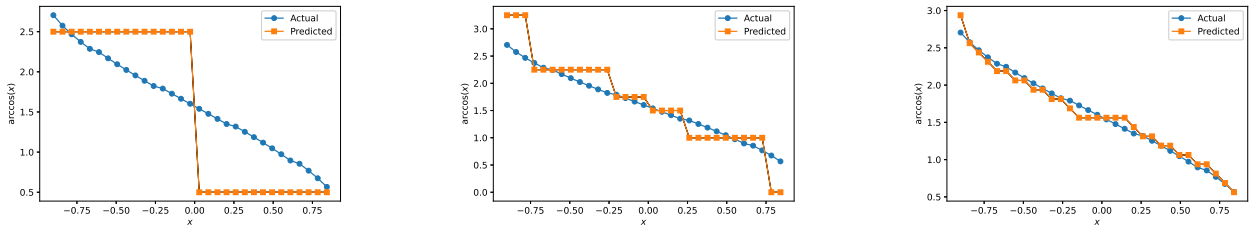
#	γ	κ	K	$\log(1+K\kappa)$	m^*
1	0	1	1	0.693147	7
2	0.0544444	1.00598	10	2.40332	15
3	0.108889	1.02459	11	2.5072	15
4	0.163333	1.05804	13	2.69155	15
5	0.217778	1.11091	14	2.80655	15
6	0.272222	1.19218	14	2.87303	15
7	0.326667	1.32088	15	3.03559	15
8	0.381111	1.54484	16	3.24717	15
9	0.435556	2.0363	17	3.57283	15
10	0.49	5.02519	17	4.45931	15

#	γ	κ	K	$\log(1+K\kappa)$	m^*
1	0.49	5.02519	17	4.45931	15
2	0.4911	5.32373	17	4.51638	13
3	0.4922	5.68359	17	4.58109	15
4	0.4933	6.12904	17	4.6558	13
5	0.4944	6.70032	17	4.74411	7
6	0.4955	7.47039	17	4.852	15
7	0.4966	8.58954	17	4.99058	12
8	0.4977	10.4377	17	5.18426	15
9	0.4988	14.4424	17	5.50745	15
10	0.4999	50.0025	17	6.74646	15

Table 6.1: Comparison of κ and $\log(1 + K\kappa)$ for two γ regimes for $N = 2^6$, $\epsilon = 10^{-12}$.

6.5.3 Quantum Arccos Subroutine

We examine the quantum implementation of the arccos subroutine using a reversible CORDIC-style routine with precision parameter p. The $\arcsin(x)$ algorithm of [BBG24] uses $5p - 1$ qubits, while our $\arccos(x)$ implementation requires $7p + 10$, preserving $\mathcal{O}(p)$ scaling but with a larger constant factor. Numerical tests with a `matrix-product-state simulator` for $p = 1, \ldots, 6$ (with $m = 6$) show that $p = 7$ exceeds simulator capacity. Results for $p = 1, 3, 5$ are shown in Figure 6.6. These results indicate a hardware bottleneck that restricts the algorithm's simulation to small parameters. Consequently, a fully quantum implementation is not feasible



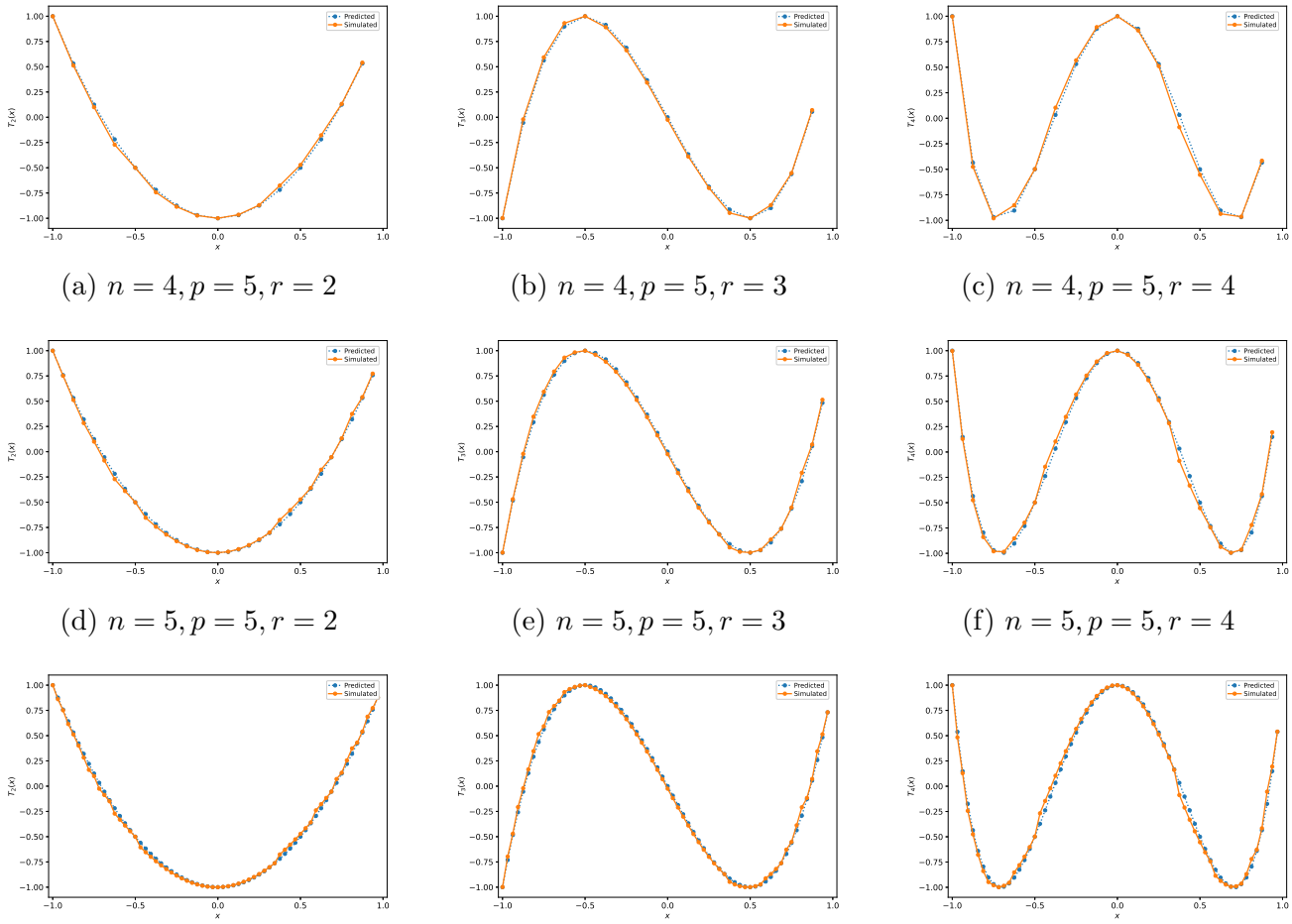
(a) $p = 1$ (b) $p = 3$ (c) $p = 5$

Figure 6.6: Comparison of actual and predicted values of $\arccos(x)$ for odd p values ($p = 1, 3, 5$). The corresponding mean absolute errors are 0.5195115053, 0.2251069658, and 0.0645826060, respectively.

with current hardware. Therefore, we adopt a hybrid classical–quantum approach, performing arithmetic operations classically while enabling quantum simulation for small parameters.

6.5.4 Hybrid Implementation of $U_{v_{\vec{r}}}$

We validate the algorithm in Section 6.3.1 implementing the unitary operator $U_{v_{\vec{r}}}$. The arccos computation is performed classically due to current hardware and simulation constraints and is included in the pre-processing stage. The resulting parameters are then loaded into a p-qubit register. To verify correctness, the circuit is simulated for small problem sizes where full statevector simulation is feasible. The simulated amplitudes are renormalized by multiplying by $\sqrt{N}$ and compared with the classical values $T_r(x_j)$ defining $v_{\vec{r}}$, computed via direct Chebyshev evaluation using `NumPy`. For various degrees r and qubit counts n, the simulated amplitudes are compared with the corresponding Chebyshev polynomial values. The results in Figure 6.7 show that $U_{v_{\vec{r}}}$ accurately reproduces the vector $v_{\vec{r}}$. Table 6.2 reports the statevector error for fixed $n = 4$ and $r = 4$ as p increases. The ratios of consecutive errors confirm that the error decreases by approximately a factor of two when p increases by one.



(a) $n = 4, p = 5, r = 2$ (b) $n = 4, p = 5, r = 3$ (c) $n = 4, p = 5, r = 4$

(d) $n = 5, p = 5, r = 2$ (e) $n = 5, p = 5, r = 3$ (f) $n = 5, p = 5, r = 4$

(g) $n = 6, p = 5, r = 2$ (h) $n = 6, p = 5, r = 3$ (i) $n = 6, p = 5, r = 4$

Figure 6.7: Comparison of computed and predicted outputs for different parameter of the quantum circuit $U_{v_{\vec{r}}}$ for $n \in \{4, 5, 6\}$ and $r \in \{2, 3, 4\}$ with fixed $p = 5$.

6.5.5 Hybrid Implementation of $U_{u_{\vec{r}}}$

We next validate the algorithm in Section 6.3.2 implementing the unitary operator $U_{u_{\vec{r}}}$. As in the previous analysis, all arithmetic computations are carried out classically, and the resulting data are subsequently loaded into the quantum registers. Let $\vec{x} = \vec{t} - \vec{s}/N$ and

$$u_q(\vec{x}) = \sqrt{N} e^{-i\pi N \vec{x}} T_q(2N\vec{x}) \tag{6.103}$$

#	n	p	r	Error	Ratio
1	4	1	4	0.9880609500	1.6475
2	4	2	4	0.5997170775	2.2256
3	4	3	4	0.2694581809	2.1944
4	4	4	4	0.1227710437	2.0449
5	4	5	4	0.0600312609	1.8409
6	4	6	4	0.0326167771	2.2405
7	4	7	4	0.0145581686	1.9220
8	4	8	4	0.0075722616	–

Table 6.2: Statevector error for fixed $n = 4$ and $r = 4$ as p increases.

denote the normalized amplitude of the $|0\rangle$ computational basis state of the circuit U_q. For a range of values of q and transform sizes N, we plot the real and imaginary parts of the simulated amplitudes and the predicted components in Figure 6.8. The results in Figure 6.8 provide numerical evidence that U_q accurately reproduces $u_{qr}(\vec{x})/\sqrt{N}$. Table 6.3 reports the statevector error for fixed $n = 4$ and $q = 2$ as p, m increases.

#	n	m	p	q	Error	Ratio
1	4	2	1	2	1.6138906535	2.00799
2	4	2	2	2	0.8036549819	1.30723
3	4	2	3	2	0.6147763375	1.28833
4	4	2	4	2	0.4771891201	1.06065
5	4	2	5	2	0.4499019429	1.03347
6	4	2	6	2	0.4353344292	1.00562
7	4	2	7	2	0.4329028110	–

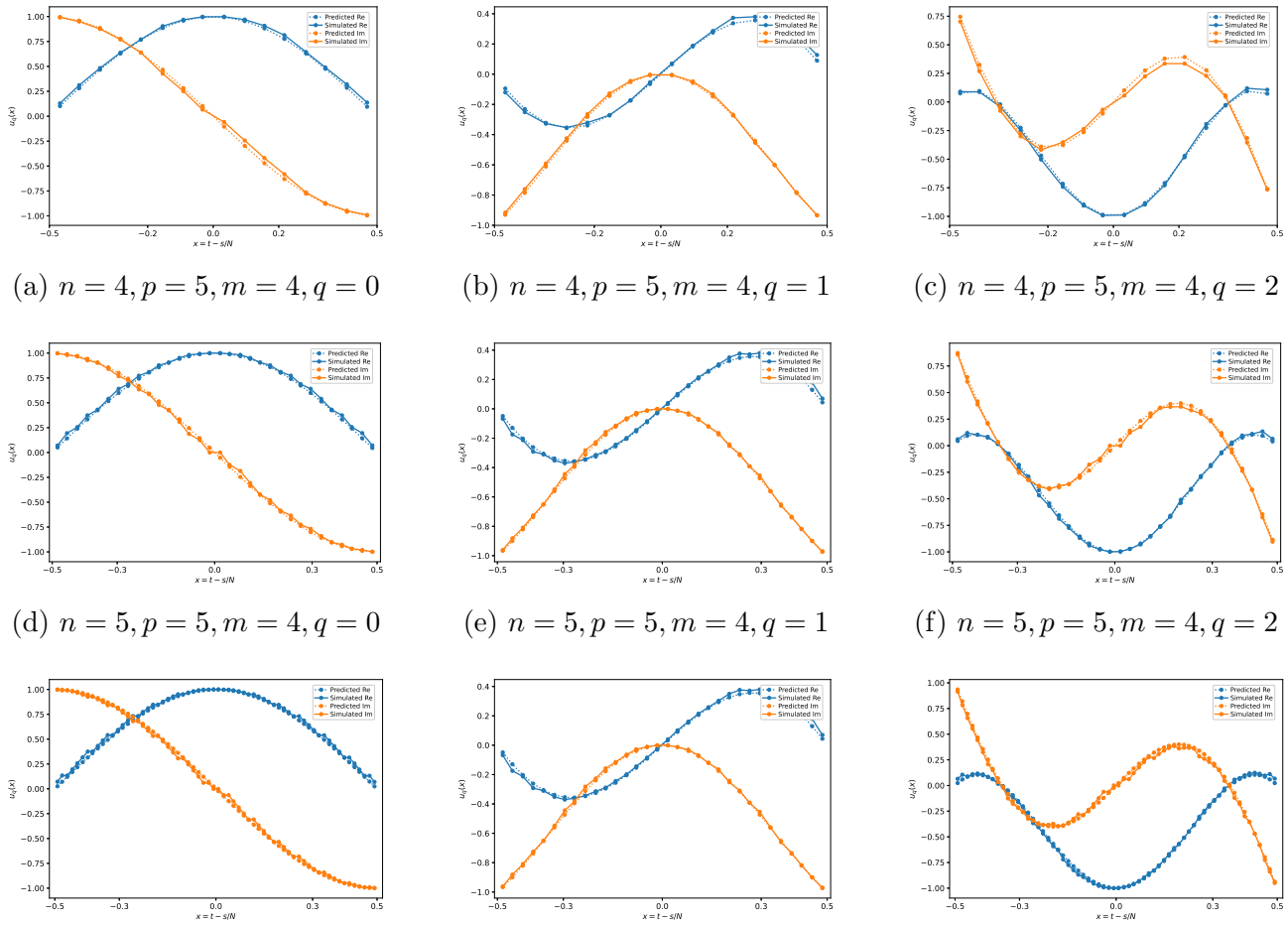
#	n	m	p	q	Error	Ratio
1	4	1	2	2	1.1343486555	1.41149
2	4	2	2	2	0.8036549819	1.12381
3	4	3	2	2	0.7151168695	1.04547
4	4	4	2	2	0.6840150547	1.01223
5	4	5	2	2	0.6757494378	1.00142
6	4	6	2	2	0.6747895628	1.00031
7	4	7	2	2	0.6745814549	–

Table 6.3: State-vector error for fixed $n = 4$ and $q = 2$. The left table varies p with $m = 2$ fixed, while the right table varies m with $p = 2$ fixed. The last column shows the ratio of consecutive rows.

We next validate $U_{u_{\vec{r}}} = \sum_{q=0}^{K-1} \alpha'_{qr} U_q$ using the LCU algorithm. Let

$$u_r(\vec{x}) = \sqrt{N}\|\vec{\alpha}_{qr}\|_1 \sum_{q=0}^{K-1} \alpha'_{qr} e^{-i\pi N\vec{x}}\, T_q(2N\vec{x}) \tag{6.104}$$

denote the normalized amplitude of the $|0\rangle$ basis state of the circuit $U_{u_{\vec{r}}}$. For $K = 2$, we report the results in Figure 6.9, which provide evidence that $U_{u_{\vec{r}}}$ accurately reproduces $u_r(\vec{x})/\|\alpha_{qr}\|_1\sqrt{N}$.



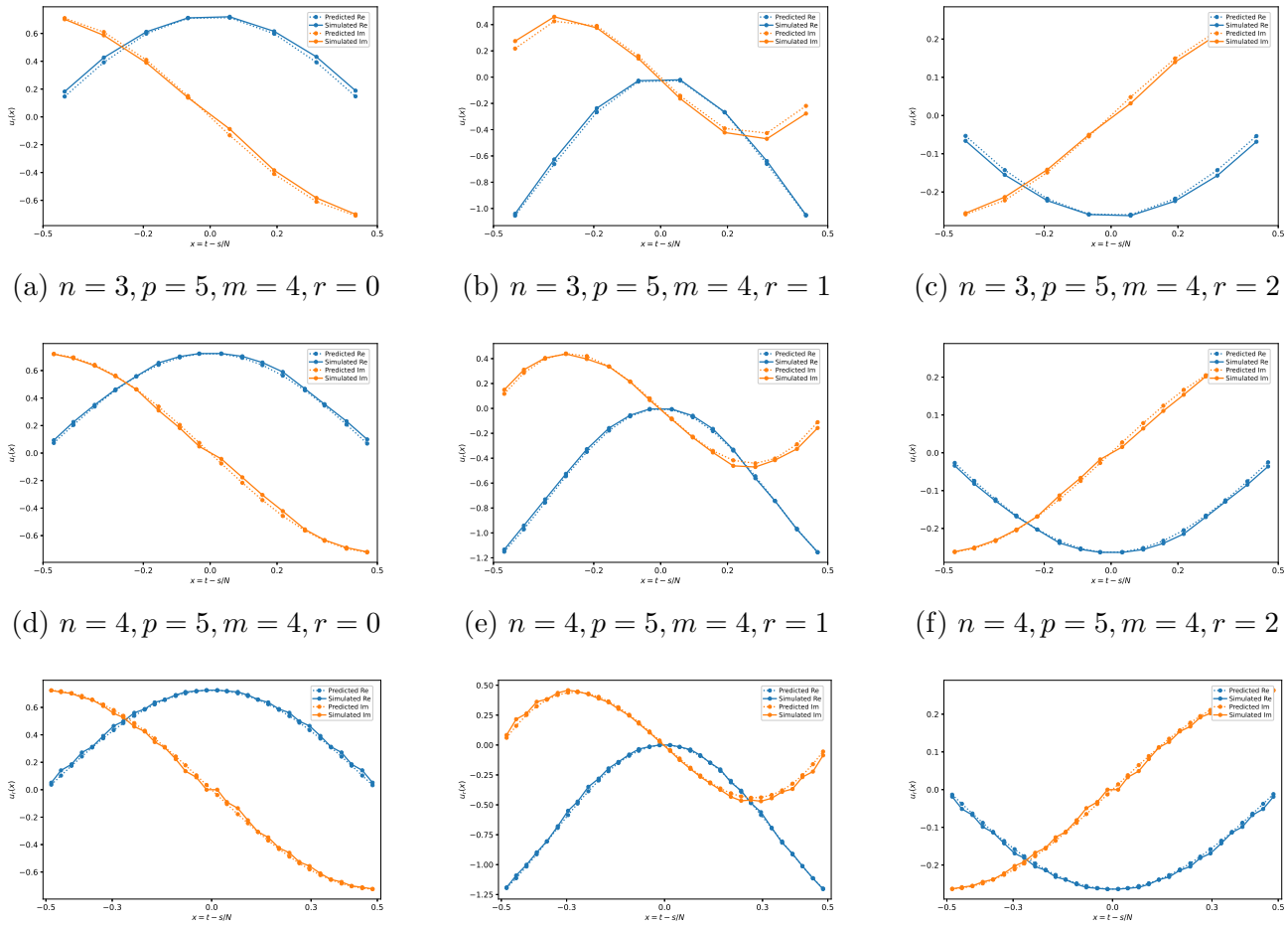
(a) $n = 4, p = 5, m = 4, q = 0$ (b) $n = 4, p = 5, m = 4, q = 1$ (c) $n = 4, p = 5, m = 4, q = 2$

(d) $n = 5, p = 5, m = 4, q = 0$ (e) $n = 5, p = 5, m = 4, q = 1$ (f) $n = 5, p = 5, m = 4, q = 2$

(g) $n = 6, p = 5, m = 4, q = 0$ (h) $n = 6, p = 5, m = 4, q = 1$ (i) $n = 6, p = 5, m = 4, q = 2$

Figure 6.8: Comparison of computed and predicted outputs for different parameters of the quantum circuit U_q for $n \in \{4, 5, 6\}$ and $q \in \{0, 1, 2\}$ with $m = 4$ and $p = 5$.

Note that the LCU algorithm dominates the computational cost of implementing the method. This is because the LCU procedure requires multi-qubit controlled unitaries which, when decomposed into elementary gates, reduce to multi-qubit controlled-X gates, since the arithmetic loader ultimately consists of such operations. Given current simulation capabilities, this approach can only be implemented efficiently for small values of K and n. In particular, we restrict to the case $K = 2$.



(a) $n = 3, p = 5, m = 4, r = 0$ (b) $n = 3, p = 5, m = 4, r = 1$ (c) $n = 3, p = 5, m = 4, r = 2$

(d) $n = 4, p = 5, m = 4, r = 0$ (e) $n = 4, p = 5, m = 4, r = 1$ (f) $n = 4, p = 5, m = 4, r = 2$

(g) $n = 5, p = 5, m = 4, r = 0$ (h) $n = 5, p = 5, m = 4, r = 1$ (i) $n = 5, p = 5, m = 4, r = 2$

Figure 6.9: Comparison of the computed and predicted for different parameters of the quantum circuit $U_{u_{\vec{r}}}$ for $n \in \{4, 5, 6\}$ and $r \in \{0, 1, 2\}$ with $m = 4$ and $p = 5$.

6.5.6 Numerical Validation of the M_s Block Encoding

Lastly, we validate the block encoding constructed for the matrix $M_{\vec{s}}$. The implemented circuit exploits the structure of $M_{\vec{s}}$. Since each column has exactly one nonzero entry, the block encoding is realized via a simplified right–isometry. Although inspired by the sparse-access block-encoding lemma, the implementation does not explicitly realize the oracle triplet (O_c, O_r, O_A). Instead, it uses the column structure of $M_{\vec{s}}$ to build a direct circuit. The resulting

unitary $U_{M_{\vec{s}}}$ uses $n+1$ ancilla qubits and satisfies

$$(\langle 0|^{n+1} \otimes I) U_{M_{\vec{s}}} (|0\rangle^{n+1} \otimes I) = \frac{M_s}{\sqrt{d_r}}, \qquad d_r = \|\vec{c}_s\|_\infty. \tag{6.105}$$

We verify Equation (6.105) by simulating $U_{M_{\vec{s}}}$, extracting the encoded matrix by projecting the ancilla register onto $|0\rangle^{n+1}$, and comparing it with the normalized target matrix $M_{\vec{s}}/\sqrt{d_r}$. We quantify the error using the maximum entrywise norm and the spectral norm. The construction is tested on representative vectors $\vec{s}$ with different sparsity patterns:

1. **Identity:** $s_j = j$, yielding the identity matrix ($d_r = 1$).
2. **Bit-Reversal:** s_j is the bit-reversal of j (permutation, $d_r = 1$).
3. **Constant:** all entries of s are identical, giving maximal multiplicity $d_r = N$.
4. **Two-Cluster:** half the columns map to one row and half to another ($d_r = N/2$).
5. **Heavy Row:** most columns map to one row, with the remainder distributed.
6. **Random:** s_j sampled uniformly from $\{0, \ldots, N-1\}$.

We report representative results for system size $n = 5$ in Table 6.4. The extracted encoded block agrees with the expected matrix $M_s/\sqrt{d_r}$ to machine precision for all tested configurations.

Remark 6.5.1. The block encoding QFT is straightforward. A complete block encoding of the NUQFT matrix can be obtained by composing the components simulated above. We defer the full construction to future work, as the simulation becomes computationally expensive for large K.

#	Family	d_r	$\alpha = \sqrt{d_r}$	Max error	Spectral error
1	Identity	1	1.000000	0.0	0.0
2	Bit reversal	1	1.000000	0.0	0.0
3	Constant ($s_j = 0$)	32	5.656854	3.9×10^{-16}	1.7×10^{-15}
4	Constant ($s_j = N - 1$)	32	5.656854	3.9×10^{-16}	1.7×10^{-15}
5	Two cluster	16	4.000000	7.3×10^{-16}	2.0×10^{-15}
6	Heavy row	17	4.123106	6.9×10^{-16}	1.9×10^{-15}
7	Random	3	1.732051	1.6×10^{-14}	2.4×10^{-14}

Table 6.4: Numerical validation of the $M_{\tilde{s}}$ block encoding for $n = 5$.

6.6 Conclusion

We have developed a quantum algorithm for implementing the Type-II non-uniform discrete Fourier transform (NUDFT), which can be readily extended to the Type-I and Type-III variants. We have also presented a rigorous error analysis that identifies the dominant sources of approximation error and derives explicit upper bounds on their respective contributions. The analysis shows that the required computational resources scale logarithmically with the inverse of the target precision and polynomially with the number of qubits.

6.6.1 Future Directions

We discuss below several future research directions:

The proposed NUQFT algorithm assumes oracle access to the non-uniform sampling points (cf. Assumption 6.3.5) and to the corresponding inverse mapping (cf. Assumption 6.3.13). An important direction for future work is to identify concrete computational scenarios in which such oracles can be implemented efficiently, either through explicit circuit constructions or through data-generation models where the required access arises naturally.

The present framework relies on Chebyshev polynomial approximations implemented via

quantum signal processing. Alternative approximation strategies, including rational approximations, Fourier–Bessel expansions, or minimax constructions tailored to restricted domains, may offer improved constants or reduced ancilla usage. Exploring how such schemes can be incorporated into QSP or the more general QSVT framework may lead to more resource-efficient NUQFT implementations.

The analysis is carried out in a fault-tolerant setting with idealized primitives, including exact QFT implementations and perfect oracle access. Future work should address explicit compilation into standard fault-tolerant gate sets, with attention to optimizing T-count and T-depth, as well as adapting the construction to hardware constraints such as limited connectivity. Extending the framework to NISQ devices is another important direction. This includes incorporating approximate implementations of the Quantum Fourier Transform (QFT) and approximate rotations (see. Equation (6.23)) by eliminating small-angle rotations to reduce circuit depth.

The numerical experiments presented here are limited by the qubit counts and memory capacity of current quantum simulators and hardware. Future work may explore more scalable implementations. In particular, more resource-efficient realizations of key arithmetic primitives—especially the arccos subroutine and the LCU stage—could substantially reduce qubit and gate overhead. Possible directions include improved polynomial approximations, QSVT-based constructions, circuit-level optimizations, or hybrid LCU implementations. Such improvements could enable larger-scale realizations and provide a clearer evaluation of the practical performance of the proposed approach.

Many applications of non-uniform Fourier analysis arise in higher dimensions. Extending the NUQFT framework to multidimensional NUDFTs, for example via tensorized or separable

low-rank decompositions, is therefore a natural next step. More broadly, the techniques developed here may generalize to other non-uniform integral transforms, including cosine and Laplace transforms, discrete Legendre transforms, and Fourier transforms defined on manifolds.

Finally, the NUQFT is intended to function as a subroutine within larger quantum algorithms rather than as an isolated primitive. Studying its integration into complete quantum workflows—including state preparation, intermediate processing, and measurement—remains an important open problem, particularly in identifying application domains where such integration yields a provable advantage.

Appendix A: Omitted Proofs in Chapter 2

This appendix contains proofs omitted from Chapter 3.

A.1 Proof of Lemma 3.4.3

Proof. Since A_N is unitarily diagonalizable, $A_N = U_N D_N U_N^*$ with unitary U_N and diagonal D_N. Since $\|\cdot\|_F$ is unitarily invariant, we have

$$\begin{aligned}\|[A_N, X_N]\|_F = \|U_N^*[A_N, X_N]U_N\|_F = \|[U_N^* A_N U_N, U_N^* X_N U_N]\|_F = \|[D_N, U_N^* X_N U_N]\|_F,\\ \|[A_N, Y_N]\|_F = \|U_N^*[A_N, Y_N]U_N\|_F = \|[U_N^* A_N U_N, U_N^* Y_N U_N]\|_F = \|[D_N, U_N^* Y_N U_N]\|_F. \quad \text{(A.1)}\end{aligned}$$

Since $\|X_N\|_F = \Theta(\|Y_N\|_F)$, we also have that $\|U_N^* X_N U_N\|_F = \Theta(\|U_N^* Y_N U_N\|_F)$. Replacing $U_N^* X_N U_N$ by X_N and $U_N^* Y_N U_N$ by Y_N, it suffices to assume that A_N is diagonal and show that $\|[A_N, X_N]\|_F = \Theta(\|[A_N, Y_N]\|_F)$, where $A = \mathrm{diag}(\lambda_1, \cdots, \lambda_N)$. Note that we have $[A_N, X_N]_{ij} = (\lambda_i - \lambda_j)X_{ij}$. If $D_{A_N} \in \mathbb{C}^{N\times N}$ such that $(D_{A_N})_{ij} = \lambda_i - \lambda_j$, we can write $[A_N, X_N]$ and $[A_N, Y_N]$ as $[A_N, X_N] := D_{A_N} \odot X_N$ and $[A_N, Y_N] := D_{A_N} \odot Y_N$ respectively, where $\odot$ is the Hadamard product. Hence, it suffices to show that $\|D_{A_N} \odot X_N\|_F = \Theta(\|D_{A_N} \odot Y_N\|_F)$. Since

$(D_{A_N} \odot X_N)_{ij} = (\lambda_i - \lambda_j)(X_N)_{ij}$, we have

$$\Omega\left(\frac{\|X_N\|_F}{N^\alpha}\right) = \sqrt{\left(\sum_{i,j=1}^{N} |\lambda_i - \lambda_j|^2 |(X_N)_{ij}|^2\right)} = \mathcal{O}\left(\frac{\|X_N\|_F}{N^\alpha}\right). \tag{A.2}$$

Therefore, we have $\|D_{A_N} \odot X_N\| = \Theta(\|X_N\|_F/N^\alpha)$ and $\|D_{A_N} \odot Y_N\| = \Theta(\|Y_N\|_F/N^\alpha)$. The first claim follows in either case since $\|X_N\|_F = \Theta(\|Y_N\|_F)$. To prove the second claim, first note that the computations above show that

$$\|[B_N, [A_N, X_N]]\|_F = \Theta\left(\frac{\|[A_N, X_N]\|_F}{N^\beta}\right) = \Theta\left(\frac{\|X_N\|_F}{N^{\alpha+\beta}}\right) = \Theta\left(\frac{\|Y_N\|_F}{N^{\alpha+\beta}}\right) \tag{A.3}$$

Moreover, if we work in the setting where A_N is unitarily diagonalizable, we have

$$\begin{aligned} \|[A_N, [B_N, Y_N]]\|_F &= \|[D_{A_N}, [U_N^* B_N U_N, U_N^* Y_N U_N]]\|_F \\ &= \Theta\left(\frac{\|[U_N^* B_N U_N, U_N^* Y_N U_N]\|_F}{N^\alpha}\right) = \Theta\left(\frac{\|[B_N, Y_N]\|_F}{N^\alpha}\right) = \Theta\left(\frac{\|Y_N\|_F}{N^{\alpha+\beta}}\right). \end{aligned} \tag{A.4}$$

The second claim now follows immediately. □

Note that Lemma 3.4.3 can be extended straightforwardly to the case in which B_N is not necessarily diagonal in the computational basis. We omit the details, since this generalization is not needed for the application in Proposition 3.4.4.

Appendix B: Omitted Proofs in Chapter 3

This appendixß contains proofs omitted from Chapter 4.

B.1 Proof of Lemma 4.4.2

Proof. Substitute $k = s\sinh t$. Then $dk = s\cosh t\,dt$ and $\sqrt{k^2+s^2} = s\cosh t$. Hence,

$$\int_{\mathbb{R}} \frac{e^{-k^2/(4\sigma^2)}}{\sqrt{k^2+s^2}}\,dk = \int_{\mathbb{R}} e^{-s^2\sinh^2 t/(4\sigma^2)}\,dt. \tag{B.1}$$

Using $\sinh^2 t = (\cosh(2t)-1)/2$, we obtain

$$\int_{\mathbb{R}} e^{-s^2\sinh^2 t/(4\sigma^2)}\,dt = e^{s^2/(8\sigma^2)}\int_{\mathbb{R}} e^{-\frac{s^2}{8\sigma^2}\cosh(2t)}\,dt. \tag{B.2}$$

Setting $u = 2t$ gives

$$\int_{\mathbb{R}} e^{-\frac{s^2}{8\sigma^2}\cosh(2t)}\,dt = \frac{1}{2}\int_{\mathbb{R}} e^{-\frac{s^2}{8\sigma^2}\cosh u}\,du = \int_0^\infty e^{-\frac{s^2}{8\sigma^2}\cosh u}\,du = K_0\left(\frac{s^2}{8\sigma^2}\right), \tag{B.3}$$

where we have used $K_0(x) = \int_0^\infty e^{-x\cosh u}\,du$ for $x > 0$. This proves Equation (4.14). □

B.2 Proof of Lemma 4.4.3

Proof. By definition, we have

$$\log B(y_0) = (a-1)\log(b+1) - \log(y_0-1) - (a-1)\log(b+y_0) + d(1-y_0) + \frac{y_0^2-1}{4c^2}. \quad \text{(B.4)}$$

Differentiating with respect to y_0 gives

$$\frac{d}{dy_0}\log B(y_0) = -d + \frac{y_0}{2c^2} - \frac{1}{y_0-1} - \frac{a-1}{b+y_0}. \quad \text{(B.5)}$$

Hence, any stationary point y_0^* of $\log B$ satisfies $\frac{d}{dy_0}\log B(y_0^*) = 0$, equivalently Equation (4.17). Moreover,

$$\frac{d^2}{dy_0^2}\log B(y_0) = \frac{1}{2c^2} + \frac{1}{(y_0-1)^2} + \frac{a-1}{(b+y_0)^2} > 0. \quad \text{(B.6)}$$

Thus, $\log B$ is strictly convex for $y_0 > 1$, so any stationary point is its unique global minimizer. Since $B(y_0) > 0$ and $\frac{d}{dy_0}B(y_0) = B(y_0)\frac{d}{dy_0}\log B(y_0)$, B and $\log B$ have the same stationary points. They also have the same because exp is strictly increasing. Such a y_0 exists because $\frac{d}{dy_0}\log B(y_0) \to -\infty$ as $y_0 \to 1$ as $\frac{d}{dy_0}\log B(y_0) \to +\infty$ as $y_0 \to \infty$. Hence, by the intermediate value theorem, there is a unique $y_0^* \in (1,\infty)$ such that $\frac{d}{dy_0}\log B(y_0^*) = 0$. Thus, y_0^* is the unique stationary point of both $\log B(y_0)$ and $B(y_0)$. □

B.3 Proof of Corollary 4.4.5

Proof. We prove the three statements separately.

1. By Proposition 4.4.4, it suffices to prove $d \geq \Phi(s_\epsilon)$, where s_ϵ is the unique solution of

$\Psi(s_\epsilon) = \epsilon_{\text{approx}}$. For $s > 1$, define

$$Q(s) := \frac{c}{\sqrt{\pi}(s-1)} \exp\left(a - \frac{(s-1)^2}{4c^2}\right). \tag{B.7}$$

Note that $\Psi(s) \le Q(s)$ for $s > 1$. Moreover,

$$Q(\widetilde{x}_0) = \frac{c}{\sqrt{\pi}(\widetilde{x}_0 - 1)} \exp\left(a - \frac{(\widetilde{x}_0 - 1)^2}{4c^2}\right) = \frac{1}{\sqrt{2\pi\,\omega_\epsilon}} \exp\left(a - \frac{\omega_\epsilon}{2}\right) = \epsilon_{\text{approx}}, \tag{B.8}$$

since $\omega_\epsilon e^{\omega_\epsilon} = e^{2a}/(2\pi\epsilon_{\text{approx}}^2)$. Hence, $\Psi(\widetilde{x}_0) \le Q(\widetilde{x}_0) = \epsilon_{\text{approx}}$. Since Ψ is strictly decreasing, $s_\epsilon \le \widetilde{x}_0$. Since Φ is strictly increasing, Equation (4.23) implies $d \ge \Phi(\widetilde{x}_0) \ge \Phi(s_\epsilon)$. The conclusion follows from Proposition 4.4.4.

2. We have $e^{2a}/(2\pi\epsilon_{\text{approx}}^2) \le e^{2a}/\epsilon_{\text{approx}}^2$. Since $e^{2a}/\epsilon_{\text{approx}}^2 \ge e^2 > e$ because $\epsilon_{\text{approx}} \in (0,1)$, we have $W\left(\frac{e^{2a}}{2\pi\epsilon_{\text{approx}}^2}\right) \le W\left(\frac{e^{2a}}{\epsilon_{\text{approx}}^2}\right) \le \log\left(\frac{e^{2a}}{\epsilon_{\text{approx}}^2}\right) = 2a + 2\log(1/\epsilon_{\text{approx}})$. It follows that

$$\widetilde{x}_0 = 1 + c\sqrt{2W\left(\frac{e^{2a}}{2\pi\epsilon_{\text{approx}}^2}\right)} \le 1 + \sqrt{2}c\sqrt{2a + 2\log(1/\epsilon_{\text{approx}})} = \overline{x}_0. \tag{B.9}$$

Since Φ is strictly increasing, $d \ge \Phi(\overline{x}_0)$ implies $d \ge \Phi(\widetilde{x}_0)$. The claim follows from part (1).

3. Substituting $\overline{x}_0$ into Φ gives $\Phi(\overline{x}_0) = \frac{1}{2c^2} + \frac{1}{c}\sqrt{a + \log(1/\epsilon_{\text{approx}})} - \frac{1}{\overline{x}_0 - 1} - \frac{a-1}{b+\overline{x}_0}$. This gives the stated asymptotic scaling for fixed b and c.

This completes the proof. □

B.4 Proof of Lemma 4.5.7

Proof. Let $\overline{R}_Q := \max\{1, R_Q\}$. For $0 \le r \le \overline{R}_Q$, define $S_n(r) := \sum_{\ell=0}^{n} r^\ell \mathcal{C}_{n,\ell}(H,L)$ for $n \ge 1$. By definition of $\mathcal{C}_n(H,L)$, we have $S_n(r) \le (n+1)\overline{R}_Q^n \mathcal{C}_n(H,L)$ for $0 \le r \le \overline{R}_Q$. For $n \ge J$, the definition of $\chi_J(H,L)$ gives $\mathcal{C}_n(H,L) \le \chi_J(H,L)^{-n}$. Since $n \mapsto (n+1)^{1/n}$ is decreasing for $n \ge 1$, we also have $n+1 \le (J+1)^{n/J}$ for $n \ge J$. Combining these bounds gives

$$S_n(r) \le \left(\frac{(J+1)^{1/J} \overline{R}_Q}{\chi_J(H,L)} \right)^n = \kappa_Q^{-n} \tag{B.10}$$

for all $n \ge J$ and $0 \le r \le \overline{R}_Q$. The estimate in Equation (B.10) applies directly only for $n \ge J$. To absorb the finitely many lower orders, define

$$B_Q := \max\left\{ 1, \max_{1 \le n < J} \kappa_Q^n \sup_{0 \le r \le \overline{R}_Q} S_n(r) \right\}. \tag{B.11}$$

If $J = 1$, the second maximum is interpreted as absent. Then, $S_n(r) \le B_Q \kappa_Q^{-n}$ for every $n \ge 1$ and every $0 \le r \le \overline{R}_Q$. Now fix $m \ge 1$. For $0 \le r \le \overline{R}_Q$, define the pointwise profile

$$\Phi_m(r) := \sum_{\substack{j \in 2\mathbb{Z}_+ \\ j \ge 2m}} \sum_{l=1}^{m} \frac{\rho_Q^{j+l-(2m+1)}}{l!} \sum_{\substack{j_1, \dots, j_l \in 2\mathbb{Z}_+ \\ j_1 + \cdots + j_l = j}} \prod_{\kappa=1}^{l} S_{j_\kappa + 1}(r). \tag{B.12}$$

Setting $n_\kappa := j_\kappa + 1$, we have $n_1 + \cdots + n_l = j + l$. We obtain $\prod_{\kappa=1}^{l} S_{j_\kappa+1}(r) \le B_Q^l \kappa_Q^{-(j+l)}$. The number of ordered decompositions of an even integer j into l positive even parts is $\binom{j/2-1}{l-1}$.

Therefore, we have

$$\Phi_m(r) \leq \rho_Q^{-(2m+1)} \sum_{l=1}^{m} \frac{B_Q^l}{l!} \sum_{\substack{j \in 2\mathbb{Z}_+ \\ j \geq 2m}} \binom{j/2-1}{l-1} \left(\frac{\rho_Q}{\kappa_Q}\right)^{j+l}. \tag{B.13}$$

Since $0 < \rho_Q < \kappa_Q$, the final series converges. Indeed, the binomial factor in Equation (B.13) grows polynomially in j, while $(\rho_Q/\kappa_Q)^j$ decays geometrically. Hence, we have $\sup_{0 \leq r \leq \overline{R}_Q} \Phi_m(r) < \infty$. We next prove convergence of the coefficient series defining $\Phi_{m,\nu}(H, L)$. Expanding the product in Equation (B.12) gives $\Phi_m(r) = \sum_{\nu \geq 0} \Phi_{m,\nu}(H, L) r^\nu$ for $0 \leq r \leq \overline{R}_Q$. All coefficients in this expansion are non-negative. Since $\overline{R}_Q \geq 1$, evaluating at $r = 1$ gives $\sum_{\nu \geq 0} \Phi_{m,\nu}(H, L) = \Phi_m(1) < \infty$. Therefore, each $\Phi_{m,\nu}(H, L)$ is finite, and its defining series is absolutely convergent. In particular, $\Phi^*_{m,Q} = \max_{i \in \mathcal{I}_Q} \Phi_m(|k_i|) < \infty$ because $|k_i| \leq R_Q \leq \overline{R}_Q$. It remains to verify convergence of $\Lambda_{m,Q}$. Since all terms in Equation (4.69) are nonnegative, Tonelli's theorem permits rearranging the sums, giving $\Lambda_{m,Q} = \sum_{\nu \geq 0} \Phi_{m,\nu}(H, L) M_\nu^Q = \sum_{i \in \mathcal{I}_Q} |v_i| \, \Phi_m(|k_i|)$. Consequently, we have $\Lambda_{m,Q} \leq \alpha_Q \Phi^*_{m,Q} < \infty$. This proves absolute convergence of the series. □

B.5 Proof of Lemma 4.5.9

Proof. Writing $j = 2m + n$ and dropping the even restriction on n only enlarges the sum, so

$$S_m(x) \leq \sum_{n=0}^{\infty} \sum_{l=1}^{m} \frac{1}{l!} \binom{2m+n-1}{l-1} x^{n+l-1}. \tag{B.14}$$

Choose $\tau \in (0, \log(1/x))$, so that $xe^{\tau} < 1$, and set $K_\tau := \max\{1, \tau^{-1}\}$. For $0 \le l-1 \le m-1$, the elementary inequality $y^r/r! \le e^{\tau y}\tau^{-r}$ gives

$$\binom{2m+n-1}{l-1} \le \frac{(2m+n)^{l-1}}{(l-1)!} \le K_\tau^m e^{\tau(2m+n)}. \tag{B.15}$$

Substituting Equation (B.15) into Equation (B.14), we obtain

$$S_m(x) \le K_\tau^m e^{2\tau m}\left(\sum_{l=1}^{m}\frac{x^{l-1}}{l!}\right)\left(\sum_{n=0}^{\infty}(xe^{\tau})^n\right) \le \frac{e}{1-xe^{\tau}}(K_\tau e^{2\tau})^m. \tag{B.16}$$

Consequently, we have $\sup_{m\ge 1} S_m(x)^{1/(2m)} \le \left(\frac{e}{1-xe^{\tau}}\right)^{1/2} K_\tau^{1/2} e^{\tau} < \infty$. This completes the proof.

□

B.6 Proof of Lemma 4.5.10

Proof. By [Miz26, Equation 8], we have the crude bound $\alpha_{\mathrm{comm},p}(H) \le (p-1)!(2qg)^{p-1}Ng$ for $p \ge 2$. Let $j_1, \dots, j_l$ be an admissible tuple in the definition of $\mu_{m,p_0}(H)$. Thus, we have $2 \le j_\nu \le p_0 - 1$, $j_1 + \cdots + j_l = j$ and $1 \le l \le \lfloor j/2 \rfloor$. Since $j_\nu + 1 \le p_0$, the bound above gives $\alpha_{\mathrm{comm},j_\nu+1}(H) \le j_\nu!(2qg)^{j_\nu}Ng \le p_0^{j_\nu}(2qg)^{j_\nu}Ng = Ng(2qgp_0)^{j_\nu}$. Therefore, we have $\prod_{\nu=1}^{l}\alpha_{\mathrm{comm},j_\nu+1}(H) \le (Ng)^l(2qgp_0)^{j_1+\cdots+j_l} = (Ng)^l(2qgp_0)^j$. The number of admissible l-tuples is bounded by $(1+p_0)^l$. Taking the $1/(j+l)$-th root gives

$$\left(\sum_{\substack{2\le j_1,\dots,j_l\le p_0-1\\ j_1+\cdots+j_l=j}}\prod_{\nu=1}^{l}\alpha_{\mathrm{comm},j_\nu+1}(H)\right)^{1/(j+l)} \le (2qgp_0)^{j/(j+l)}\left((1+p_0)Ng\right)^{l/(j+l)}. \tag{B.17}$$

Since $l \le \lfloor j/2 \rfloor$, we have $\frac{l}{j+l} \le \frac{1}{3}$ and $\frac{j}{j+l} \le 1$. Using $p_0 \le 1 + p_0$, we obtain

$$(2qgp_0)^{j/(j+l)} \left((1+p_0)Ng\right)^{l/(j+l)} \le \max\{1, 2qg\} \max\{1, Ng\}^{1/3} (1+p_0)^{4/3}. \tag{B.18}$$

Taking the supremum over all admissible j, l in the definition of $\mu_{m,p_0}(H)$ proves the first inequality. If $\{G_{k_i} : i \in \mathcal{I}_Q\}$ is uniformly q_Q-local and g_Q-extensive, then the same argument applies uniformly to each G_{k_i}. Taking the maximum over $i \in \mathcal{I}_Q$ proves the second inequality. □

B.7 Proof of Lemma 4.5.11

Proof. Let $U_z(T) := e^{-i(H+zL)T}$. By [LS25, Lemma 3], the map $z \mapsto U_z(T)$ is entire. The poles of $\hat{f}_{a,b}(z; c, d)$ occur at $z = -i$ and, when $a > 1$, at $z = ib$. Hence, if $\rho < \rho_*$, then F_T is analytic on an open set containing $\overline{S}_\rho$. Let $z = x + i\beta$, with $|\beta| \le \rho$. Since $1 - iz = (1+\beta) - ix$ and $b + iz = (b-\beta) + ix$, we have $|1 - iz| = \sqrt{(1+\beta)^2 + x^2}$ and $|b + iz| = \sqrt{(b-\beta)^2 + x^2}$. Moreover, $|e^{d(1-iz)}| = e^{d(1+\beta)}$ and $|e^{-(z^2+1)/(4c^2)}| = e^{(\beta^2-1)/(4c^2)} e^{-x^2/(4c^2)}$. By [LS25, Lemma 4], applied in the time-independent case and specialized to $L \succeq 0$, we have $\|U_{x+i\beta}(T)\| \le e^{\max\{\beta,0\}T\|L\|}$. Therefore,

$$\|F_T(x+i\beta)\| \le \frac{(b+1)^{a-1}}{2\pi} \frac{\exp\left(d(1+\beta) + \frac{\beta^2-1}{4c^2} + \max\{\beta, 0\}T\|L\|\right) e^{-x^2/(4c^2)}}{\sqrt{(1+\beta)^2 + x^2}\left((b-\beta)^2 + x^2\right)^{(a-1)/2}}. \tag{B.19}$$

Since $|\beta| \le \rho < 1$, we have $\sqrt{(1+\beta)^2 + x^2} \ge 1 - \rho$. If $a > 1$, then $\rho < b$, and therefore $\left((b-\beta)^2 + x^2\right)^{(a-1)/2} \ge (b-\rho)^{a-1}$. Thus,

$$\|F_T(x+i\beta)\| \le \frac{(b+1)^{a-1}}{2\pi(1-\rho)D_{a,b}(\rho)} \exp\left(d(1+\beta) + \frac{\beta^2-1}{4c^2} + \max\{\beta, 0\}T\|L\|\right) e^{-x^2/(4c^2)}. \tag{B.20}$$

For $\beta \in [-\rho, \rho]$, the exponent satisfies

$$d(1+\beta) + \frac{\beta^2 - 1}{4c^2} + \max\{\beta, 0\} T \|L\| \le d + \frac{\rho^2 - 1}{4c^2} + \rho \max\{-d, d + T\|L\|\}. \tag{B.21}$$

Indeed, the left-hand side is convex in β, so its maximum on $[-\rho, \rho]$ occurs at $\beta = \pm\rho$. Integrating over $\mathbb{R}$ and using $\int_{\mathbb{R}} e^{-x^2/(4c^2)}\, dx = 2c\sqrt{\pi}$ gives the L^1 bound with $M_{a,b}(\rho)$. The same Gaussian upper bound also shows that $F_T(x + i\beta)$ decays uniformly to zero on $\overline{S}_\rho$ as $|x| \to \infty$. □

B.8 Proof of Corollary 4.5.17

Proof. The choice of h controls the mesh error. Indeed, Lemma 4.5.15 gives $E_{\sinh,\eta}(h, \beta) \le \epsilon_{\text{mesh}}$. It remains to control the tail contribution. Since $Y_h \ge Y$, the choice of Y gives $e^{-\lambda_\eta^2 \sinh^2 Y_h} \le e^{-\lambda_\eta^2 \sinh^2 Y} \le \frac{\epsilon_{\text{tail}}}{2c\sqrt{\pi} C_{\ker}}$. Moreover, the choice of Y also gives $\lambda_\eta \sinh Y_h \ge 1$, so Proposition 4.5.16 implies $E^{\sinh,\eta}_{\text{tail},h}(Y_h) \le \epsilon_{\text{tail}}$. Combining the mesh and tail estimates, we obtain

$$E_{\text{quad}}(Q^{\sinh,\eta}_{Y,h}) \le E_{\sinh,\eta}(h, \beta) + E^{\sinh,\eta}_{\text{tail},h}(Y_h) \le \epsilon_{\text{mesh}} + \epsilon_{\text{tail}}. \tag{B.22}$$

Since $n_h = \lceil Y/h \rceil$, we have $|\mathcal{I}_{Q^{\sinh,\eta}_{Y,h}}| = 2n_h + 1 \le 2Y/h + 3$. Hence, $\frac{1}{h} = \max\left\{1, \frac{\ell^{\sinh,\eta}_{\text{mesh}}}{2\pi\beta}\right\} = \mathcal{O}\left(1 + \frac{\ell^{\sinh,\eta}_{\text{mesh}}}{\beta}\right)$, while $Y = \mathcal{O}\left(\log\left(1 + \frac{\sqrt{\ell^{\sinh,\eta}_{\text{tail}}}}{\lambda_\eta}\right)\right)$. Substituting these estimates gives the bound on $|\mathcal{I}_{Q^{\sinh,\eta}_{Y,h}}|$ in Equation (4.113). Moreover, $Y_h = n_h h \le Y + h \le Y + 1$, and therefore

$$\sinh(Y_h) \le \sinh(Y+1) = \mathcal{O}(\sinh Y + \cosh Y) = \mathcal{O}\left(1 + \frac{\sqrt{\ell^{\sinh,\eta}_{\text{tail}}}}{\lambda_\eta}\right). \tag{B.23}$$

This gives $R_{Q_{Y,h}^{\sinh,\eta}} = \eta \sinh(Y_h) = \mathcal{O}\left(\eta + \frac{\eta}{\lambda_\eta}\sqrt{\ell_{\text{tail}}^{\sinh,\eta}}\right) = \mathcal{O}\left(\eta + 2c\sqrt{\ell_{\text{tail}}^{\sinh,\eta}}\right)$, which proves the radius bound in Equation (4.114). The balanced-budget estimates follow from $\ell_{\text{mesh}}^{\sinh,\eta} = \ell_{\text{tail}}^{\sinh,\eta} = \mathcal{O}(\log(1/\epsilon_q))$. It remains to bound the LCU normalization and the discrete moments. For real k, Equation (4.85) implies $\frac{1}{\sqrt{2\pi}}|\hat{f}_{a,b}(k;c,d)| = \widetilde{C}_{\text{ker}} \frac{e^{-k^2/(4c^2)}}{\sqrt{1+k^2}(b^2+k^2)^{(a-1)/2}}$. For the free-scale sinh rule, $k_i = \eta \sinh(qh)$, and hence

$$M_{\nu,Y,h}^{\sinh,\eta} \le \widetilde{C}_{\text{ker}} h \sum_{q\in\mathbb{Z}} \eta \cosh(qh) \frac{e^{-\lambda_\eta^2 \sinh^2(qh)} |\eta \sinh(qh)|^\nu}{\sqrt{1+\eta^2\sinh^2(qh)}\left(b^2+\eta^2\sinh^2(qh)\right)^{(a-1)/2}}. \tag{B.24}$$

For $k > 0$, we have

$$\frac{1}{\sqrt{1+k^2}(b^2+k^2)^{(a-1)/2}} \le \min\left\{\frac{1}{b^{a-1}}, \frac{1}{k^a}\right\}. \tag{B.25}$$

Indeed, the first bound follows from $\sqrt{1+k^2} \ge 1$ and $(b^2+k^2)^{(a-1)/2} \ge b^{a-1}$, while the second follows from $\sqrt{1+k^2} \ge k$ and $(b^2+k^2)^{(a-1)/2} \ge k^{a-1}$. Let

$$\kappa_b := \begin{cases} 1, & a = 1, \\ b^{(a-1)/a}, & a > 1. \end{cases} \tag{B.26}$$

We split the full-line sum in Equation (B.24) into the regions $|\eta \sinh(qh)| \le \kappa_b$ and $|\eta \sinh(qh)| > \kappa_b$. Since $h \le 1$, the contribution from $|\eta \sinh(qh)| \le \kappa_b$ is bounded by a constant depending only on ν, a, b. More precisely,

$$h \sum_{|\eta\sinh(qh)|\le\kappa_b} \eta \cosh(qh) \frac{e^{-\lambda_\eta^2 \sinh^2(qh)} |\eta \sinh(qh)|^\nu}{\sqrt{1+\eta^2\sinh^2(qh)}\left(b^2+\eta^2\sinh^2(qh)\right)^{(a-1)/2}} \le C_{\nu,a,b}. \tag{B.27}$$

On the complementary region, Equation (B.25) gives

$$h \sum_{|\eta \sinh(qh)|>\kappa_b} \eta \cosh(qh) \frac{e^{-\lambda_\eta^2 \sinh^2(qh)} |\eta \sinh(qh)|^\nu}{\sqrt{1+\eta^2 \sinh^2(qh)} \left(b^2+\eta^2 \sinh^2(qh)\right)^{(a-1)/2}} \tag{B.28}$$

$$\leq h \sum_{|\eta \sinh(qh)|>\kappa_b} \eta \cosh(qh) e^{-\lambda_\eta^2 \sinh^2(qh)} |\eta \sinh(qh)|^{\nu-a}. \tag{B.29}$$

This transformed Riemann sum has the same three regimes as the corresponding continuous kernel moment. Indeed, using the change of variables $k = \eta \sinh x$, $dk = \eta \cosh x \, dx$, the continuous analogue is $\int_{\kappa_b}^\infty e^{-k^2/(4c^2)} k^{\nu-a} \, dk$. The full-line Riemann sums with $0 < h \leq 1$ are controlled by the corresponding integrals plus a constant multiple of the maximal summand, and hence have the same scaling. Therefore, if $\nu < a-1$, then $h \sum_{|\eta \sinh(qh)|>\kappa_b} \eta \cosh(qh) e^{-\lambda_\eta^2 \sinh^2(qh)} |\eta \sinh(qh)|^{\nu-a} \leq C_{\nu,a,b}$. If $\nu = a-1$, then $h \sum_{|\eta \sinh(qh)|>\kappa_b} \eta \cosh(qh) e^{-\lambda_\eta^2 \sinh^2(qh)} |\eta \sinh(qh)|^{-1} \leq C_{a,b}\big(1+\log(1+c)\big)$. If $\nu > a-1$, then comparison with the corresponding Gaussian integral gives

$$h \sum_{|\eta \sinh(qh)|>\kappa_b} \eta \cosh(qh) e^{-\lambda_\eta^2 \sinh^2(qh)} |\eta \sinh(qh)|^{\nu-a} \leq C_{\nu,a,b} + C_\nu \int_0^\infty e^{-k^2/(4c^2)} k^{\nu-a} \, dk \tag{B.30}$$

$$\leq C_{\nu,a,b} + C_\nu c^{\nu-a+1}. \tag{B.31}$$

Combining these three cases with Equation (B.24) proves the moment bound. The normalization bound is the case $\nu = 0$, and the asymptotic statement follows directly from the definition of $\mathfrak{M}_{\nu,a,b}(c)$. □

B.9 Proof of Lemma 4.7.2

Proof. Let $b(x) := 4\sum_{\alpha=1}^{d} \sin^2(x_\alpha/2)$ and $a_s(x) := b(x)^s$ for $x \in \mathbb{T}^d$. We claim that a_s satisfies the β_s-Hölder condition. Define $F(x) := (2|\sin(x_\alpha/2)|)_{\alpha=1}^d$. Then $b(x)^{1/2} = |F(x)|$. Since each map $x_\alpha \mapsto 2|\sin(x_\alpha/2)|$ is Lipschitz on $\mathbb{T}$, we have $|F(x) - F(y)| \le C_d|x-y|_{\text{per}}$ for some $C_d > 0$. Using the reverse triangle inequality, we have $|b(x)^{1/2} - b(y)^{1/2}| \le C_d|x-y|_{\text{per}}$. We consider two cases:

1. If $0 < s \le 1/2$, then $0 < 2s \le 1$, and the map $r \mapsto r^{2s}$ is $2s$-Hölder on $[0,\infty)$. Hence, we have

$$|a_s(x) - a_s(y)| = |(b(x)^{1/2})^{2s} - (b(y)^{1/2})^{2s}| \le |b(x)^{1/2} - b(y)^{1/2}|^{2s} \le C'_{s,d}|x-y|_{\text{per}}^{2s}, \quad \text{(B.32)}$$

   where $C'_{s,d} = C_d^{2s}$.

2. If $1/2 \le s < 1$, then $2s \ge 1$. Note that $b(x)^{1/2} \le 2\sqrt{d}$ and the map $r \mapsto r^{2s}$ is Lipschitz on $[0, 2\sqrt{d}]$. Therefore, we have

$$|a_s(x) - a_s(y)| = |(b(x)^{1/2})^{2s} - (b(y)^{1/2})^{2s}| \le C''_{s,d}|b(x)^{1/2} - b(y)^{1/2}| \le C'_{s,d}|x-y|_{\text{per}}, \quad \text{(B.33)}$$

   where $C'_{s,d} = C''_{s,d}C_d$

Combining the two cases gives $|a_s(x) - a_s(y)| \le C'_{s,d}|x-y|_{\mathrm{per}}^{\beta_s}$, where $\beta_s = \min\{2s, 1\}$ and $C'_{s,d} > 0$ is independent of N. Taking $x = 2\pi p/M$ and $y = 2\pi q/M$ gives

$$|\lambda_p^s - \lambda_q^s| = M^{2s}\left|a_s\left(\frac{2\pi p}{M}\right) - a_s\left(\frac{2\pi q}{M}\right)\right| \tag{B.34}$$

$$\le (2\pi)^{\beta_s} C'_{s,d} M^{2s-\beta_s}|p-q|_{\mathrm{per}}^{\beta_s} := C_{s,d}M^{\theta_s}|p-q|_{\mathrm{per}}^{\beta_s} = C_{s,d}N^{\theta_s/d}|p-q|_{\mathrm{per}}^{\beta_s}, \tag{B.35}$$

where $C_{s,d} = (2\pi)^{\beta_s}C'_{s,d}$. This completes the proof. □

B.10 Proof of Proposition 4.7.4

Proof. For $a \ge 0$, define the following weighted row-column norm in the Fourier basis:

$$\|A\|_{a,\mathcal{F}} := \max\left\{\max_{p\in\mathbb{Z}_M^d}\sum_{q\in\mathbb{Z}_M^d}(1+|p-q|_{\mathrm{per}})^a\,|(F_NAF_N^\dagger)_{p,q}|, \max_{q\in\mathbb{Z}_M^d}\sum_{p\in\mathbb{Z}_M^d}(1+|p-q|_{\mathrm{per}})^a\,|(F_NAF_N^\dagger)_{p,q}|\right\}. \tag{B.36}$$

It can be checked that Equation (B.36) indeed defines a norm. For $a = 0$, this norm controls the spectral norm. Indeed, since F_N is unitary and $\|A\| \le \sqrt{\|A\|_1\|A\|_\infty}$ for any matrix A, we have

$$\|A\| = \|F_NAF_N^\dagger\| \le \sqrt{\|F_NAF_N^\dagger\|_1\|F_NAF_N^\dagger\|_\infty} \le \|A\|_{0,\mathcal{F}}. \tag{B.37}$$

We first record the two commutator-type operator estimates used below.

1. Recall from Equation (4.152) that, for every matrix A, we have $(F_N\,\mathrm{ad}_{L_s}(A)F_N^\dagger)_{p,q} = (\lambda_p^s - \lambda_q^s)(F_NAF_N^\dagger)_{p,q}$. Applying the weaker consequence of Lemma 4.7.2 from Equation (4.157), we obtain

$$\|\,\mathrm{ad}_{L_s}(A)\|_{a,\mathcal{F}} \le C_{s,d}N^{\theta_s/d}\|A\|_{a+\beta_s,\mathcal{F}}. \tag{B.38}$$

Therefore, ad_{L_s} maps the Fourier-weighted norm with exponent $a+\beta_s$ into the corresponding norm with exponent a, at the cost of the factor $C_{s,d}N^{\theta_s/d}$.

2. Next, since $F_N H F_N^\dagger$ has entries $\widehat{V}_{p-q}$, multiplication by H becomes convolution in the Fourier basis. We use the submultiplicative weight estimate $1+|p-q|_{\mathrm{per}} \le \big(1+|p-r|_{\mathrm{per}}\big)\big(1+|r-q|_{\mathrm{per}}\big)$, which follows from the triangle inequality for the periodic distance. We claim that $\|HA\|_{a,\mathcal{F}} \le \mathcal{V}_a^{(N)}\|A\|_{a,\mathcal{F}}$ and $\mathcal{V}_a^{(N)}\|A\|_{a,\mathcal{F}}$. Indeed, we have

$$(F_N HAF_N^\dagger)_{p,q} = ((F_N HF_N^\dagger)(F_N AF_N^\dagger))_{p,q} = \sum_{r\in\mathbb{Z}_M^d} \widehat{V}_{p-r}(F_N AF_N^\dagger)_{r,q}. \tag{B.39}$$

Hence, we have

$$\sum_{q\in\mathbb{Z}_M^d} (1+|p-q|_{\mathrm{per}})^a |(F_N HAF_N^\dagger)_{p,q}| \tag{B.40}$$

$$\le \sum_{r\in\mathbb{Z}_M^d} (1+|p-r|_{\mathrm{per}})^a |\widehat{V}_{p-r}| \sum_{q\in\mathbb{Z}_M^d} (1+|r-q|_{\mathrm{per}})^a |(F_N AF_N^\dagger)_{r,q}| \le \mathcal{V}_a^{(N)}\|A\|_{a,\mathcal{F}}. \tag{B.41}$$

Taking the maximum over p gives the row estimate for HA. The column estimate is identical. The bound for AH follows in the same way. Thus, the desired inequalities hold. Consequently, we have

$$\|\,\mathrm{ad}_H(A)\|_{a,\mathcal{F}} = \|[H,A]\|_{a,\mathcal{F}} \le \|HA\|_{a,\mathcal{F}} + \|AH\|_{a,\mathcal{F}} \le 2\mathcal{V}_a^{(N)}\|A\|_{a,\mathcal{F}}. \tag{B.42}$$

Therefore, ad_H is bounded on each weighted Fourier norm and contributes no explicit $N^{\theta_s/d}$ factor.

We now apply these two estimates to $W_{j,\ell}(H, L_s) = [A_j, [A_{j-1}, \ldots, [A_2, A_1]\cdots]]$ for $A_i \in \{H, L_s\}$. For $m = 1, \ldots, j-1$, define the partial commutator $W^{(m)} := [A_{m+1}, [A_m, \ldots, [A_2, A_1]\cdots]]$. Equivalently, $W^{(m)} = \mathrm{ad}_{A_{m+1}}(W^{(m-1)})$ for $m = 2, \ldots, j-1$. Let $n_m := m+1$, ℓ_m be the number of occurrences of L_s among $A_1, \ldots, A_{m+1}$, and let $h_m := n_m - \ell_m$ be the number of occurrences of H among the same operators. We prove by induction on m that, for every $a \geq 0$,

$$\|W^{(m)}\|_{a,\mathcal{F}} \leq 2^{h_m} C_{s,d}^{\ell_m} (\mathcal{V}^{(N)}_{a+n_m\beta_s})^{h_m} N^{\ell_m \theta_s/d}. \tag{B.43}$$

1. Let $m = 1$. We have $W^{(1)} = [A_2, A_1]$. If $A_2 = A_1$, then $W^{(1)} = 0$, so the estimate is immediate. Otherwise, $W^{(1)} = \pm[L_s, H] = \pm \mathrm{ad}_{L_s}(H)$. Using Equation (B.38), we obtain $\|W^{(1)}\|_{a,\mathcal{F}} \leq C_{s,d} N^{\theta_s/d} \|H\|_{a+\beta_s,\mathcal{F}}$. Since $F_N H F_N^\dagger$ has entries $\widehat{V}_{p-q}$, we have $\|H\|_{a+\beta_s,\mathcal{F}} \leq \mathcal{V}^{(N)}_{a+\beta_s} \leq \mathcal{V}^{(N)}_{a+2\beta_s}$. In the non-zero base case, $n_1 = 2$, $\ell_1 = 1$, and $h_1 = 1$. Hence, we obtain Equation (B.43). This proves the base case.

2. Assume Equation (B.43) holds for $W^{(m-1)}$, where $2 \leq m \leq j-1$. Since $W^{(m)} = \mathrm{ad}_{A_{m+1}}(W^{(m-1)})$, there are two cases.

   (a) If $A_{m+1} = L_s$, then Equation (B.38) gives $\|W^{(m)}\|_{a,\mathcal{F}} = \|\mathrm{ad}_{L_s}(W^{(m-1)})\|_{a,\mathcal{F}} \leq C_{s,d} N^{\theta_s/d} \|W^{(m-1)}\|_{a+\beta_s,\mathcal{F}}$. Applying the induction hypothesis to $W^{(m-1)}$ with weight $a + \beta_s$, we obtain

$$\|W^{(m)}\|_{a,\mathcal{F}} \leq C_{s,d} N^{\theta_s/d} 2^{h_{m-1}} C_{s,d}^{\ell_{m-1}} (\mathcal{V}^{(N)}_{a+\beta_s+n_{m-1}\beta_s})^{h_{m-1}} N^{\ell_{m-1}\theta_s/d}. \tag{B.44}$$

   Because $A_{m+1} = L_s$, we have $\ell_m = \ell_{m-1} + 1, h_m = h_{m-1}$ and $n_m = n_{m-1} + 1$. Thus, $a+\beta_s+n_{m-1}\beta_s = a+n_m\beta_s$. Hence, we have $\|W^{(m)}\|_{a,\mathcal{F}} \leq 2^{h_m} C_{s,d}^{\ell_m} (\mathcal{V}^{(N)}_{a+n_m\beta_s})^{h_m} N^{\ell_m\theta_s/d}$.

(b) If $A_{m+1} = H$, then Equation (B.42) gives $\|W^{(m)}\|_{a,\mathcal{F}} = \|\operatorname{ad}_H(W^{(m-1)})\|_{a,\mathcal{F}} \leq 2\mathcal{V}_a^{(N)}\|W^{(m-1)}\|_{a,\mathcal{F}}$

Applying the induction hypothesis gives

$$\|W^{(m)}\|_{a,\mathcal{F}} \leq 2\mathcal{V}_a^{(N)} 2^{h_{m-1}} C_{s,d}^{\ell_{m-1}} (\mathcal{V}_{a+n_{m-1}\beta_s}^{(N)})^{h_{m-1}} N^{\ell_{m-1}\theta_s/d}. \tag{B.45}$$

Because $A_{m+1} = H$, we have $\ell_m = \ell_{m-1}$, $h_m = h_{m-1} + 1$ and $n_m = n_{m-1} + 1$. Moreover, $\mathcal{V}_b^{(N)}$ is non-decreasing in b. Therefore, we have $\mathcal{V}_a^{(N)} \leq \mathcal{V}_{a+n_m\beta_s}^{(N)}$ and $\mathcal{V}_{a+n_{m-1}\beta_s}^{(N)} \leq \mathcal{V}_{a+n_m\beta_s}^{(N)}$. Hence, it follows that $\|W^{(m)}\|_{a,\mathcal{F}} \leq 2^{h_m} C_{s,d}^{\ell_m} (\mathcal{V}_{a+n_m\beta_s}^{(N)})^{h_m} N^{\ell_m\theta_s/d}$. This completes the induction.

Taking $m = j-1$ and $a = 0$, we have $n_{j-1} = j$, $\ell_{j-1} = \ell$ and $h_{j-1} = j - \ell$. Thus Equation (B.43) gives $\|W_{j,\ell}(H, L_s)\|_{0,\mathcal{F}} \leq 2^{j-\ell} C_{s,d}^{\ell} (\mathcal{V}_{j\beta_s}^{(N)})^{j-\ell} N^{\ell\theta_s/d}$. The claim follows since $\|W_{j,\ell}(H, L_s)\| \leq \|W_{j,\ell}(H, L_s)\|_{0,\mathcal{F}}$. □

B.11 Proof of Lemma 4.7.7

Proof. For $0 \leq p, q \leq r$, we have $(S^T)^p P_0 S^q = e_p e_q^T$. Indeed, since $S = \sum_{j=0}^{M-1} e_j e_{j+1}^T$, we have $S e_{j+1} = e_j$ and $S^T e_j = e_{j+1}$. Thus, $(S^T)^p e_0 = e_p$. Also, we have $e_0^T S^q = e_q^T$. Since $P_0 = e_0 e_0^T$, it follows that $(S^T)^p P_0 S^q = (S^T)^p e_0 e_0^T S^q = e_p e_q^T$. Similarly, we have $S^p P_M (S^T)^q = e_{M-p} e_{M-q}^T$. Each non-zero matrix of the form $e_p e_q^T$ has operator norm one. Therefore, for every representation $X = \sum_\eta c_\eta E_\eta$ the triangle inequality gives $\|X\| \leq \sum_\eta |c_\eta|$. Taking the infimum over all such representations proves $\|X\| \leq \|X\|_{\mathcal{B}}$. We now show that for every $X \in \mathcal{B}_r$, we have $[A_1, X] \in \mathcal{B}_{r+1}$, $[B_1, X] \in \mathcal{B}_{r+1}$. It suffices to check it on the generators of $\mathcal{B}_r$. First consider $(S^T)^p P_0 S^q = e_p e_q^T$ for some $0 \leq p, q \leq r$. Since $S e_k = e_{k-1}$ for $1 \leq k \leq M$ and $S e_0 = 0$, while

$S^T e_k = e_{k+1}$ for $0 \le k \le M-1$ and $S^T e_M = 0$, we have

$$Se_p e_q^T = \begin{cases} e_{p-1}e_q^T, & p \ge 1, \\ 0, & p = 0, \end{cases} \qquad S^T e_p e_q^T = \begin{cases} e_{p+1}e_q^T, & p \le M-1, \\ 0, & p = M. \end{cases} \tag{B.46}$$

Similarly, using $e_q^T S = e_{q+1}^T$ for $q \le M-1$, $e_M^T S = 0$, $e_q^T S^T = e_{q-1}^T$ for $q \ge 1$, and $e_0^T S^T = 0$, we obtain

$$e_p e_q^T S = \begin{cases} e_p e_{q+1}^T, & q \le M-1, \\ 0, & q = M, \end{cases} \qquad e_p e_q^T S^T = \begin{cases} e_p e_{q-1}^T, & q \ge 1, \\ 0, & q = 0. \end{cases} \tag{B.47}$$

Therefore each of $Se_p e_q^T, e_p e_q^T S, S^T e_p e_q^T, e_p e_q^T S^T$ is either zero or a matrix unit $e_{p'} e_{q'}^T$ with $0 \le p', q' \le r+1$, and hence belongs to $\mathcal{B}_{r+1}$. The case $S^p P_M (S^T)^q = e_{M-p} e_{M-q}^T$ is similar. Since every $X \in \mathcal{B}_r$ is a linear combination of the generators $(S^T)^p P_0 S^q$ and $S^p P_M (S^T)^q$, we have that $SX,\ XS,\ S^T X,\ XS^T \in \mathcal{B}_{r+1}$. Using the definitions of A_1 and B_1, we have $[A_1, X] = -SX - S^T X + XS + XS^T$ and $[B_1, X] = SX - S^T X - XS + XS^T$. Hence, $[A_1, X]$ and $[B_1, X]$ belong to $\mathcal{B}_{r+1}$ for each $X \in \mathcal{B}_r$. If $X = \sum_\eta c_\eta E_\eta$ is any representation of X, then

$$SX = \sum_\eta c_\eta S E_\eta, \quad S^T X = \sum_\eta c_\eta S^T E_\eta, \quad XS = \sum_\eta c_\eta E_\eta S, \quad XS^T = \sum_\eta c_\eta E_\eta S^T. \tag{B.48}$$

By the argument above, each $SE_\eta, S^T E_\eta, E_\eta S, E_\eta S^T$ is either zero or in $\mathcal{B}_{r+1}$. Hence each of the four shifted matrices admits a representation in $\mathcal{B}_{r+1}$ with coefficient cost at most $\sum_\eta |c_\eta|$. Taking the infimum over all representations of X gives $\|SX\|_\mathcal{B},\ \|S^T X\|_\mathcal{B},\ \|XS\|_\mathcal{B},\ \|XS^T\|_\mathcal{B} \le \|X\|_\mathcal{B}$. Using the sub-additivity of $\|\cdot\|_\mathcal{B}$ gives $\|[A_1, X]\|_\mathcal{B} \le 4\|X\|_\mathcal{B}$ and $\|[B_1, X]\|_\mathcal{B} \le 4\|X\|_\mathcal{B}$. By Lemma 4.7.6, we have $[A_1, B_1] = 2(P_0 - P_M)$. Since $P_0, P_M \in \mathcal{B}_0$, this implies $\|[A_1, B_1]\|_\mathcal{B} \le 4$

and $\|[B_1, A_1]\|_{\mathcal{B}} \leq 4$. Every non-zero right-nested mixed commutator of length $j \geq 2$ is obtained from $[A_1, B_1]$ or $[B_1, A_1]$ by $j-2$ additional commutations with A_1 or B_1. Applying the argument above inductively gives $\|W_j(A_1, B_1)\|_{\mathcal{B}} \leq 4 \cdot 4^{j-2} = 4^{j-1}$. Hence, we have $\|W_j(A_1, B_1)\| \leq \|W_j(A_1, B_1)\|_{\mathcal{B}} \leq 4^{j-1}$. This completes the proof. $\square$

Bibliography

[AG22] Ahmed Abdeljawad and Philipp Grohs. "Approximations with deep neural networks in Sobolev time-space." In: *Analysis and Applications* 20.03 (2022), pp. 499–541 (cit. on p. 220).

[AS64] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions*. US Government Printing Office, 1964 (cit. on p. 260).

[AL99] Daniel S. Abrams and Seth Lloyd. "Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors." In: *Physical Review Letters* 83.24 (1999), p. 5162 (cit. on p. 224).

[Aft25] Junaid Aftab. *MPF-Commutator-Scaling*. https://github.com/Junaid-Aftab/MPF_Commutator_Scaling. 2025 (cit. on pp. 64, 93, 97).

[Aft26] Junaid Aftab. *Non-Uniform-QFT*. https://github.com/Junaid-Aftab/Non-Uniform-QFT. 2026 (cit. on pp. 223, 263).

[AAT24] Junaid Aftab, Dong An, and Konstantina Trivisa. *Multi-product Hamiltonian simulation with explicit commutator scaling*. Submitted to *Communications in Mathematical Physics*. 2024. arXiv: 2403.08922 [quant-ph] (cit. on pp. 64, 105).

[AAT26] Junaid Aftab, Dong An, and Konstantina Trivisa. *Linear Combination of Hamiltonian Simulation with Commutator Scaling*. 2026. arXiv: 2606.11475 [quant-ph] (cit. on p. 104).

[AKY26] Junaid Aftab, Yuehaw Khoo, and Haizhao Yang. *Non-Uniform Quantum Fourier Transform*. 2026. arXiv: 2602.13472 [quant-ph]. URL: https://arxiv.org/abs/2602.13472 (cit. on p. 223).

[Aft+25] Junaid Aftab, Christoph Schwab, Haizhao Yang, and Jakob Zech. *Quantum Circuit Encodings of Polynomial Chaos Expansions*. Submitted to *Quantum*. 2025. arXiv: 2506.01811 [math.NA]. URL: https://arxiv.org/abs/2506.01811 (cit. on pp. 181, 219).

[AY25] Junaid Aftab and Haizhao Yang. "Approximating Korobov functions via quantum circuits." In: *Communications in Mathematical Sciences* 23.8 (2025), pp. 2077–

2101. DOI: 10.4310/CMS.250930023153. URL: https://dx.doi.org/10.4310/CMS.250930023153 (cit. on pp. 181, 236).

[Agu+23] Lucía Agud Albesa, Marta Boix García, Laura Pla-Ferrando, and Salvador C. Cardona. "A study about the solution of convection-diffusion-reaction equation with Danckwerts boundary conditions by analytical, method of lines and Crank–Nicholson techniques." In: *Mathematical Methods in the Applied Sciences* (2023). DOI: 10.1002/mma.8633 (cit. on p. 167).

[Ale+24] Michel Alexis, Lin Lin, Gevorg Mnatsakanyan, Christoph Thiele, and Jiasu Wang. *Infinite quantum signal processing for arbitrary Szego functions*. 2024. arXiv: 2407.05634 [quant-ph]. URL: https://arxiv.org/abs/2407.05634 (cit. on p. 221).

[AMT24] Michel Alexis, Gevorg Mnatsakanyan, and Christoph Thiele. "Quantum signal processing and nonlinear Fourier analysis." In: *Rev. Mat. Complut.* 37.3 (2024), pp. 655–694. DOI: 10.1007/s13163-024-00494-5. URL: https://doi.org/10.1007/s13163-024-00494-5 (cit. on p. 221).

[Alg+22] Hedayat Alghassi, Amol Deshmukh, Noelle Ibrahim, Nicolas Robles, Stefan Woerner, and Christa Zoufal. "A variational quantum algorithm for the Feynman-Kac formula." In: *Quantum* 6 (2022), p. 730 (cit. on p. 182).

[ACL26] Dong An, Andrew M. Childs, and Lin Lin. "Quantum Algorithm for Linear Non-unitary Dynamics with Near-Optimal Dependence on All Parameters." In: *Communications in Mathematical Physics* 407.1 (2026), p. 19 (cit. on pp. 13, 18, 105, 111).

[An+24] Dong An, Andrew M. Childs, Lin Lin, and Lexing Ying. *Laplace Transform Based Quantum Eigenvalue Transformation via Linear Combination of Hamiltonian Simulation*. 2024. arXiv: 2411.04010 [quant-ph] (cit. on p. 107).

[AFL21a] Dong An, Di Fang, and Lin Lin. "Time-dependent unbounded Hamiltonian simulation with vector norm scaling." In: *Quantum* 5 (2021), p. 459 (cit. on pp. 14, 88).

[AFL21b] Dong An, Di Fang, and Lin Lin. "Time-dependent unbounded Hamiltonian simulation with vector norm scaling." In: *Quantum* 5 (2021), p. 459. DOI: 10.22331/q-2021-05-26-459 (cit. on pp. 65, 90, 93, 103).

[ALL23] Dong An, Jin-Peng Liu, and Lin Lin. "Linear combination of Hamiltonian simulation for nonunitary dynamics with optimal state preparation cost." In: *Physical Review Letters* 131.15 (2023) (cit. on pp. 13, 15, 18, 105, 161).

[AOY26] Dong An, Akwum Onwunta, and Gengzhi Yang. "Fast-forwarding quantum algorithms for linear dissipative differential equations." In: *Quantum* 10 (2026), p. 1986 (cit. on p. 167).

[ACC20] Ana Arnal, Fernando Casas, and Cristina Chiralt. *A note on the Baker–Campbell–Hausdorff series in terms of right-nested commutators*. 2020. arXiv: 2006.15869 [math-ph] (cit. on p. 74).

[Bab+18] Ryan Babbush, Nathan Wiebe, Jarrod McClean, James McClain, Hartmut Neven, and Garnet Kin-Lic Chan. "Low-depth quantum simulation of materials." In: *Physical Review X* 8.1 (2018), p. 011044 (cit. on p. 8).

[Bac+17] Markus Bachmayr, Albert Cohen, Dinh Dũng, and Christoph Schwab. "Fully discrete approximation of parametric and stochastic elliptic PDEs." In: *SIAM J. Numer. Anal.* 55.5 (2017), pp. 2151–2186. URL: https://doi.org/10.1137/17M111626X (cit. on p. 210).

[BM12] Sonali Bagchi and Sanjit K. Mitra. *The non-uniform discrete Fourier transform and its applications in signal processing*. Vol. 463. Springer Science & Business Media, 2012 (cit. on p. 223).

[BA24] Mohsen Bagherimehrab and Alán Aspuru-Guzik. "Efficient quantum algorithm for all quantum wavelet transforms." In: *Quantum Science and Technology* 9.3 (2024), p. 035010 (cit. on p. 225).

[Bar93] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function." In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945 (cit. on p. 185).

[Bau+20] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. "Quantum algorithms for quantum chemistry and quantum materials science." In: *Chemical Reviews* 120.22 (2020), pp. 12685–12717 (cit. on p. 8).

[BSW25] Elisa Bäumer, David Sutter, and Stefan Woerner. *Approximate Quantum Fourier Transform in Logarithmic Depth on a Line*. 2025. arXiv: 2504.20832 [quant-ph]. URL: https://arxiv.org/abs/2504.20832 (cit. on p. 224).

[BK59] Richard Bellman and Robert Kalaba. "A mathematical theory of adaptive control processes." In: *Proceedings of the National Academy of Sciences* 45.8 (1959), pp. 1288–1290 (cit. on p. 183).

[Ben+19] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. "Parameterized quantum circuits as machine learning models." In: *Quantum Science and Technology* 4.4 (Nov. 2019). DOI: 10.1088/2058-9565/ab4eb5 (cit. on p. 181).

[BDG23] Fred Espen Benth, Nils Detering, and Luca Galimberti. "Neural networks in Fréchet spaces." In: *Annals of Mathematics and Artificial Intelligence* 91.1 (2023), pp. 75–103 (cit. on p. 220).

[Ber14] Dominic W. Berry. "High-order quantum algorithm for solving linear differential equations." In: *Journal of Physics A: Mathematical and Theoretical* 47.10 (2014), p. 105301 (cit. on p. 104).

[Ber+07] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. "Efficient quantum algorithms for simulating sparse Hamiltonians." In: *Commun. Math. Phys.* 270.2 (2007), pp. 359–371. DOI: 10.1007/s00220-006-0150-x (cit. on p. 65).

[BC12] Dominic W. Berry and Andrew M. Childs. "Black-box Hamiltonian simulation and unitary implementation." In: *Quantum Information & Computation* 12.1-2 (2012), pp. 29–62. DOI: 10.26421/QIC12.1-2 (cit. on p. 65).

[Ber+14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. "Exponential improvement in precision for simulating sparse Hamiltonians." In: *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. 2014, pp. 283–292. DOI: 10.1145/2591796.2591854 (cit. on pp. 12, 65).

[Ber+15a] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. "Simulating Hamiltonian Dynamics with a Truncated Taylor Series." In: *Physical Review Letters* 114 (9 Mar. 2015), p. 090502. DOI: 10.1103/PhysRevLett.114.090502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.114.090502 (cit. on p. 59).

[Ber+15b] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. "Simulating Hamiltonian dynamics with a truncated Taylor series." In: *Phys. Rev. Lett.* 114 (2015), p. 090502. DOI: 10.1103/PhysRevLett.114.090502 (cit. on pp. 65, 73, 85).

[BCK15] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. "Hamiltonian Simulation with Nearly Optimal Dependence on all Parameters." In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, Oct. 2015. DOI: 10.1109/focs.2015.54 (cit. on p. 65).

[Ber+17] Dominic W. Berry, Andrew M. Childs, Aaron Ostrander, and Guoming Wang. "Quantum algorithm for linear differential equations with exponentially improved dependence on precision." In: *Communications in Mathematical Physics* 356.3 (2017), pp. 1057–1081 (cit. on p. 104).

[Ber+20] Dominic W. Berry, Andrew M. Childs, Yuan Su, Xin Wang, and Nathan Wiebe. "Time-dependent Hamiltonian simulation with L^1-norm scaling." In: *Quantum* 4 (2020), p. 254 (cit. on p. 14).

[BCG14] Dominic W. Berry, Richard Cleve, and Sevag Gharibian. "Gate-efficient discrete simulations of continuous-time quantum query algorithms." In: *Quantum Information and Computation* 14.1-2 (2014), pp. 1–30. DOI: 10.26421/QIC14.1-2-1 (cit. on p. 65).

[BC24] Dominic W. Berry and Pedro C. S. Costa. "Quantum algorithm for time-dependent differential equations using Dyson series." In: *Quantum* 8 (2024), p. 1369 (cit. on p. 104).

[BCR00] Sergio Blanes, Fernando Casas, and Javier Ros. "Improved high order integrators based on the Magnus expansion." In: *BIT Numerical Mathematics* 40.3 (2000), pp. 434–450. DOI: https://doi.org/10.1023/A:1022311628317 (cit. on p. 70).

[BFZ25] Yonah Borns-Weil, Di Fang, and Jiaqi Zhang. *Discrete Superconvergence Analysis for Quantum Magnus Algorithms of Unbounded Hamiltonian Simulation*. 2025. arXiv: 2502.20255 [math.NA]. URL: https://arxiv.org/abs/2502.20255 (cit. on p. 88).

[Boy01] John P. Boyd. *Chebyshev and Fourier spectral methods*. Dover Publications, 2001 (cit. on p. 20).

[BH95] William L. Briggs and Van E. Henson. *The DFT: an owner's manual for the discrete Fourier transform*. SIAM, 1995 (cit. on p. 223).

[BT13] Theodor Bröcker and Tammo Tom Dieck. *Representations of compact Lie groups*. Vol. 98. Springer Science & Business Media, 2013 (cit. on pp. 31, 33).

[BB17] Alfonso Bueno-Orovio and Kevin Burrage. "Exact solutions to the fractional time-space Bloch–Torrey equation for magnetic resonance imaging." In: *Communications in Nonlinear Science and Numerical Simulation* 52 (2017), pp. 91–109. DOI: 10.1016/j.cnsns.2017.04.013 (cit. on p. 161).

[BG04] Hans-Joachim Bungartz and Michael Griebel. "Sparse grids." In: *Acta numerica* 13 (2004). DOI: 10.1017/S0962492904000182 (cit. on pp. 22, 183, 196–198, 220).

[Bur+24] Daniel Burgarth, Paolo Facchi, Alexander Hahn, Mattias Johnsson, and Kazuya Yuasa. *Strong Error Bounds for Trotter & Strang-Splittings and Their Implications for Quantum Chemistry*. 2024 (cit. on p. 103).

[Bur+23] Daniel Burgarth, Niklas Galke, Alexander Hahn, and Lauritz van Luijk. "State-dependent Trotter limits and their approximations." In: *Physical Review A* 107.4 (Apr. 2023). DOI: 10.1103/physreva.107.l040201 (cit. on p. 103).

[BBG24] Iain Burge, Michel Barbeau, and Joaquin Garcia-Alfaro. *Quantum CORDIC – Arcsin on a Budget*. 2024. arXiv: 2411.14434 [quant-ph]. URL: https://arxiv.org/abs/2411.14434 (cit. on pp. 55, 235, 266).

[Cam19] Earl Campbell. "Random compiler for fast Hamiltonian simulation." In: *Phys. Rev. Lett.* 123.7 (2019), p. 070503. DOI: 10.1103/PhysRevLett.123.070503 (cit. on p. 65).

[Cao+19] Yudong Cao, Jonathan Romero, Jonathan P. Olson, et al. "Quantum chemistry in the age of quantum computing." In: *Chemical Reviews* 119.19 (2019), pp. 10856–10915 (cit. on p. 8).

[Cer+21] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, et al. "Variational quantum algorithms." In: *Nature Reviews Physics* 3.9 (2021), pp. 625–644 (cit. on p. 182).

[Cha24] Shantanav Chakraborty. "Implementing any linear combination of unitaries on intermediate-term quantum computers." In: *Quantum* 8 (2024), p. 1496 (cit. on p. 221).

[CB24] Chi-Fang Chen and Fernando GSL Brandão. "Average-case speedup for product formulas." In: *Communications in Mathematical Physics* 405.2 (2024), p. 32 (cit. on p. 103).

[Che+20] Chi-Fang Chen, Hsin-Yuan Huang, Richard Kueng, and Joel A. Tropp. "Quantum simulation via randomized product formulas: Low gate complexity with accuracy guarantees." In: (2020). arXiv: 2008.11751 [quant-ph] (cit. on p. 65).

[CR19] Alexey Chernov and Anne Reinarz. "Sparse grid approximation spaces for space–time boundary integral formulations of the heat equation." In: *Computers & Mathematics with Applications* 78.11 (2019), pp. 3605–3619 (cit. on p. 183).

[CKS17] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. "Quantum algorithm for systems of linear equations with exponentially improved dependence on precision." In: *SIAM Journal on Computing* 46.6 (2017), pp. 1920–1950 (cit. on p. 104).

[CL20] Andrew M. Childs and Jin-Peng Liu. "Quantum spectral methods for differential equations." In: *Communications in Mathematical Physics* 375.2 (2020), pp. 1427–1457 (cit. on p. 104).

[CLO21] Andrew M. Childs, Jin-Peng Liu, and Aaron Ostrander. "High-precision quantum algorithms for partial differential equations." In: *Quantum* 5 (2021), p. 574 (cit. on p. 181).

[Chi+18] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. "Toward the first quantum simulation with quantum speedup." In: *Proc. Nat. Acad. Sci.* 115 (2018), pp. 9456–9461. DOI: 10.1073/pnas.1801723115 (cit. on p. 65).

[COS19] Andrew M. Childs, Aaron Ostrander, and Yuan Su. "Faster quantum simulation by randomization." In: *Quantum* 3 (2019), p. 182. DOI: 10.22331/q-2019-09-02-182 (cit. on p. 65).

[CS19a] Andrew M. Childs and Yuan Su. "Nearly Optimal Lattice Simulation by Product Formulas." In: *Phys. Rev. Lett.* 123 (5 Aug. 2019), p. 050503. DOI: 10.1103/PhysRevLett.123.050503. URL: https://link.aps.org/doi/10.1103/PhysRevLett.123.050503 (cit. on p. 14).

[CS19b] Andrew M. Childs and Yuan Su. "Nearly optimal lattice simulation by product formulas." In: *Phys. Rev. Lett.* 123.5 (2019), p. 050503. DOI: 10.1103/PhysRevLett.123.050503 (cit. on p. 65).

[Chi+21] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. "Theory of Trotter Error with Commutator Scaling." In: *Phys. Rev. X* 11 (1 2021), p. 011020. DOI: 10.1103/PhysRevX.11.011020 (cit. on pp. 12, 65, 87, 105).

[CW12a] Andrew M. Childs and Nathan Wiebe. "Hamiltonian Simulation Using Linear Combinations of Unitary Operations." In: *Quantum Information and Computation* 12 (2012), pp. 901–924. DOI: 10.26421/qic12.11-12 (cit. on pp. 58, 105).

[CW12b] Andrew M. Childs and Nathan Wiebe. "Hamiltonian Simulation Using Linear Combinations of Unitary Operations." In: *Quantum Information and Computation* 12 (2012), pp. 901–924. DOI: 10.26421/qic12.11-12 (cit. on pp. 65, 66, 224).

[Chk+13] Abdellah Chkifa, Albert Cohen, Ronald DeVore, and Christoph Schwab. "Adaptive algorithms for sparse polynomial approximation of parametric and stochastic elliptic PDEs." In: *M2AN Math. Mod. and Num. Anal.* 47.1 (2013), pp. 253–280. DOI: http://dx.doi.org/10.1051/m2an/2012027 (cit. on p. 210).

[CCS13] Abdellah Chkifa, Albert Cohen, and Christoph Schwab. "High-Dimensional Adaptive Sparse Polynomial Interpolation and Applications to Parametric PDEs." In: *Journ. Found. Comp. Math.* 14.4 (2013), pp. 601–633. DOI: http://dx.doi.org/10.1007/s10208-013-9154-z (cit. on pp. 22, 211, 213).

[Cle+98] Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. "Quantum algorithms revisited." In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (1998), pp. 339–354. DOI: 10.1098/rspa.1998.0164. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1998.0164 (cit. on p. 59).

[CCS15] Albert Cohen, Abdellah Chkifa, and Christoph Schwab. "Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs." In: *Journ. Math. Pures et Appliquees* 103.2 (2015), pp. 400–428. DOI: http://dx.doi.org/10.1016/j.matpur.2014.04.009 (cit. on pp. 22, 23, 210, 213).

[CDS10] Albert Cohen, Ronald DeVore, and Christoph Schwab. "Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs." In: *Found. Comput. Math.* 10.6 (2010), pp. 615–646. DOI: 10.1007/s10208-010-9072-2. URL: http://dx.doi.org/10.1007/s10208-010-9072-2 (cit. on p. 22).

[CDS11] Albert Cohen, Ronald Devore, and Christoph Schwab. "Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE's." In: *Anal. Appl. (Singap.)* 9.1 (2011), pp. 11–47. DOI: 10.1142/S0219530511001728. URL: http://dx.doi.org/10.1142/S0219530511001728 (cit. on pp. 22, 208, 210, 211, 213).

[CT65] James W. Cooley and John W. Tukey. "An algorithm for the machine calculation of complex Fourier series." In: *Mathematics of Computation* 19.90 (1965), pp. 297–301 (cit. on p. 223).

[Cop02] Dan Coppersmith. *An approximate Fourier transform useful in quantum factoring.* 2002. arXiv: quant-ph/0201067 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0201067 (cit. on p. 224).

[Cos+22] Pedro C. S. Costa, Dong An, Yuval R. Sanders, Yuan Su, Ryan Babbush, and Dominic W. Berry. "Optimal Scaling Quantum Linear-Systems Solver via Discrete Adiabatic Theorem." In: *PRX Quantum* 3 (4 Oct. 2022). DOI: 10.1103/PRXQuantum.3.040303. URL: https://link.aps.org/doi/10.1103/PRXQuantum.3.040303 (cit. on p. 104).

[Cuc+04] Steven A. Cuccaro, Thomas G. Draper, Samuel A. Kutin, and David P. Moulton. *A new quantum ripple-carry addition circuit*. 2004. arXiv: quant-ph/0410184 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0410184 (cit. on p. 54).

[DCM92] Jean Dalibard, Yvan Castin, and Klaus Mølmer. "Wave-function approach to dissipative processes in quantum optics." In: *Physical Review Letters* 68.5 (1992), pp. 580–583. DOI: 10.1103/PhysRevLett.68.580 (cit. on p. 172).

[Dal26] Alexander M. Dalzell. *A shortcut to an optimal quantum linear system solver*. 2026. arXiv: 2406.12086 [quant-ph]. URL: https://arxiv.org/abs/2406.12086 (cit. on p. 104).

[DHP21] Ronald DeVore, Boris Hanin, and Guergana Petrova. "Neural network approximation." In: *Acta Numerica* 30 (2021), pp. 327–444. DOI: 10.1017/S0962492921000052 (cit. on p. 182).

[DF84] John D. Dollard and Charles N. Friedman. *Product integration with applications to differential equations*. Cambridge University Press, 1984 (cit. on pp. 6, 35).

[Don+24] Yulong Dong, Lin Lin, Hongkang Ni, and Jiasu Wang. "Infinite quantum signal processing." In: *Quantum* 8 (Dec. 2024), p. 1558. DOI: 10.22331/q-2024-12-10-1558. URL: https://doi.org/10.22331/q-2024-12-10-1558 (cit. on p. 221).

[Dra00] Thomas G. Draper. *Addition on a Quantum Computer*. 2000. arXiv: quant-ph/0008033 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0008033 (cit. on pp. 54, 224).

[Dra+06] Thomas G. Draper, Samuel A. Kutin, Eric M. Rains, and Krysta M. Svore. "A logarithmic-depth quantum carry-lookahead adder." In: *Quantum Information and Computation* 6.4 (July 2006), pp. 351–369 (cit. on p. 54).

[DR93] Alok Dutt and Vladimir Rokhlin. "Fast Fourier transforms for non-equispaced data." In: *SIAM Journal on Scientific computing* 14.6 (1993), pp. 1368–1393 (cit. on p. 223).

[EMW19] Weinan E, Chao Ma, and Lei Wu. "A Priori Estimates of the Population Risk for Two-layer Neural Networks." In: *Communications in Mathematical Sciences* 17.5 (2019), pp. 1407–1425. DOI: 10.4310/CMS.2019.v17.n5.a11. URL: https://arxiv.org/abs/1810.06397 (cit. on p. 185).

[EW22] Weinan E and Stephan Wojtowytsch. "Some Observations on High-Dimensional Partial Differential Equations with Barron Data." In: *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*. Ed. by Joan Bruna, Jan

Hesthaven, and Lenka Zdeborová. Vol. 145. Proceedings of Machine Learning Research. PMLR, 2022, pp. 253–269. URL: https://proceedings.mlr.press/v145/e22a.html (cit. on p. 185).

[Edw73] Charles H. Edwards. *Advanced Calculus of Several Variables*. Dover Publications, 1973 (cit. on p. 33).

[End+19] Suguru Endo, Qi Zhao, Ying Li, Simon Benjamin, and Xiao Yuan. "Mitigating algorithmic errors in a Hamiltonian simulation." In: *Phys. Rev. A* 99 (1 Jan. 2019), p. 012334. DOI: 10.1103/PhysRevA.99.012334 (cit. on p. 69).

[Fae+22] Paul K. Faehrmann, Mark Steudtne, Richard Kueng, Richard Kieferova, and Jens Eisert. "Randomizing multi-product formulas for Hamiltonian simulation." In: *Quantum* 6 (2022), p. 806. DOI: 10.22331/q-2022-09-19-806. URL: https://doi.org/10.22331/q-2022-09-19-806 (cit. on p. 68).

[FLT23] Di Fang, Lin Lin, and Yu Tong. "Time-marching based quantum solvers for time-dependent linear differential equations." In: *Quantum* 7 (2023), p. 955 (cit. on p. 104).

[Fan+25] Qin Fang, Lei Shi, Min Xu, and Ding-Xuan Zhou. *Two-Dimensional Deep ReLU CNN Approximation for Korobov Functions: A Constructive Approach*. 2025. arXiv: 2503.07976 [stat.ML]. URL: https://arxiv.org/abs/2503.07976 (cit. on pp. 185, 205).

[FS03] Jeffrey A. Fessler and Bradley P. Sutton. "Non-uniform fast Fourier transforms using min-max interpolation." In: *IEEE transactions on signal processing* 51.2 (2003), pp. 560–574 (cit. on p. 223).

[Fey82] Richard P. Feynman. "Simulating physics with computers." In: *International Journal of Theoretical Physics* 21 (1982), pp. 467–488 (cit. on p. 8).

[Fey86] Richard P. Feynman. "Quantum mechanical computers." In: *Foundations of Physics* 16.6 (1986), pp. 507–531 (cit. on p. 8).

[FW98] Amir Fijany and Colin P. Williams. "Quantum wavelet transforms: Fast algorithms and complete circuits." In: *NASA international conference on quantum computing and quantum communications*. Springer. 1998, pp. 10–33 (cit. on p. 225).

[Gid19] Craig Gidney. *Asymptotically Efficient Quantum Karatsuba Multiplication*. 2019. arXiv: 1904.07356 [quant-ph]. URL: https://arxiv.org/abs/1904.07356 (cit. on p. 54).

[Gil+19a] András Gilyén, Yuan Su, Guang H. Low, and Nathan Wiebe. "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics." In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. STOC '19. ACM, June 2019. DOI: 10.1145/3313276.3316366 (cit. on pp. 20, 57, 62, 187, 224, 250).

[Gil+19b] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics." In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. STOC '19. ACM, June 2019. DOI: 10.1145/3313276.3316366 (cit. on pp. 12, 65).

[GJ23] Lukas Gonon and Antoine Jacquier. "Universal Approximation Theorem and error bounds for quantum neural networks and quantum reservoirs." In: (2023). arXiv: 2307.12904 [quant-ph] (cit. on p. 182).

[GS23] Lukas Gonon and Christoph Schwab. "Deep ReLU Neural Networks Overcome the Curse of Dimensionality for Partial Integro-Differential Equations." In: *Analysis and Applications* 21.1 (2023), pp. 1–47. DOI: 10.1142/S0219530522500129. URL: https://www.worldscientific.com/doi/10.1142/S0219530522500129 (cit. on p. 185).

[Got14] Hayato Goto. "Resource requirements for a fault-tolerant quantum Fourier transform." In: *Physical Review A* 90.5 (2014), p. 052318 (cit. on p. 224).

[GTN21] Takahiro Goto, Quoc Hoan Tran, and Kohei Nakajima. "Universal Approximation Property of Quantum Machine Learning Models in Quantum-Enhanced Feature Spaces." In: *Phys. Rev. Lett.* 127 (9 Aug. 2021), p. 090506. DOI: 10.1103/PhysRevLett.127.090506 (cit. on p. 182).

[GL04] Leslie Greengard and June-Yub Lee. "Accelerating the non-uniform fast Fourier transform." In: *SIAM review* 46.3 (2004), pp. 443–454 (cit. on p. 223).

[GH07] Michael Griebel and Jan Hamaekers. "Sparse grids for the Schrödinger equation." In: *ESAIM: Mathematical Modelling and Numerical Analysis* 41.2 (2007), pp. 215–247 (cit. on p. 183).

[GGA25] Artyom M. Grigoryan, Alexis A. Gomez, and Sos S. Agaian. "New method of computing the quantum cosine transforms." In: *Multimodal Image Exploitation and Learning 2025*. Vol. 13457. SPIE. 2025, pp. 93–102 (cit. on p. 225).

[Gro+23] Philipp Grohs, Fabian Hornung, Arnulf Jentzen, and Philippe von Wurstemberger. "A Proof that Artificial Neural Networks Overcome the Curse of Dimensionality in the Numerical Approximation of Black–Scholes Partial Differential Equations." In: *Memoirs of the American Mathematical Society* 284.1410 (2023), pp. 1–106. DOI: 10.1090/memo/1410. URL: https://arxiv.org/abs/1809.02362 (cit. on p. 185).

[GI12] Laszlo Gyongyosi and Sandor Imre. *An Improvement in Quantum Fourier Transform*. 2012. arXiv: 1207.4464 [quant-ph]. URL: https://arxiv.org/abs/1207.4464 (cit. on p. 225).

[Hal15] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Vol. 222. Graduate Texts in Mathematics. Springer, 2015 (cit. on pp. 11, 31, 32, 35–38, 40–42, 45).

[HSS24] Helmut Harbrecht, Marc Schmidlin, and Christoph Schwab. "The Gevrey class implicit mapping theorem with application to UQ of semilinear elliptic PDEs." In: *Math. Models Methods Appl. Sci.* 34.5 (2024), pp. 881–917. DOI: 10.1142/S0218202524500179. URL: https://doi.org/10.1142/S0218202524500179 (cit. on p. 220).

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for linear systems of equations." In: *Physical review letters* 103.15 (2009), p. 150502 (cit. on pp. 104, 224).

[Hut+20] Martin Hutzenthaler, Arnulf Jentzen, Thomas Kruse, and Tuan Anh Nguyen. "A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations." In: *SN Partial Differential Equations and Applications* 1.2 (2020), pp. 1–34. DOI: 10.1007/s42985-019-0006-9. URL: https://link.springer.com/article/10.1007/s42985-019-0006-9 (cit. on p. 185).

[Ike+23] Tatsuhiko N. Ikeda, Asir Abrar, Isaac L. Chuang, and Sho Sugiura. "Minimum Trotterization formulas for a time-dependent Hamiltonian." In: *Quantum* 7 (2023), p. 1168 (cit. on p. 14).

[JL00] Tobias Jahnke and Christian Lubich. "Error bounds for exponential operator splittings." In: *BIT* 40.4 (2000), pp. 735–744. DOI: 10.1023/A:1022396519656 (cit. on p. 102).

[Jai+25] Siddhartha Jain, Vishnu Iyer, Rolando D. Somma, Ning Bao, and Stephen P Jordan. *Efficient Quantum Hermite Transform*. 2025. arXiv: 2510.04929 [quant-ph]. URL: https://arxiv.org/abs/2510.04929 (cit. on p. 225).

[JP25] Nitin Jha and Abhishek Parakh. *Quantum Hilbert Transform*. 2025. arXiv: 2505.23581 [quant-ph]. URL: https://arxiv.org/abs/2505.23581 (cit. on p. 225).

[JA26] Shan Jiang and Dong An. *Contour-integral Based Quantum Eigenvalue Transformation: Analysis and Applications*. 2026. arXiv: 2601.11959 [quant-ph] (cit. on p. 107).

[Jin+25] Shi Jin, Nana Liu, Chuwen Ma, and Yue Yu. *On the Schrödingerization Method for Linear Non-unitary Dynamics with Optimal Dependence on Matrix Queries*. 2025. arXiv: 2505.00370 [quant-ph] (cit. on p. 108).

[JLW25] Shi Jin, Nana Liu, and Wei Wei. *Quantum Algorithms for Stochastic Differential Equations: A Schrödingerisation Approach*. 2025. arXiv: 2412.14868 [quant-ph]. URL: https://arxiv.org/abs/2412.14868 (cit. on p. 108).

[JLY23] Shi Jin, Nana Liu, and Yue Yu. "Quantum simulation of partial differential equations: applications and detailed analysis." In: *Physical Review A* 108.3 (2023), p. 032603 (cit. on pp. 108, 181).

[JLY24] Shi Jin, Nana Liu, and Yue Yu. "Quantum simulation of partial differential equations via Schrödingerization." In: *Physical Review Letters* 133.23 (2024), p. 230602 (cit. on pp. 104, 108).

[JMZ26] Shi Jin, Chuwen Ma, and Enrique Zuazua. *Transmutation Based Quantum Simulation for Non-unitary Dynamics*. 2026. arXiv: 2601.03616 [quant-ph] (cit. on p. 108).

[Jin+24] Yuwei Jin, Xiangyu Gao, Minghao Guo, et al. "Optimizing quantum fourier transformation kernels for modern NISQ and FT architectures." In: *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2024, pp. 1–15 (cit. on p. 224).

[JN26] Naihuan Jing and Molena Nguyen. *Complexity Bounds for Hamiltonian Simulation in Unitary Representations*. 2026. arXiv: 2603.07231 [quant-ph]. URL: https://arxiv.org/abs/2603.07231 (cit. on p. 31).

[JLP12] Stephen P. Jordan, Keith S.M. Lee, and John Preskill. "Quantum algorithms for quantum field theories." In: *Science* 336.6085 (2012), pp. 1130–1133 (cit. on p. 8).

[Kit95] Alexei Y. Kitaev. *Quantum measurements and the Abelian Stabilizer Problem*. 1995. arXiv: quant-ph/9511026 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/9511026 (cit. on pp. 26, 224).

[KR01] Andreas Klappenecker and Martin Rotteler. "Discrete cosine transforms on quantum computers." In: *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis*. IEEE. 2001, pp. 464–468 (cit. on p. 225).

[Kla+24] Berend Klaver, Stefan Rombouts, Michael Fellner, et al. *SWAP-less Implementation of Quantum Algorithms*. 2024. arXiv: 2408.10907 [quant-ph]. URL: https://arxiv.org/abs/2408.10907 (cit. on p. 224).

[Kök+22] Efekan Kökcü, Thomas Steckmann, Yan Wang, J. K. Freericks, Eugene F. Dumitrescu, and Alexander F. Kemper. "Fixed depth Hamiltonian simulation via Cartan decomposition." In: *Physical Review Letters* 129.7 (2022), p. 070501 (cit. on p. 31).

[Kor59] Nikolay Korobov. "On the approximate solution of integral equations." In: *Dokl. Akad. Nauk SSSR* 128 (1959) (cit. on pp. 21, 183).

[Kro23] Hari Krovi. "Improved quantum algorithms for linear and nonlinear differential equations." In: *Quantum* 7 (2023), p. 913 (cit. on p. 104).

[LG05] June-Yub Lee and Leslie Greengard. "The type III non-uniform FFT and its applications." In: *Journal of Computational Physics* 206.1 (2005), pp. 1–5 (cit. on p. 224).

[LC13] Tony E. Lee and Ching-Kit Chan. "Dissipative transverse-field Ising model: Steady-state correlations and spin squeezing." In: *Physical Review A* 88.6 (2013), p. 063811. DOI: 10.1103/PhysRevA.88.063811 (cit. on p. 172).

[LL25] Yingzhou Li and Jingyu Liu. *A Superfast Direct Solver for Type-III Inverse Nonuniform Discrete Fourier Transform*. 2025. arXiv: 2512.03733 [math.NA]. URL: https://arxiv.org/abs/2512.03733 (cit. on p. 224).

[LZ25] Yuwen Li and Guozhi Zhang. *Higher Order Approximation Rates for ReLU CNNs in Korobov Spaces*. 2025. arXiv: 2501.11275 [cs.LG]. URL: https://arxiv.org/abs/2501.11275 (cit. on pp. 185, 220).

[Lin22] Lin Lin. *Lecture Notes on Quantum Algorithms for Scientific Computation*. 2022. arXiv: 2201.08309 [quant-ph]. URL: https://arxiv.org/abs/2201.08309 (cit. on p. 224).

[Liu+21a] Hai-Ling Liu, Yu-Sen Wu, Lin-Chun Wan, et al. "Variational quantum algorithm for the Poisson equation." In: *Physical Review A* 104.2 (2021), p. 022418 (cit. on p. 182).

[Liu+21b] Hao Liu, Minshuo Chen, Tuo Zhao, and Wenjing Liao. "Besov Function Approximation and Binary Classification on Low-Dimensional Manifolds Using Convolutional Residual Networks." In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 6770–6780. URL: https://proceedings.mlr.press/v139/liu21e.html (cit. on p. 220).

[Liu+21c] Jin-Peng Liu, Herman Ø. Kolden, Hari K. Krovi, Nuno F. Loureiro, Konstantina Trivisa, and Andrew M. Childs. "Efficient quantum algorithm for dissipative nonlinear differential equations." In: *Proceedings of the National Academy of Sciences* 118.35 (2021), e2026805118 (cit. on p. 181).

[Liu+23] Junyu Liu, Khadijeh Najafi, Kunal Sharma, Francesco Tacchino, Liang Jiang, and Antonio Mezzacapo. "Analytic theory for the dynamics of wide quantum neural networks." In: *Physical Review Letters* 130.15 (2023), p. 150601 (cit. on p. 186).

[Liu+00] Qing Huo Liu, Xue Min Xu, Bo Tian, and Zhong Qing Zhang. "Applications of non-uniform fast transform algorithms in numerical solutions of differential and integral equations." In: *IEEE Transactions on geoscience and remote sensing* 38.4 (2000), pp. 1551–1560 (cit. on p. 223).

[Llo96] Seth Lloyd. "Universal Quantum Simulators." In: *Science* 273.5278 (1996), pp. 1073–1078. DOI: 10.1126/science.273.5278.1073. URL: https://www.science.org/doi/abs/10.1126/science.273.5278.1073 (cit. on pp. 10, 105).

[LC17a] Guang H. Low and Isaac L. Chuang. "Optimal Hamiltonian Simulation by Quantum Signal Processing." In: *Physical Review Letters* 118.1 (Jan. 2017). DOI: 10.1103/PhysRevLett.118.010501 (cit. on pp. 12, 224).

[Low19] Guang Hao Low. "Hamiltonian simulation with nearly optimal dependence on spectral norm." In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. 2019, pp. 491–502. DOI: 10.1145/3313276.3316386 (cit. on p. 65).

[LC17b] Guang Hao Low and Isaac L. Chuang. "Optimal Hamiltonian Simulation by Quantum Signal Processing." In: *Physical Review Letters* 118.1 (Jan. 2017). DOI: 10.1103/physrevlett.118.010501 (cit. on pp. 20, 65).

[LC19] Guang Hao Low and Isaac L. Chuang. "Hamiltonian Simulation by Qubitization." In: *Quantum* 3 (July 2019), p. 163. DOI: 10.22331/q-2019-07-12-163 (cit. on p. 65).

[LKW19] Guang Hao Low, Vadym Kliuchnikov, and Nathan Wiebe. *Well-conditioned multiproduct Hamiltonian simulation*. 2019. arXiv: 1907.11679 [quant-ph] (cit. on pp. 11, 12, 66, 69–72, 84, 100, 105).

[LS25] Guang Hao Low and Rolando D. Somma. *Optimal quantum simulation of linear non-unitary dynamics*. 2025. arXiv: 2508.19238 [quant-ph]. URL: https://arxiv.org/abs/2508.19238 (cit. on pp. 13, 18, 105, 112, 113, 138–140, 284).

[LS24] Guang Hao Low and Yuan Su. *Quantum Eigenvalue Processing*. 2024. arXiv: 2401.06240 [quant-ph] (cit. on p. 107).

[LS26] Guang Hao Low and Yuan Su. "Quantum linear system algorithm with optimal queries to initial state preparation." In: *Quantum* 10 (2026), p. 2041 (cit. on p. 104).

[Low+23] Guang Hao Low, Yuan Su, Yu Tong, and Minh C. Tran. "Complexity of Implementing Trotter Steps." In: *PRX Quantum* 4.2 (May 2023). DOI: 10.1103/prxquantum.4.020323 (cit. on p. 65).

[LY24] Tao Luo and Haizhao Yang. "Two-Layer Neural Networks for Partial Differential Equations: Optimization and Generalization Theory." In: *Numerical Analysis Meets Machine Learning*. Ed. by Siddhartha Mishra and Alex Townsend. Vol. 25. Handbook of Numerical Analysis. Elsevier, 2024, pp. 515–554. DOI: 10.1016/bs.hna.2023.10.011. URL: https://www.sciencedirect.com/science/article/abs/pii/S1570865924000073 (cit. on p. 185).

[Mag+08] Richard L. Magin, Omar Abdullah, Dumitru Baleanu, and Xiaohong Joe Zhou. "Anomalous diffusion expressed through fractional order differential operators in the Bloch–Torrey equation." In: *Journal of Magnetic Resonance* 190.2 (2008), pp. 255–270. DOI: 10.1016/j.jmr.2007.11.005 (cit. on p. 161).

[Man+25] Alberto Manzano, David Dechant, Jordi Tura, and Vedran Dunjko. "Approximation and generalization capacities of parametrized quantum circuits for functions in Sobolev spaces." In: *Quantum* 9 (2025), p. 1658 (cit. on p. 185).

[Miz26] Kaoru Mizuta. "On the commutator scaling in Hamiltonian simulation with multi-product formulas." In: *Quantum* 10 (2026), p. 1974 (cit. on pp. 17, 102, 105, 109, 283).

[MIF24] Kaoru Mizuta, Tatsuhiko N. Ikeda, and Keisuke Fujii. *Explicit error bounds with commutator scaling for time-dependent product and multi-product formulas*. 2024. arXiv: 2410.14243 [quant-ph]. URL: https://arxiv.org/abs/2410.14243 (cit. on pp. 14, 105, 110).

[MD19] Hadrien Montanelli and Qiang Du. "New error bounds for deep ReLU networks using sparse grids." In: *SIAM Journal on Mathematics of Data Science* 1.1 (2019). DOI: 10.1137/18M1189336 (cit. on p. 185).

[Mor+22] Mauro E. S. Morales, Pedro C. S. Costa, Daniel K. Burgarth, Yuval R. Sanders, and Dominic W. Berry. *Greatly improved higher-order product formulae for quantum simulation*. 2022. arXiv: 2210.15817 [quant-ph] (cit. on p. 65).

[MMG20] Nicolas Moutal, Antoine Moutal, and Denis S. Grebenkov. "Diffusion NMR in periodic media: efficient computation and spectral properties." In: *Journal of Physics A: Mathematical and Theoretical* 53.32 (2020), p. 325201. DOI: 10.1088/1751-8121/ab977e. arXiv: 2005.06975 (cit. on p. 161).

[NSM20] Yunseong Nam, Yuan Su, and Dmitri Maslov. "Approximate quantum Fourier transform with $O(n \log n)$ T gates." In: *NPJ Quantum Information* 6.1 (2020), p. 26 (cit. on p. 224).

[NC02] Michael A Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. American Association of Physics Teachers, 2002 (cit. on pp. 30, 224).

[OSZ22] J. A. A. Opschoor, Ch. Schwab, and J. Zech. "Exponential ReLU DNN expression of holomorphic maps in high dimension." In: *Constr. Approx.* 55.1 (2022), pp. 537–582. DOI: 10.1007/s00365-021-09542-5. URL: https://doi.org/10.1007/s00365-021-09542-5 (cit. on p. 216).

[PRM17] Alex Parent, Martin Roetteler, and Michele Mosca. *Improved reversible and quantum circuits for Karatsuba-based integer multiplication*. 2017. arXiv: 1706.03419 [quant-ph]. URL: https://arxiv.org/abs/1706.03419 (cit. on p. 54).

[Par+22] Mohammad Parhizi, Gozdem Kilaz, Jason K. Ostanek, and Ankur Jain. "Analytical solution of the convection-diffusion-reaction-source (CDRS) equation using Green's function technique." In: *International Communications in Heat and Mass Transfer* 131 (2022), p. 105869. DOI: 10.1016/j.icheatmasstransfer.2021.105869 (cit. on p. 167).

[Pér+21] Adrián Pérez-Salinas, David López-Núñez, Artur García-Sáez, P. Forn-Díaz, and José I. Latorre. "One qubit as a universal approximant." In: *Phys. Rev. A* 104 (1 July 2021), p. 012405. DOI: 10.1103/PhysRevA.104.012405. URL: https://link.aps.org/doi/10.1103/PhysRevA.104.012405 (cit. on p. 182).

[Pfe23] Philipp Pfeffer. *Multidimensional Quantum Fourier Transformation*. 2023. arXiv: 2301.13835 [quant-ph]. URL: https://arxiv.org/abs/2301.13835 (cit. on p. 224).

[PK98] M. B. Plenio and P. L. Knight. "The quantum-jump approach to dissipative dynamics in quantum optics." In: *Reviews of Modern Physics* 70.1 (1998), pp. 101–144. DOI: 10.1103/RevModPhys.70.101 (cit. on p. 172).

[PST01] Daniel Potts, Gabriele Steidl, and Manfred Tasche. "Fast Fourier transforms for nonequispaced data: A tutorial." In: *Modern Sampling Theory: Mathematics and Applications* (2001), pp. 247–270 (cit. on p. 223).

[QCT13] Chuan Qin, Chin-Chen Chang, and Pei-Ling Tsou. "Robust image hashing using non-uniform sampling in discrete Fourier domain." In: *Digital Signal Processing* 23.2 (2013), pp. 578–585 (cit. on p. 223).

[Rag+24] Michael Ragone, Bojko N. Bakalov, Frédéric Sauvage, et al. "A Lie algebraic theory of barren plateaus for deep parameterized quantum circuits." In: *Nature Communications* 15.1 (2024), p. 7172 (cit. on p. 31).

[Rag+22] Michael Ragone, Paolo Braccia, Quynh T. Nguyen, et al. "Representation theory for geometric quantum machine learning." In: *arXiv preprint arXiv:2210.07980* (2022) (cit. on p. 31).

[Ram+23] Mehdi Ramezani, Morteza Nikaeen, Farnaz Farman, Seyed Mahmoud Ashrafi, and Alireza Bahrampour. "Quantum multiplication algorithm based on the convolution theorem." In: *Physical Review A* 108.5 (2023), p. 052405 (cit. on p. 54).

[RWW24] Gumaro Rendon, Jacob Watkins, and Nathan Wiebe. "Improved Accuracy for Trotter Simulations Using Chebyshev Interpolation." In: *Quantum* 8 (Feb. 2024), p. 1266. DOI: 10.22331/q-2024-02-26-1266 (cit. on p. 67).

[Riv20] Theodore J. Rivlin. *Chebyshev polynomials*. Dover Publications, 2020 (cit. on p. 20).

[RC23] David Roberts and Aashish A. Clerk. "Exact Solution of the Infinite-Range Dissipative Transverse-Field Ising Model." In: *Physical Review Letters* 131.19 (2023), p. 190403. DOI: 10.1103/PhysRevLett.131.190403 (cit. on p. 172).

[RSY+17] Zhijian Rong, Jie Shen, Haijun Yu, et al. "A nodal sparse grid spectral element method for multi-dimensional elliptic partial differential equations." In: *Int. J. Numer. Anal. Model* 14.4-5 (2017), pp. 762–783 (cit. on p. 183).

[RT18] Diego Ruiz-Antolin and Alex Townsend. "A non-uniform fast Fourier transform based on low rank approximation." In: *SIAM Journal on Scientific Computing* 40.1 (2018), A529–A547 (cit. on pp. 28, 224, 226–228, 230, 250, 251, 263, 264).

[RG17] Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. "Quantum arithmetic with the quantum Fourier transform." In: *Quantum Information Processing* 16.6 (2017), p. 152 (cit. on pp. 20, 224, 264).

[ŞS21] Burak Şahinoğlu and Rolando D. Somma. "Hamiltonian simulation in the low-energy subspace." In: *npj Quantum Information* 7.1 (July 2021). DOI: 10.1038/s41534-021-00451-w (cit. on pp. 65, 102).

[Sch19] Wolfgang Scherer. *Mathematics of Quantum Computing*. Springer, 2019 (cit. on p. 30).

[SZ19a] Christoph Schwab and Jakob Zech. "Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ." In: *Analysis and Applications* 17.01 (2019), pp. 19–55 (cit. on p. 20).

[SZ19b] Christoph Schwab and Jakob Zech. "Deep learning in high dimension: neural network expression rates for generalized polynomial chaos expansions in UQ." In: *Anal. Appl. (Singap.)* 17.1 (2019), pp. 19–55. DOI: 10.1142/S0219530518500203. URL: https://doi.org/10.1142/S0219530518500203 (cit. on pp. 208, 210, 211).

[STW11] Jie Shen, Tao Tang, and Li-Lian Wang. *Spectral methods: algorithms, analysis and applications*. Vol. 41. Springer Science & Business Media, 2011 (cit. on p. 223).

[SYZ20] Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Deep Network Approximation Characterized by Number of Neurons." In: *Communications in Computational Physics* 28.5 (2020), pp. 1768–1811. DOI: 10.4208/cicp.OA-2020-0149. URL: https://global-sci.org/intro/article_detail/cicp/18396.html (cit. on p. 185).

[SYZ21a] Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Deep Network with Approximation Error Being Reciprocal of Width to Power of Square Root of Depth." In: *Neural Computation* 33.4 (2021), pp. 1005–1036. DOI: 10.1162/neco_a_01364. URL: https://direct.mit.edu/neco/article/33/4/1005/97476/Deep-Network-With-Approximation-Error-Being (cit. on p. 185).

[SYZ21b] Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Neural Network Approximation: Three Hidden Layers Are Enough." In: *Neural Networks* 141 (2021), pp. 160–173. DOI: 10.1016/j.neunet.2021.04.011. URL: https://www.sciencedirect.com/science/article/pii/S0893608021001465 (cit. on p. 185).

[SYZ22] Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Optimal approximation rate of ReLU networks in terms of width and depth." In: *Journal de Mathématiques Pures et Appliquées* 157 (2022), pp. 101–135. DOI: https://doi.org/10.1016/j.

matpur.2021.07.009. URL: https://www.sciencedirect.com/science/article/pii/S0021782421001124 (cit. on p. 185).

[Sho94] Peter W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring." In: *Proceedings 35th annual symposium on foundations of computer science*. IEEE. 1994, pp. 124–134 (cit. on p. 26).

[Sho99] Peter W. Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." In: *SIAM review* 41.2 (1999), pp. 303–332 (cit. on p. 224).

[SP22] Adenilton J. da Silva and Daniel K. Park. "Linear-depth quantum circuits for multiqubit controlled gates." In: *Phys. Rev. A* 106 (4 Oct. 2022), p. 042602. DOI: 10.1103/PhysRevA.106.042602. URL: https://link.aps.org/doi/10.1103/PhysRevA.106.042602 (cit. on pp. 190, 243).

[Sim96] Barry Simon. *Representations of Finite and Compact Groups*. American Mathematical Society, 1996 (cit. on p. 31).

[Sin+25] Akash Kumar Singh, Ashish Kumar Patra, Anurag K. S. V., Sai Shankar P., Ruchika Bhat, and Jaiganesh G. *A Polylogarithmic-Time Quantum Algorithm for the Laplace Transform*. 2025. arXiv: 2512.17980 [quant-ph]. URL: https://arxiv.org/abs/2512.17980 (cit. on p. 225).

[Smo63] Sergey Smolyak. "Quadrature and interpolation formulas for tensor products of certain classes of functions." In: *Dokl. Akad. Nauk SSSR* 148 (1963) (cit. on pp. 21, 183).

[Ste55] S. B. Stečkin. "On absolute convergence of orthogonal series." In: *Dokl. Akad. Nauk SSSR (N.S.)* 102 (1955), pp. 37–40 (cit. on pp. 208, 214).

[SHC21] Yuan Su, Hsin-Yuan Huang, and Earl T. Campbell. "Nearly tight Trotterization of interacting electrons." In: *Quantum* 5 (July 2021), p. 495. DOI: 10.22331/q-2021-07-05-495 (cit. on pp. 65, 102).

[Suz85] Masuo Suzuki. "Decomposition formulas of exponential operators and Lie exponentials with some applications to quantum mechanics and statistical physics." In: *Journal of mathematical physics* 26.4 (1985), pp. 601–612 (cit. on p. 11).

[Suz91] Masuo Suzuki. "General theory of fractal path integrals with applications to many-body theories and statistical physics." In: *Journal of mathematical physics* 32.2 (1991), pp. 400–407 (cit. on p. 11).

[Tak+21] Souichi Takahira, Asuka Ohashi, Tomohiro Sogabe, and Tsuyoshi Sasaki Usuda. *Quantum Algorithms based on the Block-Encoding Framework for Matrix Functions by Contour Integrals*. 2021. arXiv: 2106.08076 [quant-ph] (cit. on p. 107).

[TJ18] Li Tan and Jean Jiang. *Digital signal processing: fundamentals and applications*. Academic press, 2018 (cit. on p. 223).

[Til+22] Jules Tilly, Hongxiang Chen, Shuxiang Cao, et al. "The variational quantum eigensolver: a review of methods and best practices." In: *Physics Reports* 986 (2022), pp. 1–128 (cit. on p. 182).

[TZ22] Mao T. Tong and Ding-Xuan Zhou. "Approximation of functions from Korobov spaces by deep convolutional neural networks." In: *Advances in Computational Mathematics* 48.6 (2022), p. 84. DOI: https://doi.org/10.1007/s10444-022-09991-x (cit. on pp. 185, 205).

[Tor56] H. C. Torrey. "Bloch Equations with Diffusion Terms." In: *Physical Review* 104.3 (1956), pp. 563–565. DOI: 10.1103/PhysRev.104.563 (cit. on p. 160).

[TWZ17] Hoang Tran, Clayton G. Webster, and Guannan Zhang. "Analysis of quasi-optimal polynomial approximations for parameterized PDEs with deterministic and stochastic coefficients." In: *Numer. Math.* 137.2 (2017), pp. 451–493. DOI: 10.1007/s00211-017-0878-6. URL: https://doi.org/10.1007/s00211-017-0878-6 (cit. on p. 216).

[Tra+19] Andrew Tranter, Peter J. Love, Florian Mintert, Nathan Wiebe, and Peter V. Coveney. "Ordering of trotterization: Impact on errors in quantum simulation of electronic structure." In: *Entropy* 21.12 (2019), p. 1218 (cit. on p. 174).

[Tre19] Lloyd N. Trefethen. *Approximation theory and approximation practice, extended edition*. SIAM, 2019 (cit. on p. 20).

[TW14] Lloyd N. Trefethen and J.A.C. Weideman. "The exponentially convergent trapezoidal rule." In: *SIAM review* 56.3 (2014), pp. 385–458 (cit. on pp. 140, 145).

[Vaz+23] Almudena Carrera Vazquez, Daniel Egger, David Ochsner, and Stefan Woerner. "Well-conditioned multi-product formulas for hardware-friendly Hamiltonian simulation." In: *Quantum* 7 (2023), p. 1067. DOI: 10.22331/q-2023-07-25-1067 (cit. on pp. 68, 69).

[Vil+25] Ángel P. Villota-Cadena, Iván P. Sandoval-Palis, Gabriel F. Grijalva-Villegas, and Erick P. Herrera-Granda. "CDR-Solv: Solving the convection-diffusion-reaction equation with algebraic sub-grid scale stabilization using Python." In: *Applied Sciences* 15.18 (2025), p. 10256. DOI: 10.3390/app151810256 (cit. on p. 167).

[Wan+25] Chao Wang, Huan-Yu Liu, Cheng Xue, et al. *Quantum Simulation of Non-unitary Dynamics via Contour-based Matrix Decomposition*. 2025. arXiv: 2511.10267 [quant-ph] (cit. on p. 107).

[Wan+26a] Chao Wang, Xi-Ning Zhuang, Menghan Dou, Zhao-Yun Chen, and Guo-Ping Guo. *A Unified Poisson Summation Framework for Generalized Quantum Matrix Transformations*. 2026. arXiv: 2604.02874 [quant-ph] (cit. on p. 108).

[Wan+26b] Xinzhao Wang, Shuo Zhou, Xiaoyang Wang, Yi-Cong Zheng, Shengyu Zhang, and Tongyang Li. *Lindbladian Simulation with Commutator Bounds*. 2026. arXiv: 2603.28602 [quant-ph] (cit. on p. 106).

[War98] Antony F. Ware. "Fast approximate Fourier transforms for irregularly spaced data." In: *SIAM Review* 40.4 (1998), pp. 838–856 (cit. on p. 223).

[Wat+24] Jacob Watkins, Nathan Wiebe, Alessandro Roggero, and Dean Lee. "Time-dependent Hamiltonian simulation using discrete-clock constructions." In: *PRX Quantum* 5.4 (2024), p. 040316 (cit. on p. 69).

[Wat22] George N. Watson. *A treatise on the theory of Bessel functions*. The University Press, 1922 (cit. on p. 260).

[Wat+23] James D. Watson, Jacob Bringewatt, Alexander F. Shaw, Andrew M. Childs, Alexey V. Gorshkov, and Zohreh Davoudi. *Quantum Algorithms for Simulating Nuclear Effective Field Theories*. 2023. arXiv: 2312.05344 [quant-ph] (cit. on p. 102).

[WW25] James D. Watson and Jacob Watkins. "Exponentially reduced circuit depths using Trotter error mitigation." In: *PRX Quantum* 6.3 (2025), p. 030325 (cit. on pp. 68, 69).

[Wer97] Edgar M. E. Wermuth. "A Remark on Commuting Operator Exponentials." In: *Proceedings of the American Mathematical Society* 125.6 (1997), pp. 1685–1688. URL: http://www.jstor.org/stable/2162208 (cit. on p. 77).

[Wie+10] Nathan Wiebe, Dominic Berry, Peter Høyer, and Barry C. Sanders. "Higher order decompositions of ordered operator exponentials." In: *Journal of Physics A: Mathematical and Theoretical* 43.6 (2010), p. 065203 (cit. on p. 14).

[WEB25] Heather Wilber, Ethan N. Epperly, and Alex H. Barnett. "Superfast Direct Inversion of the Non-uniform Discrete Fourier Transform via Hierarchically Semiseparable Least Squares." In: *SIAM Journal on Scientific Computing* 47.3 (2025), A1702–A1732 (cit. on p. 224).

[Wil+23] Chelsea A. Williams, Annie E. Paine, Hsin-Yu Wu, Vincent E. Elfving, and Oleksandr Kyriienko. *Quantum Chebyshev Transform: Mapping, Embedding, Learning and Sampling Distributions*. 2023. arXiv: 2306.17026 [quant-ph]. URL: https://arxiv.org/abs/2306.17026 (cit. on p. 225).

[YL23] Yahong Yang and Yulong Lu. *Optimal Deep Neural Network Approximation for Korobov Functions with respect to Sobolev Norms*. 2023. arXiv: 2311.04779 [math.NA] (cit. on p. 185).

[Yan+23] Yahong Yang, Yue Wu, Haizhao Yang, and Yang Xiang. *Nearly Optimal Approximation Rates for Deep Super ReLU Networks on Sobolev Spaces*. 2023. arXiv: 2310.10766 [math.NA] (cit. on p. 185).

[YYX23] Yahong Yang, Haizhao Yang, and Yang Xiang. "Nearly optimal VC-dimension and pseudo-dimension bounds for deep neural network derivatives." In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 21721–21756 (cit. on p. 185).

[YC22] Changhao Yi and Elizabeth Crosson. "Spectral analysis of product formulas for quantum simulation." In: *npj Quantum Information* 8.1 (2022), p. 37 (cit. on pp. 65, 103).

[YW21] Xuchen You and Xiaodi Wu. "Exponentially many local minima in quantum neural networks." In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12144–12155 (cit. on p. 186).

[Yu+13] Qiang Yu, Fawang Liu, Ian Turner, and Kevin Burrage. "Numerical investigation of three types of space and time fractional Bloch–Torrey equations in 2D." In: *Central European Journal of Physics* 11.6 (2013), pp. 646–665. DOI: 10.2478/s11534-013-0220-6 (cit. on p. 161).

[Yu+24] Zhan Yu, Qiuhao Chen, Yuling Jiao, et al. "Non-asymptotic approximation error bounds of parameterized quantum circuits." In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024 (cit. on pp. 25, 26, 182, 185, 190, 199, 211, 216).

[Yu+22] Zhan Yu, Hongshun Yao, Mujin Li, and Xin Wang. "Power and limitations of single-qubit native quantum neural networks." In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 27810–27823. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/b250de41980b58d34d6aadc3f4aedd4c-Paper-Conference.pdf (cit. on p. 182).

[Zec18] Jakob Zech. *Sparse-Grid Approximation of High-Dimensional Parametric PDEs*. Dissertation ETH Zürich 25683. ETH Zürich, 2018. DOI: https://doi.org/10.3929/ethz-b-000340651 (cit. on pp. 22, 210, 213, 214).

[Zen+25] Pei Zeng, Jinzhao Sun, Liang Jiang, and Qi Zhao. "Simple and high-precision Hamiltonian simulation by compensating Trotter error with linear combination of unitary operations." In: *PRX Quantum* 6.1 (2025), p. 010359 (cit. on p. 65).

[ZL26] Henry Zhang and Joseph Li. *Efficient Quantum Circuits for the Hilbert Transform*. 2026. arXiv: 2601.10876 [quant-ph]. URL: https://arxiv.org/abs/2601.10876 (cit. on p. 225).

[ZSY22] Shijun Zhang, Zuowei Shen, and Haizhao Yang. "Deep Network Approximation: Achieving Arbitrary Accuracy with Fixed Number of Neurons." In: *Journal of Machine Learning Research* 23.276 (2022), pp. 1–60. URL: http://jmlr.org/papers/v23/21-1404.html (cit. on p. 185).

[Zha+22] Qi Zhao, You Zhou, Alexander F. Shaw, Tongyang Li, and Andrew M. Childs. "Hamiltonian Simulation with Random Inputs." In: *Physical Review Letters* 129.27 (Dec. 2022). DOI: 10.1103/physrevlett.129.270502 (cit. on p. 103).

[ZRB24] Sergiy Zhuk, Niall F Robertson, and Sergey Bravyi. "Trotter error bounds and dynamic multi-product formulas for Hamiltonian simulation." In: *Physical Review Research* 6.3 (2024), p. 033309 (cit. on p. 69).

[Zyl24] Julien Zylberman. *Fast Laplace transforms on quantum computers*. 2024. arXiv: 2412.05173 [quant-ph]. URL: https://arxiv.org/abs/2412.05173 (cit. on p. 225).