

ABSTRACT

Title of Dissertation: QUANTUM ALGORITHMS FOR LINEAR AND NONLINEAR DIFFERENTIAL EQUATIONS

Jin-Peng Liu
Doctor of Philosophy, 2022

Dissertation Directed by: Professor Andrew M. Childs
Department of Computer Science

Quantum computers are expected to dramatically outperform classical computers for certain computational problems. Originally developed for simulating quantum physics, quantum algorithms have been subsequently developed to address diverse computational challenges.

There has been extensive previous work for linear dynamics and discrete models, including Hamiltonian simulations and systems of linear equations. However, for more complex realistic problems characterized by differential equations, the capability of quantum computing is far from well understood. One fundamental challenge is the substantial difference between the linear dynamics of a system of qubits and real-world systems with continuum and nonlinear behaviors.

My research is concerned with mathematical aspects of quantum computing. In this dissertation, I focus mainly on the design and analysis of quantum algorithms for differential equations. Systems of linear ordinary differential equations (ODEs) and linear elliptic partial differential equations (PDEs) appear ubiquitous in natural and social science,

engineering, and medicine. I propose a variety of quantum algorithms based on finite difference methods and spectral methods for producing the quantum encoding of the solutions, with an exponential improvement in the precision over previous quantum algorithms.

Nonlinear differential equations exhibit rich phenomena in many domains but are notoriously difficult to solve. Whereas previous quantum algorithms for general nonlinear equations have been severely limited due to the linearity of quantum mechanics, I give the first efficient quantum algorithm for nonlinear differential equations with sufficiently strong dissipation. I also establish a lower bound, showing that nonlinear differential equations with sufficiently weak dissipation have worst-case complexity exponential in time, giving an almost tight classification of the quantum complexity of simulating nonlinear dynamics.

Overall, utilizing advanced linear algebra techniques and nonlinear analysis, I attempt to build a bridge between classical and quantum mechanics, understand and optimize the power of quantum computation, and discover new quantum speedups over classical algorithms with provable guarantees.

QUANTUM ALGORITHMS FOR LINEAR AND
NONLINEAR DIFFERENTIAL EQUATIONS

by

Jin-Peng Liu

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
2022

Advisory Committee:
Professor Andrew M. Childs, Chair/Advisor
Professor Alexey V. Gorshkov
Professor Carl A. Miller
Professor Konstantina Trivisa
Professor Xiaodi Wu

© Copyright by
Jin-Peng Liu
2022

Acknowledgments

I would like to express my gratitude to my advisor, Andrew M. Childs, for his invaluable guidance, encouragement, and support. Andrew is an incredible researcher as well as an amazing advisor from whom I can constantly learn a lot. Without Andrew, I cannot imagine how I would get through the challenges of my studies. He taught me all I know about becoming an independent researcher, from solving research problems, writing papers, to completing a professional presentation. I will never forget the days when we worked on problems, talked about my career path, and had lunch together on occasion. I am always proud of being a Childs group member.

I am grateful to Alexey Gorshkov, Carl Miller, Konstantina Trivisa, and Xiaodi Wu for serving as my dissertation committee members. Many thanks to Konstantina for her invaluable advice and warm-hearted assistance throughout my applied mathematics studies and during my post-doctoral applications. I would like to thank Furong Huang for serving as my preliminary exam committee member and for discussing machine learning topics.

The work described in this dissertation is the product of collaborations with many people. I cherish the opportunities to work with Dong An, Di Fang, Herman Kolden, Hari Krovi, Tongyang Li, Noah Linden, Nuno Loureiro, Ashley Montanaro, Aaron Ostrander, Changpeng Shao, Chunhao Wang, Chenyi Zhang, and Ruizhe Zhang. Despite the fact

that most of the collaborations were virtual, I benefited greatly from their diverse research backgrounds and areas of expertise. I expect to meet everyone in person in the future.

At the University of Maryland, I had the privilege of learning from many fantastic faculties, including Gorjan Alagic, John Baras, Alexander Barg, Jacob Bedrossian, James Duncan, Howard Elman, Tom Goldstein, Lise-Marie Imbert-Gérard, Pierre-Emmanuel Jabin, Richard La, Brad Lackey, David Levermore, Yi-Kai Liu, Ricardo Nochetto, and Eitan Tadmor. I appreciate the instructors' willingness to share knowledge and skills with me, as well as widening my horizons.

At the Joint Center for Quantum Information and Computer Science (QuICS), I had in-depth interactions with many quantum information colleagues: Chen Bai, Aniruddha Bapat, Charles Cao, Shouvanik Chakrabarti, Nai-Hui Chia, Ze-Pei Cian, Abhinav Deshpande, Bill Fefferman, Hong Hao Fu, Andrew Guo, Kaixin Huang, Shih-Han Hung, Jiaqi Leng, Yuxiang Peng, Eddie Schoute, Yuan Su, Minh Tran, Chiao-Hsuan Wang, Daochen Wang, Guoming Wang, Xin Wang, Yidan Wang, Xinyao Wu, Penghui Yao, Qi Zhao, Daiwei Zhu, Guanyu Zhu, and Shaopeng Zhu. We had wonderful days at QuICS. I would also like to thank Andrea Svejda, the QuICS coordinator. Whenever I was looking for assistance, she was always there to offer help.

I appreciated the support from the QISE-NET Award, which allows me to collaborate with Microsoft Quantum researchers: Guang Hao Low and Stephen Jordan. Special thanks to Stephen for his generous assistance during my graduate studies, regardless of whether he works for the University of Maryland or Microsoft. I enjoyed my internship at Amazon Web Sciences (AWS) Center for Quantum Computing in the summer of 2021, and I am grateful to Fernando Brandao, Earl Campbell, Michael Kastoryano, and Nicola

Pancotti for their mentorship. Over that summer, I had a lot of fun speaking with AWS scientists: Steve Flammia, Sam McArdle, Ash Milstead, and Martin Schuetz, as well as talking with my fellow internship students: Alexander Delzell, Hsin-Yuan Huang, Noah Shutty, Thomas Bohdanowicz, and Kianna Wan.

I enjoyed many unforgettable visits to Berkeley, Harvard, MIT, Caltech, Chicago, Colorado, and Shenzhen. In the winter of 2019 and the spring of 2020, I appreciated the hosts from the University of California, Berkeley, the Lawrence Berkeley National Laboratory, and the Simons Institute for the Theory of Computing. I would like to thank Lin Lin and Chao Yang for the numerous discussions about many valuable ideas, as well as the many challenging questions. I greatly appreciated Lin's strong support during my post-doctoral applications. I am also grateful to Umesh Vazirani for organizing the Simons programs and for inviting me to participate. In the winter of 2021, I had the pleasure of visiting the MIT Center for Theoretical Physics and Plasma Science and Fusion Center in Boston. I was delighted to interact with Soonwon Choi, Issac Chuang, Edward Farhi, Aram Harrow, and Seth Lloyd. Indeed, my graduate studies relied heavily on Harrow and Lloyd's results. It is an honor for me to collaborate with and compete against MIT folks. In the winter of 2021, I was also thrilled to interview the Harvard Quantum Initiative, where I had the opportunity to speak with Arthur Jaffe, Mikhail Lukin, and Susannie Yelin. In these years, I also enjoyed attending conferences hosted by Caltech, Chicago, Colorado, and Shenzhen. I would like to express my gratitude to many individuals for their friendliness and hospitality during my visits.

I enjoyed numerous enlightening discussions with many incredible quantum information researchers apart from those mentioned above: Scott Aaronson, Anurag Anshu, Ryan

Babbush, Dominic Berry, Adam Bouland, Sergey Bravyi, Paola Cappellaro, Steve Flammia, David Gosset, Stuart Hadfield, Matthew Hastings, Patrick Hayden, Zhengfeng Ji, Robin Kothari, Debbie Liang, Shunlong Luo, Jarrod McClean, Christopher Monroe, Michele Mosca, John Preskill, Yun Shang, Fang Song, Nikitas Stamatopoulos, Nathan Wiebe, Thomas Vidick, Beni Yoshida, Henry Yuan, Bei Zeng, William Zeng, and Shenggen Zheng. I wish I could be as smart as these amazing people.

I would also like to thank my colleagues and friends in the broader quantum information community: Kaifeng Bu, Chenfeng Cao, Ningping Cao, Lijie Chen, Mo Chen, Yifang Chen, Rui Chao, Andrea Coladangelo, David Ding, Yulong Dong, Xun Gao, András Gilyén, Cupjin Huang, Yichen Huang, Hong-Ye Hu, Jiala Ji, Jiaqing Jiang, Ce Jin, Gushu Li, Jianqiang Li, Yinan Li, Jiahui Liu, Jinguo Liu, Junyu Liu, Mengke Liu, Qipeng Liu, Yunchao Liu, Yupan Liu, Ziwen Liu, Chuhan Lu, Di Luo, Chinmay Nirkhe, Luowen Qian, Yihui Quek, Yixin Shen, Qichen Song, Ewin Tang, Yuanjia Wang, Yadong Wu, Zhuqing Xu, Yuxiang Yang, Jiahao Yao, Cong Yu, Zhan Yu, Haimeng Zhang, Jiayu Zhang, Yuxuan Zhang, Zhendong Zhang, Zijian Zhang, Chen Zhao, Chunlu Zhou, Hengyun Zhou, Shangnai Zhou, Sisi Zhou, and Jiamin Zhu. I wish you all the best in your future endeavors.

During my studies at the University of Maryland, I enjoyed illuminating discussions with fellow graduate students at the Applied Mathematics & Statistics, and Scientific Computation (AMSC) Program and the Department of Mathematics: Stephanie Allen, Christopher Dock, Alexis Boleda, Muhammed Elgeballi, Luke Evans, Blake Fritz, Siming He, Gareth Johnson, Sophie Kessler, Wenbo Li, Ying Li, Yiran Li, Jiaying Liang, Yuchen Luo, Jingcheng Lu, Michael Rawson, Tengfei Su, Manyuan Tao, Cem Unsal, Peng Wan,

Qiong Wu, Shuo Yang, Anqi Ye, Yiran Zhang, and Yi Zhou. These five years could not be so enjoyable without you. I would like to thank Jessica Sadler, the AMSC program coordinator, for offering help during my graduate years. I also enjoyed insightful interactions with many fellow graduate students at the Department of Computer Science: Jingling Li, Yanchao Sun, Jiahao Su, and Xuchen You. I would like to thank Jingling in particular for her thoughtfulness and heartwarming encouragement. Thank you for being there, and I will cherish every moment with you.

I would especially thank Yanwen Zhang for giving me the courage to believe in myself, so that I can complete my Ph.D. and pursue my dream of becoming a successful professor. Yanwen, I will never stop moving forward, as you encouraged.

Finally, I would like to thank my parents for their endless love, faith, and support. You are the true heroes.

Table of Contents

Acknowledgements	ii
Table of Contents	vii
List of Tables	ix
List of Figures	x
Chapter 1: Introduction: quantum scientific computation	1
1.1 Notations and terminologies	1
1.2 Hamiltonian simulations	4
1.3 Quantum linear system algorithms	7
1.4 Quantum algorithms for linear differential equations	10
1.5 Quantum algorithms for nonlinear differential equations	16
Chapter 2: High-precision quantum algorithms for linear ordinary differential equations	21
2.1 Introduction	21
2.2 Spectral method	27
2.3 Linear system	31
2.4 Solution error	37
2.5 Condition number	41
2.6 Success probability	49
2.7 State preparation	52
2.8 Main result	54
2.9 Boundary value problems	59
2.10 Discussion	63
Chapter 3: High-precision quantum algorithms for linear elliptic partial differential equations	65
3.1 Introduction	65
3.2 Linear PDEs	68
3.3 Finite difference method	71
3.3.1 Linear system	73
3.3.2 Condition number	74
3.3.3 Error analysis	78
3.3.4 FDM algorithm	80

3.3.5	Boundary conditions via the method of images	83
3.4	Multi-dimensional spectral method	86
3.4.1	Quantum shifted Fourier transform and quantum cosine transform	90
3.4.2	Linear system	95
3.4.3	Condition number	103
3.4.4	State preparation	117
3.4.5	Main result	119
3.5	Discussion and open problems	122
Chapter 4: Efficient quantum algorithms for dissipative nonlinear differential equations		125
4.1	Introduction	125
4.2	Quadratic ODEs	129
4.3	Quantum Carleman linearization	131
4.4	Algorithm analysis	135
4.4.1	Solution error	136
4.4.2	Condition number	154
4.4.3	State preparation	158
4.4.4	Measurement success probability	161
4.4.5	Proof of Theorem 4.1	164
4.5	Lower bound	169
4.5.1	Hardness of state discrimination	170
4.5.2	State discrimination with nonlinear dynamics	171
4.5.3	Proof of Theorem 4.2	175
4.6	Applications	177
4.7	Discussion	184
Chapter 5: Conclusion and future work		188
Bibliography		192

List of Tables

3.1	Summary of the time complexities of classical and quantum algorithms for d -dimensional PDEs with error tolerance ϵ . Portions of the complexity in bold represent best known dependence on that parameter.	68
-----	--	----

List of Figures

- 4.1 Integration of the forced viscous Burgers equation using Carleman linearization on a classical computer (source code available at <https://github.com/hermannkolden/CarlemanBurgers>). The viscosity is set so that the Reynolds number $Re = U_0 L_0 / \nu = 20$. The parameters $n_x = 16$ and $n_t = 4000$ are the number of spatial and temporal discretization intervals, respectively. The corresponding Carleman convergence parameter is $R = 43.59$. Top: Initial condition and solution plotted at a third of the nonlinear time $\frac{1}{3}T_{nl} = \frac{L_0}{3U_0}$. Bottom: l_2 norm of the absolute error between the Carleman solutions at various truncation levels N (left), and the convergence of the corresponding time-maximum error (right). 182

Chapter 1: Introduction: quantum scientific computation

Quantum computing exploits quantum-mechanical phenomena such as superposition and entanglement to perform computation. Quantum computers have the potential to dramatically outperform classical computers at solving diverse computational challenges.

Quantum scientific computation is a fast-growing multidisciplinary field that combines classical numerical analysis with advanced quantum technologies to model, analyze, and solve complex problems arising in physics, chemistry, biology, engineering, social sciences, and so on. Originally developed for simulating quantum physics, various quantum algorithms have been proposed to address scientific computational problems by performing linear algebra in Hilbert space. For such problems, quantum algorithms are supposed to provide polynomial and even exponential speedups.

The basic notations and terminologies of quantum computing are covered in this chapter, followed by quantum algorithms for scientific computation problems, including topics such as Hamiltonian simulations, linear systems, and differential equations.

1.1 Notations and terminologies

To comprehend the remainder of the dissertation, we provide a brief overview of the notations and terminologies of quantum computing. More details are available in

standard textbooks, such as Nielsen and Chuang, *Quantum Computation and Quantum Information* [1], and Watrous, *The Theory of Quantum Information* [2].

Quantum mechanics can be formulated in terms of linear algebra. Throughout this dissertation, we consider a finite-dimensional complex vector space \mathbb{C}^N equipped with an inner product (the Hilbert space). We usually take $N = 2^n$ for some non-negative integer n . We use the Dirac notation $|\psi\rangle$ to represent a quantum state, and $\langle\phi| = |\phi\rangle^\dagger$ to represent its Hermitian conjugation. The scalar $\langle\phi|\psi\rangle$ gives the inner product of $|\psi\rangle$ and $|\phi\rangle$. For the quantum state, we always assume $\langle\psi|\psi\rangle = 1$, i.e. $|\psi\rangle$ is a unit complex vector. We also let $\{|j\rangle\}_{j=1}^N$ be the standard basis of space. The j -th entry of $|\psi\rangle$ can be written as $\langle j|\psi\rangle$.

Given two quantum states, $|\psi_1\rangle \in \mathbb{C}^{N_1}$, $|\psi_2\rangle \in \mathbb{C}^{N_2}$, their tensor product can be written as $|\psi_1\rangle|\psi_2\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \in \mathbb{C}^{N_1 \times N_2}$, where \otimes denotes the Kronecker product. One quantum bit (one qubit) is a quantum state in \mathbb{C}^2 , and the tensor product of one-qubit state $|\psi_j\rangle$ forms an n -qubit state $|\psi_1\rangle \otimes \dots \otimes |\psi_n\rangle \in \mathbb{C}^{2^n}$.

The n -qubit quantum gate $U \in \mathbb{C}^{2^n \times 2^n}$ is a unitary matrix, i.e. $U^\dagger U = I$, where I is the identity matrix. It is used to map one n -qubit state to the other, i.e. $U : |\psi\rangle \rightarrow |\psi'\rangle$. A sequence of quantum gates as a product forms a (reversible) quantum logic circuit.

The universality of two-qubit gates means that every n -qubit gate can be written as a composition of a sequence of two-qubit gates. Therefore, we usually count the number of two-qubit gates as the gate complexity of quantum algorithms.

For the quantum measurement, we consider a quantum observable that corresponds to a Hermitian M . It has the spectral decomposition $M = \sum_m \lambda_m P_m$, where P_m is the projection operator onto the eigenspace, i.e. $P_m^2 = P_m$, associated the eigenvalue λ_m . We

usually assume $\sum_m P_m = I$. When the quantum state $|\psi\rangle$ is measured by M , the outcome is λ_m with probability $p_m = \langle\psi| P_m |\psi\rangle$, with $\sum_m p_m = 1$. After the measurement, the post-state is $P_m |\psi\rangle / \sqrt{p_m}$.

We now discuss the notations of norms. For a vector $a = [a_1, a_2, \dots, a_n] \in \mathbb{R}^n$, we denote the vector ℓ_p norm as

$$\|a\|_p := \left(\sum_{k=1}^n |a_k|^p \right)^{1/p}. \quad (1.1)$$

For a matrix $A \in \mathbb{R}^{n \times n}$, we denote the operator norm $\|\cdot\|_{p,q}$ induced by the vector ℓ_p and ℓ_q norms as

$$\|A\|_{p,q} := \sup_{x \neq 0} \frac{\|Ax\|_q}{\|x\|_p}, \quad \|A\|_p := \|A\|_{p,p}. \quad (1.2)$$

For a continuous scalar function $f(t) : [0, T] \rightarrow \mathbb{R}$, we denote the L^∞ norm as

$$\|f\|_\infty := \max_{t \in [0, T]} |f(t)|. \quad (1.3)$$

For a continuous scalar function $u(x, t) : \Omega \times [0, T] \rightarrow \mathbb{R}$, where $\Omega \subset \mathbb{R}^d$, for a fixed t , the L^p norm of $u(\cdot, t)$ is given by

$$\|u(\cdot, t)\|_{L^p(\Omega)} := \left(\int_{\Omega} |u(x, t)|^p dx \right)^{1/p}. \quad (1.4)$$

In particular, when no subscript is used in the vector, matrix and function norms, we mean $\|\cdot\| = \|\cdot\|_2$ by default.

For real functions $f, g : \mathbb{R} \rightarrow \mathbb{R}$, we write $f = O(g)$ if there exists $c > 0$, such that

$|f(\tau)| \leq c|g(\tau)|$ for all $\tau \in \mathbb{R}$. We write $f = \Omega(g)$ if $g = O(f)$, and $f = \Theta(g)$ if both $f = O(g)$ and $g = O(f)$. We use \tilde{O} to suppress logarithmic factors in the asymptotic expression, i.e. $f = \tilde{O}(g)$ if $f = O(g \text{ poly}(\log g))$.

1.2 Hamiltonian simulations

Simulating quantum physics is one of the primary applications of quantum computers [3]. The first explicit quantum simulation algorithm was proposed by Lloyd [4] using product formulas, and numerous quantum algorithms for quantum simulations have been developed since then [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27], with various applications ranging from quantum field theory [28, 29] to quantum chemistry [18, 30, 31] and condensed matter physics [32].

We first introduce the quantum oracles. We assume there is a state preparation oracle O_ψ that produces an N -dimensional quantum state $|\psi\rangle$, i.e.

$$O_\psi|0\rangle = |\psi\rangle. \tag{1.5}$$

We then assume there is a sparse matrix oracle O_H that computes the locations and values of nonzero entries of an $N \times N$ sparse Hermitian matrix H . In detail, on input (j, l) , O_H gives the location of the l -th nonzero entry in row j , denoted as k . Then the oracle O_H gives the value $H_{j,k}$, i.e.

$$O_H(|j\rangle|k\rangle|0\rangle) = |j\rangle|k\rangle|H_{j,k}\rangle. \tag{1.6}$$

The quantum oracle O_H is reversible, and it allows access to different input elements (j, l)

in superposition, which is essential for quantum computing. Here we require the number of nonzero entries of this matrix in every row and column is at most s , where s is much smaller than the dimension N .

We count the number of querying these oracles as the query complexity. If such oracles can be implemented by one- and two-qubit gates, we usually count the number of these one- and two-qubit gates as the gate complexity of quantum algorithms.

We state the Hamiltonian simulation problem as follows.

Problem 1.1. *In the Hamiltonian simulation problem, we consider an N -dimensional Hamiltonian system*

$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle, \quad |\psi(0)\rangle = |\psi_{\text{in}}\rangle. \quad (1.7)$$

Given the ability to prepare a quantum state $|\psi(0)\rangle$, a sparse matrix oracle to provide the locations and values of nonzero entries of a Hamiltonian H , and an evolution time T , the goal is to produce a quantum state that is ϵ -close to $|\psi(T)\rangle$ in ℓ_2 norm.

When the Hamiltonian H is time-independent, we have the close-form solution $|\psi(T)\rangle = e^{-iHT} |\psi(0)\rangle$, for which a computation process attempts to produce an evolution operator as an approximation of e^{-iHT} .

This is a difficult problem in classical computation, since a classical computer cannot even represent the quantum state efficiently. All classical algorithms require time complexity at least $\Omega(N)$ to explicitly store all the entries of the quantum state. In quantum computation, we intend to design a sequence of quantum gates (a quantum circuit) to process $|\psi(t)\rangle$ from $|\psi(0)\rangle$ to $|\psi(T)\rangle$ while lowering the cost dramatically.

In [Problem 1.1](#), we aim to produce the final state $|\psi(T)\rangle$ with query and gate complexity $\text{poly}(\log N)$, an exponential improvement over classical simulations.

The first explicit Hamiltonian simulation algorithm proposed by Lloyd is based on product formulas [\[4\]](#). Specifically, if H is a time-independent k -local Hamiltonian, i.e. $H = \sum_j H_j$ and each H_j acts on $k = O(1)$ qubits, then the evolution operator e^{-iHt} for a short time t can be well approximated by the Lie-Trotter formula,

$$e^{-iHt} = \prod_j e^{-iH_j t} + O(t^2). \quad (1.8)$$

Each operator $e^{-iH_j t}$ can be efficiently implemented on a quantum computer. For a long time t , we can divide the time interval onto r subintervals, in which we simulate each $e^{-iH_j t/r}$, and finally approximate e^{-iHt} by

$$e^{-iHt} = \left(\prod_j e^{-iH_j t/r} \right)^r + O(t^2/r). \quad (1.9)$$

It takes $r = O(\|H\|^2 t^2 / \epsilon)$ to ensure the error tolerance of exact and approximated normalized states in ℓ_2 norm at most ϵ . High-order product formulas are known for a better approximation.

Using the $2k$ th-order Suzuki formula, Berry, Ahokas, Cleve, and Sanders showed that the number of exponentials required for an approximation with error at most ϵ is

$5^{2k} \|H\|^{1+1/2k} t^{1+1/2k} / \epsilon^{1/2k}$ [\[33\]](#). A large body of work has substantially developed fast quantum algorithms based on product formulas [\[33, 34, 35, 36, 37, 38, 39, 40\]](#).

Recently, some Hamiltonian simulation algorithms based on post-Trotter methods have emerged diversely. Berry, Childs, Cleve, Kothari, and Somma proposed high-precision

Hamiltonian simulations by implementing the linear combinations of unitaries (LCU) with complexity $t \text{poly}(\log(t/\epsilon))$ [35, 36], an exponential improvement over Trotter-based Hamiltonian simulations with respect to ϵ . Low and Chuang developed Hamiltonian simulations based on the quantum signal processing and qubitization [41, 42] with complexity $O(t + \frac{\log(1/\epsilon)}{\log \log(1/\epsilon)})$, realizing an optimal tradeoff in terms of t and ϵ .

Note that the Hamiltonian simulation problem can be regarded as a particular initial value problem of the differential equation. A more general problem will be introduced later.

1.3 Quantum linear system algorithms

Quantum computers are expected to outperform classical computers for characterizing the solution of a N -dimensional linear system in Hilbert space. Originally proposed by Harrow, Hassidim, and Lloyd [43], advanced quantum linear system algorithms [43, 44, 45, 46, 47, 48, 49, 50, 51] have been well developed to provide a quantum state encoding the solution with complexity $\text{poly}(\log N)$. Such algorithms have been considerably applied to address high-dimensional problems governed by linear differential equations [52, 53, 54, 55, 56, 57, 58, 59, 60, 61].

We introduce the oracles for the quantum linear system problem. Given an N -dimensional vector b , we assume there is a state preparation oracle O_b as defined in (1.5) that produces an N -dimensional quantum state $|b\rangle$, where $|b\rangle$ is proportional to the vector b . Given an $N \times N$ sparse matrix A , we then assume there is a sparse matrix oracle O_A as defined in (1.6). In detail, on input (j, l) , O_A gives the location of the l -th nonzero entry

in row j , denoted as k . Then the oracle O_A gives the value $A_{j,k}$. Here we require the number of nonzero entries of this matrix in every row and column is at most s , where s is much smaller than the dimension N .

We state the quantum linear system problem as follows.

Problem 1.2. *In the quantum linear system problem, we consider an N -dimensional linear system*

$$Ax = b. \tag{1.10}$$

Given the ability to prepare a quantum state proportional to the vector b , and a sparse matrix oracle to provide the locations and values of nonzero entries of a matrix A , the goal is to produce a quantum state that is ϵ -close to the normalized $A^{-1}b$ in ℓ_2 norm.

It takes at least $\Omega(N)$ for a classical computer (and even a quantum computer) to explicitly write down every entry of an N -dimensional vector. In [Problem 1.2](#), we aim to design a quantum circuit to provide a quantum state as an encoding of the inverse solution $A^{-1}b$ with ℓ_2 normalization, with query and gate complexity $\text{poly}(\log N)$.

The first quantum linear system algorithm (QLSA), known as the HHL algorithm, was proposed by Harrow, Hassidim, and Lloyd [\[43\]](#). The algorithm requires that A be Hermitian so that it can be converted into a unitary operator. If A is not Hermitian, the linear system in [Problem 1.2](#) is modified as

$$\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix} \begin{pmatrix} 0 \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}. \tag{1.11}$$

The algorithm requires that a quantum state $|b\rangle$ proportional to the vector b is given.

Let $\{\lambda_j, |\nu_j\rangle\}_{j=1}^N$ be eigenvalues and eigenvectors of A , and let $|b\rangle$ be expanded on the eigenbasis of A as $|b\rangle = \sum_j \beta_j |\nu_j\rangle$.

The HHL algorithm queries the unitary operator e^{iAt} with the sparse matrix oracle. Hamiltonian simulation techniques are employed to perform e^{iAt} with a superposition of different time t onto $|b\rangle$. The HHL algorithm then estimates the corresponding λ_j for each eigenbasis of $|b\rangle$ ¹, yielding the state

$$\sum_j \beta_j |0\rangle |\nu_j\rangle \rightarrow \sum_j \beta_j |\lambda_j\rangle |\nu_j\rangle, \quad (1.12)$$

where each λ_j is stored in the ancilla register. Next, the HHL algorithm performs a controlled rotation to multiply λ_j^{-1} onto each eigenbasis,

$$\sum_j \beta_j |\lambda_j\rangle |\nu_j\rangle \rightarrow \sum_j \beta_j \lambda_j^{-1} |\lambda_j\rangle |\nu_j\rangle. \quad (1.13)$$

Noticing the fact that $A^{-1}|b\rangle = \sum_j \beta_j \lambda_j^{-1} |\nu_j\rangle$, we have prepared the inverse solution encoded in the quantum state.

This algorithm achieves query and gate complexity $\kappa^2 \text{poly}(\log N)/\epsilon$, where N is the dimension, ϵ measures the error tolerance of exact and approximated normalized states in ℓ_2 norm, and κ is the condition number of the matrix A .

Subsequent work has been widely developed to improve the performance of that algorithm. Regarding the dependence on ϵ , Childs, Kothari, and Somma exponentially improved the scale of ϵ from $\text{poly}(1/\epsilon)$ to $\text{poly}(\log(1/\epsilon))$ by utilizing the linear combinations

¹The HHL algorithm uses the quantum phase estimation (QPE) [62] to estimate the eigenvalues. More details can be found in [1].

of unitaries (LCU) [46]. The LCU approach approximates the inverse of A by employing a Fourier series or a Chebyshev polynomial, which can be implemented by linear combinations of unitary operations with cost $\text{poly}(\log(1/\epsilon))$. Gilyén, Su, Low, and Wiebe provided an alternative encoding to approximate the inverse of A based on the quantum singular value transformation (QSVT) with the same ϵ scaling [47].

The dependence on the condition number κ also attracts considerable attention. Although the best-known classical linear system solver based on conjugate gradient descent can produce explicit N entries of the solution with cost scaling as $\sqrt{\kappa}$, it is known that a quantum algorithm must make at least κ queries for encoding the solution of [Problem 1.2](#) in $\text{poly}(\log N)$ [43]. Compared to the HHL algorithm, Ambainis adopted the variable time amplitude amplification (VTAA) to improve the κ^2 scaling to linear scaling[44], but it resulted in a worse $1/\epsilon^3$ scaling. Childs, Kothari, and Somma combined VTAA with LCU as introduced above to reach $\kappa \text{poly}(\log N, \log(1/\epsilon))$ [46]. Alternatively, quantum adiabatic approaches [45, 48, 49, 51] have been recently investigated to reach the same complexity without using the VTAA.

1.4 Quantum algorithms for linear differential equations

Models governed by both ordinary differential equations (ODEs) and partial differential equations (PDEs) arise extensively in natural and social science, medicine, and engineering. Such equations characterize physical and biological systems that exhibit a wide variety of complex phenomena, including turbulence and chaos. By utilizing QLSAs, quantum algorithms offer the prospect of rapidly characterizing the solutions of high-dimensional

systems of linear ODEs [52, 53, 54] and PDEs [55, 56, 57, 58, 59, 60, 61].

We introduce the oracles for the quantum linear ODE problem. Given an N -dimensional vector $u(0)$, we assume there is a state preparation oracle O_0 as defined in (1.5) that produces an N -dimensional quantum state $|u(0)\rangle$, where $|u(0)\rangle$ is proportional to the vector $u(0)$. Given an N -dimensional vector $f(t)$ with a specific t , we assume there is a state preparation oracle O_f such that

$$O_f(|t\rangle|0\rangle) = |t\rangle|f(t)\rangle, \quad (1.14)$$

which produces an N -dimensional quantum state $|f(t)\rangle$ proportional to $f(t)$. Given an $N \times N$ sparse matrix $A(t)$ with a specific t , we assume there is a sparse matrix oracle O_A as defined in (1.6). In detail, on input (t, j, l) , O_A gives the location of the l -th nonzero entry in row j , denoted as k . Then the oracle O_A gives the value $A_{j,k}(t)$, i.e.

$$O_A(|t\rangle|j\rangle|k\rangle|0\rangle) = |t\rangle|j\rangle|k\rangle|A_{j,k}(t)\rangle. \quad (1.15)$$

Here we require the number of nonzero entries of this matrix in every row and column is at most s for any time t , where s is much smaller than the dimension N .

We informally state the quantum linear ODE problem as follows.

Problem 1.3. *In the quantum linear ODE problem, we are given a system of d -dimensional differential equations*

$$\frac{du(t)}{dt} = A(t)u(t) + f(t). \quad (1.16)$$

Given the ability to prepare a quantum state proportional to the initial condition $u(0)$ and

the inhomogeneity $f(t)$ with a specific t , a sparse matrix oracle to provide the locations and values of nonzero entries of a matrix $A(t)$ with a specific t , and an evolution time T , the goal is to produce a quantum state that is ϵ -close to the normalized $u(T)$ in ℓ_2 norm.

[Problem 2.1](#) gives a formal statement of [Problem 1.3](#).

Berry presented the first quantum algorithm for general linear ODEs [52]. This work explicitly considered the time-independent case, assuming that real parts of the eigenvalues of A are non-positive, whereas in principle, it is natural to extend it to general time-dependent ODEs. Berry's algorithm discretized the system of differential equations into small time intervals as a system of linear equations using the Euler method or high-order linear multistep methods [63, 64], for which QLSAs can be applied to produce an approximated encoded solution. For instance, the first-order forward Euler method approximates the time derivative at the point $x(t)$ as

$$\frac{dx(t)}{dt} = \frac{x(t+h) - x(t)}{h} + O(h). \quad (1.17)$$

The k th-order linear multistep methods can reduce the error to $O(h^k)$. This approach achieves complexity poly-logarithmic in the dimension d . However, when solving an equation over the interval $[0, T]$, the number of iterations is $T/h = \Theta(\epsilon^{-1/k})$ for fixed k , offering a total complexity $\text{poly}(1/\epsilon)$ even using high-precision QLSAs.

Reference [53] improved Berry's result from $\text{poly}(1/\epsilon)$ to $\text{poly}(\log(1/\epsilon))$ by using a high-precision QLSA based on linear combinations of unitaries [46] to solve a linear system that encodes a truncated Taylor series. However, this approach assumes that $A(t)$ and $f(t)$ are time-independent so that the solution of the ODE can be written as an explicit

series, and it is unclear how to generalize the algorithm to time-dependent ODEs. Prior to the work [54], it was an open problem regarding whether there was a quantum algorithm for time-dependent ODEs with complexity $\text{poly}(\log(1/\epsilon))$.

In [Chapter 2](#), we introduce the quantum spectral method with such an improvement [54]. Our main contribution is to implement a method that uses a global approximation of the solution instead of locally discretizing the ODEs into small time intervals. We do this by developing a quantum version of so-called *spectral methods*, a technique from classical numerical analysis that represents the components of the solution $u(t)_i \approx \sum_j c_{ij} \phi_j(t)$ as linear combinations of basis functions $\phi_j(t)$ expressing the time dependence. Specifically, we implement a Chebyshev pseudospectral method [65, 66] using a high-precision QLSA. This approach approximates the solution by a truncated Chebyshev series with undetermined coefficients and solves for those coefficients using a linear system that interpolates the differential equations. According to the convergence theory of spectral methods, the solution error decreases exponentially provided the solution is sufficiently smooth [67, 68]. We use the LCU-based QLSA to solve this linear system with high precision [46]. To analyze the algorithm, we upper bound the solution error and condition number of the linear system and lower bound the success probability of the final measurement. Overall, we show that the total complexity of this approach is $\text{poly}(\log(1/\epsilon))$ for general time-dependent ODEs. We give a formal statement of the main result in [Theorem 2.1](#).

It is natural to extend the ordinary differential equations to the partial differential equations (PDEs) by involving multivariate derivatives. Prominent examples include Maxwell's equations for electromagnetism, Boltzmann's equation and the Fokker-Planck equation in thermodynamics, and Schrödinger's equation in continuum quantum mechanics.

For solving PDEs on a digital computer, it is common to consider a system of linear equations that approximates the PDE on the grid space, and produce the solution on those grid points within the ℓ_2 discretization error ϵ .

We introduce the oracles for the quantum linear PDE problem. Given an N -dimensional vector $f(\mathbf{x})$ with a specific \mathbf{x} , we assume there is a state preparation oracle O_b as defined in (1.14) that produces an N -dimensional quantum state $|f(\mathbf{x})\rangle$, where $|f(\mathbf{x})\rangle$ is proportional to the vector $f(\mathbf{x})$. Given $N \times N$ sparse matrices $A_{j_1 j_2}(\mathbf{x})$, $A_j(\mathbf{x})$, and $A_0(\mathbf{x})$ with a specific \mathbf{x} , we then assume there are sparse matrix oracles as defined in (1.15). For instance, $A_{j_1 j_2}(\mathbf{x})$ is modeled by a sparse matrix oracle that, on input (m, l) , gives the location of the l -th nonzero entry in row m , denoted as n , and gives the value $A_{j_1 j_2}(\mathbf{x})_{m,n}$.

We informally state the quantum linear PDE problem as follows.

Problem 1.4. *In the quantum linear PDE problem, we are given a system of second-order d -dimensional equations*

$$\sum_{j_1, j_2=1}^d A_{j_1 j_2}(\mathbf{x}) \frac{\partial^2 u(\mathbf{x})}{\partial x_{j_1} \partial x_{j_2}} + \sum_{j=1}^d A_j(\mathbf{x}) \frac{\partial u(\mathbf{x})}{\partial x_j} + A_0(\mathbf{x}) u(\mathbf{x}) = f(\mathbf{x}). \quad (1.18)$$

Given the ability to prepare a quantum state proportional to the inhomogeneity $f(\mathbf{x})$, sparse matrix oracles to provide the locations and values of nonzero entries of a matrix $A_{j_1 j_2}(\mathbf{x})$, $A_j(\mathbf{x})$, and $A_0(\mathbf{x})$ on a set of interpolation nodes \mathbf{x} , the goal is to produce a quantum state $|u(\mathbf{x})\rangle$ that is ϵ -close to the normalized $u(\mathbf{x})$ on a set of interpolation nodes \mathbf{x} in ℓ_2 norm.

Problem 3.1 gives a formal statement of **Problem 1.4**, for which we consider additional technical assumptions introduced in **Chapter 3**.

The discretized solution $u(\boldsymbol{x})$ on a set of interpolation nodes \boldsymbol{x} is a multi-dimensional vector function. If each spatial coordinate has n discrete values, then n^d points are needed to discretize a d -dimensional problem. Simply producing the solution on these grid points takes time $\Omega(n^d)$.

Compared to classical algorithms, quantum algorithms can produce a quantum state proportional to the solution on the grid, which requires only $\text{poly}(d, \log n)$ space. There are a large variety of quantum algorithms using different numerical schemes and the QLSA to reach $\text{poly}(d, 1/\epsilon)$ [55, 56, 57, 58, 59, 60, 61], because of the additional approximation errors in the numerical schemes. It was unknown how to achieve $\text{poly}(d, \log(1/\epsilon))$ prior to the work [59].

In [Chapter 3](#), we introduce two classes of quantum algorithms for linear elliptic PDEs with such an improvement [59]. Our first algorithm is based on a quantum version of the FDM approach: we use a finite-difference approximation to produce a system of linear equations and then solve that system using the QLSA. We analyze our FDM algorithm as applied to Poisson's equation under periodic, Dirichlet, and Neumann boundary conditions. Whereas previous FDM approaches [69, 70] considered fixed orders of truncation, we adapt the order of truncation depending on ϵ , inspired by the classical adaptive FDM [71]. As the order increases, the eigenvalues of the FDM matrix approach the eigenvalues of the continuous Laplacian, allowing for more precise approximations. The main algorithm we present uses the quantum Fourier transform (QFT) and takes advantage of the high-precision LCU-based QLSA [46]. This quantum adaptive FDM approach produces a quantum state that approximates the solution of the Poisson's equation with complexity $d^{6.5} \text{poly}(\log d, \log(1/\epsilon))$.

We also propose a quantum algorithm for more general second-order elliptic PDEs under periodic or non-periodic Dirichlet boundary conditions. This algorithm is based on the quantum spectral method [54] that globally approximates the solution of a PDE by a truncated Fourier or Chebyshev series with undetermined coefficients, and then finds the coefficients by solving a linear system. This system is exponentially large in d , so solving it is infeasible for classical algorithms but feasible in a quantum context. To be able to apply the QLSA efficiently, we show how to make the system sparse using variants of the quantum Fourier transform. Our bound on the condition number of the linear system uses global strict diagonal dominance, and introduces a factor in the complexity that measures the extent to which this condition holds. We give a complexity of $d^2 \text{poly}(\log(1/\epsilon))$ for producing a quantum state approximating the solution of general second-order elliptic PDEs with Dirichlet boundary conditions.

Both of these approaches have complexity $\text{poly}(d, \log(1/\epsilon))$, providing optimal dependence on ϵ and an exponential improvement over classical methods with respect to d . We state our main results in [Theorem 3.1](#) and [Theorem 3.2](#).

1.5 Quantum algorithms for nonlinear differential equations

We now turn attention to the nonlinear generalization of [Problem 1.3](#). We focus here on differential equations with nonlinearities that can be expressed with quadratic polynomials. Note that polynomials of degree higher than two, and even more general nonlinearities, can be reduced to the quadratic case by introducing additional variables [72, 73]. The quadratic case also directly includes many archetypal models, such as the

logistic equation in biology, the Lorenz system in atmospheric dynamics, and the Navier–Stokes equations in fluid dynamics.

We introduce the oracles for the quantum quadratic PDE problem. Given an N -dimensional vector $u(0)$, we assume there is a state preparation oracle O_0 as defined in (1.5) that produces an N -dimensional quantum state $|u(0)\rangle$, where $|u(0)\rangle$ is proportional to the vector $u(0)$. Given $N \times N$ sparse matrices F_2 , F_1 , and $F_0(t)$ with a specific t , we then assume there are sparse matrix oracles as defined in (1.6) and (1.15). For instance, F_2 is modeled by a sparse matrix oracle O_{F_2} that, on input (j, l) , gives the location of the l -th nonzero entry in row j , denoted as k , and gives the value $(F_2)_{j,k}$.

We informally state the quantum quadratic ODE problem as follows.

Problem 1.5. *In the quantum quadratic ODE problem, we are given a system of N -dimensional differential equations*

$$\frac{du(t)}{dt} = F_2 u^2(t) + F_1 u(t) + F_0(t). \quad (1.19)$$

Given the ability to prepare a quantum state proportional to the initial condition $u(0)$ and sparse matrix oracles to provide the locations and values of nonzero entries of matrices F_2 , F_1 , and $F_0(t)$ for any specified t , and an evolution time T , the goal is to produce a quantum state that is ϵ -close to the normalized $u(T)$ in ℓ_2 norm.

[Problem 4.1](#) gives a formal statement of [Problem 1.5](#).

Early work on quantum algorithms for differential equations already considered the nonlinear case by Leyton and Osborne [74]. It gave a quantum algorithm that simulates the nonlinear ODE by storing and maintaining multiple copies of the solution. In each

iteration from $t \rightarrow t + \Delta t$, it costs multiple copies $|x(t)\rangle$ to represent the nonlinearity $F(x(t))$ to obtain one copy of $|x(t + \Delta t)\rangle$. The complexity of this approach is polynomial in the logarithm of the dimension but exponential in the evolution time, scaling as $O(1/\epsilon^T)$ due to exponentially increasing resources used to maintain sufficiently many copies of the solution throughout the evolution.

Recently, heuristic quantum algorithms for nonlinear ODEs have been studied. Reference [75] explores a linearization technique known as the Koopman–von Neumann method that might be amenable to the quantum linear system algorithm. In [76], the authors provide a high-level description of how linearization can help solve nonlinear equations on a quantum computer. However, neither paper makes precise statements about concrete implementations or running times of quantum algorithms. The recent preprint [77] also describes a quantum algorithm to solve a nonlinear ODE by linearizing it using a different approach from the one taken here. However, neither paper makes precise statements about concrete implementations or rigorous time complexities of quantum algorithms. The authors also do not describe how barriers such as those of [78] could be avoided in their approach.

While quantum mechanics is described by linear dynamics, possible nonlinear modifications of the theory have been widely studied. Generically, such modifications enable quickly solving hard computational problems (e.g., solving unstructured search among n items in time $\text{poly}(\log n)$), making nonlinear dynamics exponentially difficult to simulate in general [78, 79, 80]. Therefore, constructing efficient quantum algorithms for general classes of nonlinear dynamics has been considered largely out of reach.

Prior to the work [81], it was a long-standing and celebrated open problem regarding

whether quantum computing can efficiently characterize nonlinear differential equations.

In [Chapter 4](#), we design and analyze a quantum algorithm that overcomes this limitation using Carleman linearization [[73](#), [82](#), [83](#)]. This approach embeds polynomial nonlinearities into an infinite-dimensional system of linear ODEs, and then truncates it to obtain a finite-dimensional linear approximation. We discretize the finite ODE system in time using the forward Euler method and solve the resulting linear equations with the quantum linear system algorithm [[46](#), [84](#)]. We control the approximation error of this approach by combining a novel convergence theorem with a bound for the global error of the Euler method. Furthermore, we upper bound the condition number of the linear system and lower bound the success probability of the final measurement. Subject to the condition $R < 1$, where the quantity R characterizes the relative strength of the nonlinear and dissipative linear terms, we show that the total complexity of this *quantum Carleman linearization algorithm* is $sT^2q \text{poly}(\log T, \log n, \log 1/\epsilon)/\epsilon$, where s is the sparsity, T is the evolution time, q quantifies the decay of the final solution relative to the initial condition, n is the dimension, and ϵ is the allowed error (see [Theorem 4.1](#)). In the regime $R < 1$, this is an exponential improvement over [[74](#)], which has complexity exponential in T . We state our main algorithmic result in [Theorem 4.1](#).

We also provide a quantum lower bound for the worst-case complexity of simulating strongly nonlinear dynamics, demonstrating that the algorithm's condition $R < 1$ cannot be significantly improved in general. Following the approach of [[78](#), [79](#)], we construct a protocol for distinguishing two states of a qubit driven by a certain quadratic ODE. Provided $R \geq \sqrt{2}$, this procedure distinguishes states with overlap $1-\epsilon$ in time $\text{poly}(\log(1/\epsilon))$. Since nonorthogonal quantum states are hard to distinguish, this implies a lower bound

on the complexity of the quantum ODE problem. We state our main lower bound result in [Theorem 4.2](#).

Our quantum algorithm could potentially be applied to study models governed by quadratic ODEs arising in biology and epidemiology as well as in fluid and plasma dynamics. In particular, the celebrated Navier–Stokes equation with linear damping, which describes many physical phenomena, can be treated by our approach provided the Reynolds number is sufficiently small. We also note that while the formal validity of our arguments assumes $R < 1$, we find in one numerical experiment that our proposed approach remains valid for larger R .

The remainder of this dissertation is outlined as follows. [Chapter 2](#) covers high-precision quantum algorithms for linear ordinary differential equations. [Chapter 3](#) presents high-precision quantum algorithms for linear elliptic partial differential equations. [Chapter 4](#) introduces an efficient quantum algorithm for dissipative nonlinear differential equations. Finally, we conclude the results of the dissertation and discuss future work in [Chapter 5](#).

Chapter 2: High-precision quantum algorithms for linear ordinary differential equations

2.1 Introduction

In this chapter, we focus on systems of first-order linear ordinary differential equations (ODEs)¹. As earlier introduced in [Problem 1.3](#), such equations can be written in the form

$$\frac{dx(t)}{dt} = A(t)x(t) + f(t) \quad (2.1)$$

where $t \in [0, T]$ for some $T > 0$, the solution $x(t) \in \mathbb{C}^d$ is a d -dimensional vector, and the system is determined by a time-dependent matrix $A(t) \in \mathbb{C}^{d \times d}$ and a time-dependent inhomogeneity $f(t) \in \mathbb{C}^d$. Provided $A(t)$ and $f(t)$ are continuous functions of t , the initial value problem (i.e., the problem of determining $x(t)$ for a given initial condition $x(0)$) has a unique solution [\[85\]](#).

Recent work has developed quantum algorithms with the potential to extract information about solutions of systems of differential equations even faster than is possible classically. This body of work grew from the quantum linear systems algorithm (QLSA) [\[84\]](#), which produces a quantum state proportional to the solution of a sparse system of d linear

¹This chapter is based on the paper [\[54\]](#).

equations in time $\text{poly}(\log d)$. We have introduced the quantum linear system algorithm in [Chapter 1](#).

Berry presented the first efficient quantum algorithm for general linear ODEs [52]. His algorithm represents the system of differential equations as a system of linear equations using a linear multistep method and solves that system using the QLSA. This approach achieves complexity logarithmic in the dimension d and, by using a high-order integrator, close to quadratic in the evolution time T . While this method could in principle be applied to handle time-dependent equations, the analysis of [52] only explicitly considers the time-independent case for simplicity.

Since it uses a finite difference approximation, the complexity of Berry's algorithm as a function of the solution error ϵ is $\text{poly}(1/\epsilon)$ [52]. Reference [53] improved this to $\text{poly}(\log(1/\epsilon))$ by using a high-precision QLSA based on linear combinations of unitaries [46] to solve a linear system that encodes a truncated Taylor series. However, this approach assumes that $A(t)$ and $f(t)$ are time-independent so that the solution of the ODE can be written as an explicit series, and it is unclear how to generalize the algorithm to time-dependent ODEs.

Most of the aforementioned algorithms use a local approximation: they discretize the differential equations into small time intervals to obtain a system of linear equations or linear differential equations that can be solved by the QLSA or Hamiltonian simulation. For example, the central difference scheme approximates the time derivative at the point $x(t)$ as

$$\frac{dx(t)}{dt} = \frac{x(t+h) - x(t-h)}{2h} + O(h^2). \quad (2.2)$$

High-order finite difference or finite element methods can reduce the error to $O(h^k)$, where $k - 1$ is the order of the approximation. However, when solving an equation over the interval $[0, T]$, the number of iterations is $T/h = \Theta(\epsilon^{-1/k})$ for fixed k , giving a total complexity that is $\text{poly}(1/\epsilon)$ even using high-precision methods for the QLSA or Hamiltonian simulation.

For ODEs with special structure, some prior results already show how to avoid a local approximation and thereby achieve complexity $\text{poly}(\log(1/\epsilon))$. When $A(t)$ is anti-Hermitian and $f(t) = 0$, we can directly apply Hamiltonian simulation [37]; if A and f are time-independent, then [53] uses a Taylor series to achieve complexity $\text{poly}(\log(1/\epsilon))$. However, the case of general time-dependent linear ODEs had remained elusive.

In this chapter, we use a nonlocal representation of the solution of a system of differential equations to give a new quantum algorithm with complexity $\text{poly}(\log(1/\epsilon))$ even for time-dependent equations. While this is an exponential improvement in the dependence on ϵ over previous work, it does not necessarily give an exponential runtime improvement in the context of an algorithm with classical output. In general, statistical error will introduce an overhead of $\text{poly}(1/\epsilon)$ when attempting to measure an observable with precision ϵ . However, achieving complexity $\text{poly}(\log(1/\epsilon))$ can result in a polynomial improvement in the overall running time. In particular, if an algorithm is used as a subroutine k times, we should ensure error $O(1/k)$ for each subroutine to give an overall algorithm with bounded error. A subroutine with complexity $\text{poly}(\log(1/\epsilon))$ can potentially give significant polynomial savings in such a case.

Time-dependent linear differential equations describe a wide variety of systems in

science and engineering. Examples include the wave equation and the Stokes equation (i.e., creeping flow) in fluid dynamics [86], the heat equation and the Boltzmann equation in thermodynamics [87, 88], the Poisson equation and Maxwell’s equations in electromagnetism [89, 90], and of course Schrödinger’s equation in quantum mechanics. Moreover, some nonlinear differential equations can be studied by linearizing them to produce time-dependent linear equations (e.g., the linearized advection equation in fluid dynamics [91]).

We focus our discussion on first-order linear ODEs. Higher-order ODEs can be transformed into first-order ODEs by standard methods. Also, by discretizing space, PDEs with both time and space dependence can be regarded as sparse linear systems of time-dependent ODEs. Thus we focus on an equation of the form (2.1) with initial condition

$$x(0) = \gamma \tag{2.3}$$

for some specified $\gamma \in \mathbb{C}^d$. We assume that $A(t)$ is s -sparse (i.e., has at most s nonzero entries in any row or column) for any $t \in [0, T]$. Furthermore, we assume that $A(t)$, $f(t)$, and γ are provided by black-box subroutines (which serve as abstractions of efficient computations). In particular, following essentially the same model as in [53] (see also Section 1.1 of [46]), suppose we have an oracle $O_A(t)$ that, for any $t \in [0, T]$ and any given row or column specified as input, computes the locations and values of the nonzero entries of $A(t)$ in that row or column. We also assume oracles O_x and $O_f(t)$ that, for any $t \in [0, T]$, prepare normalized states $|\gamma\rangle$ and $|f(t)\rangle$ proportional to γ and $f(t)$, and that also compute $\|\gamma\|$ and $\|f(t)\|$, respectively. Given such a description of the instance, the goal is to produce a quantum state ϵ -close to $|x(T)\rangle$ (a normalized quantum state

proportional to $x(T)$).

As mentioned above, our main contribution is to implement a method that uses a global approximation of the solution. We do this by developing a quantum version of so-called *spectral methods*, a technique from classical numerical analysis that (approximately) represents the components of the solution $x(t)_i \approx \sum_j c_{ij} \phi_j(t)$ as linear combinations of basis functions $\phi_j(t)$ expressing the time dependence. Specifically, we implement a Chebyshev pseudospectral method [65, 66] using the QLSA. This approach approximates the solution by a truncated Chebyshev series with undetermined coefficients and solves for those coefficients using a linear system that interpolates the differential equations. According to the convergence theory of spectral methods, the solution error decreases exponentially provided the solution is sufficiently smooth [67, 68]. We use the LCU-based QLSA to solve this linear system with high precision [46]. To analyze the algorithm, we upper bound the solution error and condition number of the linear system and lower bound the success probability of the final measurement. Overall, we show that the total complexity of this approach is $\text{poly}(\log(1/\epsilon))$ for general time-dependent ODEs. Informally, we show the following:

Theorem 2.1 (Informal). *Consider a linear ODE (2.1) with given initial conditions. Assume $A(t)$ is s -sparse and diagonalizable, and $\text{Re}(\lambda_i(t)) \leq 0$ for all eigenvalues of $A(t)$. Then there exists a quantum algorithm that produces a state ϵ -close in l_2 norm to the exact solution, succeeding with probability $\Omega(1)$, with query and gate complexity $O(s\|A\|T \text{ poly}(\log(s\|A\|T/\epsilon)))$.*

In addition to initial value problems (IVPs), our approach can also address boundary

value problems (BVPs). Given an oracle for preparing a state $\alpha|x(0)\rangle+\beta|x(T)\rangle$ expressing a general boundary condition, the goal of the quantum BVP is to produce a quantum state ϵ -close to $|x(t)\rangle$ (a normalized state proportional to $x(t)$) for any desired $t \in [0, T]$. We also give a quantum algorithm for this problem with complexity $\text{poly}(\log(1/\epsilon))$, as follows:

Theorem 2.2 (Informal). *Consider a linear ODE (2.1) with given boundary conditions. Assume $A(t)$ is s -sparse and diagonalizable, and $\text{Re}(\lambda_i(t)) \leq 0$ for all eigenvalues of $A(t)$. Then there exists a quantum algorithm that produces a state ϵ -close in l_2 norm to the exact solution, succeeding with probability $\Omega(1)$, with query and gate complexity $O(s\|A\|^4 T^4 \text{poly}(\log(s\|A\|T/\epsilon)))$.*

We give formal statements of [Theorem 2.1](#) and [Theorem 2.2](#) in [Section 2.8](#) and [Section 2.9](#), respectively. Note that the dependence of the complexity on $\|A\|$ and T is worse for BVPs than for IVPs. This is because a rescaling approach that we apply for IVPs (introduced in [Section 2.3](#)) cannot be extended to BVPs.

The remainder of this chapter is organized as follows. [Section 2.2](#) introduces the spectral method and [Section 2.3](#) shows how to encode it into a quantum linear system. Then [Section 2.4](#) analyzes the exponential decrease of the solution error, [Section 2.5](#) bounds the condition number of the linear system, [Section 2.6](#) lower bounds the success probability of the final measurement, and [Section 2.7](#) describes how to prepare the initial quantum state. We combine these bounds in [Section 2.8](#) to establish the main result. We then extend the analysis for initial value problems to boundary value problems in [Section 2.9](#). Finally, we conclude in [Section 2.10](#) with a discussion of the results and

some open problems.

2.2 Spectral method

Spectral methods provide a way of solving differential equations using global approximations [67, 68]. The main idea of the approach is as follows. First, express an approximate solution as a linear combination of certain basis functions with undetermined coefficients. Second, construct a system of linear equations that such an approximate solution should satisfy. Finally, solve the linear system to determine the coefficients of the linear combination.

Spectral methods offer a flexible approach that can be adapted to different settings by careful choice of the basis functions and the linear system. A Fourier series provides an appropriate basis for periodic problems, whereas Chebyshev polynomials can be applied more generally. The linear system can be specified using Gaussian quadrature (giving a *spectral element method* or *Tau method*), or one can simply interpolate the differential equations using quadrature nodes (giving a *pseudo-spectral method*) [68]. Since general linear ODEs are non-periodic, and interpolation facilitates constructing a straightforward linear system, we develop a quantum algorithm based on the Chebyshev pseudo-spectral method [65, 66].

In this approach, we consider a truncated Chebyshev approximation $x(t)$ of the exact solution $\hat{x}(t)$, namely

$$x_i(t) = \sum_{k=0}^n c_{i,k} T_k(t), \quad i \in [d]_0 := \{0, 1, \dots, d-1\} \quad (2.4)$$

for any $n \in \mathbb{Z}^+$. Here $T_k(t) = \cos(k \arccos x)$ is the Chebyshev polynomial of the first kind. (See [54, Appendix A] for its properties.) The coefficients $c_{i,k} \in \mathbb{C}$ for all $i \in [d]_0$ and $k \in [n+1]_0$ are determined by demanding that $x(t)$ satisfies the ODE and initial conditions at a set of *interpolation nodes* $\{t_l\}_{l=0}^n$ (with $1 = t_0 > t_1 > \dots > t_n = -1$), where $x(t_0)$ and $x(t_n)$ are the initial and final states, respectively. In other words, we require

$$\frac{dx(t_l)}{dt} = A(t_l)x(t_l) + f(t_l), \quad \forall l \in [n+1], t \in [-1, 1], \quad (2.5)$$

and

$$x_i(t_0) = \gamma_i, \quad i \in [d]_0. \quad (2.6)$$

We choose the domain $[-1, 1]$ in (2.5) because this is the natural domain for Chebyshev polynomials. Correspondingly, in the following section, we rescale the domain of initial value problems to be $[-1, 1]$. We would like to be able to increase the accuracy of the approximation by increasing n , so that

$$\|\hat{x}(t) - x(t)\| \rightarrow 0 \quad \text{as} \quad n \rightarrow \infty. \quad (2.7)$$

There are many possible choices for the interpolation nodes. Here we use the Chebyshev-Gauss-Lobatto quadrature nodes, $t_l = \cos \frac{l\pi}{n}$ for $l \in [n+1]_0$, since these nodes achieve the highest convergence rate among all schemes with the same number of nodes [63, 64]. These nodes also have the convenient property that $T_k(t_l) = \cos \frac{kl\pi}{n}$, making it easy to compute the values $x_i(t_l)$.

To evaluate the condition (2.5), it is convenient to define coefficients $c_{i,k}^\ell$ for $i \in [d]_0$

and $k \in [n + 1]_0$ such that

$$\frac{dx_i(t)}{dt} = \sum_{k=0}^n c_{i,k}^\ell T_k(t). \quad (2.8)$$

We can use the differential property of Chebyshev polynomials,

$$2T_k(t) = \frac{T_{k+1}^\ell(t)}{k+1} - \frac{T_{k-1}^\ell(t)}{k-1}, \quad (2.9)$$

to determine the transformation between $c_{i,k}$ and $c_{i,k}^\ell$. Based on the property of derivatives of Chebyshev polynomials (as detailed in [54, Appendix A]), we have

$$c_{i,k}^\ell = \sum_{j=0}^n [D_n]_{kj} c_{i,j}, \quad i \in [d]_0, k \in [n + 1]_0, \quad (2.10)$$

where D_n is the $(n + 1) \times (n + 1)$ upper triangular matrix with nonzero entries

$$[D_n]_{kj} = \frac{2j}{\sigma_k}, \quad k + j \text{ odd}, j > k, \quad (2.11)$$

where

$$\sigma_k := \begin{cases} 2 & k = 0 \\ 1 & k \in [n] := \{1, 2, \dots, n\}. \end{cases} \quad (2.12)$$

Using this expression in (2.5), (2.10), and (2.11), we obtain the following linear equations:

$$\sum_{k=0}^n T_k(t_l) c_{i,k}^\ell = \sum_{j=0}^{d-1} A_{ij}(t_l) \sum_{k=0}^n T_k(t_l) c_{j,k} + f(t_l)_i, \quad i \in [d]_0, l \in [n + 1]_0. \quad (2.13)$$

We also demand that the Chebyshev series satisfies the initial condition $x_i(1) = \gamma_i$ for all $i \in [d]_0$. This system of linear equations gives a global approximation of the underlying system of differential equations. Instead of locally approximating the ODE at discretized times, these linear equations use the behavior of the differential equations at the $n + 1$ times $\{t_l\}_{l=0}^n$ to capture their behavior over the entire interval $[-1, 1]$.

Our algorithm solves this linear system using the high-precision QLSA [46]. Given an encoding of the Chebyshev coefficients c_{ik} , we can obtain the approximate solution $x(t)$ as a suitable linear combination of the c_{ik} , a computation that can also be captured within a linear system. The resulting approximate solution $x(t)$ is close to the exact solution $\hat{x}(t)$:

Lemma 2.1 (Lemma 19 of [67]). *Let $\hat{x}(t) \in C^{r+1}(-1, 1)$ be the solution of the differential equations (2.1) and let $x(t)$ satisfy (2.5) and (2.6) for $\{t_l = \cos \frac{l\pi}{n}\}_{l=0}^n$. Then there is a constant C , independent of n , such that*

$$\max_{t \in [-1, 1]} \|\hat{x}(t) - x(t)\| \leq C \max_{t \in [-1, 1]} \frac{\|\hat{x}^{(n+1)}(t)\|}{n^{r-2}}. \quad (2.14)$$

This shows that the convergence behavior of the spectral method is related to the smoothness of the solution. For a solution in C^{r+1} , the spectral method approximates the solution with $n = \text{poly}(1/\epsilon)$. Furthermore, if the solution is smoother, we have an even tighter bound:

Lemma 2.2 (Eq. (1.8.28) of [68]). *Let $\hat{x}(t) \in C^1(-1, 1)$ be the solution of the differential*

equations (2.1) and let $x(t)$ satisfy (2.5) and (2.6) for $\{t_l = \cos \frac{l\pi}{n}\}_{l=0}^n$. Then

$$\max_{t \in [-1,1]} \|\hat{x}(t) - x(t)\| \leq \sqrt{\frac{2}{\pi}} \max_{t \in [-1,1]} \|\hat{x}^{(n+1)}(t)\| \left(\frac{e}{2n}\right)^n. \quad (2.15)$$

For simplicity, we replace the value $\sqrt{2/\pi}$ by the upper bound of 1 in the following analysis.

This result implies that if the solution is in C^1 , the spectral method approximates the solution to within ϵ using only $n = \text{poly}(\log(1/\epsilon))$ terms in the Chebyshev series. Consequently, this approach gives a quantum algorithm with complexity $\text{poly}(\log(1/\epsilon))$.

2.3 Linear system

In this section we construct a linear system that encodes the solution of a system of differential equations via the Chebyshev pseudospectral method introduced in [Section 2.2](#). We consider a system of linear, first-order, time-dependent ordinary differential equations, and focus on the following initial value problem. This is a formal statement of [Problem 1.3](#).

Problem 2.1. *In the quantum ODE problem, we are given a system of equations*

$$\frac{dx(t)}{dt} = A(t)x(t) + f(t) \quad (2.16)$$

where $x(t) \in \mathbb{C}^d$, $A(t) \in \mathbb{C}^{d \times d}$ is s -sparse, and $f(t) \in \mathbb{C}^d$ for all $t \in [0, T]$. We assume that $A_{ij}, f_i \in C^1(0, T)$ for all $i, j \in [d]$. We are also given an initial condition $x(0) = \gamma \in \mathbb{C}^d$. Given oracles that compute the locations and values of nonzero entries

of $A(t)$ for any t^2 , and that prepare normalized states $|\gamma\rangle$ proportional to γ and $|f(t)\rangle$ proportional to $f(t)$ for any $t \in [0, T]$, the goal is to output a quantum state $|x(T)\rangle$ that is ϵ -close to the normalized $x(T)$ in ℓ_2 norm.

Without loss of generality, we rescale the interval $[0, T]$ onto $[-1, 1]$ by the linear map $t \mapsto 1 - 2t/T$. Under this rescaling, we have $\frac{d}{dt} \mapsto -\frac{T}{2} \frac{d}{dt}$, so $A \mapsto -\frac{T}{2}A$, which can increase the spectral norm. To reduce the dependence on T —specifically, to give an algorithm with complexity close to linear in T —we divide the interval $[0, T]$ into subintervals $[0, \Gamma_1], [\Gamma_1, \Gamma_2], \dots, [\Gamma_{m-1}, T]$ with $\Gamma_0 := 0, \Gamma_m := T$. Each subinterval $[\Gamma_h, \Gamma_{h+1}]$ for $h \in [m]_0$ is then rescaled onto $[-1, 1]$ with the linear map $K_h: [\Gamma_h, \Gamma_{h+1}] \rightarrow [-1, 1]$ defined by

$$K_h: t \mapsto 1 - \frac{2(t - \Gamma_h)}{\Gamma_{h+1} - \Gamma_h}, \quad (2.17)$$

which satisfies $K_h(\Gamma_h) = 1$ and $K_h(\Gamma_{h+1}) = -1$. To solve the overall initial value problem, we simply solve the differential equations for each successive interval (as encoded into a single system of linear equations).

Now let $\tau_h := |\Gamma_{h+1} - \Gamma_h|$ and define

$$A_h(t) := -\frac{\tau_h}{2} A(K_h(t)) \quad (2.18)$$

$$x_h(t) := x(K_h(t)) \quad (2.19)$$

$$f_h(t) := -\frac{\tau_h}{2} f(K_h(t)). \quad (2.20)$$

² $A(t)$ is modeled by a sparse matrix oracle O_A that, on input (j, l) , gives the location of the l -th nonzero entry in row j , denoted as k , and gives the value $A(t)_{j,k}$.

Then, for each $h \in [m]_0$, we have the rescaled differential equations

$$\frac{dx_h}{dt} = A_h(t)x_h(t) + f_h(t) \quad (2.21)$$

for $t \in [-1, 1]$ with the initial conditions

$$x_h(1) = \begin{cases} \gamma & h = 0 \\ x_{h-1}(-1) & h \in [m]. \end{cases} \quad (2.22)$$

By taking

$$\tau_h \leq \frac{2}{\max_{t \in [\Gamma_h, \Gamma_{h+1}]} \|A(t)\|} \quad (2.23)$$

where $\|\cdot\|$ denotes the spectral norm, we can ensure that $\|A_h(t)\| \leq 1$ for all $t \in [-1, 1]$.

In particular, it suffices to take

$$\tau := \max_{h \in \{0, 1, \dots, m-1\}} \tau_h \leq \frac{2}{\max_{t \in [0, T]} \|A(t)\|}. \quad (2.24)$$

Having rescaled the equations to use the domain $[-1, 1]$, we now apply the Chebyshev pseudospectral method. Following [Section 2.2](#), we substitute the truncated Chebyshev series of $x(t)$ into the differential equations with interpolating nodes $\{t_l = \cos \frac{l\pi}{n} : l \in [n]\}$, giving the linear system

$$\frac{dx(t_l)}{dt} = A_h(t_l)x(t_l) + f_h(t_l), \quad h \in [m]_0, l \in [n+1] \quad (2.25)$$

with initial condition

$$x(t_0) = \gamma. \quad (2.26)$$

Note that in the following, terms with $l = 0$ refer to this initial condition.

We now describe a linear system

$$L|X\rangle = |B\rangle \quad (2.27)$$

that encodes the Chebyshev pseudospectral approximation and uses it to produce an approximation of the solution at time T .

The vector $|X\rangle \in \mathbb{C}^{m+p} \otimes \mathbb{C}^d \otimes \mathbb{C}^{n+1}$ represents the solution in the form

$$|X\rangle = \sum_{h=0}^{m-1} \sum_{i=0}^{d-1} \sum_{l=0}^n c_{i,l}(\Gamma_{h+1}) |hil\rangle + \sum_{h=m}^{m+p} \sum_{i=0}^{d-1} \sum_{l=0}^n x_i |hil\rangle \quad (2.28)$$

where $c_{i,l}(\Gamma_{h+1})$ are the Chebyshev series coefficients of $x(\Gamma_{h+1})$ and $x_i := x(\Gamma_m)_i$ is the i th component of the final state $x(\Gamma_m)$.

The right-hand-side vector $|B\rangle$ represents the input terms in the form

$$|B\rangle = \sum_{h=0}^{m-1} |h\rangle |B(f_h)\rangle \quad (2.29)$$

where

$$|B(f_h)\rangle = \sum_{i=0}^{d-1} \gamma_i |i0\rangle + \sum_{i=0}^{d-1} \sum_{l=1}^n f_h(\cos \frac{l\pi}{n})_i |il\rangle, \quad h \in [m-1]. \quad (2.30)$$

Here γ is the initial condition and $f_h(\cos \frac{l\pi}{n})_i$ is i th component of f_h at the interpolation point $t_l = \cos \frac{l\pi}{n}$.

We decompose the matrix L in the form

$$L = \sum_{h=0}^{m-1} |h\rangle\langle h| \otimes (L_1 + L_2(A_h)) + \sum_{h=1}^m |h\rangle\langle h-1| \otimes L_3 + \sum_{h=m}^{m+p} |h\rangle\langle h| \otimes L_4 + \sum_{h=m+1}^{m+p} |h\rangle\langle h-1| \otimes L_5. \quad (2.31)$$

We now describe each of the matrices L_i for $i \in [5]$ in turn.

The matrix L_1 is a discrete representation of $\frac{dx}{dt}$, satisfying

$$|h\rangle\langle h| \otimes L_1 |X\rangle = \sum_{i=0}^{d-1} \sum_{k=0}^n T_k(t_0) c_{i,k} |hi0\rangle + \sum_{i=0}^{d-1} \sum_{l=1, k, r=0}^n T_k(t_l) [D_n]_{kr} c_{i,r} |hil\rangle \quad (2.32)$$

(recall from (2.8) and (2.10) that D_n encodes the action of the time derivative on a Chebyshev expansion). Thus L_1 has the form

$$L_1 = \sum_{i=0}^{d-1} \sum_{k=0}^n T_k(t_0) |i0\rangle\langle ik| + \sum_{i=0}^{d-1} \sum_{l=1, k, r=0}^n \cos \frac{kl\pi}{n} [D_n]_{kr} |il\rangle\langle ir| \quad (2.33)$$

$$= I_d \otimes (|0\rangle\langle 0| P_n + \sum_{l=1}^n |l\rangle\langle l| P_n D_n) \quad (2.34)$$

where the interpolation matrix is a *discrete cosine transform* matrix:

$$P_n := \sum_{l, k=0}^n \cos \frac{kl\pi}{n} |l\rangle\langle k|. \quad (2.35)$$

The matrix $L_2(A_h)$ discretizes $A_h(t)$, i.e.,

$$|h\rangle\langle h| \otimes L_2(A_h) |X\rangle = - \sum_{i, j=0}^{d-1} \sum_{l=1, k=0}^n A_h(t_l)_{ij} T_k(t_l) c_{j,k} |hil\rangle. \quad (2.36)$$

Thus

$$L_2(A_h) = - \sum_{i,j=0}^{d-1} \sum_{l=1, k=0}^n A_h(t_l)_{ij} \cos \frac{kl\pi}{n} |il\rangle\langle jk| \quad (2.37)$$

$$= - \sum_{l=1}^n A_h(t_l) \otimes |l\rangle\langle l| P_n. \quad (2.38)$$

Note that if A_h is time-independent, then

$$L_2(A_h) = -A_h \otimes P_n. \quad (2.39)$$

The matrix L_3 combines the Chebyshev series coefficients $c_{i,l}$ to produce x_i for each $i \in [d]_0$. To express the final state $x(-1)$, L_3 represents the linear combination $x_i(-1) = \sum_{k=0}^n c_{i,k} T_k(-1) = \sum_{k=0}^n (-1)^k c_{i,k}$. Thus we take

$$L_3 = \sum_{i=0}^{d-1} \sum_{k=0}^n (-1)^k |i0\rangle\langle ik|. \quad (2.40)$$

Notice that L_3 has zero rows for $l \in [n]$.

When $h = m$, L_4 is used to construct x_i from the output of L_3 for $l = 0$, and to repeat x_i n times for $l \in [n]$. When $m+1 \leq h \leq m+p$, both L_4 and L_5 are used to repeat x_i $(n+1)p$ times for $l \in [n]$. This repetition serves to increase the success probability of the final measurement. In particular, we take

$$L_4 = - \sum_{i=0}^{d-1} \sum_{l=1}^n |il\rangle\langle il-1| + \sum_{i=0}^{d-1} \sum_{l=0}^n |il\rangle\langle il| \quad (2.41)$$

and

$$L_5 = - \sum_{i=0}^{d-1} |i0\rangle \langle in|. \quad (2.42)$$

In summary, the linear system is as follows. For each $h \in [m]_0$, $(L_1 + L_2(A_h))|X\rangle = |B_h\rangle$ solves the differential equations over $[\Gamma_h, \Gamma_{h+1}]$, and the coefficients $c_{i,l}(\Gamma_{h+1})$ are combined by L_3 into the $(h+1)$ st block as initial conditions. When $h = m$, the final coefficients $c_{i,l}(\Gamma_m)$ are combined by L_3 and L_4 into the final state with coefficients x_i , and this solution is repeated $(p+1)(n+1)$ times by L_4 and L_5 .

2.4 Solution error

In this section, we bound how well the solution of the linear system defined above approximates the actual solution of the system of differential equations.

Lemma 2.3. *For the linear system $L|X\rangle = |B\rangle$ defined in (2.27), let x be the approximate ODE solution specified by the linear system and let \hat{x} be the exact ODE solution. Then for n sufficiently large, the error in the solution at time T satisfies*

$$\|\hat{x}(T) - x(T)\| \leq m \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \frac{e^{n+1}}{(2n)^n}. \quad (2.43)$$

Proof. First we carefully choose n satisfying

$$n \geq \frac{e}{2} \left\lceil \frac{\log(\omega)}{\log(\log(\omega))} \right\rceil \quad (2.44)$$

where

$$\omega := \max_{t \in [0, T]} \frac{\|\hat{x}^{(n+1)}(t)\|}{\|\gamma\|} (m+1) \quad (2.45)$$

to ensure that

$$\max_{t \in [0, T]} \frac{\|\hat{x}^{(n+1)}(t)\|}{\|\gamma\|} \left(\frac{e}{2n}\right)^n \leq \frac{1}{m+1}. \quad (2.46)$$

According to the quantum spectral method defined in [Section 2.3](#), we solve

$$\frac{dx}{dt} = A_h(t)x(t) + f_h(t), \quad h \in [m]_0. \quad (2.47)$$

We denote the exact solution by $\hat{x}(\Gamma_{h+1})$, and we let $x(\Gamma_{h+1}) = \sum_{i=0}^d \sum_{l=0}^n (-1)^n c_{i,l}(\Gamma_{h+1})$, where $c_{i,l}(\Gamma_{h+1})$ is defined in [\(2.28\)](#). Define

$$\Delta_{h+1} := \|\hat{x}(\Gamma_{h+1}) - x(\Gamma_{h+1})\|. \quad (2.48)$$

For $h = 0$, [Lemma 2.2](#) implies

$$\Delta_1 = \|\hat{x}(\Gamma_1) - x(\Gamma_1)\| \leq \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \left(\frac{e}{2n}\right)^n. \quad (2.49)$$

For $h \in [m]$, the error in the approximate solution of $\frac{dx}{dt} = A_h(t)x(t) + f_h(t)$ has two contributions: the error from the linear system and the error in the initial condition. We let $\tilde{x}(\Gamma_{h+1})$ denote the solution of the linear system $(L_1 + L_2(A_h))|\tilde{x}(\Gamma_{h+1})\rangle = |B(f_h)\rangle$ under the initial condition $\hat{x}(\Gamma_h)$. Then

$$\Delta_{h+1} \leq \|\hat{x}(\Gamma_{h+1}) - \tilde{x}(\Gamma_{h+1})\| + \|\tilde{x}(\Gamma_{h+1}) - x(\Gamma_{h+1})\|. \quad (2.50)$$

The first term can be bounded using [Lemma 2.2](#), giving

$$\|\hat{x}(\Gamma_{h+1}) - \tilde{x}(\Gamma_{h+1})\| \leq \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \left(\frac{e}{2n}\right)^n. \quad (2.51)$$

The second term comes from the initial error Δ_h , which is transported through the linear system. Let

$$E_{h+1} = \hat{E}_{h+1} + \delta_{h+1} \quad (2.52)$$

where E_{h+1} is the solution of the linear system with input Δ_h and \hat{E}_{h+1} is the exact solution of $\frac{dx}{dt} = A_{h+1}(t)x(t) + f_{h+1}(t)$ with initial condition $x(\Gamma_h) = \Delta_h$. Then by [Lemma 2.2](#),

$$\|\delta_{h+1}\| = \|\hat{E}_{h+1} - E_{h+1}\| \leq \frac{\Delta_h}{\|\gamma\|} \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \left(\frac{e}{2n}\right)^n, \quad (2.53)$$

so

$$\|\tilde{x}(\Gamma_{h+1}) - x(\Gamma_{h+1})\| \leq \Delta_h + \frac{\Delta_h}{\|\gamma\|} \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \left(\frac{e}{2n}\right)^n. \quad (2.54)$$

Thus, we have an inequality recurrence for bounding Δ_h :

$$\Delta_{h+1} \leq \left(1 + \max_{t \in [0, T]} \frac{\|\hat{x}^{(n+1)}(t)\|}{\|\gamma\|} \left(\frac{e}{2n}\right)^n\right) \Delta_h + \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \left(\frac{e}{2n}\right)^n. \quad (2.55)$$

Now we iterate h from 1 to m . Equation (2.46) implies

$$\max_{t \in [0, T]} \frac{\|\hat{x}^{(n+1)}(t)\|}{\|\gamma\|} \left(\frac{e}{2n}\right)^n \leq \frac{1}{m+1} \leq \frac{1}{m}, \quad (2.56)$$

so

$$\left(1 + \max_{t \in [0, T]} \frac{\|\hat{x}^{(n+1)}(t)\|}{\|\gamma\|} \left(\frac{e}{2n}\right)^{m-1}\right) \leq \left(1 + \frac{1}{m}\right)^m \leq e. \quad (2.57)$$

Therefore

$$\begin{aligned} \Delta_m &\leq \left(1 + \max_{t \in [0, T]} \frac{\|\hat{x}^{(n+1)}(t)\|}{\|\gamma\|} \left(\frac{e}{2n}\right)^{m-1}\right) \Delta_1 \\ &\quad + \sum_{h=1}^{m-1} \left(1 + \max_{t \in [0, T]} \frac{\|\hat{x}^{(n+1)}(t)\|}{\|\gamma\|} \left(\frac{e}{2n}\right)^{h-1}\right) \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \left(\frac{e}{2n}\right)^h \\ &\leq \left(1 + \frac{1}{m}\right)^{m-1} \Delta_1 + (m-1) \left(1 + \frac{1}{m}\right)^{m-1} \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \left(\frac{e}{2n}\right)^m \\ &\leq \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \frac{e^{n+1}}{(2n)^n} + (m-1) \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \frac{e^{n+1}}{(2n)^n} \\ &= m \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \frac{e^{n+1}}{(2n)^n}, \end{aligned} \quad (2.58)$$

which shows that the solution error decreases exponentially with n . In other words, the linear system approximates the solution with error ϵ using $n = \text{poly}(\log(1/\epsilon))$. \square

Note that for time-independent differential equations, we can directly estimate $\|\hat{x}^{(n+1)}(t)\|$ using

$$\hat{x}^{(n+1)}(t) = A_h^{n+1} \hat{x}(t) + A_h^n f_h. \quad (2.59)$$

Writing $A_h = V_h \Lambda_h V_h^{-1}$ where $\Lambda_h = \text{diag}(\lambda_0, \dots, \lambda_{d-1})$, we have $e^{A_h} = V_h e^{\Lambda_h} V_h^{-1}$.

Thus the exact solution of time-independent equation with initial condition $\hat{x}(1) = \gamma$ is

$$\begin{aligned} \hat{x}(t) &= e^{A_h(1-t)} \gamma + (e^{A_h(1-t)} - I) A_h^{-1} f_h \\ &= V_h e^{\Lambda_h} V_h^{-1} \gamma + V_h (e^{\Lambda_h(1-t)} - I) \Lambda_h^{-1} V_h^{-1} f_h. \end{aligned} \quad (2.60)$$

Since $\text{Re}(\lambda_i) \leq 0$ for all eigenvalues λ_i of A_h for $i \in [d]_0$, we have $\|e^{A_h}\| \leq 1$. Therefore

$$\|\hat{x}(t)\| \leq \kappa_V(\|\gamma\| + 2\|f_h\|). \quad (2.61)$$

Furthermore, since $\max_{h,t} \|A_h(t)\| \leq 1$, we have

$$\begin{aligned} \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| &\leq \max_{t \in [0, T]} (\|\hat{x}(t)\| + \|f_h(t)\|) \\ &\leq \kappa_V(\|\gamma\| + 3\|f_h\|) \\ &\leq \kappa_V(\|\gamma\| + 2\tau\|f\|). \end{aligned} \quad (2.62)$$

Thus the solution error satisfies

$$\|\hat{x}(T) - x(T)\| \leq m\kappa_V(\|\gamma\| + 2\tau\|f\|) \frac{e^{n+1}}{(2n)^n}. \quad (2.63)$$

Note that, although we represent the solution differently, this bound is similar to the corresponding bound in [53, Theorem 6].

2.5 Condition number

We now analyze the condition number of the linear system.

Lemma 2.4. *Consider an instance of the quantum ODE problem as defined in [Problem 2.1](#).*

For all $t \in [0, T]$, assume $A(t)$ can be diagonalized as $A(t) = V(t)\Lambda(t)V^{-1}(t)$ for some $\Lambda(t) = \text{diag}(\lambda_0(t), \dots, \lambda_d(t))$, with $\text{Re}(\lambda_i(t)) \leq 0$ for all $i \in [d]_0$. Let $\kappa_V := \max_{t \in [0, T]} \kappa_V(t)$ be an upper bound on the condition number of $V(t)$. Then for $m, p \in \mathbb{Z}^+$

and n sufficiently large, the condition number of L in the linear system (2.27) satisfies

$$\kappa_L \leq (\pi m + p + 2)(n + 1)^{3.5}(2\kappa_V + e\|\gamma\|). \quad (2.64)$$

Proof. We begin by bounding the norms of some operators that appear in the definition of L . First we consider the l_1 norm of D_n since this is straightforward to calculate:

$$\|D_n\|_1 := \max_{1 \leq i \leq n} \sum_{j=0}^n |[D_n]_{ij}| = \begin{cases} \frac{n(n+2)}{2} & n \text{ even,} \\ \frac{(n+1)^2}{2} - 2 & n \text{ odd.} \end{cases} \quad (2.65)$$

Thus we have the upper bound

$$\|D_n\| \leq \sqrt{n+1} \|D_n\|_1 \leq \frac{(n+1)^{2.5}}{2}. \quad (2.66)$$

Next we upper bound the spectral norm of the discrete cosine transform matrix P_n :

$$\|P_n\|^2 \leq \max_{0 \leq l \leq n} \sum_{k=0}^n \cos^2 \frac{kl\pi}{n} \leq \max_{0 \leq l \leq n} \{n+1\} = n+1. \quad (2.67)$$

Therefore

$$\|P_n\| \leq \sqrt{n+1}. \quad (2.68)$$

Thus we can upper bound the norm of L_1 as

$$\|L_1\| \leq \|D_n\| \|P_n\| \leq \frac{(n+1)^3}{2}. \quad (2.69)$$

Next we consider the spectral norm of $L_2(A_h)$ for any $h \in [m]_0$. We have

$$L_2(A_h) = - \sum_{l=1}^n A_h(t_l) \otimes |l\rangle\langle l| P_n. \quad (2.70)$$

Since the eigenvalues of each $A_h(t_l)$ for $l \in [n+1]_0$ are all eigenvalues of

$$\sum_{l=0}^n A_h(t_l) \otimes |l\rangle\langle l|, \quad (2.71)$$

we have

$$\left\| \sum_{l=1}^n A_h(t_l) \otimes |l\rangle\langle l| \right\| \leq \left\| \sum_{l=0}^n A_h(t_l) \otimes |l\rangle\langle l| \right\| \leq \max_{t \in [1,1]} \|A_h(t)\| \leq 1 \quad (2.72)$$

by (2.23). Therefore

$$\|L_2(A_h)\| \leq \|P_n\| \leq \sqrt{n+1}. \quad (2.73)$$

By direct calculation, we have

$$\|L_3\| = \sqrt{n+1}, \quad (2.74)$$

$$\|L_4\| \leq 2, \quad (2.75)$$

$$\|L_5\| = 1. \quad (2.76)$$

Thus, for $n \geq 5$, we find

$$\|L\| \leq \frac{(n+1)^3}{2} + \sqrt{n+1} + \sqrt{n+1} + 2 + 1 \leq (n+1)^3. \quad (2.77)$$

Next we upper bound $\|L^{-1}\|$. By definition,

$$\|L^{-1}\| = \sup_{k \in [d]_0} \|L^{-1}|B\rangle\|. \quad (2.78)$$

We express $|B\rangle$ as

$$|B\rangle = \sum_{h=0}^{m+p} \sum_{l=0}^n \sum_{i=0}^{d-1} \beta_{hil} |hil\rangle = \sum_{h=0}^{m+p} \sum_{l=0}^n |b_{hl}\rangle \quad (2.79)$$

where $|b_{hl}\rangle := \sum_{i=0}^{d-1} \beta_{hil} |hil\rangle$ satisfies $\| |b_{hl}\rangle \|^2 = \sum_{i=0}^{d-1} |\beta_{hil}|^2 \leq 1$. For any fixed $h \in [m+p+1]_0$ and $l \in [n+1]_0$, we first upper bound $\|L^{-1}|b_{hl}\rangle\|$ and use this to upper bound the norm of L^{-1} applied to linear combinations of such vectors.

Recall that the linear system comes from (2.13), which is equivalent to

$$\sum_{k=0}^n T_k^\theta(t_r) c_{i,k}(\Gamma_h) = \sum_{j=0}^{d-1} A_h(t_r)_{ij} \sum_{k=0}^n T_k(t_r) c_{j,k}(\Gamma_h) + f_h(t_r)_i, \quad i \in [d]_0, r \in [n+1]_0. \quad (2.80)$$

For fixed $h \in [m+p+1]_0$ and $r \in [n+1]_0$, define vectors $x_{hr}, x_{hr}^\theta \in \mathbb{C}^d$ with

$$(x_{hr})_i := \sum_{k=0}^n T_k(t_r) c_{i,k}(\Gamma_h), \quad (x_{hr}^\theta)_i := \sum_{k=0}^n T_k^\theta(t_r) c_{i,k}(\Gamma_h) \quad (2.81)$$

for $i \in [d]_0$. We claim that $x_{hr} = x_{hr}^\theta = 0$ for any $r \neq l$. Combining only the equations from (2.80) with $r \neq l$ gives the system

$$x_{hr}^\theta = A_h(t_r) x_{hr}. \quad (2.82)$$

Consider a corresponding system of differential equations

$$\frac{d\hat{x}_{hr}(t)}{dt} = A_h(t_r)\hat{x}(t) + b \quad (2.83)$$

where $\hat{x}_{hr}(t) \in \mathbb{C}^d$ for all $t \in [-1, 1]$. The solution of this system with $b = 0$ and initial condition $\hat{x}_{hr}(1) = 0$ is clearly $\hat{x}_{hr}(t) = 0$ for all $t \in [-1, 1]$. Then the n th-order truncated Chebyshev approximation of (2.83), which should satisfy the linear system (2.82) by (2.4) and (2.5), is exactly x_{hr} . Using Lemma 2.3 and observing that $\hat{x}^{(n+1)}(t) = 0$, we have

$$x_{hr} = \hat{x}_{hr}(t) = 0. \quad (2.84)$$

When $t = t_l$, we let $|B\rangle = |b_{hl}\rangle$ denote the first nonzero vector. Combining only the equations from (2.80) with $r = l$ gives the system

$$x_{hl}^0 = A_h(t_l)x_{hl}. \quad (2.85)$$

Consider a corresponding system of differential equations

$$\frac{d\hat{x}_{hr}(t)}{dt} = A_h(t_r)\hat{x}(t) + b, \quad (2.86)$$

with $\gamma = b_{h0}, b = 0$ for $l = 0$; or $\gamma = 0, b = b_{hl}$ for $l \in [n]$.

Using the diagonalization $A_h(t_l) = V(t_l)\Lambda_h(t_l)V^{-1}(t_l)$, we have $e^A = V(t_l)e^{\Lambda_h(t_l)}V^{-1}(t_l)$.

Thus the exact solution of the differential equations (2.83) with $r = l$ and initial condition

$\hat{x}_{hr}(1) = \gamma$ is

$$\begin{aligned}\hat{x}_{hr}(t) &= e^{A_h(t_i)(1-t)}\gamma + (e^{A_h(t_i)(1-t)} - I)A_h(t_i)^{-1}b \\ &= V(t_i)e^{\Lambda_h(t_i)(1-t)}V^{-1}(t_i)\gamma + V(e^{\Lambda_h(t_i)(1-t)} - I)\Lambda_h(t_i)^{-1}V^{-1}b.\end{aligned}\tag{2.87}$$

According to equation (2.46) in the proof of Lemma 2.3, we have

$$x_{hl} = \hat{x}_{hl}(-1) + \delta_{hl}\tag{2.88}$$

where

$$\|\delta_{hl}\| \leq \max_{t \in [0, T]} \|\hat{x}_{hl}^{(n+1)}(t)\| \frac{e^{n+1}}{(2n)^n} \leq \frac{e\|\gamma\|}{m+1}.\tag{2.89}$$

Now for $h \in [m+1]_0$, we take x_{hl} to be the initial condition γ for the next subinterval to obtain $x_{(h+1)l}$. Using (2.87) and (2.88), starting from $\gamma = b_{h0}, b = 0$ for $l = 0$, we find

$$x_{ml} = V(t_l) \left(\prod_{j=1}^{m-h+1} e^{2\Lambda_h(t_j)} \right) V^{-1}(t_l)\gamma + \sum_{k=0}^{m-h} V(t_l) \left(\prod_{j=1}^k e^{2\Lambda_h(t_j)} \right) V^{-1}(t_l)\delta_{(m-k)l}.\tag{2.90}$$

Since $\|\Lambda_h(t_i)\| \leq \|\Lambda\| \leq 1$ and $\Lambda_h(t_i) = \text{diag}(\lambda_0, \dots, \lambda_{d-1})$ with $\text{Re}(\lambda_i) \leq 0$ for $i \in [d]_0$, we have $\|e^{2\Lambda_h(t_i)}\| \leq 1$. Therefore

$$\|x_{hl}\| \leq \|x_{ml}\| \leq \kappa_V(t_l)\|b_{hl}\| + (m-h+1)\kappa_V(t_l)\|\delta_{hl}\| \leq \kappa_V(t_l) + e\|\gamma\| \leq \kappa_V + e\|\gamma\|.\tag{2.91}$$

On the other hand, with $\gamma = 0, b = b_{hl}$ for $l \in [n]$, we have

$$\begin{aligned} x_{ml} &= V(t_l) \left(\prod_{j=1}^{m-h} e^{2\Lambda_h(t_l)} \right) \left((e^{2\Lambda_h(t_l)} - I) \Lambda_h(t_l)^{-1} \right) V^{-1}(t_l) b \\ &\quad + \sum_{k=0}^{m-h} V(t_l) \left(\prod_{j=1}^k e^{2\Lambda_h(t_l)} \right) V^{-1}(t_l) \delta_{(m-k)l}, \end{aligned} \quad (2.92)$$

so

$$\|x_{hl}\| \leq 2\kappa_V(t_l) \|b_{hl}\| + (m-h+1)\kappa_V(t_l) \|\delta_{hl}\| \leq 2\kappa_V(t_l) + e\|\gamma\| \leq 2\kappa_V + e\|\gamma\|. \quad (2.93)$$

For $h \in \{m, m+1, \dots, m+p\}$, according to the definition of L_4 and L_5 , we similarly have

$$\|x_{hl}\| = \|x_{ml}\| \leq 2\kappa_V + e\|\gamma\|. \quad (2.94)$$

According to (2.87), $\hat{x}_{hl}(t)$ is a monotonic function of $t \in [-1, 1]$, which implies

$$\|\hat{x}_{hl}(t)\|^2 \leq \max\{\|\hat{x}_{hl}(-1)\|^2, \|\hat{x}_{hl}(1)\|^2\} \leq (2\kappa_V + e\|\gamma\|)^2. \quad (2.95)$$

Using the identity

$$\int_{-1}^1 \frac{dt}{\sqrt{1-t^2}} = \pi, \quad (2.96)$$

we have

$$\int_{-1}^1 \|\hat{x}_{hl}(t)\|^2 \frac{dt}{\sqrt{1-t^2}} \leq (2\kappa_V + e\|\gamma\|)^2 \int_{-1}^1 \frac{dt}{\sqrt{1-t^2}} = \pi(2\kappa_V + e\|\gamma\|)^2. \quad (2.97)$$

Consider the Chebyshev expansion of $\hat{x}_{hl}(t)$ as in (2.4):

$$\hat{x}_{hl}(t) = \sum_{i=0}^{d-1} \sum_{l=0}^7 c_{i,l}(\Gamma_{h+1}) T_l(\bar{t}). \quad (2.98)$$

By the orthogonality of Chebyshev polynomials (as specified in [54, Appendix A]), we have

$$\begin{aligned} \int_{-1}^1 \|\hat{x}_{hl}(t)\|^2 \frac{dt}{\sqrt{1-t^2}} &= \int_{-1}^1 \left(\sum_{i=0}^{d-1} \sum_{l=0}^7 c_{i,l}(\Gamma_{h+1}) T_l(\bar{t}) \right)^2 \frac{dt}{\sqrt{1-t^2}} \\ &= \sum_{i=0}^{d-1} \sum_{l=1}^7 c_{i,l}^2(\Gamma_{h+1}) + 2 \sum_{i=0}^{d-1} c_{i,0}^2(\Gamma_{h+1}) \geq \sum_{i=0}^{d-1} \sum_{l=1}^n c_{i,l}^2(\Gamma_{h+1}) + 2 \sum_{i=0}^{d-1} c_{i,0}^2(\Gamma_{h+1}). \end{aligned} \quad (2.99)$$

Using (2.97), this gives

$$\sum_{i=0}^{d-1} \sum_{l=0}^n c_{i,l}^2(\Gamma_{h+1}) \leq \int_{-1}^1 \hat{x}_{hl}^2(t) \frac{dt}{\sqrt{1-t^2}} \leq \pi(2\kappa_V + e\|\gamma\|)^2. \quad (2.100)$$

Now we compute $\|X\|$, summing the contributions from all $c_{i,r}(\Gamma_h)$ and x_{mr} , and notice that $c_{i,r} = 0$ and $x_{mr} = 0$ for all $r \neq l$, giving

$$\begin{aligned} \|X\|^2 &= \sum_{h=0}^{m-1} \sum_{i=0}^{d-1} c_{i,l}^2(\Gamma_{h+1}) + (p+1)(x_{ml})^2 \\ &\leq \pi m(2\kappa_V + e\|\gamma\|)^2 + (p+1)(\kappa_V + e\|\gamma\|)^2 \\ &\leq (\pi m + p+1)(2\kappa_V + e\|\gamma\|)^2. \end{aligned} \quad (2.101)$$

Finally, considering all $h \in [m+p+1]_0$ and $l \in [n+1]_0$, from (2.79) we have

$$\|B\|^2 = \sum_{h=0}^{m+p} \sum_{l=0}^n \|b_{hl}\|^2 \leq 1, \quad (2.102)$$

so

$$\begin{aligned}
\|L^{-1}\|^2 &= \sup_{kjBik-1} \|L^{-1}|B\rangle\|^2 = \sup_{kjBik-1} \sum_{h=0}^{m+p} \sum_{l=0}^n \|L^{-1}|b_{hl}\rangle\|^2 \\
&\leq (\pi m + p + 1)(m + p + 1)(n + 1)(2\kappa_V + e\|\gamma\|)^2 \\
&\leq (\pi m + p + 1)^2(n + 1)(2\kappa_V + e\|\gamma\|)^2,
\end{aligned} \tag{2.103}$$

and therefore

$$\|L^{-1}\| \leq (\pi m + p + 1)(n + 1)^{0.5}(2\kappa_V + e\|\gamma\|). \tag{2.104}$$

Finally, combining (2.77) and (2.104) gives

$$\kappa_L = \|L\|\|L^{-1}\| \leq (\pi m + p + 1)(n + 1)^{3.5}(2\kappa_V + e\|\gamma\|) \tag{2.105}$$

as claimed. □

2.6 Success probability

We now evaluate the success probability of our approach to the quantum ODE problem.

Lemma 2.5. *Consider an instance of the quantum ODE problem as defined in [Problem 2.1](#) with the exact solution $\hat{x}(t)$ for $t \in [0, T]$, and its corresponding linear system (2.27) with $m, p \in \mathbb{Z}^+$ and n sufficiently large. When applying the QLSA to this system, the probability of measuring a state proportional to $|x(T)\rangle = \sum_{i=0}^d x_i|i\rangle$ is*

$$P_{\text{measure}} \geq \frac{(p + 1)(n + 1)}{\pi m q^2 + (p + 1)(n + 1)}, \tag{2.106}$$

where x_i is defined in (2.28), τ is defined in (2.24), and

$$q := \max_{t \in [0, T]} \frac{\|\hat{x}(t)\|}{\|x(T)\|}. \quad (2.107)$$

Proof. After solving the linear system (2.27) using the QLSA, we measure the first and third registers of $|X\rangle$ (as defined in (2.28)). We decompose this state as

$$|X\rangle = |X_{\text{bad}}\rangle + |X_{\text{good}}\rangle, \quad (2.108)$$

where

$$|X_{\text{bad}}\rangle = \sum_{h=0}^{m-1} \sum_{i=0}^{d-1} \sum_{l=0}^{n-1} c_{i,l}(\Gamma_{h+1}) |hil\rangle, \quad (2.109)$$

$$|X_{\text{good}}\rangle = \sum_{h=m}^{m+p} \sum_{i=0}^{d-1} \sum_{l=0}^{n-1} x_i |hil\rangle. \quad (2.110)$$

When the first register is observed in some $h \in \{m, m+1, \dots, m+p\}$ (no matter what outcome is seen for the third register), we output the second register, which is then in a normalized state proportional to the final state:

$$|X_{\text{measure}}\rangle = \frac{|x(T)\rangle}{\| |x(T)\rangle \|}, \quad (2.111)$$

with

$$|x(T)\rangle = \sum_{i=0}^{d-1} x_i |i\rangle = \sum_{i=0}^{d-1} \sum_{k=0}^{n-1} c_{i,k} T_k(t) |i\rangle. \quad (2.112)$$

Notice that

$$\|x(T)\|^2 = \sum_{i=0}^{d-1} x_i^2 \quad (2.113)$$

and

$$\|X_{\text{good}}\|^2 = (p+1)(n+1) \sum_{i=0}^{d-1} x_i^2 = (p+1)(n+1) \|x(T)\|^2. \quad (2.114)$$

Considering the definition of q , the contribution from time interval h under the rescaling (2.17), and the identity (2.96), we have

$$\begin{aligned} q^2 \|x(T)\|^2 &= \max_{t \in [0, T]} \|\hat{x}(t)\|^2 = \frac{1}{\pi} \int_{-1}^1 \frac{d\tau}{\sqrt{1-\tau^2}} \max_{t \in [0, T]} \|\hat{x}(t)\|^2 \\ &\geq \frac{1}{\pi} \int_{-1}^1 \frac{d\tau}{\sqrt{1-\tau^2}} \max_{t \in [\Gamma_h, \Gamma_{h+1}]} \|\hat{x}(t)\|^2 \\ &= \frac{1}{\pi} \int_{-1}^1 \frac{d\tau}{\sqrt{1-\tau^2}} \max_{\bar{t} \in [-1, 1]} \|\hat{x}_h(\bar{t})\|^2 \\ &\geq \frac{1}{\pi} \int_{-1}^1 \|\hat{x}_h(\bar{t})\|^2 \frac{d\bar{t}}{\sqrt{1-\bar{t}^2}}, \end{aligned} \quad (2.115)$$

where $\hat{x}_h(\bar{t})$ is the solution of (2.47) with the rescaling in (2.19). By the orthogonality of Chebyshev polynomials (as specified in [54, Appendix A]),

$$\begin{aligned} q^2 \|x(T)\|^2 &\geq \frac{1}{\pi} \int_{-1}^1 \|\hat{x}_h(\bar{t})\|^2 \frac{d\bar{t}}{\sqrt{1-\bar{t}^2}} = \frac{1}{\pi} \int_{-1}^1 \left(\sum_{i=0}^{d-1} \sum_{k=0}^{\lceil \frac{1}{2} \rceil} c_{i,k}(\Gamma_{h+1}) T_k(\bar{t}) \right)^2 \frac{d\bar{t}}{\sqrt{1-\bar{t}^2}} \\ &= \frac{1}{\pi} \left(\sum_{i=0}^{d-1} \sum_{k=1}^{\lceil \frac{1}{2} \rceil} c_{i,k}^2(\Gamma_{h+1}) + 2 \sum_{i=0}^{d-1} c_{i,0}^2(\Gamma_{h+1}) \right) \geq \frac{1}{\pi} \sum_{i=0}^{d-1} \sum_{k=0}^n c_{i,k}^2(\Gamma_{h+1}). \end{aligned} \quad (2.116)$$

For all $h \in [m]_0$, we have

$$mq^2 \|x(T)\|^2 \geq \sum_{h=0}^{m-1} \frac{1}{\pi} \sum_{i=0}^{d-1} \sum_{k=0}^{n-1} c_{i,k}^2(\Gamma_{h+1}) = \frac{1}{\pi} \|X_{\text{bad}}\|^2, \quad (2.117)$$

and therefore

$$\|X_{\text{good}}\|^2 = (p+1)(n+1) \|x(T)\|^2 \geq \frac{(p+1)(n+1)}{\pi mq^2} \|X_{\text{bad}}\|^2. \quad (2.118)$$

Thus we see that the success probability of the measurement satisfies

$$P_{\text{measure}} \geq \frac{(p+1)(n+1)}{\pi mq^2 + (p+1)(n+1)} \quad (2.119)$$

as claimed. □

2.7 State preparation

We now describe a procedure for preparing the vector $|B\rangle$ in the linear system (2.27) (defined in (2.29) and (2.30)) using the given ability to prepare the initial state of the system of differential equations. We also evaluate the complexity of this procedure.

Lemma 2.6. *Consider state preparation oracles acting on a state space with basis vectors $|h\rangle|i\rangle|l\rangle$ for $h \in [m]_0, i \in [d]_0, l \in [n]_0$, where $m, d, n \in \mathbb{N}$, encoding an initial condition $\gamma \in \mathbb{C}^d$ and function $f_h(\cos \frac{l\pi}{n}) \in \mathbb{C}^d$ as in (2.30). Specifically, for any $h \in [m]_0$ and $l \in [n]$, let O_x be a unitary oracle that maps $|0\rangle|0\rangle|0\rangle$ to a state proportional to $|0\rangle|\gamma\rangle|0\rangle$ and $|h\rangle|\phi\rangle|l\rangle$ to $|h\rangle|\phi\rangle|l\rangle$ for any $|\phi\rangle$ orthogonal to $|0\rangle$; let $O_f(h, l)$ be a unitary that maps*

$|h\rangle|0\rangle|l\rangle$ to a state proportional to $|h\rangle|f_h(\cos \frac{l\pi}{n})\rangle|l\rangle$ and maps $|0\rangle|\phi\rangle|0\rangle$ to $|0\rangle|\phi\rangle|0\rangle$ for any $|\phi\rangle$ orthogonal to $|0\rangle$. Suppose $\|\gamma\|$ and $\|f_h(\cos \frac{l\pi}{n})\|$ are known. Then the normalized quantum state

$$|B\rangle \propto |0\rangle|\gamma\rangle|0\rangle + \sum_{h=0}^{m-1} \sum_{l=1}^n |h\rangle|f_h(\cos \frac{l\pi}{n})\rangle|l\rangle \quad (2.120)$$

can be prepared with gate and query complexity $O(mn)$.

Proof. We normalize the components of the state using the coefficients

$$\begin{aligned} b_{00} &= \frac{\|\gamma\|}{\sqrt{\|\gamma\|^2 + \sum_{l=1}^n \|f_h(\cos \frac{l\pi}{n})\|^2}}, \\ b_{hl} &= \frac{\|f_h(\cos \frac{l\pi}{n})\|}{\sqrt{\|\gamma\|^2 + \sum_{l=1}^n \|f_h(\cos \frac{l\pi}{n})\|^2}}, \quad h \in [m]_0, l \in [n] \end{aligned} \quad (2.121)$$

so that

$$\sum_{h=0}^{m-1} \sum_{l=0}^n b_{hl}^2 = 1. \quad (2.122)$$

First we perform a unitary transformation mapping

$$|0\rangle|0\rangle|0\rangle \mapsto b_{00}|0\rangle|0\rangle|0\rangle + b_{01}|0\rangle|0\rangle|1\rangle + \cdots + b_{(m-1)n}|m-1\rangle|0\rangle|n\rangle. \quad (2.123)$$

This can be done in time complexity $O(mn)$ by standard techniques [92]. Then we perform O_x and $O_f(h, l)$ for all $h \in [m]_0, l \in [n]$, giving

$$|0\rangle|\gamma\rangle|0\rangle + \sum_{h=0}^{m-1} \sum_{l=1}^n |h\rangle|f_h(\cos \frac{l\pi}{n})\rangle|l\rangle \quad (2.124)$$

using $O(mn)$ queries. □

2.8 Main result

Having analyzed the solution error, condition number, success probability, and state preparation procedure for our approach, we are now ready to establish the main result.

Theorem 2.1. *Consider an instance of the quantum ODE problem as defined in [Problem 2.1](#).*

Assume $A(t)$ can be diagonalized as the form $A(t) = V(t)\Lambda(t)V^{-1}(t)$ where $\Lambda(t) = \text{diag}(\lambda_1(t), \dots, \lambda_d(t))$ with $\text{Re}(\lambda_i(t)) \leq 0$ for each $i \in [d]_0$ and $t \in [0, T]$. Then there exists a quantum algorithm that produces a state $x(T)/\|x(T)\|$ ϵ -close to $\hat{x}(T)/\|\hat{x}(T)\|$ in l^2 norm, succeeding with probability $\Omega(1)$, with a flag indicating success, using

$$O(\kappa_V s \|A\| T q \text{poly}(\log(\kappa_V s \|A\| g^\theta T / \epsilon g))) \quad (2.125)$$

queries to oracles $O_A(h, l)$ (a sparse matrix oracle for $A_h(t_l)$ as defined in [\(2.18\)](#)) and O_x and $O_f(h, l)$ (as defined in [Lemma 2.6](#)). Here $\|A\| := \max_{t \in [0, T]} \|A(t)\|$; $\kappa_V := \max_t \kappa_V(t)$, where $\kappa_V(t)$ is the condition number of $V(t)$; and

$$g := \|\hat{x}(T)\|, \quad g^\theta := \max_{t \in [0, T]} \max_{n \in \mathbb{N}} \|\hat{x}^{(n+1)}(t)\|, \quad q := \max_{t \in [0, T]} \frac{\|\hat{x}(t)\|}{\|x(T)\|}. \quad (2.126)$$

The gate complexity is larger than the query complexity by a factor of $\text{poly}(\log(\kappa_V ds \|A\| g^\theta T / \epsilon))$.

Proof. We first present the algorithm and then analyze its complexity.

Statement of the algorithm. First, we choose m to guarantee

$$\frac{\|A\| T}{2m} \leq 1. \quad (2.127)$$

Then, as in [Section 2.3](#), we divide the interval $[0, T]$ into small subintervals

$[0, \Gamma_1], [\Gamma_1, \Gamma_2], \dots, [\Gamma_{m-1}, T]$ with $\Gamma_0 = 0, \Gamma_m = T$, and define

$$\tau := \max_{h=0}^{m-1} \{\tau_h\}, \quad \tau_h := |\Gamma_{h+1} - \Gamma_h| = \frac{T}{m}. \quad (2.128)$$

Each subinterval $[\Gamma_h, \Gamma_{h+1}]$ for $h \in [m-1]$ is mapped onto $[-1, 1]$ with a linear mapping K_h satisfying $K_h(\Gamma_h) = 1, K_h(\Gamma_{h+1}) = -1$:

$$K_h: t \mapsto \bar{t} = 1 - \frac{2(t - \Gamma_h)}{\Gamma_{h+1} - \Gamma_h}. \quad (2.129)$$

We choose

$$n = \frac{e}{2} \max \left\{ \left\lfloor \frac{\log(\Omega)}{\log(\log(\Omega))} \right\rfloor, \left\lfloor \frac{\log(\omega)}{\log(\log(\omega))} \right\rfloor \right\} \quad (2.130)$$

where

$$\Omega := \frac{g^\ell e m}{\delta} = \frac{g^\ell e m (1 + \epsilon)}{g \epsilon} \quad (2.131)$$

and

$$\omega := \frac{g^\ell}{\|\gamma\|} (m + 1). \quad (2.132)$$

Since $\max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \leq g^\ell$, by [Lemma 2.3](#), this choice guarantees

$$\|\hat{x}(T) - x(T)\| \leq m \max_{t \in [0, T]} \|\hat{x}^{(n+1)}(t)\| \frac{e^{n+1}}{(2n)^n} \leq \delta \quad (2.133)$$

and

$$\max_{t \in [0, T]} \frac{\|\hat{x}^{(n+1)}(t)\|}{\|\gamma\|} \left(\frac{e}{2n}\right)^n \leq \frac{1}{m+1}. \quad (2.134)$$

Now $\|\hat{x}(T) - x(T)\| \leq \delta$ implies

$$\left\| \frac{\hat{x}(T)}{\|\hat{x}(T)\|} - \frac{x(T)}{\|x(T)\|} \right\| \leq \frac{\delta}{\min\{\|\hat{x}(T)\|, \|x(T)\|\}} \leq \frac{\delta}{g - \delta} =: \epsilon, \quad (2.135)$$

so we can choose such n to ensure that the normalized output state is ϵ -close to $\hat{x}(T)/\|\hat{x}(T)\|$.

Following [Section 2.3](#), we build the linear system $L|X\rangle = |B\rangle$ (see [\(2.27\)](#)) that encodes the quantum spectral method. By [Lemma 2.4](#), the condition number of this linear system is at most $(\pi m + p + 1)(n + 1)^{3.5}(2\kappa_V + e\|\gamma\|_1)$. Then we use the QLSA from reference [\[46\]](#) to obtain a normalized state $|X\rangle$ and measure the first and third register of $|X\rangle$ in the standard basis. If the measurement outcome for the first register belongs to

$$S = \{m, m + 1, \dots, m + p\}, \quad (2.136)$$

we output the state of the second register, which is a normalized state $|x(T)\rangle/\|x(T)\|$ satisfying [\(2.135\)](#). By [Lemma 2.5](#), the probability of this event happening is at least $\frac{(p+1)(n+1)}{\pi m q^2 + (p+1)(n+1)}$. To ensure $m + p = O(\|A\|T)$, we can choose

$$p = O(m) = O(\|A\|T), \quad (2.137)$$

so we can achieve success probability $\Omega(1)$ with $O(q/\sqrt{n})$ repetitions of the above procedure.

Analysis of the complexity. The matrix L is an $(m + p + 1)d(n + 1) \times (m + p + 1)d(n + 1)$ matrix with $O(ns)$ nonzero entries in any row or column. By [Lemma 2.4](#) and our choice of parameters, the condition number of L is $O(\kappa_V(m + p)n^{3.5})$. Consequently, by Theorem

5 of [46], the QLSA produces the state $|x(T)\rangle$ with

$$O(\kappa_V(m+p)n^{4.5}s \text{ poly}(\log(\kappa_V m n s / \delta))) = O(\kappa_V s \|A\| T \text{ poly}(\log(\kappa_V s \|A\| g^\theta T / \epsilon g))) \quad (2.138)$$

queries to the oracles $O_A(h, l)$, O_x , and $O_f(h, l)$, and its gate complexity is larger by a factor of $\text{poly}(\log(\kappa_V m n s / \delta))$. Using $O(q/\sqrt{n})$ steps of amplitude amplification to achieve success probability $\Omega(1)$, the overall query complexity of our algorithm is

$$O(\kappa_V(m+p)n^4 s q \text{ poly}(\log(\kappa_V m n s / \delta))) = O(\kappa_V s \|A\| T q \text{ poly}(\log(\kappa_V s \|A\| g^\theta T / \epsilon g))), \quad (2.139)$$

and the gate complexity is larger by a factor of

$$\text{poly}(\log(\kappa_V d s \|A\| g^\theta T / \epsilon g)) \quad (2.140)$$

as claimed. □

In general, g^θ could be unbounded above as $n \rightarrow \infty$. However, we could obtain a useful bound in such a case by solving the implicit equations (2.133) and (2.134).

Note that for time-independent differential equations, we can replace g^θ by $\|\gamma\| + 2\tau\|f\|$ as shown in (2.62). In place of (2.131) and (2.132), we choose

$$\Omega := \frac{(\|\gamma\| + 2\tau\|f\|)em\kappa_V}{\delta} = \frac{(\|\gamma\| + 2\tau\|f\|)em\kappa_V(1 + \epsilon)}{g\epsilon} \quad (2.141)$$

and

$$\omega := \frac{\|\gamma\| + 2\tau\|f\|}{\|\gamma\|} (m+1)\kappa_V. \quad (2.142)$$

By [Lemma 2.3](#), this choice guarantees

$$\|\hat{x}(T) - x(T)\| \leq \max_{t \in [0,1]} \|\hat{x}(t) - x(t)\| \leq m\kappa_V (\|\gamma\| + 2\tau\|f\|) \frac{e^{n+1}}{(2n)^n} \leq \delta \quad (2.143)$$

and

$$\max_{t \in [0,T]} \frac{\|\hat{x}^{(n+1)}(t)\|}{\|\gamma\|} \left(\frac{e}{2n}\right)^n \leq \frac{\kappa_V (\|\gamma\| + 2\tau\|f\|)}{\|\gamma\|} \left(\frac{e}{2n}\right)^n \leq \frac{1}{m+1}. \quad (2.144)$$

Thus we have the following:

Corollary 2.1. *For time-independent differential equations, under the same assumptions of [Theorem 2.1](#), there exists a quantum algorithm using*

$$O(\kappa_V s \|A\| T q \text{ poly}(\log(\kappa_V s \gamma \|A\| \|f\| T / \epsilon g))) \quad (2.145)$$

queries to $O_A(h, l)$, O_x , and $O_f(h, l)$. The gate complexity of this algorithm is larger than its query complexity by a factor of $\text{poly}(\log(\kappa_V d s \gamma \|A\| \|f\| T / \epsilon))$.

The complexity of our algorithm depends on the parameter q in defined in [\(2.126\)](#), which characterizes the decay of the final state relative to the initial state. As discussed in Section 8 of [\[53\]](#), it is unlikely that the dependence on q can be significantly improved, since renormalization of the state effectively implements postselection and an efficient procedure for performing this would have the unlikely consequence $\text{BQP} = \text{PP}$.

We also require the real parts of the eigenvalues of $A(t)$ to be non-positive for all $t \in [0, T]$ so that the solution cannot grow exponentially. This requirement is essentially the same as in the time-independent case considered in [53] and improves upon the analogous condition in [52] (which requires an additional stability condition). Also as in [53], our algorithm can produce approximate solutions for non-diagonalizable $A(t)$, although the dependence on ϵ degrades to $\text{poly}(1/\epsilon)$. For further discussion of these considerations, see Sections 1 and 8 of [53].

2.9 Boundary value problems

So far we have focused on initial value problems (IVPs). Boundary value problems (BVPs) are another widely studied class of differential equations that appear in many applications, but that can be harder to solve than IVPs.

Consider a sparse, linear, time-dependent system of differential equations as in [Problem 2.1](#) but with a constraint on some linear combination of the initial and final states:

Problem 2.2. *In the quantum BVP, we are given a system of equations*

$$\frac{dx(t)}{dt} = A(t)x(t) + f(t), \quad (2.146)$$

where $x(t) \in \mathbb{C}^d$, $A(t) \in \mathbb{C}^{d \times d}$ is s -sparse, and $f(t) \in \mathbb{C}^d$ for all $t \in [0, T]$, and a boundary condition $\alpha x(0) + \beta x(T) = \gamma$ with $\alpha, \beta, \gamma \in \mathbb{C}^d$. Suppose there exists a unique solution $\hat{x} \in C^1(0, T)$ of this boundary value problem. Given oracles that compute the

locations and values of nonzero entries of $A(t)$ for any t , and that prepare quantum states $\alpha|x(0)\rangle + \beta|x(T)\rangle = |\gamma\rangle$ and $|f(t)\rangle$ for any t , the goal is to output a quantum state $|x(t)\rangle$ that is proportional to $x(t)$ for some specified $t \in [0, T]$.

As before, we can rescale $[0, T]$ onto $[-1, 1]$ by a linear mapping. However, since we have boundary conditions at $t = 0$ and $t = T$, we cannot divide $[0, T]$ into small subintervals. Instead, we directly map $[0, T]$ onto $[-1, 1]$ with a linear map K satisfying $K(0) = 1$ and $K(T) = -1$:

$$K: t \mapsto \bar{t} = 1 - \frac{2t}{T}. \quad (2.147)$$

Now the new differential equations are

$$\frac{dx}{d\bar{t}} = -\frac{T}{2}(A(t)x + f(t)). \quad (2.148)$$

If we define $A_K(\bar{t}) := -\frac{T}{2}A(t)$ and $f_K(\bar{t}) = -\frac{T}{2}f(t)$, we have

$$\frac{dx}{d\bar{t}} = A_K(\bar{t})x(\bar{t}) + f_K(\bar{t}) \quad (2.149)$$

for $\bar{t} \in [-1, 1]$. Now the boundary condition takes the form

$$\alpha x(1) + \beta x(-1) = \gamma. \quad (2.150)$$

Since we only have one solution interval, we need to choose a larger order n of the

Chebyshev series to reduce the solution error. In particular, we take

$$n = \frac{e}{2} \|A\| T \max \left\{ \left\lfloor \frac{\log(\Omega)}{\log(\log(\Omega))} \right\rfloor, \left\lfloor \frac{\log(\omega)}{\log(\log(\omega))} \right\rfloor \right\} \quad (2.151)$$

where Ω and ω are the same as [Theorem 2.1](#).

As in [Section 2.3](#), we approximate $x(t)$ by a finite Chebyshev series with interpolating nodes $\{t_l = \cos \frac{l\pi}{n} : l \in [n]\}$ and thereby obtain a linear system

$$\frac{dx(t_l)}{dt} = A_K(t_l)x(t_l) + f(t_l), \quad l \in [n] \quad (2.152)$$

with the boundary condition

$$\alpha x(t_0) + \beta x(t_n) = \gamma. \quad (2.153)$$

Observe that the linear equations have the same form as in [\(2.25\)](#). Instead of [\(2.26\)](#), the term with $l = 0$ encodes the condition [\(2.153\)](#) expanded in a Chebyshev series, namely

$$\alpha_i \sum_{k=0}^n c_{i,k} T_k(t_0) + \beta_i \sum_{k=0}^n c_{i,k} T_k(t_n) = \gamma_i \quad (2.154)$$

for each $i \in [d]_0$. Since $T_k(t_0) = 1$ and $T_k(t_n) = (-1)^k$, this can be simplified as

$$\sum_{k=0}^n (\alpha_i + (-1)^k \beta_i) c_{i,k} = \gamma_i. \quad (2.155)$$

If $\alpha_i + (-1)^k \beta_i = 0$, the element of $|il\rangle \langle ik|$ of $L_2(A_K)$ is zero; if $\alpha_i + (-1)^k \beta_i \neq 0$, without loss of generality, the two sides of this equality can be divided by $\alpha_i + (-1)^k \beta_i$

to guarantee that the terms with $l = 0$ can be encoded as in (2.26).

Now this system can be written in the form of equation (2.27) with $m = 1$. Here L , $|X\rangle$, and $|B\rangle$ are the same as in (2.31), (2.28), and (2.29), respectively, with $m = 1$, except for adjustments to L_3 that we now describe.

The matrix L_3 represents the linear combination $x_i(t) = \sum_{k=0}^n c_{i,k} T_k(t)$. Thus we take

$$L_3 = \sum_{i=0}^{d-1} \sum_{k=0}^n T_k(t) |i0\rangle \langle ik|. \quad (2.156)$$

Since $|T_k(t)| \leq 1$, we have

$$\|L_3\| \leq n + 1, \quad (2.157)$$

and it follows that Lemma 2.3 also holds for boundary value problems. Similarly, Lemma 2.4 still holds with $m = 1$.

We are now ready to analyze the complexity of the quantum BVP algorithm. The matrix L defined above is a $(p+2)d(n+1) \times (p+2)d(n+1)$ matrix with $O(ns)$ nonzero entries in any row or column, with condition number $O(\kappa_V pn^{3.5})$. By Lemma 2.5 with $p = O(1)$, $O(q/\sqrt{n})$ repetitions suffice to ensure success probability $\Omega(1)$. By (2.151), n is linear in $\|A\|T$ and poly-logarithmic in Ω and ω . Therefore, we have the following:

Theorem 2.2. *Consider an instance of the quantum BVP as defined in Problem 2.2. Assume $A(t)$ can be diagonalized as the form $A(t) = V(t)\Lambda(t)V^{-1}(t)$ where $\Lambda(t) = \text{diag}(\lambda_1(t), \dots, \lambda_d(t))$ with $\text{Re}(\lambda_i(t)) \leq 0$ for each $i \in [d]_0$ and $t \in [0, T]$. Then there exists a quantum algorithm that produces a state $x(t) / \|x(t)\|$ ϵ -close to $\hat{x}(t) / \|\hat{x}(t)\|$*

in l^2 norm, succeeding with probability $\Omega(1)$, with a flag indicating success, using

$$O(\kappa_V s \|A\|^4 T^4 q \text{ poly}(\log(\kappa_V s \|A\| g^0 T / \epsilon g))) \quad (2.158)$$

queries to $O_A(h, l)$, O_x , and $O_f(h, l)$. Here $\|A\|$, κ_V , g , g^0 and q are defined as in [Theorem 2.1](#). The gate complexity is larger than the query complexity by a factor of $\text{poly}(\log(\kappa_V ds \|A\| g^0 T / \epsilon))$.

As for initial value problems, we can simplify this result in the time-independent case.

Corollary 2.2. *For a time-independent boundary value problem, under the same assumptions of [Theorem 2.2](#), there exists a quantum algorithm using*

$$O(\kappa_V s \|A\|^4 T^4 q \text{ poly}(\log(\kappa_V s \gamma \|A\| \|f\| T / \epsilon g))) \quad (2.159)$$

queries to $O_A(h, l)$, O_x , and $O_f(h, l)$. The gate complexity of this algorithm is larger than its query complexity by a factor of $\text{poly}(\log(\kappa_V ds \gamma \|A\| \|f\| T / \epsilon))$.

2.10 Discussion

In this chapter, we presented a quantum algorithm to solve linear, time-dependent ordinary differential equations. Specifically, we showed how to employ a global approximation based on the spectral method as an alternative to the more straightforward finite difference method. Our algorithm handles time-independent differential equations with almost the same complexity as [\[53\]](#), but unlike that approach, can also handle time-dependent differential

equations. Compared to [52], our algorithm improves the complexity of solving time-dependent linear differential equations from $\text{poly}(1/\epsilon)$ to $\text{poly}(\log(1/\epsilon))$.

This work raises several natural open problems. First, our algorithm must assume that the solution is smooth. If the solution is in C^r , the solution error is $O(\frac{1}{n^{r-2}})$ by [Lemma 2.1](#). Can we improve the complexity to $\text{poly}(\log(1/\epsilon))$ under such weaker smoothness assumptions?

Second, the complexity of our algorithm is logarithmic in the parameter g^θ defined in (2.126), which characterizes the amount of fluctuation in the solution. However, the query complexity of Hamiltonian simulation is independent of that parameter [34, 35]. Can we develop quantum algorithms for general differential equations with query complexity independent of g^θ ?

Third, our algorithm has nearly optimal dependence on T , scaling as $O(T \text{poly}(\log T))$. According to the no-fast-forwarding theorem [33], the complexity must be at least linear in T , and indeed linear complexity is achievable for the case of Hamiltonian simulation [93]. Can we handle general differential equations with complexity linear in T ? Furthermore, can we achieve an optimal tradeoff between T and ϵ as shown for Hamiltonian simulation in [42]?

Chapter 3: High-precision quantum algorithms for linear elliptic partial differential equations

3.1 Introduction

In this chapter, we study high-precision quantum algorithms for linear elliptic partial differential equations (PDEs)¹. As earlier introduced in [Problem 1.4](#), the problem we address can be stated as follows: Given a linear PDE with boundary conditions and an error parameter ϵ , output a quantum state that is ϵ -close to one whose amplitudes are proportional to the solution of the PDE at a set of grid points in the domain of the PDE. We focus on elliptic PDEs, and we assume a technical condition that we call *global strict diagonal dominance* (defined in [\(3.8\)](#)).

Our first algorithm is based on a quantum version of the FDM approach: we use a finite-difference approximation to produce a system of linear equations and then solve that system using the QLSA. We analyze our FDM algorithm as applied to Poisson's equation (which automatically satisfies global strict diagonal dominance) under periodic, Dirichlet, and Neumann boundary conditions. Whereas previous FDM approaches [\[69, 70\]](#) considered fixed orders of truncation, we adapt the order of truncation depending on ϵ , inspired by the classical adaptive FDM [\[71\]](#). As the order increases, the eigenvalues

¹This chapter is based on the paper [\[59\]](#).

of the FDM matrix approach the eigenvalues of the continuous Laplacian, allowing for more precise approximations. The main algorithm we present uses the quantum Fourier transform (QFT) and takes advantage of the high-precision LCU-based QLSA [46]. We first consider periodic boundary conditions, but by restricting to appropriate subspaces, this approach can also be applied to homogeneous Dirichlet and Neumann boundary conditions. We state our result in [Theorem 3.1](#), which (informally) says that this quantum adaptive FDM approach produces a quantum state approximating the solution of Poisson’s equation with complexity $d^{6.5} \text{poly}(\log d, \log(1/\epsilon))$.

We also propose a quantum algorithm for more general second-order elliptic PDEs under periodic or non-periodic Dirichlet boundary conditions. This algorithm is based on quantum spectral methods [54]. The spectral method globally approximates the solution of a PDE by a truncated Fourier or Chebyshev series (which converges exponentially for smooth functions) with undetermined coefficients, and then finds the coefficients by solving a linear system. This system is exponentially large in d , so solving it is infeasible for classical algorithms but feasible in a quantum context. To be able to apply the QLSA efficiently, we show how to make the system sparse using variants of the quantum Fourier transform. Our bound on the condition number of the linear system uses global strict diagonal dominance, and introduces a factor in the complexity that measures the extent to which this condition holds. We state our result in [Theorem 3.2](#), which (informally) gives a complexity of $d^2 \text{poly}(\log(1/\epsilon))$ for producing a quantum state approximating the solution of general second-order elliptic PDEs with Dirichlet boundary conditions.

Both of these approaches have complexity $\text{poly}(d, \log(1/\epsilon))$, providing optimal dependence on ϵ and an exponential improvement over classical methods as a function of

the spatial dimension d . Bounding the complexities of these algorithms requires analyzing how d and ϵ affect the condition numbers of the relevant linear systems (finite difference matrices and matrices relating the spectral coefficients) and accounting for errors in the approximate solution provided by the QLSA. Furthermore, the complexities of both approaches scale logarithmically with high-order derivatives of the solution and the inhomogeneity. The detailed complexity dependence is presented in [Theorem 3.1](#) and [Theorem 3.2](#), and is further discussed in [Section 3.5](#).

[Table 3.1](#) compares the performance of our approaches to other classical and quantum algorithms for PDEs. Compared to classical algorithms, quantum algorithms improve the dependence on spatial dimension from exponential to polynomial (with the significant caveat that they produce a different representation of the solution). Compared to previous quantum FDM/FEM/FVM algorithms [[57](#), [69](#), [70](#), [94](#)], the quantum adaptive FDM and quantum spectral method improve the error dependence from $\text{poly}(1/\epsilon)$ to $\text{poly}(\log(1/\epsilon))$. Our approaches achieve the best known dependence on the parameter ϵ for the Poisson equation with homogeneous boundary conditions. Furthermore, our quantum spectral method approach not only achieves the best known dependence on d and ϵ for elliptic PDEs with inhomogeneous Dirichlet boundary conditions, but also improves the dependence on d for the Poisson equation with inhomogeneous Dirichlet boundary conditions, as compared to previous quantum algorithms.

The remainder of the chapter is structured as follows. [Section 3.2](#) introduces technical details about linear PDEs and formally states the problem we solve. [Section 3.3](#) covers our FDM algorithm for Poisson's equation. [Section 3.4](#) details the spectral algorithm for elliptic PDEs. Finally, [Section 3.5](#) concludes with a brief discussion of the results, their

	Algorithm	Equation	Boundary conditions	Complexity
Classical	FDM/FEM/FVM	general	general	$\text{poly}((1/\epsilon)^d)$
	Adaptive FDM/FEM [71]	general	general	$\text{poly}((\log(1/\epsilon))^d)$
	Spectral method [67, 68]	general	general	$\text{poly}((\log(1/\epsilon))^d)$
	Sparse grid FDM/FEM [95, 96]	general	general	$\text{poly}((1/\epsilon)(\log(1/\epsilon))^d)$
	Sparse grid spectral method [97, 98]	elliptic	general	$\text{poly}(\log(1/\epsilon)(\log \log(1/\epsilon))^d)$
Quantum	FEM [57]	Poisson	homogeneous	$\text{poly}(d, 1/\epsilon)$
	FDM [69]	Poisson	homogeneous Dirichlet	d $\text{poly}(\log d, 1/\epsilon)$
	FDM [70]	wave	homogeneous	$d^{5/2}$ $\text{poly}(1/\epsilon)$
	FVM [94]	hyperbolic	periodic	d $\text{poly}(1/\epsilon)$
	Adaptive FDM [59]	Poisson	periodic, homogeneous	$d^{13/2} \text{poly}(\log d, \log(1/\epsilon))$
	Spectral method [59]	Poisson	homogeneous Dirichlet	d $\text{poly}(\log d, \log(1/\epsilon))$
	Spectral method [59]	elliptic	inhomogeneous Dirichlet	d^2 $\text{poly}(\log(1/\epsilon))$

Table 3.1: Summary of the time complexities of classical and quantum algorithms for d -dimensional PDEs with error tolerance ϵ . Portions of the complexity in bold represent best known dependence on that parameter.

possible applications, and some open problems.

3.2 Linear PDEs

In this chapter, we focus on systems of linear PDEs. Such equations can be written in the form

$$L(u(\mathbf{x})) = f(\mathbf{x}), \quad (3.1)$$

where the variable $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{C}^d$ is a d -dimensional vector, the solution $u(\mathbf{x}) \in \mathbb{C}$ and the inhomogeneity $f(\mathbf{x}) \in \mathbb{C}$ are scalar functions, and L is a linear differential operator acting on $u(\mathbf{x})$. In general, L can be written in a linear combination of $u(\mathbf{x})$ and its derivatives. A linear differential operator L of order h has the form

$$L(u(\mathbf{x})) = \sum_{k_j k_1 \leq h} A_j(\mathbf{x}) \frac{\partial^j}{\partial \mathbf{x}^j} u(\mathbf{x}), \quad (3.2)$$

where $\mathbf{j} = (j_1, \dots, j_d)$ is a d -dimensional non-negative vector with $\|\mathbf{j}\|_1 = j_1 + \dots + j_d \leq h$, $A_j(\mathbf{x}) \in \mathbb{C}$, and

$$\frac{\partial^{\mathbf{j}}}{\partial \mathbf{x}^{\mathbf{j}}} u(\mathbf{x}) = \frac{\partial^{j_1}}{\partial x_1^{j_1}} \cdots \frac{\partial^{j_d}}{\partial x_d^{j_d}} u(\mathbf{x}). \quad (3.3)$$

The problem reduces to a system of linear ordinary differential equations (ODEs) when $d = 1$. For $d \geq 2$, we call (3.1) a (multi-dimensional) PDE.

For example, systems of first-order linear PDEs can be written in the form

$$\sum_{j=1}^d A_j(\mathbf{x}) \frac{\partial u(\mathbf{x})}{\partial x_j} + A_0(\mathbf{x}) u(\mathbf{x}) = f(\mathbf{x}), \quad (3.4)$$

where $A_j(\mathbf{x}), A_0(\mathbf{x}), f(\mathbf{x}) \in \mathbb{C}$ for $j \in [d] := \{1, \dots, d\}$. Similarly, systems of second-order linear PDEs can be expressed in the form

$$\sum_{j_1, j_2=1}^d A_{j_1 j_2}(\mathbf{x}) \frac{\partial^2 u(\mathbf{x})}{\partial x_{j_1} \partial x_{j_2}} + \sum_{j=1}^d A_j(\mathbf{x}) \frac{\partial u(\mathbf{x})}{\partial x_j} + A_0(\mathbf{x}) u(\mathbf{x}) = f(\mathbf{x}), \quad (3.5)$$

where $A_{j_1, j_2}(\mathbf{x}), A_j(\mathbf{x}), A_0(\mathbf{x}), f(\mathbf{x}) \in \mathbb{C}$ for $j_1, j_2, j \in [d]$. A well-known second-order linear PDEs is the *Poisson equation*

$$\Delta u(\mathbf{x}) := \sum_{j=1}^d \frac{\partial^2}{\partial x_j^2} u(\mathbf{x}) = f(\mathbf{x}). \quad (3.6)$$

A linear PDE of order h is called *elliptic* if its differential operator (3.2) satisfies

$$\sum_{k \mathbf{j} k_1 = h} A_{\mathbf{j}}(\mathbf{x}) \xi^{\mathbf{j}} \neq 0, \quad (3.7)$$

for all nonzero $\boldsymbol{\xi}^j = \xi_1^{j_1} \dots \xi_d^{j_d}$ with $\xi_1, \dots, \xi_d \in \mathbb{R}^m$ and all \boldsymbol{x} . Note that ellipticity only depends on the highest-order terms. When $h = 2$, the linear PDE (3.5) is called a second-order elliptic PDE if and only if $A_{j_1 j_2}(\boldsymbol{x})$ is positive-definite or negative-definite for any \boldsymbol{x} . In particular, the Poisson equation (3.6) is a second-order elliptic PDE.

We consider a class of elliptic PDEs that also satisfy the condition

$$C := 1 - \sum_{j_1=1}^d \frac{1}{|A_{j_1, j_1}(\boldsymbol{x})|} \sum_{j_2 \in [d]^{n f_{j_1} g}} |A_{j_1, j_2}(\boldsymbol{x})| > 0 \quad (3.8)$$

for all \boldsymbol{x} . We call this condition *global strict diagonal dominance*, since it is a strengthening of the standard (strict) diagonal dominance condition

$$d - \sum_{j_1=1}^d \frac{1}{|A_{j_1, j_1}(\boldsymbol{x})|} \sum_{j_2 \in [d]^{n f_{j_1} g}} |A_{j_1, j_2}(\boldsymbol{x})| > 0. \quad (3.9)$$

Observe that (3.8) holds for the Poisson equation (3.6) with $C = 1$.

In this chapter, we focus on the following boundary value problem. This is a formal statement of [Problem 1.4](#).

Problem 3.1. *In the quantum PDE problem, we are given a system of second-order elliptic equations*

$$L(u(\boldsymbol{x})) = \sum_{k^j k_1=2} A_j \frac{\partial^j}{\partial \boldsymbol{x}^j} u(\boldsymbol{x}) = \sum_{j_1, j_2=1}^d A_{j_1 j_2} \frac{\partial^2 u(\boldsymbol{x})}{\partial x_{j_1} \partial x_{j_2}} = f(\boldsymbol{x}) \quad (3.10)$$

satisfying the global strict diagonal dominance condition (3.8), where the variable $\boldsymbol{x} = (x_1, \dots, x_d) \in D = [-1, 1]^d$ is a d -dimensional vector, the inhomogeneity $f(\boldsymbol{x}) \in \mathbb{C}$ is a

scalar function of \mathbf{x} satisfying $f(\mathbf{x}) \in C^1$, and the linear coefficients $A_j \in \mathbb{C}$. We are also given boundary conditions $u(\mathbf{x}) = \gamma(\mathbf{x}) \in \partial D$ or $\frac{\partial u(\mathbf{x})}{\partial x_j} \Big|_{x_j=1} = \gamma(\mathbf{x}) \Big|_{x_j=1} \in \partial D$ where $\gamma(\mathbf{x}) \in C^1$. We assume there exists a weak solution $\hat{u}(\mathbf{x}) \in \mathbb{C}$ for the boundary value problem (see Reference [99, Section 6.1.2]). Given sparse matrix oracles to provide the locations and values of nonzero entries of a matrix $A_{j_1 j_2}(\mathbf{x})$, $A_j(\mathbf{x})$, and $A_0(\mathbf{x})$ on a set of interpolation nodes \mathbf{x}^2 , and that prepare normalized states $|\gamma(\mathbf{x})\rangle$ and $|f(\mathbf{x})\rangle$ whose amplitudes are proportional to $\gamma(\mathbf{x})$ and $f(\mathbf{x})$ on a set of interpolation nodes \mathbf{x} , the goal is to output a quantum state $|u(\mathbf{x})\rangle$ that is ϵ -close to the normalized $u(\mathbf{x})$ on a set of interpolation nodes \mathbf{x} in ℓ_2 norm.

3.3 Finite difference method

We now describe our first approach to quantum algorithms for linear PDEs, based on the finite difference method (FDM). Using this approach, we show the following.

Theorem 3.1. *There exists a quantum algorithm that outputs a state ϵ -close to $|u\rangle$ that runs in time*

$$\tilde{O}\left(d^{6.5} \log^{4.5}\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right) \sqrt{\log\left[d^4 \log^3\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right)/\epsilon}\right)} \quad (3.11)$$

and makes

$$\tilde{O}\left(d^4 \log^3\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right) \sqrt{\log\left[d^4 \log^3\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right)/\epsilon}\right)} \quad (3.12)$$

²For instance, $A_{j_1 j_2}(\mathbf{x})$ is modeled by a sparse matrix oracle that, on input (m, l) , gives the location of the l -th nonzero entry in row m , denoted as n , and gives the value $A_{j_1 j_2}(\mathbf{x})_{m,n}$.

queries to the oracle for f .

To show this, we first construct a linear system corresponding to the finite difference approximation of Poisson's equation with periodic boundary conditions and bound the error of this high-order FDM in [Section 3.3.1 \(Lemma 3.1\)](#). Then we bound the condition number of this system in [Section 3.3.2 \(Lemma 3.2 and Lemma 3.3\)](#) and bound the error of approximation in [Section 3.3.3 \(Lemma 3.4\)](#). We use these results to give an efficient quantum algorithm in [Section 3.3.4](#), establishing [Theorem 3.1](#). We conclude by discussing how to use the method of images to apply this algorithm for Neumann and Dirichlet boundary conditions in [Section 3.3.5](#).

The FDM approximates the derivative of a function f at a point \mathbf{x} in terms of the values of f on a finite set of points near \mathbf{x} . Generally there are no restrictions on where these points are located relative to \mathbf{x} , but they are typically taken to be uniformly spaced points with respect to a certain coordinate. This corresponds to discretizing $[-1, 1]^d$ (or $[0, 2\pi)^d$) to a d -dimensional rectangular lattice (where we use periodic boundary conditions).

For a scalar field, in which $u(\mathbf{x}) \in \mathbb{C}$, the canonical elliptic PDE is Poisson's equation (3.6), which we consider solving on $[0, 2\pi)^d$ with periodic boundary conditions. This also implies results for the domain $\Omega = [-1, 1]^d$ under Dirichlet ($u(\partial\Omega) = 0$) and Neumann ($\hat{n} \cdot \nabla u(\partial\Omega) = 0$ where \hat{n} denotes the normal direction to $\partial\Omega$, which for domain $\Omega = [-1, 1]^d$ is equivalent to $\frac{\partial u}{\partial x_j} \Big|_{x_j = -1} = 0$ for $j \in [d]$) boundary conditions.

3.3.1 Linear system

To approximate the second derivatives appearing in Poisson's equation, we apply the central finite difference formula of order $2k$. Taking $x_j = jh$ for a lattice with spacing h , this formula gives the approximation

$$f''(0) \approx \frac{1}{h^2} \sum_{j=-k}^k r_j f(jh) \quad (3.13)$$

where the coefficients are [6, 100]

$$r_j := \begin{cases} \frac{2(-1)^{j+1}(k!)^2}{j^2(k-j)!(k+j)!} & j \in [k] \\ -2 \sum_{j=1}^k r_j & j = 0 \\ r_{-j} & j \in -[k]. \end{cases} \quad (3.14)$$

We leave the dependence on k implicit in this notation. The following lemma characterizes the error of this formula.

Lemma 3.1 ([6, Theorem 7]). *Let $k \geq 1$ and suppose $f(x) \in C^{2k+1}$ for $x \in \mathbb{R}$. Define the coefficients r_j as in (3.14). Then*

$$\frac{d^2 u(x_0)}{dx^2} = \frac{1}{h^2} \sum_{j=-k}^k r_j f(x_0 + jh) + O\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right| \left(\frac{eh}{2}\right)^{2k-1}\right) \quad (3.15)$$

where

$$\left| \frac{d^{2k+1}u}{dx^{2k+1}} \right| := \max_{y \in [x_0 - kh, x_0 + kh]} \left| \frac{d^{2k+1}u}{dx^{2k+1}}(y) \right|. \quad (3.16)$$

Since we assume periodic boundary conditions and apply the same FDM formula at each lattice site, the matrices we consider are circulant. Define the $2n \times 2n$ matrix S to have entries $S_{i,j} = \delta_{i,j+1 \bmod 2n}$. If we represent the solution $u(x)$ as a vector $\mathbf{u} = \sum_{j=1}^{2n} u(\pi j/n) \mathbf{e}_j$, then we can approximate Poisson's equation using a central difference formula as

$$\frac{1}{h^2} L \mathbf{u} = \frac{1}{h^2} \left(r_0 I + \sum_{j=1}^k r_j (S^j + S^{-j}) \right) \mathbf{u} = \mathbf{f} \quad (3.17)$$

where $\mathbf{f} = \sum_{j=1}^{2n} f(\pi j/n) \mathbf{e}_j$. The solution \mathbf{u} corresponds exactly with the quantum state we want to produce, so we do not have to perform any post-processing such as in Reference [70] and other quantum differential equation algorithms. The matrix in this linear system is just the finite difference matrix, so it suffices to bound its condition number and approximation error (whereas previous quantum algorithms involved more complicated linear systems).

3.3.2 Condition number

The following lemma characterizes the condition number of a circulant Laplacian on $2n$ points.

Lemma 3.2. *For $k < (6/\pi^2)^{1/3} n^{2/3}$, the matrix $L = r_0 I + \sum_{j=1}^k r_j (S^j + S^{-j})$ with r_j as*

in (3.14) has condition number $\kappa(L) = O(n^2)$.

Proof. We first upper bound $\|L\|$ using Gershgorin's circle theorem [101] (a similar argument appears in Reference [6]). Note that

$$|r_j| = \frac{2(k!)^2}{j^2(k-j)!(k+j)!} \leq \frac{2}{j^2} \quad (3.18)$$

since

$$\frac{(k!)^2}{(k-j)!(k+j)!} = \frac{k(k-1)\cdots(k-j+1)}{(k+j)(k+j-1)\cdots(k+1)} < 1. \quad (3.19)$$

The radii of the Gershgorin discs are

$$2 \sum_{j=1}^k |r_j| \leq 2 \sum_{j=1}^k \frac{2}{j^2} \leq \frac{2\pi^2}{3}. \quad (3.20)$$

The discs are centered at r_0 , and

$$|r_0| \leq 2 \sum_{j=1}^k |r_j| \leq \frac{2\pi^2}{3}, \quad (3.21)$$

so $\|L\| \leq \frac{4\pi^2}{3}$.

To lower bound $\|L^{-1}\|$ we lower bound the (absolute value of the) smallest non-zero eigenvalue of L (since by construction the all-ones vector is a zero eigenvector). Let

$\omega := \exp(\pi i/n)$. Since L is circulant, its eigenvalues are

$$\lambda_l = r_0 + \sum_{j=1}^k r_j (\omega^{lj} + \omega^{-lj}) \quad (3.22)$$

$$= r_0 + \sum_{j=1}^k 2r_j \cos\left(\frac{\pi lj}{n}\right) \quad (3.23)$$

$$= r_0 + \sum_{j=1}^k 2r_j \left(1 - \frac{\pi^2 l^2 j^2}{2n^2} + \frac{(\pi c_j)^4}{4!n^4} \cos\left(\frac{\pi c_j}{n}\right)\right) \quad (3.24)$$

$$= \sum_{j=1}^k 2r_j \left(-\frac{\pi^2 l^2 j^2}{2n^2} + \frac{(\pi c_j)^4}{4!n^4} \cos\left(\frac{\pi c_j}{n}\right)\right) \quad (3.25)$$

where the $c_j \in [0, lj]$ arise from the Taylor remainder theorem. Using (3.18), we have

$$\left| \lambda_1 + \frac{\pi^2}{n^2} \sum_{j=1}^k r_j j^2 \right| \leq \frac{\pi^4 k^3}{6n^4}. \quad (3.26)$$

We now compute the sum

$$-\sum_{j=1}^k r_j j^2 = \sum_{j=1}^k j^2 \frac{2(-1)^j (k!)^2}{j^2 (k+j)! (k-j)!} \quad (3.27)$$

$$= 2(k!)^2 \sum_{j=1}^k \frac{(-1)^j}{(k+j)! (k-j)!} \quad (3.28)$$

$$= \frac{2(k!)^2}{(2k)!} \sum_{j=1}^k (-1)^j \binom{2k}{k+j} \quad (3.29)$$

$$= \frac{2(k!)^2}{(2k)!} \sum_{j=k+1}^{2k} (-1)^{j+k} \binom{2k}{j} \quad (3.30)$$

$$= (-1)^k \frac{(k!)^2}{(2k)!} \sum_{j=0, j \neq k}^{2k} (-1)^j \binom{2k}{j} \quad (3.31)$$

$$= (-1)^k \frac{(k!)^2}{(2k)!} \left((1-1)^{2k} - (-1)^k \binom{2k}{k} \right) \quad (3.32)$$

$$= -1. \quad (3.33)$$

Therefore, we have

$$\lambda_1 \leq -\frac{\pi^2}{n^2} + \frac{\pi^4 k^3}{6n^4}. \quad (3.34)$$

Finally, we see that

$$\kappa(L) = \|L\| \|L^{-1}\| \quad (3.35)$$

$$\leq \frac{4\pi^2}{3} \left(\frac{\pi^2}{n^2} - \frac{\pi^4 k^3}{6n^4} \right)^{-1} \quad (3.36)$$

$$= \frac{4}{3} n^2 \left(1 - \frac{\pi^2 k^3}{6n^2} \right)^{-1} \quad (3.37)$$

which is $O(n^2)$ provided $k < (6/\pi^2)^{1/3} n^{2/3}$. \square

In d dimensions, a similar analysis holds.

Lemma 3.3. For $k < (6/\pi^2)^{1/3}n^{2/3}$, let $L := r_0I + \sum_{j=1}^k r_j(S^j + S^{-j})$ with r_j as in (3.14). The matrix $L^\theta := L \otimes I^{d-1} + I \otimes L \otimes I^{d-2} + \dots + I^{d-1} \otimes L$ has condition number $\kappa(L^\theta) = O(dn^2)$.

Proof. By the triangle inequality for spectral norms, $\|L^\theta\| \leq d\|L\|$. Since L has zero-sum rows by construction, the all-ones vector lies in its kernel, and thus the smallest non-zero eigenvalue of L is the same as that of L^θ . Therefore we have

$$\kappa(L^\theta) \leq \frac{4}{3}dn^2 \left(1 - \frac{\pi^2 k^3}{6n^2}\right)^{-1} \quad (3.38)$$

which is $O(dn^2)$ provided $k < (6/\pi^2)^{1/3}n^{2/3}$. \square

3.3.3 Error analysis

There are two types of error relevant to our analysis: the FDM error and the QLSA error. We assume that we are able to perfectly generate states proportional to \mathbf{f} . The FDM errors arise from the remainder terms in the finite difference formulas and from inexact approximations of the eigenvalues.

We introduce several states for the purpose of error analysis. Let $|u\rangle$ be the quantum state that is proportional to $\mathbf{u} = \sum_{j \in \mathbb{Z}_{2n}^d} u(\pi j/n) \bigotimes_{i=1}^d e_{j_i}$ for the exact solution of the differential equation. Let $|\bar{u}\rangle$ be the state output by a QLSA that exactly solves the linear system. Let $|\tilde{u}\rangle$ be the state output by a QLSA with error. Then the total error

of approximating $|u\rangle$ by $|\tilde{u}\rangle$ is bounded by

$$\| |u\rangle - |\tilde{u}\rangle \| \leq \| |u\rangle - |\bar{u}\rangle \| + \| |\bar{u}\rangle - |\tilde{u}\rangle \| \quad (3.39)$$

$$= \epsilon_{\text{FDM}} + \epsilon_{\text{QLSA}} \quad (3.40)$$

and without loss of generality we can take ϵ_{FDM} and ϵ_{QLSA} to be of the same order of magnitude.

Lemma 3.4. *Let $u(\mathbf{x})$ be the exact solution of $(\sum_{i=1}^d \frac{d^2}{dx_i^2})u(\mathbf{x}) = f(\mathbf{x})$. Let $\mathbf{u} \in \mathbb{R}^{(2n)^d}$ encode the exact solution in the sense that $\mathbf{u} = \sum_{j \in \mathbb{Z}_{2n}^d} u(\pi j/n) \otimes_{i=1}^d e_{j_i}$. Let $\bar{\mathbf{u}} \in \mathbb{R}^{(2n)^d}$ be the exact solution of the FDM linear system $\frac{1}{h^2} L^\theta \bar{\mathbf{u}} = \mathbf{f}$, where L^θ is a d -dimensional $(2k)$ th-order Laplacian as above with $k < (6/\pi^2)^{1/3} n^{3/2}$, and $\mathbf{f} = \sum_{j=1}^{2n} f(\pi j/n) \mathbf{e}_j$. Then $\| \mathbf{u} - \bar{\mathbf{u}} \| \leq O(2^{d/2} n^{(d/2) - 2k+1} \left| \frac{d^{2k+1} u}{dx^{2k+1}} \right| (e^2/4)^k)$.*

Proof. The remainder term of the central difference formula is $O\left(\left| \frac{d^{2k+1} u}{dx^{2k+1}} \right| h^{2k-1} (e/2)^{2k}\right)$,

so

$$\frac{1}{h^2} L^\theta \mathbf{u} = \mathbf{f} + O\left(\left| \frac{d^{2k+1} u}{dx^{2k+1}} \right| (eh/2)^{2k-1}\right) \epsilon \quad (3.41)$$

where ϵ is a $(2n)^d$ dimensional vector whose entries are $O(1)$. This implies

$$\frac{1}{h^2} L^\theta (\mathbf{u} - \bar{\mathbf{u}}) = O\left(\left| \frac{d^{2k+1} u}{dx^{2k+1}} \right| (eh/2)^{2k-1}\right) \epsilon \quad (3.42)$$

and therefore

$$\|\mathbf{u} - \bar{u}\| = O\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|(eh/2)^{2k+1}\right)\|(L^\theta)^{-1}\epsilon\| \quad (3.43)$$

$$= O\left((2n)^{d/2}\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|(eh/2)^{2k+1}/\lambda_1\right). \quad (3.44)$$

By [Lemma 3.2](#) we have $\lambda_1 = \Theta(1/n^2)$, and since $h = \Theta(1/n)$, we have

$$\|\mathbf{u} - \bar{u}\| = O\left(2^{d/2}n^{(d/2)-2k+1}\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|(e/2)^{2k}\right) \quad (3.45)$$

as claimed. □

3.3.4 FDM algorithm

To apply QLSAs, we must consider the complexity of simulating Hamiltonians that correspond to Laplacian FDM operators. For periodic boundary conditions, the Laplacians are circulant, so they can be diagonalized by the QFT F (or a tensor product of QFTs for the multi-dimensional Laplacian L^θ), i.e., $D = F^\dagger L F$ is diagonal. In this case the simplest way to simulate $\exp(iLt)$ is to perform the inverse QFT, apply controlled phase rotations to implement $\exp(iDt)$, and perform the QFT. Reference [\[102\]](#) shows how to exactly implement arbitrary diagonal unitaries on m qubits using $O(2^m)$ gates. Since we consider Laplacians on n lattice sites, simulating $\exp(iLt)$ takes $O(n)$ gates with the dominant contribution coming from the phase rotations (alternatively, the methods of Reference [\[103\]](#) or Reference [\[104\]](#) could also be used). Using this Hamiltonian simulation algorithm in a QLSA for the FDM linear system gives us the following theorem.

We restate [Theorem 3.1](#) as follows.

Theorem 3.1. *There exists a quantum algorithm that outputs a state ϵ -close to $|u\rangle$ that runs in time*

$$\tilde{O}\left(d^{6.5} \log^{4.5}\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right) \sqrt{\log\left[d^4 \log^3\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right)/\epsilon}\right) \quad (3.11)$$

and makes

$$\tilde{O}\left(d^4 \log^3\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right) \sqrt{\log\left[d^4 \log^3\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right)/\epsilon}\right) \quad (3.12)$$

queries to the oracle for \mathbf{f} .

Proof. We use the Fourier series based QLSA from Reference [46]. By Theorem 3 of that work, the QLSA makes $O(\kappa\sqrt{\log(\kappa/\epsilon_{\text{QLSA}})})$ uses of a Hamiltonian simulation algorithm and uses of the oracle for the inhomogeneity. For Hamiltonian simulation we use d parallel QFTs and phase rotations as described in Reference [102], for a total of $O(dn\kappa\sqrt{\log(\kappa/\epsilon_{\text{QLSA}})})$ gates. The condition number for the d -dimensional Laplacian scales as $\kappa = O(dn^2)$.

We take ϵ_{FDM} and ϵ_{QLSA} to be of the same order and just write ϵ . Then the QLSA has time complexity $O(d^2n^3\sqrt{\log(dn^2/\epsilon)})$ and query complexity $O(dn^2\log(dn^2/\epsilon))$. The adjustable parameters are the number of lattice sites n and the order $2k$ of the finite difference formula. To keep the error below the target error of ϵ we require

$$2^{d/2}n^{(d/2)-2k+1}\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|(e/2)^{2k} = O(\epsilon), \quad (3.46)$$

or equivalently,

$$(-d/2) + (2k - 1 - (d/2)) \log(n) - 2k \log(e/2) = \Omega\left(\log\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right)\right). \quad (3.47)$$

Now we focus on the choice of adjustable n and k relying on ϵ . This procedure is inspired by the classical adaptive FDM [71], so we call it the adaptive FDM approach. We must have $2k - 1 > d/2$ for the left-hand side of (3.47) to be positive for large n . Indeed, we find the best performance by taking k as large as possible subject to the assumption of Lemma 3.2, i.e., $k = cn^{2/3}$ where $c := (6/\pi^2)^{1/3}$. For this choice of k and for n sufficiently large, (3.47) is equivalent to

$$k \log(n) = cn^{2/3} \log(n) = \Omega\left(\log\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right)\right). \quad (3.48)$$

To satisfy the condition $2cn^{2/3} - 1 > d/2$, we must have $n = \Omega(d^{3/2})$. Combining this observation with (3.48), we choose

$$n = \Theta\left(d^{3/2} \log^{3/2}\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right)\right) \quad (3.49)$$

so that

$$k = cn^{2/3} = \Theta\left(d \log\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right)\right). \quad (3.50)$$

The QLSA then has the stated time complexity

$$\tilde{O}(d^2 n^3 \sqrt{\log(dn^2/\epsilon)}) = O\left(d^{6.5} \log^{4.5}\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right) \sqrt{\log\left[d^4 \log^3\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right)/\epsilon}\right), \quad (3.51)$$

and makes

$$\tilde{O}(dn^2 \log(dn^2/\epsilon)) = O\left(d^4 \log^3\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right) \sqrt{\log\left[d^4 \log^3\left(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon\right)/\epsilon}\right). \quad (3.52)$$

queries to the oracle for f . □

This can be compared to the cost of using the conjugate gradient method to solve the same linear system classically. The sparse conjugate gradient algorithm for an $N \times N$ matrix has time complexity $O(Ns\sqrt{\kappa} \log(1/\epsilon))$. For arbitrary dimension $N = \Theta(n^d)$, we have $s = dk =cdn^{2/3}$ and $\kappa = O(dn^2)$, so that the time complexity is $O(d^{4+3d/2} \log(1/\epsilon) \log^{5/2+3d/2}(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon))$. Alternatively, d fast Fourier transforms could be used, although this will generally take $\Omega(n^d) = \Omega(d^{3d/2} \log^{3d/2}(\left|\frac{d^{2k+1}u}{dx^{2k+1}}\right|/\epsilon))$ time.

3.3.5 Boundary conditions via the method of images

We can apply the method of images to deal with homogeneous Neumann and Dirichlet boundary conditions using the algorithm for periodic boundary conditions described above. In the method of images, the domain $[-1, 1]$ is extended to include all of \mathbb{R} , and the boundary conditions are related to symmetries of the solutions. For a pair of Dirichlet

boundary conditions there are two symmetries: the solutions are anti-symmetric about -1 (i.e., $f(-x - 1) = -f(x - 1)$) and anti-symmetric about 1 (i.e., $f(1 + x) = -f(1 - x)$). Continuity and anti-symmetry about -1 and 1 imply $f(-1) = f(1) = 0$, and furthermore that $f(x) = 0$ for all odd $x \in \mathbb{Z}$ and that $f(x + 4) = f(x)$ for all $x \in \mathbb{R}$. For Neumann boundary conditions, the solutions are instead symmetric about -1 and 1 , which also implies $f(x + 2) = f(x)$ for all $x \in \mathbb{R}$.

We would like to combine the method of images with the FDM to arrive at finite difference formulas for this special case. In both cases, the method of images implies that the solutions are periodic, so without loss of generality we can consider a lattice on $[0, 2\pi)$ instead of a lattice on \mathbb{R} . It is useful to think of this lattice in terms of the cycle graph on $2n$ vertices, i.e., $(V, E) = (\mathbb{Z}_{2n}, \{(i, i + 1) \mid i \in \mathbb{Z}_{2n}\})$, which means that the vectors encoding the solution $u(x)$ will lie in \mathbb{R}^{2n} . Let each vector e_j correspond to the vertex j . Then we divide \mathbb{R}^{2n} into a symmetric and an anti-symmetric subspace, namely $\text{span}\{e_j + e_{2n+1-j}\}_{j=1}^n$ and $\text{span}\{e_j - e_{2n+1-j}\}_{j=1}^n$, respectively. Vectors lying in the symmetric subspace correspond to solutions that are symmetric about 0 and π , so they obey Neumann boundary conditions at 0 and π ; similarly, vectors in the anti-symmetric space correspond to solutions obeying Dirichlet boundary conditions at 0 and π .

Restricting to a subspace of vectors reduces the size of the FDM vectors and matrices we consider, and the symmetry of that subspace indicates how to adjust the coefficients.

If the FDM linear system is $L^{00}\mathbf{u}^{00} = \mathbf{f}^{00}$ then L^{00} has entries

$$L_{i,j}^{00} = \begin{cases} r_{j\bar{i}} - r_{j\bar{j}} \pm r_{i+j-1} & i \leq k \\ r_{j\bar{i}} - r_{j\bar{j}} & k < i \leq n-k \\ r_{j\bar{i}} - r_{j\bar{j}} \pm r_{2n-i-j+1} & n-k \leq i \end{cases} \quad (3.53)$$

where $+$ ($-$) is chosen for Neumann (Dirichlet) boundary conditions and due to the truncation order k , $r_j = 0$ for any $j > k$. This is similar to how Laplacian coefficients are modified when imposing boundary conditions in discrete variable representations [105].

For the purpose of solving the new linear systems using quantum algorithms, we still treat these cases as obeying periodic boundary conditions. We assume access to an oracle that produces states $|f^{00}\rangle$ proportional to the inhomogeneity $f^{00}(x)$. Then we apply the QLSA for periodic boundary conditions using $|f^{00}\rangle|\pm\rangle$ to encode the inhomogeneity, which will output solutions of the form $|u^{00}\rangle|\pm\rangle$. Here the ancillary state is chosen to be $|+\rangle$ ($|-\rangle$) for Neumann (Dirichlet) boundary conditions.

Typically, the (second-order) graph Laplacian for the path graph with Dirichlet boundary conditions has diagonal entries that are all equal to 2; however, using the above specification for the entries of L leads to the $(1, 1)$ and (n, n) entries being 3 while the rest of the diagonal entries are 2.

To reproduce this case, we consider an alternative subspace restriction used in Reference [106] to diagonalize the Dirichlet graph Laplacian. In this case it is easiest to consider the lattice of a cycle graph on $2n + 2$ vertices, where the vertices 0 and $n + 1$ are selected as boundary points where the field takes the value 0. The relevant antisymmetric

subspace is now $\text{span}(\{e_j - e_{2n+2-j}\}_{j=1}^n)$ (which has no support on e_0 and e_{n+1}).

If we again write the linear system as $L^{00}\mathbf{u}^{00} = \mathbf{f}^{00}$, then the Laplacian has entries

$$L_{i,j}^{00} = \begin{cases} r_{ji} - r_{i+j} & i \leq k \\ r_{ji} & k < i \leq n - k \\ r_{ji} - r_{2n-i-j+2} & n - k \leq i. \end{cases}$$

We again assume access to an oracle producing states proportional to $f^{00}(x)$; however, we assume that this oracle operates in a Hilbert space with one additional dimension compared to the previous approaches (i.e., whereas previously we considered implementing U , here we consider implementing $\begin{pmatrix} U & 0 \\ 0^T & 1 \end{pmatrix}$). With this oracle we again prepare the state $|f^{00}\rangle|-\rangle$ and solve Poisson's equation for periodic boundary conditions to output a state $|u^{00}\rangle|-\rangle$ (where $|u^{00}\rangle$ lies in an $(n + 1)$ -dimensional Hilbert space but has no support on the $(n + 1)$ st basis state).

3.4 Multi-dimensional spectral method

We now turn our attention to the spectral method for multi-dimensional PDEs. Since interpolation facilitates constructing a straightforward linear system, we develop a quantum algorithm based on the *pseudo-spectral method* [67, 68, 107] for second-order elliptic equations with global strict diagonal dominance, under various boundary conditions. Using this approach, we show the following.

Theorem 3.2. *Consider an instance of the quantum PDE problem as defined in [Problem 3.1](#) with Dirichlet boundary conditions (3.81). Then there exists a quantum algorithm that*

produces a state in the form of (3.82) whose amplitudes are proportional to $u(\mathbf{x})$ on a set of interpolation nodes \mathbf{x} (with respect to the uniform grid nodes for periodic boundary conditions or the Chebyshev-Gauss-Lobatto quadrature nodes for non-periodic boundary conditions, as defined in (3.60)), where $u(\mathbf{x})/\|u(\mathbf{x})\|$ is ϵ -close to $\hat{u}(\mathbf{x})/\|\hat{u}(\mathbf{x})\|$ in l_2 norm for all nodes \mathbf{x} , succeeding with probability $\Omega(1)$, with a flag indicating success, using

$$\left(\frac{d\|A\|_{\Sigma}}{C\|A\|} + qd^2 \right) \text{poly}(\log(g^{\theta}/g\epsilon)) \quad (3.54)$$

queries to oracles as defined in Section 3.4.4. Here $\|A\|_{\Sigma} := \sum_{k,j,k_1} \|A_j\|$, $\|A\| := \sum_{j=1}^d |A_{j,j}|$, $C > 0$ is defined in (3.8), and

$$g = \min_{\mathbf{x}} \|\hat{u}(\mathbf{x})\|, \quad g^{\theta} := \max_{\mathbf{x}} \max_{n \in \mathbb{N}} \|\hat{u}^{(n+1)}(\mathbf{x})\|, \quad (3.55)$$

$$q = \sqrt{\frac{\sum_{k,k_1} \sum_{j=1}^d \hat{f}_{\mathbf{k}}^2 + (A_{j,j} \hat{\gamma}_{\mathbf{k}}^{j+})^2 + (A_{j,j} \hat{\gamma}_{\mathbf{k}}^j)^2}{\sum_{k,k_1} \sum_{j=1}^d (\hat{f}_{\mathbf{k}} + A_{j,j} \hat{\gamma}_{\mathbf{k}}^{j+} + A_{j,j} \hat{\gamma}_{\mathbf{k}}^j)^2}}. \quad (3.56)$$

The gate complexity is larger than the query complexity by a factor of $\text{poly}(\log(d\|A\|_{\Sigma}/\epsilon))$.

After introducing the method, we discuss the complexity of the quantum shifted Fourier transform (Lemma 3.5) and the quantum cosine transform (Lemma 3.6) in Section 3.4.1. These transforms are used as subroutines in our algorithm. Then we construct a linear system whose solution encodes the solution of the PDE in Section 3.4.2, analyze its condition number in Section 3.4.3 (Lemma 3.10, established using Lemma 3.7, Lemma 3.8, and Lemma 3.9), and consider the complexity of state preparation in Section 3.4.4 (Lemma 3.11). Finally, we prove our main result (Theorem 3.2) in Section 3.4.5.

In the spectral approach, we approximate the exact solution $\hat{u}(\mathbf{x})$ by a linear combination

of basis functions

$$u(\mathbf{x}) = \sum_{\mathbf{k} \in [n+1]_0} c_{\mathbf{k}} \phi_{\mathbf{k}}(\mathbf{x}) \quad (3.57)$$

for some $n \in \mathbb{Z}^+$. Here $\mathbf{k} = (k_1, \dots, k_d)$ with $k_j \in [n+1]_0 := \{0, 1, \dots, n\}$, $c_{\mathbf{k}} \in \mathbb{C}$, and

$$\phi_{\mathbf{k}}(\mathbf{x}) = \prod_{j=1}^d \phi_{k_j}(x_j), \quad j \in [d]. \quad (3.58)$$

We choose different basis functions for the case of periodic boundary conditions and for the more general case of non-periodic boundary conditions. When the boundary conditions are periodic, the algorithm implementation is more straightforward, and in some cases (e.g., for the Poisson equation), can be faster. Specifically, for any $k_j \in [n+1]_0$ and $x_j \in [-1, 1]$, we take

$$\phi_{k_j}(x_j) = \begin{cases} e^{i(k_j - bn/2c)\pi x_j}, & \text{periodic conditions,} \\ T_{k_j}(x_j) := \cos(k_j \arccos x_j), & \text{non-periodic conditions.} \end{cases} \quad (3.59)$$

Here T_k is the degree- k Chebyshev polynomial of the first kind.

The coefficients $c_{\mathbf{k}}$ are determined by demanding that $u(\mathbf{x})$ satisfies the ODE and boundary conditions at a set of *interpolation nodes* $\{\chi_l = (\chi_{l_1}, \dots, \chi_{l_d})\}_{\mathbf{l} \in [n+1]_0}$ with $l_j \in [n+1]_0$, where

$$\chi_{l_j} = \begin{cases} \frac{2l_j}{n+1} - 1, & \text{periodic conditions,} \\ \cos \frac{\pi l_j}{n}, & \text{non-periodic conditions.} \end{cases} \quad (3.60)$$

Here $\{\frac{2l}{n+1} - 1 : l \in [n+1]_0\}$ are called the *uniform grid nodes*, and $\{\cos \frac{\pi l}{n} : l \in [n+1]_0\}$ are called the *Chebyshev-Gauss-Lobatto quadrature nodes*.

We require the numerical solution $u(\mathbf{x})$ to satisfy

$$L(u(\mathbf{x}_l)) = f(\mathbf{x}_l), \quad \forall l_j \in [n+1]_0, j \in [d]. \quad (3.61)$$

We would like to be able to increase the accuracy of the approximation by increasing n , so that

$$\|\hat{u}(\mathbf{x}) - u(\mathbf{x})\| \rightarrow 0 \quad \text{as } n \rightarrow \infty. \quad (3.62)$$

The convergence behavior of the spectral method is related to the smoothness of the solution. For a solution in C^{r+1} , the spectral method approximates the solution with $n = \text{poly}(1/\epsilon)$. Furthermore, if the solution is in C^1 , the spectral method approximates the solution to within ϵ using only $n = \text{poly}(\log(1/\epsilon))$ [68]. Since we require $k_j \in [n+1]_0$ for all $j \in [d]$, we have $(n+1)^d$ terms in total. Consequently, a classical pseudo-spectral method solves multi-dimensional PDEs with complexity $\text{poly}(\log^d(1/\epsilon))$. Such classical spectral methods rapidly become infeasible since the number of coefficients $(n+1)^d$ grows exponentially with d .

Here we develop a quantum algorithm for multi-dimensional PDEs. The algorithm applies techniques from the quantum spectral method for ODEs [54]. However, in the case of PDEs, the linear system to be solved is non-sparse. We address this difficulty using a quantum transform that restores sparsity.

3.4.1 Quantum shifted Fourier transform and quantum cosine transform

The well-known *quantum Fourier transform* (QFT) can be regarded as an analogue of the discrete Fourier transform (DFT) acting on the amplitudes of a quantum state. The QFT maps the $(n + 1)$ -dimensional quantum state $v = (v_0, v_1, \dots, v_n) \in \mathbb{C}^{n+1}$ to the state $\hat{v} = (\hat{v}_0, \hat{v}_1, \dots, \hat{v}_n) \in \mathbb{C}^{n+1}$ with

$$\hat{v}_l = \frac{1}{\sqrt{n+1}} \sum_{k=0}^n \exp\left(\frac{2\pi i k l}{n+1}\right) v_k, \quad l \in [n+1]_0. \quad (3.63)$$

In other words, the QFT is the unitary transform

$$F_n := \frac{1}{\sqrt{n+1}} \sum_{k,l=0}^n \exp\left(\frac{2\pi i k l}{n+1}\right) |l\rangle\langle k|. \quad (3.64)$$

Here we also consider the *quantum shifted Fourier transform* (QSFT), an analogue of the classical shifted discrete Fourier transform, which maps $v \in \mathbb{C}^{n+1}$ to $\hat{v} \in \mathbb{C}^{n+1}$ with

$$\hat{v}_l = \frac{1}{\sqrt{n+1}} \sum_{k=0}^n \exp\left(\frac{2\pi i (k - \lfloor n/2 \rfloor)(l - (n+1)/2)}{n+1}\right) v_k, \quad l \in [n+1]_0. \quad (3.65)$$

In other words, the QSFT is the unitary transform

$$F_n^s := \frac{1}{\sqrt{n+1}} \sum_{k,l=0}^n \exp\left(\frac{2\pi i (k - \lfloor n/2 \rfloor)(l - (n+1)/2)}{n+1}\right) |l\rangle\langle k|. \quad (3.66)$$

We define the multi-dimensional QSFT by the tensor product, namely

$$\mathbf{F}_n^s := \frac{1}{\sqrt{(n+1)^d}} \sum_{\mathbf{k}, \mathbf{l}} \prod_{j=1}^d \exp\left(\frac{2\pi i(k_j - bn/2c)(l_j - (n+1)/2)}{n+1}\right) |\mathbf{l}\rangle \dots |\mathbf{l}_d\rangle \langle \mathbf{k}_1| \dots \langle \mathbf{k}_d|, \quad (3.67)$$

where $\mathbf{k} = (k_1, \dots, k_d)$ and $\mathbf{l} = (l_1, \dots, l_d)$ are d -dimensional vectors with $k_j, l_j \in [n]_0$.

The QSFT can be efficiently implemented as follows:

Lemma 3.5. *The QSFT F_n^s defined by (3.66) can be performed with gate complexity $O(\log n \log \log n)$. More generally, the d -dimensional QSFT \mathbf{F}_n^s defined by (3.67) can be performed with gate complexity $O(d \log n \log \log n)$.*

Proof. The unitary matrix F_n^s can be written as the product of three unitary matrices

$$F_n^s = S_n F_n R_n, \quad (3.68)$$

where

$$R_n = \sum_{k=0}^n \exp\left(-\frac{2\pi i k(n+1)/2}{n+1}\right) |k\rangle \langle k| \quad (3.69)$$

and

$$S_n = \sum_{l=0}^n \exp\left(-\frac{2\pi i \lfloor n/2 \rfloor (l - (n+1)/2)}{n+1}\right) |l\rangle \langle l|. \quad (3.70)$$

It is well known that F_n can be implemented with gate complexity $O(\log n \log \log n)$, and it is straightforward to implement R_n and S_n with gate complexity $O(\log n)$. Thus the total complexity is $O(\log n \log \log n)$.

We rewrite v in the form

$$v = \sum_{\mathbf{k} \in [n]^d} v_{\mathbf{k}} |k_1\rangle \dots |k_d\rangle, \quad (3.71)$$

where $v_{\mathbf{k}} \in \mathbb{C}$ with $\mathbf{k} = (k_1, \dots, k_d)$, and each $k_j \in [n]_0$ for $j \in [d]$. The unitary matrix F_n^s can be written as the tensor product

$$F_n^s = \bigotimes_{j=1}^d F_n^s. \quad (3.72)$$

Performing the multi-dimensional QSFT is equivalent to performing the one-dimensional QSFT on each register. Thus, the gate complexity of performing F_n^s is $O(d \log n \log \log n)$.

□

Another efficient quantum transformation is the *quantum cosine transform* (QCT) [108, 109]. The QCT can be regarded as an analogue of the discrete cosine transform (DCT). The QCT maps $v \in \mathbb{C}^{n+1}$ to $\hat{v} \in \mathbb{C}^{n+1}$ with

$$\hat{v}_l = \sqrt{\frac{2}{n}} \sum_{k=0}^n \delta_k \delta_l \cos \frac{kl\pi}{n} v_k, \quad l \in [n+1]_0, \quad (3.73)$$

where

$$\delta_l := \begin{cases} \frac{1}{\sqrt{2}} & l = 0, n \\ 1 & l \in [n-1]. \end{cases} \quad (3.74)$$

In other words, the QCT is the orthogonal transform

$$C_n := \sqrt{\frac{2}{n}} \sum_{k,l=0}^n \delta_l \delta_k \cos \frac{kl\pi}{n} |l\rangle \langle k|. \quad (3.75)$$

Again we define the multi-dimensional QCT by the tensor product, namely

$$\mathbf{C}_n := \sqrt{\left(\frac{2}{n}\right)^d} \sum_{\mathbf{k}, \mathbf{l}} \prod_{j=1}^d \delta_{k_j} \delta_{l_j} \cos \frac{k_j l_j \pi}{n} |l_1\rangle \dots |l_d\rangle \langle k_1| \dots \langle k_d|, \quad (3.76)$$

where $\mathbf{k} = (k_1, \dots, k_d)$ and $\mathbf{l} = (l_1, \dots, l_d)$ are d -dimensional vectors with $k_j, l_j \in [n+1]_0$.

The classical DCT on $(n+1)$ -dimensional vectors takes $\Theta(n \log n)$ gates, while the QCT on $(n+1)$ -dimensional quantum states can be implemented with complexity $\text{poly}(\log n)$. According to Theorem 1 of Reference [108], the gate complexity of performing C_n is $O(\log^2 n)$. We observe that this can be improved as follows.

Lemma 3.6. *The quantum cosine transform C_n defined by (3.75) can be performed with gate complexity $O(\log n \log \log n)$. More generally, the multi-dimensional QCT \mathbf{C}_n defined by (3.76) can be performed with gate complexity $O(d \log n \log \log n)$.*

Proof. According to the quantum circuit in Figure 2 of Reference [108], C_n can be

decomposed into a QFT F_{n+1} , a permutation

$$P_n = \begin{pmatrix} & & & 1 \\ & 1 & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{pmatrix}, \quad (3.77)$$

and additional operations with $O(1)$ cost. The QFT F_{n+1} has gate complexity $O(\log n \log \log n)$. We then consider an alternative way to implement P_n that improves over the approach in [110].

The permutation P_n can be decomposed as

$$P_n = F_n T_n F_n^{-1}, \quad (3.78)$$

where F_n is the Fourier transform (3.64) and $T_n = \sum_{k=0}^n e^{\frac{2\pi i k}{n+1}} |k\rangle\langle k|$ is diagonal. The gate complexities of performing F_n and T_n are $O(\log n \log \log n)$ and $O(\log n)$, respectively.

It follows that C_n can be implemented with circuit complexity $O(\log n \log \log n)$.

The matrix C_n can be written as the tensor product

$$C_n = \bigotimes_{j=1}^d C_n. \quad (3.79)$$

As in Lemma 3.5, performing the multi-dimensional QCT is equivalent to performing a QCT on each register. Thus, the gate complexity of performing C_n is $O(d \log n \log \log n)$.

□

3.4.2 Linear system

In this section we introduce the quantum PDE solver for the problem (3.1). We construct a linear system that encodes the solution of (3.1) according to the pseudo-spectral method introduced above, using the QSFT/QCT introduced in Section 3.4.1 to ensure sparsity.

We consider a linear PDE problem (Problem 3.1) with periodic boundary conditions

$$u(\mathbf{x} + 2\mathbf{v}) = u(\mathbf{x}) \quad \forall \mathbf{x} \in D, \forall \mathbf{v} \in \mathbb{Z}^d \quad (3.80)$$

or non-periodic Dirichlet boundary conditions

$$u(\mathbf{x}) = \gamma(\mathbf{x}) \quad \forall \mathbf{x} \in \partial D. \quad (3.81)$$

According to the elliptic regularity theorem (Theorem 6 in Section 6.3 of Reference [99]), there exists a unique solution $\hat{u}(\mathbf{x})$ in C^1 for Problem 3.1.

We now show how to apply the Fourier and Chebyshev pseudo-spectral methods to this problem. Our goal is to obtain the quantum state

$$|u\rangle \propto \sum_{\mathbf{k} \in \mathbb{Z}^d} c_{\mathbf{k}} \phi_{\mathbf{k}}(\chi_1) |l_1\rangle \cdots |l_d\rangle, \quad (3.82)$$

where $\phi_{\mathbf{k}}(\chi_l)$ is defined by (3.58) using (3.59) for the appropriate boundary conditions

(periodic or non-periodic). This state corresponds to a truncated Fourier/Chebyshev approximation and is ϵ -close to the exact solution $\hat{u}(\chi_l)$ with $n = \text{poly}(\log(1/\epsilon))$ [68]. Note that this state encodes the values of the solution at the interpolation nodes (3.60) appropriate to the boundary conditions (the uniform grid nodes in the Fourier approach, for periodic boundary conditions, and the Chebyshev-Gauss-Lobatto quadrature nodes in the Chebyshev approach, for non-periodic boundary conditions).

Instead of developing our algorithm for the standard basis, we aim to produce a state

$$|c\rangle \propto \sum_{k \mathbf{k} k_1} c_{\mathbf{k}} |k_1\rangle \dots |k_d\rangle \quad (3.83)$$

that is the inverse QSFT/QCT of $|u\rangle$. We then apply the QSFT/QCT to transform back into the interpolation node basis.

The truncated spectral series of the inhomogeneity $f(\mathbf{x})$ and the boundary conditions $\gamma(\mathbf{x})$ can be expressed as

$$f(\mathbf{x}) = \sum_{k \mathbf{k} k_1} \hat{f}_{\mathbf{k}} \phi_{\mathbf{k}}(\mathbf{x}) \quad (3.84)$$

and

$$\gamma(\mathbf{x}) = \sum_{k \mathbf{k} k_1} \hat{\gamma}_{\mathbf{k}} \phi_{\mathbf{k}}(\mathbf{x}), \quad (3.85)$$

respectively. We define quantum states $|f\rangle$ and $|\gamma\rangle$ by interpolating the nodes $\{\chi_l\}$ defined by (3.60) as

$$|f\rangle \propto \sum_{k \mathbf{k} k_1, k_l k_1} \phi_{k_j}(\chi_l) \hat{f}_{\mathbf{k}} |l_1\rangle \dots |l_d\rangle, \quad (3.86)$$

and

$$|\gamma\rangle \propto \sum_{\mathbf{k} \mathbf{k}_1, \mathbf{l} \mathbf{l}_1} \phi_{k_j}(\boldsymbol{\chi}_l) \hat{\gamma}_{\mathbf{k}} |l_1\rangle \dots |l_d\rangle, \quad (3.87)$$

respectively. These are the states that we assume we can produce using oracles. We perform the multi-dimensional inverse QSFT/QCT to obtain the states

$$|\hat{f}\rangle \propto \sum_{\mathbf{k} \mathbf{k}_1} \hat{f}_{\mathbf{k}} |k_1\rangle \dots |k_d\rangle, \quad (3.88)$$

and

$$|\hat{\gamma}\rangle \propto \sum_{\mathbf{k} \mathbf{k}_1} \hat{\gamma}_{\mathbf{k}} |k_1\rangle \dots |k_d\rangle. \quad (3.89)$$

Having defined these states, we now detail the construction of the linear system. At a high level, we construct two linear systems: one system $A\mathbf{x} = \mathbf{f}$ (where \mathbf{x} corresponds to (3.83)) describes the differential equation, and another system $B\mathbf{x} = \mathbf{g}$ describes the boundary conditions. We combine these into a linear system with the form

$$L\mathbf{x} = (A + B)\mathbf{x} = \mathbf{f} + \mathbf{g}. \quad (3.90)$$

Even though we do not impose the two linear systems separately, we show that there exists a unique solution of (3.90) (which is therefore the solution of the simultaneous equations $A\mathbf{x} = \mathbf{f}$ and $B\mathbf{x} = \mathbf{g}$), since we show that L has full rank, and indeed we upper bound its condition number in [Section 3.4.3](#).

Part of this linear system will correspond to just the differential equation

$$L(u(\boldsymbol{\chi}_l)) = \sum_{k \in \mathcal{K}, k_1=2} A_k \frac{\partial^j}{\partial \boldsymbol{x}^j} u(\boldsymbol{\chi}_l) = f(\boldsymbol{\chi}_l), \quad (3.91)$$

while another part will come from imposing the boundary conditions on $\partial D = \bigcup_{j \in [d]} \partial D_j$, where $\partial D_j := \{\boldsymbol{x} \in D \mid x_j = \pm 1\}$ is a $(d-1)$ -dimensional subspace. More specifically, the boundary conditions

$$u(\boldsymbol{\chi}_l) = \gamma(\boldsymbol{\chi}_l) \quad \forall \boldsymbol{\chi}_l \in \partial D \quad (3.92)$$

can be expressed as conditions on each boundary:

$$\begin{aligned} u(x_1, \dots, x_{j-1}, 1, x_{j+1}, \dots, x_d) &= \gamma^{j+}, & \boldsymbol{x} \in \partial D_j, & \quad j \in [d] \\ u(x_1, \dots, x_{j-1}, -1, x_{j+1}, \dots, x_d) &= \gamma^{j-}, & \boldsymbol{x} \in \partial D_j, & \quad j \in [d]. \end{aligned} \quad (3.93)$$

3.4.2.1 Linear system from the differential equation

To evaluate the matrix corresponding to the differential operator from (3.91), it is convenient to define coefficients $c_{\boldsymbol{k}}^{(j)}$ and $\|\boldsymbol{k}\|_1 \leq n$ such that

$$\frac{\partial^j}{\partial \boldsymbol{x}^j} u(\boldsymbol{x}) = \sum_{\substack{\boldsymbol{k} \in \mathcal{K} \\ \|\boldsymbol{k}\|_1 \leq n}} c_{\boldsymbol{k}}^{(j)} \phi_{\boldsymbol{k}}(\boldsymbol{x}) \quad (3.94)$$

for some fixed $\boldsymbol{j} \in \mathbb{N}^d$ (as we explain below, such a decomposition exists for the choices of basis functions in (3.59)). Using this expression, we obtain the following linear equations

for $c_{\mathbf{k}}^{(j)}$:

$$\sum_{k_j k_1=2} A_j \sum_{\mathbf{k} \mathbf{k} k_1, k_l k_1} \phi_{\mathbf{k}}(\chi_l) c_{\mathbf{k}}^{(j)} |l_1\rangle \dots |l_d\rangle = \sum_{\mathbf{k} \mathbf{k} k_1, k_l k_1} \phi_{\mathbf{k}}(\chi_l) \hat{f}_{\mathbf{k}} |l_1\rangle \dots |l_d\rangle. \quad (3.95)$$

To determine the transformation between $c_{\mathbf{k}}^{(j)}$ and $c_{\mathbf{k}}$, we can make use of the differential properties of Fourier and Chebyshev series, namely

$$\frac{d}{dx} e^{ik\pi x} = ik\pi e^{ik\pi x} \quad (3.96)$$

and

$$2T_k(t) = \frac{T_{k+1}^\theta(t)}{k+1} - \frac{T_{k-1}^\theta(t)}{k-1}, \quad (3.97)$$

respectively. We have

$$c_{\mathbf{k}}^{(j)} = \sum_{\mathbf{k} \mathbf{k} k_1} [D_n^{(j)}]_{\mathbf{k} \mathbf{k} k_1} c_{\mathbf{k}}, \quad \|\mathbf{k}\|_1 \leq n, \quad (3.98)$$

where $D_n^{(j)}$ can be expressed as the tensor product

$$D_n^{(j)} = D_n^{j_1} \otimes D_n^{j_2} \otimes \dots \otimes D_n^{j_d}, \quad (3.99)$$

with $\mathbf{j} = (j_1, \dots, j_d)$. The matrix D_n for the Fourier basis functions in (3.59) can be written as the $(n+1) \times (n+1)$ diagonal matrix with entries

$$[D_n]_{kk} = i(k - \lfloor n/2 \rfloor)\pi. \quad (3.100)$$

As detailed in Appendix A of Reference [54], the matrix D_n for the Chebyshev polynomials in (3.59) can be expressed as the $(n+1) \times (n+1)$ upper triangular matrix with nonzero entries

$$[D_n]_{kr} = \frac{2r}{\sigma_k}, \quad k+r \text{ odd}, r > k, \quad (3.101)$$

where

$$\sigma_k := \begin{cases} 2 & k = 0 \\ 1 & k \in [n]. \end{cases} \quad (3.102)$$

Substituting (3.99) into (3.95), with D_n defined by (3.100) in the periodic case or (3.101) in the non-periodic case, and performing the multi-dimensional inverse QSFT/QCT (for a reason that will be explained in the next section), we obtain the following linear equations for c_r :

$$\sum_{k_j k_1=2} A_j \sum_{k k k_1, k l k_1, k r k_1} [D_n^{(j)}]_{k r} c_r |l_1\rangle \dots |l_d\rangle = \sum_{k k k_1, k l k_1} \hat{f}_k |l_1\rangle \dots |l_d\rangle. \quad (3.103)$$

Notice that the matrices (3.100) and (3.101) are not full rank. More specifically, there exists at least one zero row in the matrix of (3.103) when using either (3.100) ($k = \lfloor n/2 \rfloor$) or (3.101) ($k = n$). To obtain an invertible linear system, we next introduce the boundary conditions.

3.4.2.2 Adding the linear system from the boundary conditions

When we use the form (3.82) of $u(\mathbf{x})$ to write linear equations describing the boundary conditions (3.93), we obtain a non-sparse linear system. Thus, for each $\mathbf{x} \in \partial D_j$ in (3.93), we perform the $(d - 1)$ -dimensional inverse QSFT/QCT on the $d - 1$ registers except the j th register to obtain the linear equations

$$\sum_{\substack{k\mathbf{k}k_1 \dots k_n \\ k_j=n}} c_{\mathbf{k}} |k_1\rangle \dots |k_d\rangle = \sum_{\substack{k\mathbf{k}k_1 \dots k_n \\ k_j=n}} \hat{\gamma}_{\mathbf{k}}^{j+} |k_1\rangle \dots |k_d\rangle, \quad \hat{\gamma}_{\mathbf{k}}^{j+} \in \partial D_j, \quad (3.104)$$

$$\sum_{\substack{k\mathbf{k}k_1 \dots k_n \\ k_j=n-1}} (-1)^{k_j} c_{\mathbf{k}} |k_1\rangle \dots |k_d\rangle = \sum_{\substack{k\mathbf{k}k_1 \dots k_n \\ k_j=n-1}} \hat{\gamma}_{\mathbf{k}}^j |k_1\rangle \dots |k_d\rangle, \quad \hat{\gamma}_{\mathbf{k}}^j \in \partial D_j$$

for all $j \in [d]$, where the values of k_j indicate that we place these constraints in the last two rows with respect to the j th coordinate. We combine these equations with (3.103) to obtain the linear system

$$\sum_{k_j k_1=2} A_j \sum_{\substack{k\mathbf{k}k_1 \dots k_r k_1 \dots k_n \\ k_r k_1}} [\overline{D}_n^{(j)}]_{k_r c_r} |k_1\rangle \dots |k_d\rangle = \sum_{k\mathbf{k}k_1} \sum_{n \ j=1}^d (A_{j,j} \hat{\gamma}_{\mathbf{k}}^{j+} + A_{j,j} \hat{\gamma}_{\mathbf{k}}^j + \hat{f}_{\mathbf{k}}) |k_1\rangle \dots |k_d\rangle, \quad (3.105)$$

where

$$\overline{D}_n^{(j)} = \begin{cases} D_n^{(j)} + G_n^{(j)}, & \|\mathbf{j}\|_1 = 2, \|\mathbf{j}\|_7 = 2; \\ D_n^{(j)}, & \|\mathbf{j}\|_1 = 2, \|\mathbf{j}\|_7 = 1 \end{cases} \quad (3.106)$$

with $G_n^{(j)}$ defined below. In other words, $\overline{D}_n^{(j)} = D_n^{(j)} + G_n^{(j)}$ for each \mathbf{j} that has exactly one entry equal to 2 and all other entries 0, whereas $\overline{D}_n^{(j)} = D_n^{(j)}$ for each \mathbf{j} that has exactly two entries equal to 1 and all other entries 0. Here $G_n^{(j)}$ can be expressed as the

tensor product

$$\mathbf{G}_n^{(j)} = I^{r-1} \otimes G_n \otimes I^{d-r} \quad (3.107)$$

where the r th entry of \mathbf{j} is 2 and all other entries are 0. For the Fourier case in (3.59) used for periodic boundary conditions, D_n comes from (3.100), and the nonzero entries of G_n are

$$[G_n]_{bn/2c,k} = 1, \quad k \in [n+1]_0. \quad (3.108)$$

Alternatively, for the Chebyshev case in (3.59) used for non-periodic boundary conditions, D_n comes from (3.101), and the nonzero entries of G_n are

$$\begin{aligned} [G_n]_{n,k} &= 1, & k \in [n+1]_0, \\ [G_n]_{n-1,k} &= (-1)^k, & k \in [n+1]_0. \end{aligned} \quad (3.109)$$

The system (3.105) has the form of (3.90). For instance, the matrix in (3.90) for Poisson's equation (3.6) is

$$L_{\text{Poisson}} := \overline{\mathbf{D}}_n^{(2,0,\dots,0)} + \overline{\mathbf{D}}_n^{(0,2,\dots,0)} + \dots + \overline{\mathbf{D}}_n^{(0,0,\dots,2)} \quad (3.110)$$

$$= \bigoplus_{j=1}^d \overline{\mathbf{D}}_n^{(2)} = \overline{\mathbf{D}}_n^{(2)} \otimes I^{d-1} + I \otimes \overline{\mathbf{D}}_n^{(2)} \otimes I^{d-2} + \dots + I^{d-1} \otimes \overline{\mathbf{D}}_n^{(2)}. \quad (3.111)$$

For periodic boundary conditions, using (3.98), (3.100), and (3.108), the second-order

differential matrix $\overline{D}_n^{(2)}$ has nonzero entries

$$\begin{aligned} [\overline{D}_n^{(2)}]_{k,k} &= -((k - \lfloor n/2 \rfloor)\pi)^2, \quad k \in [n+1]_0 \setminus \{\lfloor n/2 \rfloor\}, \\ [\overline{D}_n^{(2)}]_{bn/2c,k} &= 1, \quad k \in [n+1]_0. \end{aligned} \quad (3.112)$$

For non-periodic boundary conditions, using (3.98), (3.101), and (3.109), $\overline{D}_n^{(2)}$ has nonzero entries

$$\begin{aligned} [\overline{D}_n^{(2)}]_{kr} &= \sum_{\substack{l=k+1 \\ k+l \text{ odd} \\ l+r \text{ odd}}}^{r-1} [D_n]_{kl} [D_n]_{lr} = \frac{2r}{\sigma_k} \sum_{\substack{l=k+1 \\ k+l \text{ odd} \\ l+r \text{ odd}}}^{r-1} \frac{2l}{\sigma_l} = \frac{r(r^2 - k^2)}{\sigma_k}, \quad k+r \text{ even}, r > k+1, \\ [\overline{D}_n^{(2)}]_{n,k} &= 1, \quad k \in [n+1]_0, \\ [\overline{D}_n^{(2)}]_{n-1,k} &= (-1)^k, \quad k \in [n+1]_0. \end{aligned} \quad (3.113)$$

We discuss the invertible linear system (3.105) and upper bound its condition number in the following section.

3.4.3 Condition number

We now analyze the condition number of the linear system. We begin with two lemmas bounding the singular values of the matrices (3.112) and (3.113) that appear in the linear system.

Lemma 3.7. *Consider the case of periodic boundary conditions. Then for $n \geq 4$, the*

largest and smallest singular values of $\overline{D}_n^{(2)}$ defined in (3.112) satisfy

$$\begin{aligned}\sigma_{\max}(\overline{D}_n^{(2)}) &\leq (2n)^{2.5}, \\ \sigma_{\min}(\overline{D}_n^{(2)}) &\geq \frac{1}{\sqrt{2}}.\end{aligned}\tag{3.114}$$

Proof. By direct calculation of the l_1 norm (i.e., the maximum absolute column sum) of (3.112), for $n \geq 4$, we have

$$\|\overline{D}_n^{(2)}\|_1 \leq \left(\frac{(n+1)\pi}{2}\right)^2 \leq (2n)^2.\tag{3.115}$$

Then the inverse of the matrix (3.112) is

$$\begin{aligned}[(\overline{D}_n^{(2)})^{-1}]_{k,k} &= -\frac{1}{((k - \lfloor n/2 \rfloor)\pi)^2}, & k \in [n+1]_0 \setminus \{\lfloor n/2 \rfloor\}, \\ [(\overline{D}_n^{(2)})^{-1}]_{bn/2c,k} &= \frac{1}{((k - \lfloor n/2 \rfloor)\pi)^2}, & k \in [n+1]_0\end{aligned}\tag{3.116}$$

as can easily be verified by a direct calculation.

By direct calculation of the Frobenius norm of (3.112), we have

$$\|(\overline{D}_n^{(2)})^{-1}\|_F^2 \leq 1 + 2 \sum_{k=1}^{\lfloor n/2 \rfloor} \frac{1}{k^4 \pi^4} = 1 + \frac{2}{\pi^4} \frac{\pi^4}{90} \leq 2.\tag{3.117}$$

Thus we have the result in (3.114):

$$\begin{aligned}\sigma_{\max}(\overline{D}_n^{(2)}) &\leq \sqrt{n+1} \|\overline{D}_n^{(2)}\|_1 \leq (2n)^{2.5}, \\ \sigma_{\min}(\overline{D}_n^{(2)}) &\geq \frac{1}{\|(\overline{D}_n^{(2)})^{-1}\|_F} \geq \frac{1}{\sqrt{2}}\end{aligned}\tag{3.118}$$

as claimed. □

Lemma 3.8. *Consider the case of non-periodic boundary conditions. Then the largest and smallest singular values of $\overline{D}_n^{(2)}$ defined in (3.113) satisfy*

$$\begin{aligned}\sigma_{\max}(\overline{D}_n^{(2)}) &\leq n^4, \\ \sigma_{\min}(\overline{D}_n^{(2)}) &\geq \frac{1}{16}.\end{aligned}\tag{3.119}$$

Proof. By direct calculation of the Frobenius norm of (3.113), we have

$$\|\overline{D}_n^{(2)}\|_F^2 \leq n^2 \max_{k,r} \left(\frac{r(r^2 - k^2)}{\sigma_k} \right)^2 \leq n^2 \cdot n^6 = n^8.\tag{3.120}$$

Next we upper bound $\|(\overline{D}_n^{(2)})^{-1}\|$. By definition,

$$\|(\overline{D}_n^{(2)})^{-1}\| = \sup_{\|b\| \leq 1} \|(\overline{D}_n^{(2)})^{-1}b\|.\tag{3.121}$$

Given any vector b satisfying $\|b\| \leq 1$, we estimate $\|x\|$ defined by the full-rank linear system

$$\overline{D}_n^{(2)}x = b.\tag{3.122}$$

Notice that $\overline{D}_n^{(2)}$ is the sum of the upper triangular matrix $D_n^{(2)}$ and (3.109), the coordinates x_2, \dots, x_n are only defined by coordinates b_0, \dots, b_{n-2} . So we only focus on the partial system

$$D_n^{(2)}[0, 0, x_2, \dots, x_n]^T = [b_0, \dots, b_{n-2}, 0, 0]^T.\tag{3.123}$$

Given the same b , we also define the vector y by

$$D_n[0, y_1, \dots, y_{n-1}, 0]^T = [b_0, \dots, b_{n-2}, 0, 0]^T, \quad (3.124)$$

where each coordinate of y can be expressed by

$$b_k = \sum_{l=1}^{n-1} [D_n]_{kl} y_l = \sum_{l=1}^{n-1} \frac{2l}{\sigma_k} y_l, \quad k+l \text{ odd}, \quad l > k, \quad k \in [n-1]_0. \quad (3.125)$$

Using this equation with $k = l - 1$ and $k = l + 1$, we can express y_l in terms of b_{l-1} and b_{l+1} :

$$\frac{2l}{\sigma_{l-1}} y_l = b_{l-1} - \frac{1}{\sigma_{l-1}} b_{l+1}, \quad l \in [n-1], \quad (3.126)$$

where we let $b_{n-1} = b_n = 0$. Thus we have

$$\begin{aligned} \sum_{l=1}^{n-1} y_l^2 &= \sum_{l=1}^{n-1} \left(\frac{\sigma_{l-1}}{2l} \left(b_{l-1} - \frac{1}{\sigma_{l-1}} b_{l+1} \right) \right)^2 \\ &\leq \sum_{l=1}^{n-1} \frac{\sigma_{l-1}^2}{4l^2} \left(1 + \frac{1}{\sigma_{l-1}^2} \right) (b_{l-1}^2 + b_{l+1}^2) \\ &\leq \frac{5}{4} (b_0^2 + b_2^2) + \frac{1}{16} \sum_{l=2}^{n-2} (b_{l-1}^2 + b_{l+1}^2) \\ &\leq 2 \sum_{l=0}^{n-2} b_l^2. \end{aligned} \quad (3.127)$$

We notice that y also satisfies

$$[0, y_1, \dots, y_{n-1}, 0]^T = D_n[0, 0, x_2, \dots, x_n]^T, \quad (3.128)$$

where each coordinate of y can be expressed by

$$y_l = \sum_{r=1}^n [D_n]_{lr} x_r = \sum_{r=1}^n \frac{2r}{\sigma_l} x_r, \quad l+r \text{ odd}, r > l, l \in [n-1]. \quad (3.129)$$

Substituting the $(r-1)$ st and the $(r+1)$ st equations of (3.129), we can express x in terms of y :

$$\frac{2r}{\sigma_{r-1}} x_r = y_{r-1} - \frac{1}{\sigma_{r-1}} y_{r+1}, \quad r \in [n] \setminus \{1\}, \quad (3.130)$$

where we let $y_n = y_{n+1} = 0$. Similarly, according to (3.130), we also have

$$\sum_{l=2}^n x_l^2 \leq 2 \sum_{l=1}^{n-1} y_l^2. \quad (3.131)$$

Then we calculate $x_0^2 + x_1^2$ based on the last two equations of (3.122), (3.127), and (3.130),

giving

$$\begin{aligned}
x_0^2 + x_1^2 &= \frac{1}{2}[(x_0 + x_1)^2 + (x_0 - x_1)^2] \\
&= \frac{1}{2} \left[\left(b_n - \sum_{l=2}^n x_l \right)^2 + \left(b_{n-1} - \sum_{l=2}^n (-1)^l x_l \right)^2 \right] \\
&= \frac{1}{2} \left[\left(b_n - \sum_{l=2}^n \frac{\sigma_{l-1}}{2l} \left(y_{l-1} - \frac{1}{\sigma_{l-1}} y_{l+1} \right) \right)^2 \right. \\
&\quad \left. + \left(b_{n-1} - \sum_{l=2}^n (-1)^l \frac{\sigma_{l-1}}{2l} \left(y_{l-1} - \frac{1}{\sigma_{l-1}} y_{l+1} \right) \right)^2 \right] \\
&\leq \frac{1}{2} \left(1 + \sum_{l=2}^n \frac{\sigma_{l-1}^2}{4l^2} \right) \left[b_n^2 + \sum_{l=2}^n \left(y_{l-1} - \frac{1}{\sigma_{l-1}} y_{l+1} \right)^2 + b_{n-1}^2 \right. \\
&\quad \left. + \sum_{l=2}^n \left(y_{l-1} - \frac{1}{\sigma_{l-1}} y_{l+1} \right)^2 \right] \\
&\leq \frac{1}{2} \left(1 + \frac{1}{4} \sum_{l=2}^n \frac{1}{l^2} \right) \left[b_n^2 + b_{n-1}^2 + \sum_{l=2}^n \left(1 + \frac{1}{\sigma_{l-1}^2} \right) (y_{l-1}^2 + y_{l+1}^2) \right] \\
&\leq \frac{1}{2} \left(1 + \frac{\pi^2}{24} \right) \left[b_n^2 + b_{n-1}^2 + 4 \sum_{l=1}^{n-1} y_l^2 \right] \\
&\leq b_n^2 + b_{n-1}^2 + 8 \sum_{l=0}^{n-2} b_l^2.
\end{aligned} \tag{3.132}$$

Thus, based on (3.127), (3.131), and (3.132), the inequality

$$\begin{aligned}
\sum_{l=0}^n x_l^2 &= x_0^2 + x_1^2 + \sum_{l=2}^n x_l^2 \\
&\leq b_n^2 + b_{n-1}^2 + 8 \sum_{l=0}^{n-2} b_l^2 + 4 \sum_{l=0}^{n-2} b_l^2 \\
&\leq b_n^2 + b_{n-1}^2 + 12 \sum_{l=0}^{n-2} b_l^2 \leq 12
\end{aligned} \tag{3.133}$$

holds for any vectors b satisfying $\|b\| \leq 1$. Thus

$$\|(\overline{D}_n^{(2)})^{-1}\| = \sup_{\|x\|=1} \|x\| \leq 12 < 16. \quad (3.134)$$

Altogether, we have

$$\begin{aligned} \sigma_{\max}(\overline{D}_n^{(2)}) &\leq \|\overline{D}_n^2\|_F \leq n^4, \\ \sigma_{\min}(\overline{D}_n^{(2)}) &\geq \frac{1}{\|(\overline{D}_n^2)^{-1}\|} \geq \frac{1}{16} \end{aligned} \quad (3.135)$$

as claimed in (3.119). □

Using these two lemmas, we first upper bound the condition number of the linear system for Poisson's equation, and then extend the result to general elliptic PDEs.

For the case of the Poisson equation, we use the following simple bounds on the extreme singular values of a Kronecker sum.

Lemma 3.9. *Let*

$$L = \bigoplus_{j=1}^d M_j = M_1 \otimes I^{d-1} + I \otimes M_2 \otimes I^{d-2} + \dots + I^{d-1} \otimes M_d, \quad (3.136)$$

where $\{M_j\}_{j=1}^d$ are square matrices. If the largest and smallest singular values of M_j satisfy

$$\begin{aligned} \sigma_{\max}(M_j) &\leq s_j^{\max}, \\ \sigma_{\min}(M_j) &\geq s_j^{\min}, \end{aligned} \quad (3.137)$$

respectively, then the condition number of L satisfies

$$\kappa_L \leq \frac{\sum_{j=1}^d s_j^{\max}}{\sum_{j=1}^d s_j^{\min}}. \quad (3.138)$$

Proof. We bound the singular values of the matrix exponential $\exp(M_j)$ by

$$\begin{aligned} \sigma_{\max}(\exp(M_j)) &\leq e^{s_j^{\max}}, \\ \sigma_{\min}(\exp(M_j)) &\geq e^{s_j^{\min}} \end{aligned} \quad (3.139)$$

using (3.137). The singular values of the Kronecker product $\bigotimes_{j=1}^d \exp(M_j)$ are

$$\sigma_{k_1, \dots, k_d} \left(\bigotimes_{j=1}^d \exp(M_j) \right) = \prod_{j=1}^d \sigma_{k_j}(\exp(M_j)) \quad (3.140)$$

where $\sigma_{k_j}(\exp(M_j))$ are the singular values of the matrix $\exp(M_j)$ for each $j \in [d]$, where k_j runs from 1 to the dimension of M_j . Using the property of the Kronecker sum that

$$\exp(L) = \exp \left(\bigoplus_{j=1}^d M_j \right) = \bigotimes_{j=1}^d \exp(M_j), \quad (3.141)$$

we bound the singular values of the matrix exponential of (3.111) by

$$\begin{aligned} \sigma_{\max}(\exp(L)) &\leq e^{\sum_{j=1}^d s_j^{\max}}, \\ \sigma_{\min}(\exp(L)) &\geq e^{\sum_{j=1}^d s_j^{\min}}. \end{aligned} \quad (3.142)$$

Finally, we bound the singular values of the matrix logarithm of (3.142) by

$$\begin{aligned}\sigma_{\max}(L) &\leq \sum_{j=1}^d s_j^{\max}, \\ \sigma_{\min}(L) &\geq \sum_{j=1}^d s_j^{\min}.\end{aligned}\tag{3.143}$$

Thus the condition number of L satisfies

$$\kappa_L \leq \frac{\sum_{j=1}^d s_j^{\max}}{\sum_{j=1}^d s_j^{\min}}\tag{3.144}$$

as claimed. □

This lemma easily implies a bound on the condition number of the linear system for Poisson's equation:

Corollary 3.1. *Consider an instance of the quantum PDE problem as defined in [Problem 3.1](#) for Poisson's equation (3.6) with Dirichlet boundary conditions (3.81). Then for $n \geq 4$, the condition number of L_{Poisson} in the linear system (3.90) satisfies*

$$\kappa_{L_{\text{Poisson}}} \leq (2n)^4.\tag{3.145}$$

Proof. The matrix in (3.90) for Poisson's equation (3.6) is L_{Poisson} defined in (3.111). For both the periodic and the non-periodic case, we have

$$\begin{aligned}\sigma_{\max}(\overline{D}_n^{(2)}) &\leq n^4, \\ \sigma_{\min}(\overline{D}_n^{(2)}) &\geq \frac{1}{16}\end{aligned}\tag{3.146}$$

by [Lemma 3.7](#) and [Lemma 3.8](#). Let $M_j = \overline{D}_n^{(2)}$ for $j \in [d]$ in (3.136), and apply [Lemma 3.9](#) with $s_j^{\max} = n^4$ and $s_j^{\min} = 1/16$ in (3.138). Then the condition number of L_{Poisson} is bounded by

$$\kappa_{L_{\text{Poisson}}} \leq \frac{\sigma_{\max}(\overline{D}_n^{(2)})}{\sigma_{\min}(\overline{D}_n^{(2)})} \leq (2n)^4 \quad (3.147)$$

as claimed. \square

We now consider the condition number of the linear system for general elliptic PDEs.

Lemma 3.10. *Consider an instance of the quantum PDE problem as defined in [Problem 3.1](#) with Dirichlet boundary conditions (3.81). Then for $n \geq 4$, the condition number of L in the linear system (3.90) satisfies*

$$\kappa_L \leq \frac{\|A\|_{\Sigma}}{C\|A\|} (2n)^4, \quad (3.148)$$

where $\|A\|_{\Sigma} := \sum_{k_j k_1=2} |A_j| = \sum_{j_1, j_2=1}^d |A_{j_1, j_2}|$, $\|A\| := \sum_{j=1}^d |A_{j, j}|$, and $C > 0$ is defined in (3.8).

Recall that C quantifies the extent to which the global strict diagonal dominance condition holds.

Proof. According to (3.105), the matrix in (3.90) is

$$L = \sum_{k_j k_1=2} A_j \overline{D}_n^{(j)}. \quad (3.149)$$

We upper bound the spectral norm of the matrix L by

$$\|L\| \leq \sum_{kj k_1=2} |A_j| \|\overline{D}_n^{(j)}\|. \quad (3.150)$$

For the matrix $\overline{D}_n^{(j)}$ defined by (3.106), Lemma 3.7 (in the periodic case) and Lemma 3.8 (in the non-periodic case) give the inequality

$$\|\overline{D}_n^{(j)}\| \leq n^4, \quad (3.151)$$

so we have

$$\|L\| \leq \sum_{kj k_1=2} |A_j| n^4 = \|A\|_{\Sigma} n^4. \quad (3.152)$$

Next we lower bound $\|L\xi\|$ for any $\|\xi\| = 1$.

It is non-trivial to directly compute the singular values of a sum of non-normal matrices. Instead, we write L as a sum of terms L_1 and L_2 , where L_1 is a tensor sum similar to (3.111) that can be bounded by Lemma 3.9, and L_2 is a sum of tensor products that are easily bounded. Specifically, we have

$$\begin{aligned} L_1 &= A_{1,1} \overline{D}_n^{(2)} \otimes I^{d-1} + \dots + A_{d,d} I^{d-1} \otimes \overline{D}_n^{(2)} \\ L_2 &= L - L_1. \end{aligned} \quad (3.153)$$

The ellipticity condition (3.7), $\forall \xi \sum_{kj k_1=h} A_j(\mathbf{x}) \xi^j \neq 0$, can only hold if the $A_{j,j}$ for $j \in [d]$ are either all positive or all negative; we consider $A_{j,j} > 0$ without loss of

generality, so

$$\|A\| = \sum_{j=1}^d |A_{j,j}| = \sum_{j=1}^d A_{j,j}. \quad (3.154)$$

Also, the global strict diagonal dominance condition (3.8) simplifies to

$$C = 1 - \sum_{j_1=1}^d \frac{1}{A_{j_1,j_1}} \sum_{j_2 \in [d] \setminus \{j_1\}} |A_{j_1,j_2}| > 0, \quad (3.155)$$

where $0 < C \leq 1$.

We now upper bound $\|L_2 L_1^{-1}\|$ by bounding $\|D_n^{(j)} L_1^{-1}\|$ for each $\mathbf{j} = (j_1, \dots, j_d)$ that has exactly two entries equal to 1 and all other entries 0. Specifically, consider $j_{r_1} = j_{r_2} = 1$ for $r_1, r_2 \in [d]$, $r_1 \neq r_2$, and $j_r = 0$ for $r \in [d] \setminus \{r_1, r_2\}$. We denote

$$L^{(j)} := I^{r_1-1} \otimes D_n^2 \otimes I^{d-r_1} + I^{r_2-1} \otimes D_n^2 \otimes I^{d-r_2}. \quad (3.156)$$

We first upper bound $\|D_n^{(j)}\|$ by $\frac{1}{2}\|L^{(j)}\|$. Notice the matrices $D_n^{(j)}$ and $L^{(j)}$ share the same singular vectors. For $k \in [n+1]_0$, we let v_k and λ_k denote the right singular vectors and corresponding singular values of D_n , respectively. Then the right singular vectors of $D_n^{(j)}$ and $L^{(j)}$ are $\mathbf{v}_k := \bigotimes_{j=1}^d v_{k_j}$, where $\mathbf{k} = (k_1, \dots, k_d)$ with $k_j \in [n+1]_0$ for $j \in [d]$. For any vector $v = \sum_{\mathbf{k} \in [n+1]_0^d} \alpha_{\mathbf{k}} \mathbf{v}_k$, we have

$$\|D_n^{(j)} v\|^2 = \sum_{\mathbf{k} \in [n+1]_0^d} |\alpha_{\mathbf{k}}|^2 \|D_n^{(j)} \mathbf{v}_k\|^2 = \sum_{\mathbf{k} \in [n+1]_0^d} |\alpha_{\mathbf{k}}|^2 (\lambda_{k_{j_{r_1}}} \lambda_{k_{j_{r_2}}})^2, \quad (3.157)$$

$$\|L^{(j)} v\|^2 = \sum_{\mathbf{k} \in [n+1]_0^d} |\alpha_{\mathbf{k}}|^2 \|L^{(j)} \mathbf{v}_k\|^2 = \sum_{\mathbf{k} \in [n+1]_0^d} |\alpha_{\mathbf{k}}|^2 (\lambda_{k_{j_{r_1}}}^2 + \lambda_{k_{j_{r_2}}}^2)^2, \quad (3.158)$$

which implies $\|D_n^{(j)} v\| \leq \frac{1}{2}\|L^{(j)} v\|$ by the inequality of arithmetic and geometric means

(also known as the AM-GM inequality). Since this holds for any vector v , we have

$$\|D_n^{(j)} L_1^{-1}\| \leq \frac{1}{2} \|L^{(j)} L_1^{-1}\|. \quad (3.159)$$

Next we upper bound $\|D_n^2\|$ by $\|\bar{D}_n^{(2)}\|$. For any vector $u = [u_0, \dots, u_n]^T$, define two vectors $w = [w_0, \dots, w_n]^T$ and $\bar{w} = [\bar{w}_0, \dots, \bar{w}_n]^T$ such that

$$D_n^2[u_0, \dots, u_n]^T = [w_0, \dots, w_n]^T \quad (3.160)$$

and

$$\bar{D}_n^2[u_0, \dots, u_n]^T = [\bar{w}_0, \dots, \bar{w}_n]^T. \quad (3.161)$$

Notice that $w_{bn/2c} = 0$ and $w_k = \bar{w}_k$ for $k \in [n+1]_0 \setminus \{[n/2]\}$ for periodic conditions, and $w_{n-1} = w_n = 0$ and $w_k = \bar{w}_k$ for $k \in [n+1]_0 \setminus \{n-1, n\}$ for non-periodic conditions. Thus, for any vector v ,

$$\|D_n^2 v\|^2 = \|w\|^2 = \sum_{k=0}^n w_k^2 \leq \sum_{k=0}^n \bar{w}_k^2 = \|\bar{w}\|^2 = \|\bar{D}_n^{(2)} v\|^2. \quad (3.162)$$

Therefore,

$$\|L^{(j)} L_1^{-1}\| \leq \sum_{s=1}^2 \|I^{r_s-1} \otimes D_n^2 \otimes I^{d-r_s} L_1^{-1}\| \leq \sum_{s=1}^2 \|I^{r_s-1} \otimes \bar{D}_n^{(2)} \otimes I^{d-r_s} L_1^{-1}\|. \quad (3.163)$$

We also have

$$\|D_n^{(j)} L_1^{-1}\| \leq \frac{1}{2} \sum_{s=1}^2 \|I^{r_s-1} \otimes \bar{D}_n^{(2)} \otimes I^{d-r_s} L_1^{-1}\|. \quad (3.164)$$

We can rewrite $I^{r_s-1} \otimes \overline{D}_n^{(2)} \otimes I^{d-r_s} L_1^{-1}$ in the form

$$I^{r_s-1} \otimes \overline{D}_n^{(2)} \otimes I^{d-r_s} \left(\sum_{h=1}^d A_{h,h} I^{r_h-1} \otimes \overline{D}_n^{(2)} \otimes I^{d-r_h} \right)^{-1}. \quad (3.165)$$

The matrices $I^{r_h-1} \otimes \overline{D}_n^{(2)} \otimes I^{d-r_h}$ share the same singular values and singular vectors,

so

$$\|I^{r_s-1} \otimes \overline{D}_n^{(2)} \otimes I^{d-r_s} L_1^{-1}\| = \max_{\bar{\lambda}_{k_r}} \frac{\bar{\lambda}_{k_{r_s}}}{\sum_{h=1}^d A_{h,h} \bar{\lambda}_{k_h}} < \frac{1}{A_{r_s,r_s}}, \quad (3.166)$$

where $\bar{\lambda}_{k_h}$ are singular values of $I^{r_h-1} \otimes \overline{D}_n^{(2)} \otimes I^{d-r_h}$ for $k_h \in [n]_0$, $h \in [d]$. This implies

$$\|\mathbf{D}_n^{(j)} L_1^{-1}\| \leq \frac{1}{2} \left(\frac{1}{A_{r_1,r_1}} + \frac{1}{A_{r_2,r_2}} \right). \quad (3.167)$$

Using (3.155), considering each instance of $\mathbf{D}_n^{(j)}$ in L_2 , we have

$$\|L_2 L_1^{-1}\| \leq \sum_{j_1 \neq j_2} |A_{j_1,j_2}| \|\mathbf{D}_n^{(j)} L_1^{-1}\| \leq \sum_{j_1=1}^d \frac{1}{A_{j_1,j_1}} \sum_{j_2 \in [d] \setminus \{j_1\}} |A_{j_1,j_2}| \leq 1 - C. \quad (3.168)$$

Since L and L_1 are invertible, $\|L_1^{-1} L_2\| \leq 1 - C < 1$, and by Lemma 3.9 applied to

$\|L_1^{-1}\|$, we have

$$\|L^{-1}\| = \|(L_1 + L_2)^{-1}\| \leq \|(I + L_2 L_1^{-1})^{-1}\| \|L_1^{-1}\| \leq \frac{\|L_1^{-1}\|}{1 - \|L_2 L_1^{-1}\|} \leq \frac{1/\frac{1}{16} \|A\|}{C} = \frac{16}{C \|A\|}. \quad (3.169)$$

Thus we have

$$\kappa_L = \|L\| \|L^{-1}\| \leq \frac{\|A\|_{\Sigma}}{C \|A\|} (2n)^4 \quad (3.170)$$

as claimed. □

3.4.4 State preparation

We now describe a state preparation procedure for the vector $\mathbf{f} + \mathbf{g}$ in the linear system (3.90).

Lemma 3.11. *Let O_f be a unitary oracle that maps $|0\rangle|0\rangle$ to a state proportional to $|0\rangle|f\rangle$, and $|\phi\rangle|0\rangle$ to $|\phi\rangle|0\rangle$ for any $|\phi\rangle$ orthogonal to $|0\rangle$; let O_x be a unitary oracle that maps $|0\rangle|0\rangle$ to $|0\rangle|0\rangle$, $|j\rangle|0\rangle$ to a state proportional to $|j\rangle|\gamma^{j+}\rangle$ for $j \in [d]$, and $|j+d\rangle|0\rangle$ to a state proportional to $|j+d\rangle|\gamma^j\rangle$ for $j \in [d]$. Suppose $\| |f\rangle \|, \| |\gamma^{j+}\rangle \|, \| |\gamma^j\rangle \|$ and $A_{j,j}$ for $j \in [d]$ are known. Define the parameter*

$$q := \sqrt{\frac{\sum_{\mathbf{k}} \sum_{k_1} \sum_{j=1}^d [\hat{f}_{\mathbf{k}}^2 + (A_{j,j} \hat{\gamma}_{\mathbf{k}}^{j+})^2 + (A_{j,j} \hat{\gamma}_{\mathbf{k}}^j)^2]}{\sum_{\mathbf{k}} \sum_{k_1} \sum_{j=1}^d |\hat{f}_{\mathbf{k}} + A_{j,j} \hat{\gamma}_{\mathbf{k}}^{j+} + A_{j,j} \hat{\gamma}_{\mathbf{k}}^j|^2}}. \quad (3.171)$$

Then the normalized quantum state

$$|B\rangle \propto \sum_{\mathbf{k}} \sum_{k_1} \sum_{j=1}^d (\hat{f}_{\mathbf{k}} + A_{j,j} \hat{\gamma}_{\mathbf{k}}^{j+} + A_{j,j} \hat{\gamma}_{\mathbf{k}}^j) |k_1\rangle \dots |k_d\rangle, \quad (3.172)$$

with coefficients defined as in (3.88) and (3.89), can be prepared with gate and query complexity $O(qd^2 \log n \log \log n)$.

Proof. Starting from the initial state $|0\rangle|0\rangle$, we first perform a unitary transformation U

satisfying

$$\begin{aligned}
U|0\rangle &= \frac{\|f\rangle\|}{\sqrt{\|f\rangle\|^2 + \sum_{j=1}^d (A_{j,j}^2 \|\gamma^{j+}\rangle\|^2 + A_{j,j}^2 \|\gamma^j\rangle\|^2)} |0\rangle \\
&+ \sum_{j=1}^d \frac{A_{j,j} \|\gamma^{j+}\rangle\|}{\sqrt{\|f\rangle\|^2 + \sum_{j=1}^d (A_{j,j}^2 \|\gamma^{j+}\rangle\|^2 + A_{j,j}^2 \|\gamma^j\rangle\|^2)} |j\rangle \\
&+ \sum_{j=1}^d \frac{A_{j,j} \|\gamma^j\rangle\|}{\sqrt{\|f\rangle\|^2 + \sum_{j=1}^d (A_{j,j}^2 \|\gamma^{j+}\rangle\|^2 + A_{j,j}^2 \|\gamma^j\rangle\|^2)} |j+d\rangle
\end{aligned} \tag{3.173}$$

on the first register to obtain

$$\frac{\|f\rangle\| |0\rangle + A_{1,1} \|\gamma^{1+}\rangle\| |1\rangle + \dots + A_{d,d} \|\gamma^d\rangle\| |2d\rangle}{\sqrt{\|f\rangle\|^2 + \sum_{j=1}^d (A_{j,j}^2 \|\gamma^{j+}\rangle\|^2 + A_{j,j}^2 \|\gamma^j\rangle\|^2)} |0\rangle. \tag{3.174}$$

This can be done in time $O(2d + 1)$ by standard techniques [92]. Then we apply O_x and O_f to obtain

$$\begin{aligned}
&|0\rangle|f\rangle + A_{1,1}|1\rangle|\gamma^{1+}\rangle + \dots + A_{d,d}|2d\rangle|\gamma^d\rangle, \\
&\propto \sum_{k \in k_1, k_1, k_1, \dots, k_1} \phi_k(\chi_l) (\hat{f}_k |0\rangle + A_{1,1} \hat{\gamma}_k^{1+} |1\rangle + \dots + A_{d,d} \hat{\gamma}_k^d |2d\rangle) |l_1\rangle \dots |l_d\rangle,
\end{aligned} \tag{3.175}$$

according to (3.86) and (3.87). We then perform the d -dimensional inverse QSFT (for periodic boundary conditions) or inverse QCT (for non-periodic boundary conditions) on the last d registers, obtaining

$$\sum_{k \in k_1, k_1, k_1, \dots, k_1} (\hat{f}_k |0\rangle + A_{1,1} \hat{\gamma}_k^{1+} |1\rangle + \dots + A_{d,d} \hat{\gamma}_k^d |2d\rangle) |k_1\rangle \dots |k_d\rangle. \tag{3.176}$$

Finally, observe that if we measure the first register in a basis containing the uniform

superposition $|0\rangle + |1\rangle + \dots + |2d\rangle$ (say, the Fourier basis) and obtain the outcome corresponding to the uniform superposition, we produce the state

$$\sum_{\mathbf{k}} \sum_{n=1}^d (\hat{f}_{\mathbf{k}} + A_{j,j} \hat{\gamma}_{\mathbf{k}}^{j+} + A_{j,j} \hat{\gamma}_{\mathbf{k}}^j) |k_1\rangle \dots |k_d\rangle. \quad (3.177)$$

Since this outcome occurs with probability $1/q^2$, we can prepare this state with probability close to 1 using $O(q)$ steps of amplitude amplification. According to [Lemma 3.5](#) and [Lemma 3.6](#), the d -dimensional (inverse) QSFT or QCT can be performed with gate complexity $O(d \log n \log \log n)$. Thus the total gate and query complexity is $O(qd^2 \log n \log \log n)$. \square

Alternatively, if it is possible to directly prepare the quantum state $|B\rangle$, then we may be able to avoid the factor of q in the complexity of the overall algorithm.

3.4.5 Main result

Having analyzed the condition number and state preparation procedure for our approach, we are now ready to establish the main result. [Theorem 3.2](#) as follows.

Theorem 3.2. *Consider an instance of the quantum PDE problem as defined in [Problem 3.1](#) with Dirichlet boundary conditions [\(3.81\)](#). Then there exists a quantum algorithm that produces a state in the form of [\(3.82\)](#) whose amplitudes are proportional to $u(\mathbf{x})$ on a set of interpolation nodes \mathbf{x} (with respect to the uniform grid nodes for periodic boundary conditions or the Chebyshev-Gauss-Lobatto quadrature nodes for non-periodic boundary conditions, as defined in [\(3.60\)](#)), where $u(\mathbf{x})/\|u(\mathbf{x})\|$ is ϵ -close to $\hat{u}(\mathbf{x})/\|\hat{u}(\mathbf{x})\|$ in l_2*

norm for all nodes \mathbf{x} , succeeding with probability $\Omega(1)$, with a flag indicating success, using

$$\left(\frac{d\|A\|_{\Sigma}}{C\|A\|} + qd^2 \right) \text{poly}(\log(g^\theta/g\epsilon)) \quad (3.54)$$

queries to oracles as defined in [Section 3.4.4](#). Here $\|A\|_{\Sigma} := \sum_{k,j,k_1} \|A_j\|$, $\|A\| := \sum_{j=1}^d |A_{j,j}|$, $C > 0$ is defined in [\(3.8\)](#), and

$$g = \min_{\mathbf{x}} \|\hat{u}(\mathbf{x})\|, \quad g^\theta := \max_{\mathbf{x}} \max_{n \geq \mathbb{N}} \|\hat{u}^{(n+1)}(\mathbf{x})\|, \quad (3.55)$$

$$q = \sqrt{\frac{\sum_{k,k_1} \sum_{j=1}^d \hat{f}_{\mathbf{k}}^2 + (A_{j,j} \hat{\gamma}_{\mathbf{k}}^{j+})^2 + (A_{j,j} \hat{\gamma}_{\mathbf{k}}^j)^2}{\sum_{k,k_1} \sum_{j=1}^d (\hat{f}_{\mathbf{k}} + A_{j,j} \hat{\gamma}_{\mathbf{k}}^{j+} + A_{j,j} \hat{\gamma}_{\mathbf{k}}^j)^2}}. \quad (3.56)$$

The gate complexity is larger than the query complexity by a factor of $\text{poly}(\log(d\|A\|_{\Sigma}/\epsilon))$.

Proof. We analyze the complexity of the algorithm presented in [Section 3.4.2](#).

First we choose

$$n := \left\lceil \frac{\log(\Omega)}{\log(\log(\Omega))} \right\rceil, \quad (3.178)$$

where

$$\Omega = \frac{g^\theta(1+\epsilon)}{g\epsilon}. \quad (3.179)$$

By Eq. (1.8.28) of Reference [\[107\]](#), this choice guarantees

$$\|\hat{u}(\mathbf{x}) - u(\mathbf{x})\| \leq \max_{\mathbf{x}} \|\hat{u}^{(n+1)}(\mathbf{x})\| \frac{e^n}{(2n)^n} \leq \frac{g^\theta}{\Omega} = \frac{g\epsilon}{1+\epsilon} =: \delta. \quad (3.180)$$

Now $\|\hat{u}(\mathbf{x}) - u(\mathbf{x})\| \leq \delta$ implies

$$\left\| \frac{\hat{u}(\mathbf{x})}{\|\hat{u}(\mathbf{x})\|} - \frac{u(\mathbf{x})}{\|u(\mathbf{x})\|} \right\| \leq \frac{\delta}{\min\{\|\hat{u}(\mathbf{x})\|, \|u(\mathbf{x})\|\}} \leq \frac{\delta}{g - \delta} = \epsilon, \quad (3.181)$$

so we can choose n to ensure that the normalized output state is ϵ -close to $\hat{u}(\mathbf{x})/\|\hat{u}(\mathbf{x})\|$.

As described in [Section 3.4.2](#), the algorithm uses the high-precision QLSA from Reference [\[46\]](#) and the multi-dimensional QSFT/QCT (and its inverse). According to [Lemma 3.5](#) and [Lemma 3.6](#), the d -dimensional (inverse) QSFT or QCT can be performed with gate complexity $O(d \log n \log \log n)$. According to [Lemma 3.11](#), the query and gate complexity for state preparation is $O(qd^2 \log n \log \log n)$.

For the linear system $L\mathbf{x} = \mathbf{f} + \mathbf{g}$ in [\(3.90\)](#), the matrix L is an $(n+1)^d \times (n+1)^d$ matrix with $(n+1)$ or $(n+1)d$ nonzero entries in any row or column for periodic or non-periodic conditions, respectively. According to [Lemma 3.10](#), the condition number of L is upper bounded by $\frac{kAK_\Sigma}{CkAK} (2n)^4$. Consequently, by Theorem 5 of Reference [\[46\]](#), the QLSA produces a state proportional to \mathbf{x} with $O(\frac{dKAK_\Sigma}{CkAK} (2n)^5)$ queries to the oracles, and its gate complexity is larger by a factor of $\text{poly}(\log(d\|A\|_\Sigma n))$. Using the value of n specified in [\(3.178\)](#), the overall query complexity of our algorithm is

$$\left(\frac{d\|A\|_\Sigma}{C\|A\|} + qd^2 \right) \text{poly}(\log(g^\ell/g\epsilon)), \quad (3.182)$$

and the gate complexity is

$$\left(\frac{d\|A\|_\Sigma}{C\|A\|} \text{poly}(\log(d\|A\|_\Sigma/\epsilon)) + qd^2 \right) \text{poly}(\log(g^\ell/g\epsilon)) \quad (3.183)$$

which is larger by a factor of $\text{poly}(\log(d\|A\|_{\Sigma}/\epsilon))$, as claimed. \square

Note that we can establish a more efficient algorithm in the special case of the Poisson equation with homogeneous boundary conditions. In this case, $\|A\|_{\Sigma} = \|A\| = d$ and $C = 1$. Under homogeneous boundary conditions, the complexity of state preparation can be reduced to $d \text{poly}(\log(g^{\ell}/g\epsilon))$, since we can remove $2d$ applications of the QSFT or QCT for preparing a state depending on the boundary conditions, and since $\gamma = 0$ there is no need to postselect on the uniform superposition to incorporate the boundary conditions. In summary, the query complexity of the Poisson equation with homogeneous boundary conditions is

$$d \text{poly}(\log(g^{\ell}/g\epsilon)); \tag{3.184}$$

again the gate complexity is larger by a factor of $\text{poly}(\log(d\|A\|_{\Sigma}/\epsilon))$.

3.5 Discussion and open problems

We have presented high-precision quantum algorithms for d -dimensional PDEs using the FDM and spectral methods. These algorithms use high-precision QLSAs to solve Poisson's equation and other second-order elliptic equations. Whereas previous algorithms scaled as $\text{poly}(d, 1/\epsilon)$, our algorithms scale as $\text{poly}(d, \log(1/\epsilon))$.

This work raises several natural open problems. First, for the quantum adaptive FDM, we only deal with Poisson's equation with homogeneous boundary conditions. Can we apply the adaptive FDM to other linear equations or to inhomogeneous boundary conditions? The quantum spectral algorithm applies to second-order elliptic PDEs with Dirichlet boundary conditions. Can we generalize it to other linear PDEs with Neumann

or mixed boundary conditions? Also, can we develop algorithms for space- and time-dependent PDEs? These cases are more challenging since the quantum Fourier transform cannot be directly applied to ensure sparsity. Finally, can we improve the dependence on d ?

Second, the complexity scales logarithmically with high-order derivatives (of the inhomogeneity or solution) for both the adaptive FDM and the spectral method. In particular, [Theorem 3.1](#) shows that the complexity of the quantum adaptive FDM scales logarithmically with $\left| \frac{d^{2k+1}u}{dx^{2k+1}} \right|$, and [Theorem 3.2](#) shows that the complexity of the quantum spectral method is $\text{poly}(\log g^\theta)$, where g^θ upper bounds $\|\hat{u}^{(n+1)}(\mathbf{x})\|$ (see [\(3.55\)](#)). Such a logarithmic dependence on high-order derivatives of the solution is typical for classical algorithms, including the classical adaptive FDM (see for example [Theorem 7](#) of [Reference \[6\]](#)) and spectral methods (see for example [Eq. \(1.8.28\)](#) of [Reference \[107\]](#)), both of which have the same logarithmic dependence on $\left| \frac{d^{2k+1}u}{dx^{2k+1}} \right|$ and g^θ . This logarithmic dependence means that the algorithm is efficient even when faced with a highly oscillatory solution with an exponentially large derivative.

However, the query complexity of time-dependent Hamiltonian simulation only depends on the first-order derivatives of the Hamiltonian [\[34, 35\]](#). Can we develop quantum algorithms for PDEs with query complexity independent of high-order derivatives, and henceforth develop an unexpected advantage of quantum algorithms for PDEs?

Third, can we use quantum algorithms for PDEs as a subroutine of other quantum algorithms? For example, some PDE algorithms have state preparation steps that require inverting finite difference matrices (such as [Reference \[70\]](#) using certain oracles for the initial conditions); are there other scenarios in which state preparation can be done using

the solution of another system of PDEs? Can quantum algorithms for PDEs be applied to other algorithmic tasks, such as optimization?

Finally, how should these algorithms be applied? While PDEs have broad applications, much more work remains to understand the extent to which quantum algorithms can be of practical value. Answering this question will require careful consideration of various technical aspects of the algorithms. In particular: What measurements give useful information about the solutions, and how can those measurements be efficiently implemented? How should the oracles encoding the equations and boundary conditions be implemented in practice? And with these aspects taken into account, what are the resource requirements for quantum computers to solve classically intractable problems related to PDEs?

Chapter 4: Efficient quantum algorithms for dissipative nonlinear differential equations

4.1 Introduction

In this chapter, we study efficient quantum algorithm for dissipative nonlinear differential equations¹. As earlier introduced in [Problem 1.5](#), we focus here on differential equations with nonlinearities that can be expressed with quadratic polynomials, as described in [\(4.1\)](#). Note that polynomials of degree higher than two, and even more general nonlinearities, can be reduced to the quadratic case by introducing additional variables [\[72, 73\]](#). The quadratic case also directly includes many archetypal models, such as the logistic equation in biology, the Lorenz system in atmospheric dynamics, and the Navier–Stokes equations in fluid dynamics.

As discussed in [Chapter 1](#), quantum algorithms offer the prospect of rapidly characterizing the solutions of high-dimensional systems of linear ODEs [\[52, 53, 54\]](#) and PDEs [\[55, 56, 57, 58, 59, 60, 61\]](#). Such algorithms can produce a quantum state proportional to the solution of a sparse (or block-encoded) n -dimensional system of linear differential equations in time $\text{poly}(\log n)$ using the quantum linear system algorithm [\[84\]](#).

Early work on quantum algorithms for differential equations already considered the

¹This chapter is based on the paper [\[81\]](#).

nonlinear case [74]. It gave a quantum algorithm for ODEs that simulates polynomial nonlinearities by storing multiple copies of the solution. The complexity of this approach is polynomial in the logarithm of the dimension but exponential in the evolution time, scaling as $O(1/\epsilon^T)$ due to exponentially increasing resources used to maintain sufficiently many copies of the solution to represent the nonlinearity throughout the evolution.

Recently, heuristic quantum algorithms for nonlinear ODEs have been studied. Reference [75] explores a linearization technique known as the Koopman–von Neumann method that might be amenable to the quantum linear system algorithm. In [76], the authors provide a high-level description of how linearization can help solve nonlinear equations on a quantum computer. However, neither paper makes precise statements about concrete implementations or running times of quantum algorithms. The recent preprint [77] also describes a quantum algorithm to solve a nonlinear ODE by linearizing it using a different approach from the one taken here. However, a proof of correctness of their algorithm involving a bound on the condition number and probability of success is not given. The authors also do not describe how barriers such as those of [78] could be avoided in their approach.

While quantum mechanics is described by linear dynamics, possible nonlinear modifications of the theory have been widely studied. Generically, such modifications enable quickly solving hard computational problems (e.g., solving unstructured search among n items in time $\text{poly}(\log n)$), making nonlinear dynamics exponentially difficult to simulate in general [78, 79, 80]. Therefore, constructing efficient quantum algorithms for general classes of nonlinear dynamics has been considered largely out of reach.

We design and analyze a quantum algorithm that overcomes this limitation using

Carleman linearization [73, 82, 83]. This approach embeds polynomial nonlinearities into an infinite-dimensional system of linear ODEs, and then truncates it to obtain a finite-dimensional linear approximation. The Carleman method has previously been used in the analysis of dynamical systems [111, 112, 113] and the design of control systems [114, 115, 116], but to the best of our knowledge it has not been employed in the context of quantum algorithms. We discretize the finite ODE system in time using the forward Euler method and solve the resulting linear equations with the quantum linear system algorithm [46, 84]. We control the approximation error of this approach by combining a novel convergence theorem with a bound for the global error of the Euler method. Furthermore, we provide an upper bound for the condition number of the linear system and lower bound the success probability of the final measurement. Subject to the condition $R < 1$, where the quantity R (defined in [Problem 4.1](#) below) characterizes the relative strength of the nonlinear and dissipative linear terms, we show that the total complexity of this *quantum Carleman linearization algorithm* is $sT^2q \text{poly}(\log T, \log n, \log 1/\epsilon)/\epsilon$, where s is the sparsity, T is the evolution time, q quantifies the decay of the final solution relative to the initial condition, n is the dimension, and ϵ is the allowed error (see [Theorem 4.1](#)). In the regime $R < 1$, this is an exponential improvement over [74], which has complexity exponential in T .

Note that the solution cannot decay exponentially in T for the algorithm to be efficient, as captured by the dependence of the complexity on q —a known limitation of quantum ODE algorithms [53]. For homogeneous ODEs with $R < 1$, the solution necessarily decays exponentially in time (see equation (4.30)), so the algorithm is not asymptotically efficient. However, even for solutions with exponential decay, we still

find an improvement over the best previous result $O(1/\epsilon^T)$ [74] for sufficiently small ϵ . Thus our algorithm might provide an advantage over classical computation for studying evolution for short times. More significantly, our algorithm can handle inhomogeneous quadratic ODEs, for which it can remain efficient in the long-time limit since the solution can remain asymptotically nonzero (for an explicit example, see the discussion just before the proof of [Lemma 4.2](#)), or can decay slowly (i.e., q can be $\text{poly}(T)$). Inhomogeneous equations arise in many applications, including for example the discretization of PDEs with nontrivial boundary conditions.

We also provide a quantum lower bound for the worst-case complexity of simulating strongly nonlinear dynamics, showing that the algorithm’s condition $R < 1$ cannot be significantly improved in general ([Theorem 4.2](#)). Following the approach of [78, 79], we construct a protocol for distinguishing two states of a qubit driven by a certain quadratic ODE. Provided $R \geq \sqrt{2}$, this procedure distinguishes states with overlap $1 - \epsilon$ in time $\text{poly}(\log(1/\epsilon))$. Since nonorthogonal quantum states are hard to distinguish, this implies a lower bound on the complexity of the quantum ODE problem.

Our quantum algorithm could potentially be applied to study models governed by quadratic ODEs arising in biology and epidemiology as well as in fluid and plasma dynamics. In particular, the celebrated Navier–Stokes equation with linear damping, which describes many physical phenomena, can be treated by our approach provided the Reynolds number is sufficiently small. We also note that while the formal validity of our arguments assumes $R < 1$, we find in one numerical experiment that our proposed approach remains valid for larger R (see [Section 4.6](#)).

We emphasize that, as in related quantum algorithms for linear algebra and differential

equations, instantiating our approach requires an implicit description of the problem that allows for efficient preparation of the initial state and implementation of the dynamics. Furthermore, since the output is encoded in a quantum state, readout is restricted to features that can be revealed by efficient quantum measurements. More work remains to understand how these methods might be applied, as we discuss further in [Section 4.7](#).

The remainder of this chapter is structured as follows. [Section 4.2](#) introduces the quantum quadratic ODE problem. [Section 4.3](#) presents the Carleman linearization procedure and describes its performance. [Section 4.4](#) gives a detailed analysis of the quantum Carleman linearization algorithm. [Section 4.5](#) establishes a quantum lower bound for simulating quadratic ODEs. [Section 4.6](#) describes how our approach could be applied to several well-known ODEs and PDEs and presents numerical results for the case of the viscous Burgers equation. Finally, we conclude with a discussion of the results and some possible future directions in [Section 4.7](#).

4.2 Quadratic ODEs

We focus on an initial value problem described by the n -dimensional quadratic ODE

$$\frac{du}{dt} = F_2 u^2 + F_1 u + F_0(t), \quad u(0) = u_{\text{in}}. \quad (4.1)$$

Here $u = [u_1, \dots, u_n]^T \in \mathbb{R}^n$, $u^2 = [u_1^2, u_1 u_2, \dots, u_1 u_n, u_2 u_1, \dots, u_n u_{n-1}, u_n^2]^T \in \mathbb{R}^{n^2}$, each $u_j = u_j(t)$ is a function of t on the interval $[0, T]$ for $j \in [n] := \{1, \dots, n\}$, $F_2 \in \mathbb{R}^{n \times n^2}$, $F_1 \in \mathbb{R}^{n \times n}$ are time-independent matrices, and the inhomogeneity $F_0(t) \in \mathbb{R}^n$ is a C^1 continuous function of t . We let $\|\cdot\|$ denote the spectral norm.

The main computational problem we consider is as follows.

Problem 4.1. *In the quantum quadratic ODE problem, we consider an n -dimensional quadratic ODE as in (4.1). We assume F_2 , F_1 , and F_0 are s -sparse (i.e., have at most s nonzero entries in each row and column), F_1 is diagonalizable, and that the eigenvalues λ_j of F_1 satisfy $\text{Re}(\lambda_n) \leq \dots \leq \text{Re}(\lambda_1) < 0$. We parametrize the problem in terms of the quantity*

$$\mathbb{R} := \frac{1}{|\text{Re}(\lambda_1)|} \left(\|u_{\text{in}}\| \|F_2\| + \frac{\|F_0\|}{\|u_{\text{in}}\|} \right). \quad (4.2)$$

For some given $T > 0$, we assume the values $\text{Re}(\lambda_1)$, $\|F_2\|$, $\|F_1\|$, $\|F_0(t)\|$ for each $t \in [0, T]$, and $\|F_0\| := \max_{t \in [0, T]} \|F_0(t)\|$, $\|F_0^0\| := \max_{t \in [0, T]} \|F_0^0(t)\|$ are known, and that we are given oracles O_{F_2} , O_{F_1} , and O_{F_0} that provide the locations and values of the nonzero entries of F_2 , F_1 , and $F_0(t)$ for any specified t , respectively, for any desired row or column². We are also given the value $\|u_{\text{in}}\|$ and an oracle O_x that maps $|00 \dots 0\rangle \in \mathbb{C}^n$ to a quantum state proportional to u_{in} . Our goal is to produce a quantum state $|u(T)\rangle$ that is ϵ -close to the normalized $u(T)$ for some given $T > 0$ in ℓ_2 norm.

When $F_0(t) = 0$ (i.e., the ODE is homogeneous), the quantity $\mathbb{R} = \frac{\|u_{\text{in}}\| \|F_2\|}{|\text{Re}(\lambda_1)|}$ is qualitatively similar to the Reynolds number, which characterizes the ratio of the (nonlinear) convective forces to the (linear) viscous forces within a fluid [117, 118]. More generally, \mathbb{R} quantifies the combined strength of the nonlinearity and the inhomogeneity relative to dissipation.

Note that without loss of generality, given a quadratic ODE satisfying (4.1) with

²For instance, F_1 is modeled by a sparse matrix oracle O_{F_1} that, on input (j, l) , gives the location of the l -th nonzero entry in row j , denoted as k , and gives the value $(F_1)_{j,k}$.

$R < 1$, we can modify it by rescaling $u \rightarrow \gamma u$ with a suitable constant γ to satisfy

$$\|F_2\| + \|F_0\| < |\operatorname{Re}(\lambda_1)| \quad (4.3)$$

and

$$\|u_{\text{in}}\| < 1, \quad (4.4)$$

with R left unchanged by the rescaling. We use this rescaling in our algorithm and its analysis. With this rescaling, a small R implies both small $\|u_{\text{in}}\| \|F_2\|$ and small $\|F_0\| / \|u_{\text{in}}\|$ relative to $|\operatorname{Re}(\lambda_1)|$.

4.3 Quantum Carleman linearization

It is challenging to directly simulate quadratic ODEs using quantum computers, and indeed the complexity of the best known quantum algorithm is exponential in the evolution time T [74]. However, for a dissipative nonlinear ODE without a source, any quadratic nonlinear effect will only be significant for a finite time because of the dissipation. To exploit this, we can create a linear system that approximates the initial nonlinear evolution within some controllable error. After the nonlinear effects are no longer important, the linear system properly captures the almost-linear evolution from then on.

We develop such a quantum algorithm using the concept of Carleman linearization [73, 82, 83]. Carleman linearization is a method for converting a finite-dimensional system of nonlinear differential equations into an infinite-dimensional linear one. This is

Note that A is a $(3Ns)$ -sparse matrix. The dimension of (4.5) is

$$\Delta := n + n^2 + \dots + n^N = \frac{n^{N+1} - n}{n - 1} = O(n^N). \quad (4.10)$$

To construct a system of linear equations, we next divide $[0, T]$ into $m = T/h$ time steps and apply the forward Euler method on (4.5), letting

$$y^{k+1} = [I + A(kh)h]y^k + b(kh) \quad (4.11)$$

where $y^k \in \mathbb{R}^\Delta$ approximates $\hat{y}(kh)$ for each $k \in [m+1]_0 := \{0, 1, \dots, m\}$, with $y^0 = y_{\text{in}} := \hat{y}(0) = \hat{y}_{\text{in}}$, and letting all y^k be equal for $k \in [m+p+1]_0 \setminus [m+1]_0$, for some sufficiently large integer p . (It is unclear whether another discretization could improve performance, as discussed further in Section 4.7.) This gives an $(m+p+1)\Delta \times (m+p+1)\Delta$ linear system

$$L|Y\rangle = |B\rangle \quad (4.12)$$

that encodes (4.11) and uses it to produce a numerical solution at time T , where

$$L = \sum_{k=0}^{m+p} |k\rangle\langle k| \otimes I - \sum_{k=1}^m |k\rangle\langle k-1| \otimes [I + A((k-1)h)h] - \sum_{h=m+1}^{m+p} |k\rangle\langle k-1| \otimes I \quad (4.13)$$

and

$$|B\rangle = \frac{1}{\sqrt{B_m}} \left(\|y_{\text{in}}\| |0\rangle \otimes |y_{\text{in}}\rangle + \sum_{k=1}^m \|b((k-1)h)\| |k\rangle \otimes |b((k-1)h)\rangle \right) \quad (4.14)$$

as defined in (4.5). Assume $R < 1$. Let

$$g := \|u(T)\|, \quad q := \frac{\|u_{\text{in}}\|}{\|u(T)\|}. \quad (4.17)$$

There exists a quantum algorithm producing a state that approximates $u(T)/\|u(T)\|$ with error at most $\epsilon \leq 1$, succeeding with probability $\Omega(1)$, with a flag indicating success, using at most

$$\frac{sT^2q[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{(1 - \|u_{\text{in}}\|)^2(\|F_2\| + \|F_0\|)g\epsilon} \text{poly}\left(\log \frac{sT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|}{(1 - \|u_{\text{in}}\|)g\epsilon} / \log(1/\|u_{\text{in}}\|)\right) \quad (4.18)$$

queries to the oracles O_{F_2} , O_{F_1} , O_{F_0} and O_x . The gate complexity is larger than the query complexity by a factor of $\text{poly}(\log(nsT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|/(1 - \|u_{\text{in}}\|)g\epsilon)/\log(1/\|u_{\text{in}}\|))$. Furthermore, if the eigenvalues of F_1 are all real, the query complexity is

$$\frac{sT^2q[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{g\epsilon} \text{poly}\left(\log \frac{sT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|}{g\epsilon} / \log(1/\|u_{\text{in}}\|)\right) \quad (4.19)$$

and the gate complexity is larger by $\text{poly}(\log(nsT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|/g\epsilon)/\log(1/\|u_{\text{in}}\|))$.

4.4 Algorithm analysis

In this section we establish several lemmas and use them to prove [Theorem 4.1](#).

4.4.1 Solution error

The solution error has three contributions: the error from applying Carleman linearization to (4.1), the error in the time discretization of (4.5) by the forward Euler method, and the error from the QLSA. Since the QLSA produces a solution with error at most ϵ with complexity $\text{poly}(\log(1/\epsilon))$ [46], we focus on bounding the first two contributions.

4.4.1.1 Error from Carleman linearization

First, we provide an upper bound for the error from Carleman linearization for arbitrary evolution time T . To the best of our knowledge, the first and only explicit bound on the error of Carleman linearization appears in [73]. However, they only consider homogeneous quadratic ODEs; and furthermore, to bound the error for arbitrary T , they assume the logarithmic norm of F_1 is negative (see Theorems 4.2 and 4.3 of [73]), which is too strong for our case. Instead, we give a novel analysis under milder conditions, providing the first convergence guarantee for general inhomogeneous quadratic ODEs.

We begin with a lemma that describes the decay of the solution of (4.1).

Lemma 4.1. *Consider an instance of the quadratic ODE (4.1), and assume $R < 1$ as defined in (4.2). Let*

$$r_{\pm} := \frac{-\text{Re}(\lambda_1) \pm \sqrt{\text{Re}(\lambda_1)^2 - 4\|F_2\|\|F_0\|}}{2\|F_2\|}. \quad (4.20)$$

Then r_{\pm} are distinct real numbers with $0 \leq r_{-} < r_{+}$, and the solution $u(t)$ of (4.1) satisfies $\|u(t)\| < \|u_{\text{in}}\| < r_{+}$ for any $t > 0$.

Proof. Consider the derivative of $\|u(t)\|$. We have

$$\begin{aligned} \frac{d\|u\|^2}{dt} &= u^y F_2(u \otimes u) + (u^y \otimes u^y) F_2^y u + u^y (F_1 + F_1^y) u + u^y F_0(t) + F_0(t)^y u, \\ &\leq 2\|F_2\| \|u\|^3 + 2\operatorname{Re}(\lambda_1) \|u\|^2 + 2\|F_0\| \|u\|. \end{aligned} \quad (4.21)$$

If $\|u\| \neq 0$, then

$$\frac{d\|u\|}{dt} \leq \|F_2\| \|u\|^2 + \operatorname{Re}(\lambda_1) \|u\| + \|F_0\|. \quad (4.22)$$

Letting $a = \|F_2\| > 0$, $b = \operatorname{Re}(\lambda_1) < 0$, and $c = \|F_0\| > 0$ with $a, b, c > 0$, we consider a 1-dimensional quadratic ODE

$$\frac{dx}{dt} = ax^2 + bx + c, \quad x(0) = \|u_{\text{in}}\|. \quad (4.23)$$

Since $R < 1 \Leftrightarrow b > a\|u_{\text{in}}\| + \frac{c}{\|u_{\text{in}}\|}$, the discriminant satisfies

$$b^2 - 4ac > \left(a\|u_{\text{in}}\| + \frac{c}{\|u_{\text{in}}\|}\right)^2 - 4a\|u_{\text{in}}\| \cdot \frac{c}{\|u_{\text{in}}\|} \geq \left(a\|u_{\text{in}}\| - \frac{c}{\|u_{\text{in}}\|}\right)^2 \geq 0. \quad (4.24)$$

Thus, r_{\pm} defined in (4.20) are distinct real roots of $ax^2 + bx + c$. Since $r_{-} + r_{+} = -\frac{b}{a} > 0$ and $r_{-} r_{+} = \frac{c}{a} \geq 0$, we have $0 \leq r_{-} < r_{+}$. We can rewrite the ODE as

$$\frac{dx}{dt} = ax^2 + bx + c = a(x - r_{-})(x - r_{+}), \quad x(0) = \|u_{\text{in}}\|. \quad (4.25)$$

Letting $y = x - r_-$, we obtain an associated homogeneous quadratic ODE

$$\frac{dy}{dt} = -a(r_+ - r_-)y + ay^2 = ay[y - (r_+ - r_-)], \quad y(0) = \|u_{\text{in}}\| - r_- . \quad (4.26)$$

Since the homogeneous equation has the closed-form solution

$$y(t) = \frac{r_+ - r_-}{1 - e^{a(r_+ - r_-)t} [1 - (r_+ - r_-)/(\|u_{\text{in}}\| - r_-)]}, \quad (4.27)$$

the solution of the inhomogeneous equation can be obtained as

$$x(t) = \frac{r_+ - r_-}{1 - e^{a(r_+ - r_-)t} [1 - (r_+ - r_-)/(\|u_{\text{in}}\| - r_-)]} + r_- . \quad (4.28)$$

Therefore we have

$$\|u(t)\| \leq \frac{r_+ - r_-}{1 - e^{a(r_+ - r_-)t} [1 - (r_+ - r_-)/(\|u_{\text{in}}\| - r_-)]} + r_- . \quad (4.29)$$

Since $R < 1 \Leftrightarrow a\|u_{\text{in}}\| + \frac{c}{k\|u_{\text{in}}\|^k} < -b \Leftrightarrow a\|u_{\text{in}}\|^2 + b\|u_{\text{in}}\| + c < 0$, $\|u_{\text{in}}\|$ is located between the two roots r_- and r_+ , and thus $1 - (r_+ - r_-)/(\|u_{\text{in}}\| - r_-) < 0$. This implies $\|u(t)\|$ in (4.29) decreases from $u(0) = \|u_{\text{in}}\|$, so we have $\|u(t)\| < \|u_{\text{in}}\| < r_+$ for any $t > 0$. □

We remark that $\lim_{t \rightarrow \infty} \frac{d\|u\|}{dt} = 0$ since $\frac{d\|u\|}{dt} < 0$ and $\|u(t)\| \geq 0$, so $u(t)$ approaches to a stationary point of the right-hand side of (4.1) (called an attractor in the theory of dynamical systems).

Note that for a homogeneous equation (i.e., $\|F_0\| = 0$), this shows that the dissipation

inevitably leads to exponential decay. In this case we have $r_+ = 0$, so (4.29) gives

$$\|u(t)\| = \frac{\|u_{\text{in}}\| r_+}{e^{ar_+ t} (r_+ - \|u_{\text{in}}\|) + \|u_{\text{in}}\|}, \quad (4.30)$$

which decays exponentially in t .

On the other hand, as mentioned in the introduction, the solution of a dissipative inhomogeneous quadratic ODE can remain asymptotically nonzero. Here we present an example of this. Consider a time-independent uncoupled system with $\frac{du_j}{dt} = f_2 u_j^2 + f_1 u_j + f_0$, $j \in [n]$, with $u_j(0) = x_0 > 0$, $f_2 > 0$, $f_1 < 0$, $f_0 > 0$, and $R < 1$. We see that each $u_j(t)$ decreases from x_0 to $x_1 := \frac{f_1}{2f_2} \sqrt{f_1^2 + 4f_2 f_0} > 0$, with $0 < x_1 < u_j(t) < x_0$. Hence, the norm of $u(t)$ is bounded as $0 < \sqrt{n} x_1 < \|u(t)\| < \sqrt{n} x_0$ for any $t > 0$. In general, it is hard to lower bound $\|u(t)\|$, but the above example shows that a nonzero inhomogeneity can prevent the norm of the solution from decreasing to zero.

We now give an upper bound on the error of Carleman linearization.

Lemma 4.2. *Consider an instance of the quadratic ODE (4.1), with its corresponding Carleman linearization as defined in (4.5). As in Problem 4.1, assume that the eigenvalues λ_j of F_1 satisfy $\text{Re}(\lambda_n) \leq \dots \leq \text{Re}(\lambda_1) < 0$. Assume that R defined in (4.2) satisfies $R < 1$. Then for any $j \in [N]$, the error $\eta_j(t) := u^j(t) - \hat{y}_j(t)$ satisfies*

$$\|\eta_j(t)\| \leq \|\eta(t)\| \leq tN \|F_2\| \|u_{\text{in}}\|^{N+1}. \quad (4.31)$$

Consider the derivative of $\|\eta(t)\|$. We have

$$\frac{d\|\eta\|^2}{dt} = \eta^\gamma(A(t) + A^\gamma(t))\eta + \eta^\gamma\hat{b}(t) + \hat{b}(t)^\gamma\eta. \quad (4.35)$$

For $\eta^\gamma(A(t) + A^\gamma(t))\eta$, we bound each term as

$$\begin{aligned} \eta_j^\gamma A_{j+1}^j \eta_{j+1} + \eta_{j+1}^\gamma (A_{j+1}^j)^\gamma \eta_j &\leq j \|F_2\| \|\eta_{j+1}\| \|\eta_j\|, \\ \eta_j^\gamma [A_j^j + (A_j^j)^\gamma] \eta_j &\leq 2j \operatorname{Re}(\lambda_1) \|\eta_j\|^2, \\ \eta_j^\gamma A_{j-1}^j \eta_{j-1} + \eta_{j-1}^\gamma (A_{j-1}^j)^\gamma \eta_j &\leq 2j \|F_0\| \|\eta_{j-1}\| \|\eta_j\| \end{aligned} \quad (4.36)$$

using the definitions in (4.7)–(4.9).

Define a matrix $G \in \mathbb{R}^{n \times n}$ with nonzero entries $G_{j-1,j} = j \|F_0\|$, $G_{j,j} = j \operatorname{Re}(\lambda_1)$, and $G_{j+1,j} = j \|F_2\|$. Then

$$\eta^\gamma(A(t) + A^\gamma(t))\eta \leq \eta^\gamma(G + G^\gamma)\eta. \quad (4.37)$$

Since $\|F_2\| + \|F_0\| < |\operatorname{Re}(\lambda_1)|$, G is strictly diagonally dominant and thus the eigenvalues ν_j of G satisfy $\operatorname{Re}(\nu_n) \leq \dots \leq \operatorname{Re}(\nu_1) < 0$. Thus we have

$$\eta^\gamma(G + G^\gamma)\eta \leq 2 \operatorname{Re}(\nu_1) \|\eta\|^2. \quad (4.38)$$

For $\eta^\gamma\hat{b}(t) + \hat{b}(t)^\gamma\eta$, we have

$$\eta^\gamma\hat{b}(t) + \hat{b}(t)^\gamma\eta \leq 2 \|\hat{b}\| \|\eta\| \leq 2 \|A_{N+1}^N\| \|u^{(N+1)}\| \|\eta\|. \quad (4.39)$$

Since $\|A_{N+1}^N\| = N\|F_2\|$, and $\|u^{(N+1)}\| = \|u\|^{N+1} \leq \|u_{\text{in}}\|^{N+1}$, we have

$$\eta^{\hat{b}(t)} + \hat{b}(t)^{\flat}\eta \leq 2N\|F_2\|\|u_{\text{in}}\|^{N+1}\|\eta\|. \quad (4.40)$$

Using (4.37), (4.38), and (4.40) in (4.35), we find

$$\frac{d\|\eta\|^2}{dt} \leq 2\operatorname{Re}(\nu_1)\|\eta\|^2 + 2N\|F_2\|\|u_{\text{in}}\|^{N+1}\|\eta\|, \quad (4.41)$$

so elementary calculus gives

$$\frac{d\|\eta\|}{dt} = \frac{1}{2\|\eta\|} \frac{d\|\eta\|^2}{dt} \leq \operatorname{Re}(\nu_1)\|\eta\| + N\|F_2\|\|u_{\text{in}}\|^{N+1}. \quad (4.42)$$

Solving the differential inequality as an equation with $\eta(0) = 0$ gives us a bound on $\|\eta\|$:

$$\|\eta(t)\| \leq \int_0^t e^{\operatorname{Re}(\nu_1)(t-s)} N\|F_2\|\|u_{\text{in}}\|^{N+1} ds \leq N\|F_2\|\|u_{\text{in}}\|^{N+1} \int_0^t e^{\operatorname{Re}(\nu_1)(t-s)} ds. \quad (4.43)$$

Finally, using

$$\int_0^t e^{\operatorname{Re}(\nu_1)(t-s)} ds = \frac{1 - e^{\operatorname{Re}(\nu_1)t}}{|\operatorname{Re}(\nu_1)|} \leq t \quad (4.44)$$

(where we used the inequality $1 - e^{at} \leq -at$ with $a < 0$), (4.43) gives the bound

$$\|\eta_j(t)\| \leq \|\eta(t)\| \leq tN\|F_2\|\|u_{\text{in}}\|^{N+1} \quad (4.45)$$

as claimed. □

Note that (4.44) can be bounded alternatively by

$$\int_0^t e^{\operatorname{Re}(\nu_1)(t-s)} ds = \frac{1 - e^{\operatorname{Re}(\nu_1)t}}{|\operatorname{Re}(\nu_1)|} \leq \frac{1}{|\operatorname{Re}(\nu_1)|}, \quad (4.46)$$

and thus $\|\eta_j(t)\| \leq \|\eta(t)\| \leq \frac{1}{j|\operatorname{Re}(\nu_1)|} N \|F_2\| \|u_{\text{in}}\|^{N+1}$. We select (4.44) because it avoids including an additional parameter $\operatorname{Re}(\nu_1)$.

We also give an improved analysis that works for homogeneous quadratic ODEs ($F_0(t) = 0$) under milder conditions. This analysis follows the proof in [73] closely.

Corollary 4.1. *Under the same setting of Lemma 4.2, assume $F_0(t) = 0$ in (4.1). Then for any $j \in [N]$, the error $\eta_j(t) := u^j(t) - \hat{y}_j(t)$ satisfies*

$$\|\eta_j(t)\| \leq \|u_{\text{in}}\|^j \mathbb{R}^{N+1-j}. \quad (4.47)$$

For $j = 1$, we have the tighter bound

$$\|\eta_1(t)\| \leq \|u_{\text{in}}\| \mathbb{R}^N (1 - e^{\operatorname{Re}(\lambda_1)t})^N. \quad (4.48)$$

Proof. We again consider η satisfying (4.33). Since $F_0(t) = 0$, (4.33) reduces to a time-independent ODE with an upper triangular block structure,

$$\frac{d\eta_j}{dt} = A_j^j \eta_j + A_{j+1}^j \eta_{j+1}, \quad j \in [N-1] \quad (4.49)$$

and

$$\frac{d\eta_N}{dt} = A_N^N \eta_N + A_{N+1}^N u^{(N+1)}. \quad (4.50)$$

We proceed by backward substitution. Since $\eta_N(0) = 0$, we have

$$\eta_N(t) = \int_0^t e^{A_N^N(t-s_0)} A_{N+1}^N u^{(N+1)}(s_0) ds_0. \quad (4.51)$$

For $j \in [N]$, (4.8) gives $\|e^{A_j^j t}\| = e^{j \operatorname{Re}(\lambda_1)t}$ and (4.7) gives $\|A_{j+1}^j\| = j\|F_2\|$. By Lemma 4.1, $\|u^{(N+1)}\| = \|u\|^{N+1} \leq \|u_{\text{in}}\|^{N+1}$. We can therefore upper bound (4.51)

by

$$\begin{aligned} \|\eta_N(t)\| &\leq \int_0^t \|e^{A_N^N(t-s_0)}\| \cdot \|A_{N+1}^N u^{(N+1)}(s_0)\| ds_0 \\ &\leq N\|F_2\| \|u_{\text{in}}\|^{N+1} \int_0^t e^{N \operatorname{Re}(\lambda_1)(t-s_0)} ds_0. \end{aligned} \quad (4.52)$$

For $j = N - 1$, (4.49) gives

$$\frac{d\eta_{N-1}}{dt} = A_{N-1}^N \eta_{N-1} + A_N^{N-1} \eta_N. \quad (4.53)$$

Again, since $\eta_{N-1}(0) = 0$, we have

$$\eta_{N-1}(t) = \int_0^t e^{A_{N-1}^N(t-s_1)} A_N^{N-1} \eta_N(s_1) ds_1 \quad (4.54)$$

which has the upper bound

$$\begin{aligned} \|\eta_{N-1}(t)\| &\leq \int_0^t \|e^{A_{N-1}^N(t-s_1)}\| \cdot \|A_N^{N-1} \eta_N(s_1)\| ds_1 \\ &\leq (N-1)\|F_2\| \int_0^t e^{(N-1) \operatorname{Re}(\lambda_1)(t-s_1)} \|\eta_N(s_1)\| ds_1 \\ &\leq N(N-1)\|F_2\|^2 \|u_{\text{in}}\|^{N+1} \int_0^t e^{(N-1) \operatorname{Re}(\lambda_1)(t-s_1)} \int_0^{s_1} e^{N \operatorname{Re}(\lambda_1)(s_1-s)} ds_0 ds_1, \end{aligned} \quad (4.55)$$

where we used (4.52) in the last step. Iterating this procedure for $j = N - 2, \dots, 1$, we

find

$$\begin{aligned}
\|\eta_j(t)\| &\leq \frac{N!}{(j-1)!} \|F_2\|^{N+1-j} \|u_{\text{in}}\|^{N+1} \int_0^t e^{j \operatorname{Re}(\lambda_1)(t-s_{N-j})} \int_0^{s_{N-j}} e^{(j+1) \operatorname{Re}(\lambda_1)(s_{N-j}-s_{N-1-j})} \\
&\quad \dots \int_0^{s_2} e^{(N-1) \operatorname{Re}(\lambda_1)(s_2-s_1)} \int_0^{s_1} e^{N \operatorname{Re}(\lambda_1)(s_1-s_0)} ds_0 \dots ds_{N-j} \\
&= \frac{N!}{(j-1)!} \|F_2\|^{N+1-j} \|u_{\text{in}}\|^{N+1} \int_0^{s_{N+1-j}} \dots \int_0^{s_2} \int_0^{s_1} e^{\operatorname{Re}(\lambda_1)(-Ns_0 + \sum_{k=1}^{j+1} s_k)} ds_0 \dots ds_{N-j}.
\end{aligned} \tag{4.56}$$

Finally, using

$$\int_0^{s_{k+1}} e^{(N-k) \operatorname{Re}(\lambda_1)(s_{k+1}-s_k)} ds_k = \frac{1 - e^{(N-k) \operatorname{Re}(\lambda_1)s_{k+1}}}{(N-k) |\operatorname{Re}(\lambda_1)|} \leq \frac{1}{(N-k) |\operatorname{Re}(\lambda_1)|} \tag{4.57}$$

for $k = 0, \dots, N-j$, (4.56) can be bounded by

$$\begin{aligned}
\|\eta_j(t)\| &\leq \frac{N!}{(j-1)!} \|F_2\|^{N+1-j} \|u_{\text{in}}\|^{N+1} \frac{(j-1)!}{N! |\operatorname{Re}(\lambda_1)|^{N+1-j}} \\
&= \frac{\|u_{\text{in}}\|^{N+1} \|F_2\|^{N+1-j}}{|\operatorname{Re}(\lambda_1)|^{N+1-j}} = \|u_{\text{in}}\|^j R^{N+1-j}.
\end{aligned} \tag{4.58}$$

For $j = 1$, the bound can be further improved. By Lemma 5.2 of [73], if $a \neq 0$,

$$\int_0^{s_N} \dots \int_0^{s_2} \int_0^{s_1} e^{a(-Ns_0 + \sum_{k=1}^N s_k)} ds_0 ds_1 \dots ds_{N-1} = \frac{(e^{as_N} - 1)^N}{N! a^N}. \tag{4.59}$$

With $s_N = t$ and $a = \text{Re}(\lambda_1)$, we find

$$\begin{aligned}
\|\eta_1(t)\| &\leq N! \|F_2\|^N \|u_{\text{in}}\|^{N+1} \int_0^{s_N} \cdots \int_0^{s_2} \int_0^{s_1} e^{\text{Re}(\lambda_1)(Ns_0 + \sum_{k=1}^N s_k)} ds_0 ds_1 \cdots ds_{N-1} \\
&\leq N! \|F_2\|^N \|u_{\text{in}}\|^{N+1} \frac{(e^{\text{Re}(\lambda_1)t} - 1)^N}{N! \text{Re}(\lambda_1)^N} \\
&= \|u_{\text{in}}\| R^N (1 - e^{at})^N,
\end{aligned} \tag{4.60}$$

which is tighter than the $j = 1$ case in (4.58). \square

While [Problem 4.1](#) makes some strong assumptions about the system of differential equations, they appear to be necessary for our analysis. Specifically, the conditions $\text{Re}(\lambda_1) < 0$ and $R < 1$ are required to ensure arbitrary-time convergence.

Since the Euler method for (4.5) is unstable if $\text{Re}(\lambda_1) > 0$ [[52](#), [119](#)], we only consider the case $\text{Re}(\lambda_1) \leq 0$. If $\text{Re}(\lambda_1) = 0$, (4.59) reduces to

$$\int_0^{s_N} \cdots \int_0^{s_2} \int_0^{s_1} e^{a(Ns_0 + \sum_{k=1}^N s_k)} ds_0 ds_1 \cdots ds_{N-1} = \frac{t^N}{N!}, \tag{4.61}$$

giving the error bound

$$\|\eta_1(t)\| \leq \|u_{\text{in}}\| (\|u_{\text{in}}\| \|F_2\| t)^N \tag{4.62}$$

instead of (4.60). Then the error bound can be made arbitrarily small for a finite time by increasing N , but after $t > 1/\|u_{\text{in}}\| \|F_2\|$, the error bound diverges.

Furthermore, if $R \geq 1$, Bernoulli's inequality gives

$$\|u_{\text{in}}\| (1 - Ne^{\text{Re}(\lambda_1)t}) \leq \|u_{\text{in}}\| (1 - Ne^{\text{Re}(\lambda_1)t}) R^N \leq \|u_{\text{in}}\| (1 - e^{\text{Re}(\lambda_1)t})^N R^N, \tag{4.63}$$

where the right-hand side upper bounds $\|\eta_1(t)\|$ as in (4.60). Assuming $\|\eta_1(t)\| = \|u(t) - \hat{y}_1(t)\|$ is smaller than $\|u_{\text{in}}\|$, we require

$$N = \Omega(e^{j\text{Re}(\lambda_1)t}). \quad (4.64)$$

In other words, to apply (4.48) for the Carleman linearization procedure, the truncation order given by Lemma 4.2 must grow exponentially with t .

In fact, we prove in Section 4.5 that for $R \geq \sqrt{2}$, no quantum algorithm (even one based on a technique other than Carleman linearization) can solve Problem 4.1 efficiently. It remains open to understand the complexity of the problem for $1 \leq R < \sqrt{2}$.

On the other hand, if $R < 1$, both (4.47) and (4.48) decrease exponentially with N , making the truncation efficient. We discuss the specific choice of N in (4.149) below.

4.4.1.2 Error from forward Euler method

Next, we provide an upper bound for the error incurred by approximating (4.5) with the forward Euler method. This problem has been well studied for general ODEs. Given an ODE $\frac{dz}{dt} = f(z)$ on $[0, T]$ with an inhomogeneity f that is an L -Lipschitz continuous function of z , the global error of the solution is upper bounded by e^{LT} , although in most cases this bound overestimates the actual error [120]. To remove the exponential dependence on T in our case, we derive a tighter bound for time discretization of (4.5) in Lemma 4.3 below. This lemma is potentially useful for other ODEs as well and can be straightforwardly adapted to other problems.

Lemma 4.3. *Consider an instance of the quantum quadratic ODE problem as defined in*

Problem 4.1, with $R < 1$ as defined in (4.2). Choose a time step

$$h \leq \min \left\{ \frac{1}{N\|F_1\|}, \frac{2(|\operatorname{Re}(\lambda_1)| - \|F_2\| - \|F_0\|)}{N(|\operatorname{Re}(\lambda_1)|^2 - (\|F_2\| + \|F_0\|)^2 + \|F_1\|^2)} \right\} \quad (4.65)$$

in general, or

$$h \leq \frac{1}{N\|F_1\|} \quad (4.66)$$

if the eigenvalues of F_1 are all real. Suppose the error from Carleman linearization $\eta(t)$ as defined in [Lemma 4.2](#) is bounded by

$$\|\eta(t)\| \leq \frac{g}{4}, \quad (4.67)$$

where g is defined in (4.17). Then the global error from the forward Euler method (4.11) on the interval $[0, T]$ for (4.5) satisfies

$$\|\hat{y}_1(T) - y_1^m\| \leq \|\hat{y}(T) - y^m\| \leq 3N^{2.5}Th[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^0\|]. \quad (4.68)$$

Proof. We define a linear system that locally approximates the initial value problem (4.5) on $[kh, (k+1)h]$ for $k \in [m]_0$ as

$$z^k = [I + A((k-1)h)h]\hat{y}((k-1)h) + b((k-1)h), \quad (4.69)$$

where $\hat{y}(t)$ is the exact solution of (4.5). For $k \in [m]$, we denote the local truncation error

by

$$e^k := \|\hat{y}(kh) - z^k\| \quad (4.70)$$

and the global error by

$$g^k := \|\hat{y}(kh) - y^k\|, \quad (4.71)$$

where y^k in (4.11) is the numerical solution. Note that $g^m = \|\hat{y}(T) - y^m\|$.

For the local truncation error, we Taylor expand $\hat{y}((k-1)h)$ with a second-order Lagrange remainder, giving

$$\hat{y}(kh) = \hat{y}((k-1)h) + \hat{y}'((k-1)h)h + \frac{\hat{y}''(\xi)h^2}{2} \quad (4.72)$$

for some $\xi \in [(k-1)h, kh]$. Since $\hat{y}'((k-1)h) = A((k-1)h)\hat{y}((k-1)h) + b((k-1)h)$ by (4.5), we have

$$\hat{y}(kh) = [I + A((k-1)h)h]\hat{y}((k-1)h) + b((k-1)h)h + \frac{\hat{y}''(\xi)h^2}{2} = z^k + \frac{\hat{y}''(\xi)h^2}{2}, \quad (4.73)$$

and thus the local error (4.70) can be bounded as

$$e^k = \|\hat{y}(kh) - z^k\| = \|\hat{y}''(\xi)h\| \frac{h^2}{2} \leq \frac{Mh^2}{2}, \quad (4.74)$$

where $M := \max_{t \in [0, T]} \|\hat{y}''(\tau)\|$.

By the triangle inequality, the global error (4.71) can therefore be bounded as

$$g^k = \|\hat{y}(kh) - y^k\| \leq \|\hat{y}(kh) - z^k\| + \|z^k - y^k\| \leq e^k + \|z^k - y^k\|. \quad (4.75)$$

Since y^k and z^k are obtained by the same linear system with different right-hand sides, we have the upper bound

$$\begin{aligned}
\|z^k - y^k\| &= \|[I + A((k-1)h)h][\hat{y}((k-1)h) - y^{k-1}]\| \\
&\leq \|I + A((k-1)h)h\| \cdot \|\hat{y}((k-1)h) - y^{k-1}\| \\
&= \|I + A((k-1)h)h\| g^{k-1}.
\end{aligned} \tag{4.76}$$

In order to provide an upper bound for $\|I + A(t)h\|$ for all $t \in [0, T]$, we write

$$I + A(t)h = H_2 + H_1 + H_0(t) \tag{4.77}$$

where

$$H_2 = \sum_{j=1}^{N-1} |j\rangle\langle j+1| \otimes A_{j+1}^j h, \tag{4.78}$$

$$H_1 = I + \sum_{j=1}^N |j\rangle\langle j| \otimes A_j^j h, \tag{4.79}$$

$$H_0(t) = \sum_{j=2}^N |j\rangle\langle j-1| \otimes A_{j-1}^j h. \tag{4.80}$$

We provide upper bounds separately for $\|H_2\|$, $\|H_1\|$, and $\|H_0\| := \max_{t \in [0, T]} \|H_0(t)\|$, and use the bound $\max_{t \in [0, T]} \|I + A(t)h\| \leq \|H_2\| + \|H_1\| + \|H_0\|$.

The eigenvalues of A_j^j consist of all j -term sums of the eigenvalues of F_1 . More precisely, they are $\{\sum_{\ell \in 2[j]} \lambda_{I_\ell^j}\}_{I^j \in 2[n]^j}$ where $\{\lambda_\ell\}_{\ell \in 2[n]}$ are the eigenvalues of F_1 and $I^j \in [n]^j$ is a j -tuple of indices. The eigenvalues of H_1 are thus $\{1 + h \sum_{\ell \in 2[j]} \lambda_{I_\ell^j}\}_{I^j \in 2[n]^j, j \in 2[N]}$.

With $J := \max_{\ell \in [n]} |\operatorname{Im}(\lambda_\ell)|$, we have

$$\begin{aligned} \left| 1 + h \sum_{\ell \in [j]} \lambda_{\ell^j} \right|^2 &= \left| 1 + h \sum_{\ell \in [j]} \operatorname{Re}(\lambda_{\ell^j}) \right|^2 + \left| h \sum_{\ell \in [j]} \operatorname{Im}(\lambda_{\ell^j}) \right|^2 \\ &\leq 1 - 2Nh |\operatorname{Re}(\lambda_1)| + N^2 h^2 (|\operatorname{Re}(\lambda_1)|^2 + J^2) \end{aligned} \quad (4.81)$$

where we used $Nh |\operatorname{Re}(\lambda_1)| \leq 1$ by (4.65). Therefore

$$\|H_1\| = \max_{j \in [N]} \max_{\ell \in [n]^j} \left| 1 + h \sum_{\ell \in [j]} \lambda_{\ell^j} \right| \leq \sqrt{1 - 2Nh |\operatorname{Re}(\lambda_1)| + N^2 h^2 (|\operatorname{Re}(\lambda_1)|^2 + J^2)}. \quad (4.82)$$

We also have

$$\|H_2\| = \left\| \sum_{j=1}^{N-1} |j\rangle \langle j+1| \otimes A_{j+1}^j h \right\| \leq \max_{j \in [N]} \|A_{j+1}^j\| h \leq N \|F_2\| h \quad (4.83)$$

and

$$\|H_0\| = \max_{t \in [0, T]} \left\| \sum_{j=1}^{N-1} |j\rangle \langle j| \otimes A_j^j h \right\| \leq \max_{t \in [0, T]} \max_{j \in [N]} \|A_j^j\| h \leq N \max_{t \in [0, T]} \|F_0(t)\| h \leq N \|F_0\| h. \quad (4.84)$$

Using the bounds (4.82) and (4.83), we aim to select the value of h to ensure

$$\max_{t \in [0, T]} \|I + A(t)h\| \leq \|H_2\| + \|H_1\| + \|H_0\| \leq 1. \quad (4.85)$$

The assumption (4.65) implies

$$h \leq \frac{2(|\operatorname{Re}(\lambda_1)| - \|F_2\| - \|F_0\|)}{N(|\operatorname{Re}(\lambda_1)|^2 - (\|F_2\| + \|F_0\|)^2 + J^2)} \quad (4.86)$$

(note that the denominator is non-zero due to (4.3)). Then we have

$$\begin{aligned}
& N^2 h^2 (|\operatorname{Re}(\lambda_1)|^2 - \|F_2\|^2 + J^2) \leq 2Nh(|\operatorname{Re}(\lambda_1)| - \|F_2\| - \|F_0\|) \\
\implies & \sqrt{1 - 2Nh|\operatorname{Re}(\lambda_1)| + N^2 h^2 (|\operatorname{Re}(\lambda_1)|^2 + J^2)} \leq 1 - N(\|F_2\| + \|F_0\|)h,
\end{aligned} \tag{4.87}$$

so $\max_{t \in [0, T]} \|I + A(t)h\| \leq 1$ as claimed.

The choice (4.65) can be improved if an upper bound on J is known. In particular, if $J = 0$, (4.86) simplifies to

$$h \leq \frac{2}{N(|\lambda_1| + \|F_2\| + \|F_0\|)}, \tag{4.88}$$

which is satisfied by (4.66) using $\|F_2\| + \|F_0\| < |\lambda_1| \leq \|F_1\|$.

Using this in (4.76), we have

$$\|z^k - y^k\| \leq \|I + A((k-1)h)h\| g^{k-1} \leq g^{k-1}. \tag{4.89}$$

Plugging (4.89) into (4.75) iteratively, we find

$$g^k \leq g^{k-1} + e^k \leq g^{k-2} + e^{k-1} + e^k \leq \dots \leq \sum_{j=1}^k e^j, \quad k \in [m+1]_0. \tag{4.90}$$

Using (4.74), this shows that the global error from the forward Euler method is bounded by

$$\|\hat{y}^1(kh) - y_1^k\| \leq \|\hat{y}(kh) - y^k\| = g^k \leq \sum_{j=1}^k e^j \leq \frac{Mkh^2}{2}, \tag{4.91}$$

and when $k = m$, $mh = T$,

$$\|\hat{y}^1(T) - y_1^m\| \leq g^m \leq m \frac{Mh^2}{2} = \frac{MTh}{2}. \quad (4.92)$$

Finally, we remove the dependence on $M = \max_{t \in [0, T]} \|\hat{y}^{\theta\theta}(\tau)\|$. Since

$$\begin{aligned} \hat{y}^{\theta\theta}(t) &= [A(t)\hat{y}(t) + b(t)]^\theta = A(t)\hat{y}^\theta(t) + A^\theta(t)\hat{y}(t) + b^\theta(t) \\ &= A(t)[A(t)\hat{y}(t) + b(t)] + A^\theta(t)\hat{y}(t) + b^\theta(t), \end{aligned} \quad (4.93)$$

we have

$$\|\hat{y}^{\theta\theta}(t)\| = \|A(t)\|^2 \|\hat{y}(t)\| + \|A(t)\| \|b(t)\| + \|A^\theta(t)\| \|\hat{y}(t)\| + \|b^\theta(t)\|. \quad (4.94)$$

Maximizing each term for $t \in [0, T]$, we have

$$\begin{aligned} \max_{t \in [0, T]} \|A(t)\| &= \left\| \sum_{j=1}^{N-1} |j\rangle\langle j+1| \otimes A_{j+1}^j + \sum_{j=1}^N |j\rangle\langle j| \otimes A_j^j + \sum_{j=2}^N |j\rangle\langle j-1| \otimes A_{j-1}^j \right\| \\ &\leq N(\|F_2\| + \|F_1\| + \|F_0\|), \end{aligned} \quad (4.95)$$

$$\max_{t \in [0, T]} \|A^\theta(t)\| = \left\| \sum_{j=2}^N |j\rangle\langle j-1| \otimes (A_{j-1}^j)^\theta \right\| \leq N\|F_0^\theta(t)\| \leq N\|F_0^\theta\|, \quad (4.96)$$

$$\max_{t \in [0, T]} \|b(t)\| = \max_{t \in [0, T]} \|F_0(t)\| \leq \|F_0\|, \quad (4.97)$$

$$\max_{t \in [0, T]} \|b^\theta(t)\| = \max_{t \in [0, T]} \|F_0^\theta(t)\| \leq \|F_0^\theta\|, \quad (4.98)$$

and using $\|u\| \leq \|u_{\text{in}}\| < 1$, $\|\eta_j(t)\| \leq \|\eta(t)\| \leq g/4 < \|u_{\text{in}}\|/4$ by (4.67), and $R < 1$,

we have

$$\begin{aligned}
\|\hat{y}(t)\|^2 &\leq \sum_{j=1}^N \|\hat{y}_j(t)\|^2 = \sum_{j=1}^N \|u^j(t) - \eta_j(t)\|^2 \leq 2 \sum_{j=1}^N (\|u^j(t)\|^2 + \|\eta_j(t)\|^2) \\
&\leq 2 \sum_{j=1}^N \left(\|u_{\text{in}}\|^{2j} + \frac{\|u_{\text{in}}\|^2}{16} \right) < 2 \sum_{j=1}^N \left(1 + \frac{1}{16} \right) \|u_{\text{in}}\|^2 < 4N \|u_{\text{in}}\|^2 < 4N
\end{aligned} \tag{4.99}$$

for all $t \in [0, T]$. Therefore, substituting the bounds (4.95)–(4.99) into (4.94), we find

$$\begin{aligned}
M &\leq \max_{t \in [0, T]} \|A(t)\|^2 \|\hat{y}(t)\| + \|A(t)\| \|b(t)\| + \|A^\theta(t)\| \|\hat{y}(t)\| + \|b^\theta(t)\| \\
&\leq 2N^{2.5} (\|F_2\| + \|F_1\| + \|F_0\|)^2 + N (\|F_2\| + \|F_1\| + \|F_0\|) \|F_0\| + 2N^{1.5} \|F_0^\theta\| + \|F_0^\theta\| \\
&\leq 6N^{2.5} [(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|].
\end{aligned} \tag{4.100}$$

Thus, (4.92) gives

$$\|\hat{y}^1(kh) - y_1^m\| \leq 3N^{2.5} kh^2 [(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|], \tag{4.101}$$

and when $k = m$, $mh = T$,

$$\|\hat{y}^1(T) - y_1^m\| \leq 3N^{2.5} Th [(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|] \tag{4.102}$$

as claimed. □

4.4.2 Condition number

Now we upper bound the condition number of the linear system.

Lemma 4.4. *Consider an instance of the quantum quadratic ODE problem as defined in [Problem 4.1](#). Apply the forward Euler method [\(4.11\)](#) with time step [\(4.65\)](#) to the Carleman linearization [\(4.5\)](#). Then the condition number of the matrix L defined in [\(4.12\)](#) satisfies*

$$\kappa \leq 3(m + p + 1). \quad (4.103)$$

Proof. We begin by upper bounding $\|L\|$. We write

$$L = L_1 + L_2 + L_3, \quad (4.104)$$

where

$$L_1 = \sum_{k=0}^{m+p} |k\rangle\langle k| \otimes I, \quad (4.105)$$

$$L_2 = - \sum_{k=1}^m |k\rangle\langle k-1| \otimes [I + A((k-1)h)h], \quad (4.106)$$

$$L_3 = - \sum_{k=m+1}^{m+p} |k\rangle\langle k-1| \otimes I. \quad (4.107)$$

Clearly $\|L_1\| = \|L_3\| = 1$. Furthermore, $\|L_2\| \leq \max_{t \in [0, T]} \|I + A(t)h\| \leq 1$ by [\(4.85\)](#), which follows from the choice of time step [\(4.65\)](#). Therefore,

$$\|L\| \leq \|L_1\| + \|L_2\| + \|L_3\| \leq 3. \quad (4.108)$$

Next we upper bound

$$\|L^{-1}\| = \sup_{k \in [m+p]_0} \|L^{-1}|B\rangle\|. \quad (4.109)$$

We express $|B\rangle$ as

$$|B\rangle = \sum_{k=0}^{m+p} \beta_k |k\rangle = \sum_{k=0}^{m+p} |b^k\rangle, \quad (4.110)$$

where $|b^k\rangle := \beta_k |k\rangle$ satisfies

$$\sum_{k=0}^{m+p} \| |b^k\rangle \|^2 = \| |B\rangle \|^2 \leq 1. \quad (4.111)$$

Given any $|b^k\rangle$ for $k \in [m+p+1]_0$, we define

$$|Y^k\rangle := L^{-1}|b^k\rangle = \sum_{l=0}^{m+p} \gamma_l^k |l\rangle = \sum_{l=0}^{m+p} |Y_l^k\rangle, \quad (4.112)$$

where $|Y_l^k\rangle := \gamma_l^k |l\rangle$. We first upper bound $\| |Y^k\rangle \| = \| L^{-1}|b^k\rangle \|$, and then use this to upper bound $\| L^{-1}|B\rangle \|$.

We consider two cases. First, for fixed $k \in [m+p+1]_0$, we directly calculate $|Y_l^k\rangle$ for each $l \in [m+p+1]_0$ by the forward Euler method (4.11), giving

$$|Y_l^k\rangle = \begin{cases} 0, & \text{if } l \in [k]_0; \\ |b^k\rangle, & \text{if } l = k; \\ \Pi_{j=k}^{l-1} [I + A(jh)h] |b^k\rangle, & \text{if } l \in [m+p+1]_0 \setminus [k+1]_0; \\ \Pi_{j=k}^{m+p} [I + A(jh)h] |b^k\rangle, & \text{if } l \in [m+p+1]_0 \setminus [m+p+1]_0. \end{cases} \quad (4.113)$$

Since $\max_{t \in [0, T]} \|I + A(t)h\| \leq 1$ by (4.85), (4.112) gives

$$\begin{aligned} \| |Y^k\rangle \|^2 &= \sum_{l=0}^{m+p} \| |Y_l^k\rangle \|^2 \leq \sum_{l=k}^m \| |b^k\rangle \|^2 + \sum_{l=m+1}^{m+p} \| |b^k\rangle \|^2 \\ &\leq (m+p+1-k) \| |b^k\rangle \|^2 \leq (m+p+1) \| |b^k\rangle \|^2. \end{aligned} \quad (4.114)$$

Second, for fixed $k \in [m+p+1]_0 \setminus [m+1]_0$, similarly to (4.113), we directly calculate $|Y_l^k\rangle$ using (4.11), giving

$$|Y_l^k\rangle = \begin{cases} 0, & \text{if } l \in [k]_0; \\ |b^k\rangle, & \text{if } l \in [m+p+1]_0 \setminus [k]_0. \end{cases} \quad (4.115)$$

Similarly to (4.114), we have (again using (4.112))

$$\| |Y^k\rangle \|^2 = \sum_{l=0}^{m+p} \| |Y_l^k\rangle \|^2 = \sum_{l=k}^{m+p} \| |b^k\rangle \|^2 = (m+p+1-k) \| |b^k\rangle \|^2 \leq (m+p+1) \| |b^k\rangle \|^2. \quad (4.116)$$

Combining (4.114) and (4.116), for any $k \in [m+p+1]_0$, we have

$$\| |Y^k\rangle \|^2 = \| L^{-1} |b^k\rangle \|^2 \leq (m+p+1) \| |b^k\rangle \|^2. \quad (4.117)$$

By the definition of $|Y^k\rangle$ in (4.112), (4.117) gives

$$\| L^{-1} \|^2 = \sup_{kjB^k \ 1} \| L^{-1} |B\rangle \|^2 \leq (m+p+1) \sup_{kjB^k \ 1} \| L^{-1} |b^k\rangle \|^2 \leq (m+p+1)^2, \quad (4.118)$$

and therefore

$$\|L^{-1}\| \leq (m + p + 1). \quad (4.119)$$

Finally, combining (4.108) with (4.119), we conclude

$$\kappa = \|L\| \|L^{-1}\| \leq 3(m + p + 1) \quad (4.120)$$

as claimed. □

4.4.3 State preparation

We now describe a procedure for preparing the right-hand side $|B\rangle$ of the linear system (4.12), whose entries are composed of $|y_{\text{in}}\rangle$ and $|b((k-1)h)\rangle$ for $k \in [m]$.

The initial vector y_{in} is a direct sum over spaces of different dimensions, which is cumbersome to prepare. Instead, we prepare an equivalent state that has a convenient tensor product structure. Specifically, we embed y_{in} into a slightly larger space and prepare the normalized version of

$$z_{\text{in}} = [u_{\text{in}} \otimes v_0^{N-1}; u_{\text{in}} \otimes v_0^{N-2}; \dots; u_{\text{in}}^N], \quad (4.121)$$

where v_0 is some standard vector (for simplicity, we take $v_0 = |0\rangle$). If u_{in} lives in a vector space of dimension n , then z_{in} lives in a space of dimension Nn^N while y_{in} lives in a slightly smaller space of dimension $\Delta = n + n^2 + \dots + n^N = (n^{N+1} - n)/(n - 1)$. Using standard techniques, all the operations we would otherwise apply to y_{in} can be applied instead to z_{in} , with the same effect.

Lemma 4.5. *Assume we are given the value $\|u_{\text{in}}\|$, and let O_x be an oracle that maps $|00\dots 0\rangle \in \mathbb{C}^n$ to a normalized quantum state $|u_{\text{in}}\rangle$ proportional to u_{in} . Assume we are also given the values $\|F_0(t)\|$ for each $t \in [0, T]$, and let O_{F_0} be an oracle that provides the locations and values of the nonzero entries of $F_0(t)$ for any specified t . Then the quantum state $|B\rangle$ defined in (4.14) (with y_{in} replaced by z_{in}) can be prepared using $O(N)$ queries to O_x and $O(m)$ queries to O_{F_0} , with gate complexity larger by a $\text{poly}(\log N, \log n)$ factor.*

Proof. We first show how to prepare the state

$$|z_{\text{in}}\rangle = \frac{1}{\sqrt{V}} \sum_{j=1}^N \|u_{\text{in}}\|^j |j\rangle |u_{\text{in}}\rangle^{\otimes j} |0\rangle^{\otimes N-j}, \quad (4.122)$$

where

$$V := \sum_{j=1}^N \|u_{\text{in}}\|^{2j}. \quad (4.123)$$

This state can be prepared using N queries to the initial state oracle O_x applied in superposition to the intermediate state

$$|\psi_{\text{int}}\rangle := \frac{1}{\sqrt{V}} \sum_{j=1}^N \|u_{\text{in}}\|^j |j\rangle \otimes |0\rangle^{\otimes N-j}. \quad (4.124)$$

To efficiently prepare $|\psi_{\text{int}}\rangle$, notice that

$$|\psi_{\text{int}}\rangle = \frac{\|u_{\text{in}}\|}{\sqrt{V}} \sum_{j_0, j_1, \dots, j_{k-1}=0}^1 \prod_{\ell=0}^{k-1} \|u_{\text{in}}\|^{j_\ell 2^\ell} |j_0 j_1 \dots j_{k-1}\rangle \otimes |0\rangle^{\otimes N}, \quad (4.125)$$

where $k := \log_2 N$ (assuming for simplicity that N is a power of 2) and $j_{k-1} \dots j_1 j_0$ is

the k -bit binary expansion of $j - 1$. Observe that

$$|\psi_{\text{int}}\rangle = \bigotimes_{\ell=0}^{k-1} \left(\frac{1}{\sqrt{V_\ell}} \sum_{j_\ell=0}^1 \|u_{\text{in}}\|^{j_\ell 2^\ell} |j_\ell\rangle \right) \otimes |0\rangle^N \quad (4.126)$$

where

$$V_\ell := 1 + \|u_{\text{in}}\|^{2^{\ell+1}}. \quad (4.127)$$

(Notice that $\prod_{\ell=0}^{k-1} V_\ell = V/\|u_{\text{in}}\|^2$.) Each tensor factor in (4.126) is a qubit that can be produced in constant time. Overall, we prepare these $k = \log_2 N$ qubit states and then apply O_x N times.

We now discuss how to prepare the state

$$|B\rangle = \frac{1}{\sqrt{B_m}} \left(\|z_{\text{in}}\| |0\rangle \otimes |z_{\text{in}}\rangle + \sum_{k=1}^m \|b((k-1)h)\| |k\rangle \otimes |b((k-1)h)\rangle \right), \quad (4.128)$$

in which we replace y_{in} by z_{in} in (4.14), and define

$$B_m := \|z_{\text{in}}\|^2 + \sum_{k=1}^m \|b((k-1)h)\|^2. \quad (4.129)$$

This state can be prepared using the above procedure for $|0\rangle \mapsto |z_{\text{in}}\rangle$ and m queries to O_{F_0} with $t = (k-1)h$ that implement $|0\rangle \mapsto |b((k-1)h)\rangle$ for $k \in \{1, \dots, m\}$, applied in superposition to the intermediate state

$$|\phi_{\text{int}}\rangle = \frac{1}{\sqrt{B_m}} \left(\|z_{\text{in}}\| |0\rangle \otimes |0\rangle + \sum_{k=1}^m \|b((k-1)h)\| |k\rangle \otimes |0\rangle \right). \quad (4.130)$$

Here the queries are applied conditionally upon the value in the first register: we prepare $|z_{\text{in}}\rangle$ if the first register is $|0\rangle$ and $|b((k-1)h)\rangle$ if the first register is $|k\rangle$ for $k \in \{1, \dots, m\}$. We can prepare $|\phi_{\text{int}}\rangle$ (i.e., perform a unitary transform mapping $|0\rangle|0\rangle \mapsto |\phi_{\text{int}}\rangle$) in time complexity $O(m)$ [92] using the known values of $\|u_{\text{in}}\|$ and $\|b((k-1)h)\|$.

Overall, we use $O(N)$ queries to O_x and $O(m)$ queries to O_{F_0} to prepare $|B\rangle$. The gate complexity is larger by a $\text{poly}(\log N, \log n)$ factor. \square

4.4.4 Measurement success probability

After applying the QLSA to (4.12), we perform a measurement to extract a final state of the desired form. We now consider the probability of this measurement succeeding.

Lemma 4.6. *Consider an instance of the quantum quadratic ODE problem defined in Problem 4.1, with the QLSA applied to the linear system (4.12) using the forward Euler method (4.11) with time step (4.65). Suppose the error from Carleman linearization satisfies $\|\eta(t)\| \leq \frac{g}{4}$ as in (4.67), and the global error from the forward Euler method as defined in Lemma 4.3 is bounded by*

$$\|\hat{y}(T) - y^m\| \leq \frac{g}{4}, \quad (4.131)$$

where g is defined in (4.17). Then the probability of measuring a state $|y_1^k\rangle$ for $k = [m+p+1]_0 \setminus [m+1]_0$ satisfies

$$P_{\text{measure}} \geq \frac{p+1}{9(m+p+1)Nq^2}, \quad (4.132)$$

where q is also defined in (4.17).

Proof. The idealized quantum state produced by the QLSA applied to (4.12) has the form

$$|Y\rangle = \sum_{k=0}^{m+p} |y^k\rangle |k\rangle = \sum_{k=0}^{m+p} \sum_{j=1}^N |y_j^k\rangle |j\rangle |k\rangle \quad (4.133)$$

where the states $|y^k\rangle$ and $|y_j^k\rangle$ for $k \in [m+p+1]_0$ and $j \in [N]$ are subnormalized to ensure $\| |Y\rangle \| = 1$.

We decompose the state $|Y\rangle$ as

$$|Y\rangle = |Y_{\text{bad}}\rangle + |Y_{\text{good}}\rangle, \quad (4.134)$$

where

$$\begin{aligned} |Y_{\text{bad}}\rangle &:= \sum_{k=0}^{m-1} \sum_{j=1}^N |y_j^k\rangle |j\rangle |k\rangle + \sum_{k=m}^{m+p} \sum_{j=2}^N |y_j^k\rangle |j\rangle |k\rangle, \\ |Y_{\text{good}}\rangle &:= \sum_{k=m}^{m+p} |y_1^k\rangle |1\rangle |k\rangle. \end{aligned} \quad (4.135)$$

Note that $|y_1^k\rangle = |y_1^m\rangle$ for all $k \in \{m, m+1, \dots, m+p\}$. We lower bound

$$P_{\text{measure}} := \frac{\| |Y_{\text{good}}\rangle \|^2}{\| |Y\rangle \|^2} = \frac{(p+1) \| |y_1^m\rangle \|^2}{\| |Y\rangle \|^2} \quad (4.136)$$

by lower bounding the terms of the product

$$\frac{\| |y_1^m\rangle \|^2}{\| |Y\rangle \|^2} = \frac{\| |y_1^m\rangle \|^2}{\| |y_1^0\rangle \|^2} \cdot \frac{\| |y_1^0\rangle \|^2}{\| |Y\rangle \|^2}. \quad (4.137)$$

First, according to (4.67) and (4.131), the exact solution $u(T)$ and the approximate

solution y_1^m defined in (4.12) satisfy

$$\|u(T) - y_1^m\| \leq \|u(T) - \hat{y}_1(T)\| + \|\hat{y}_1(T) - y_1^m\| \leq \|\eta(t)\| + \|\hat{y}(T) - y^m\| \leq \frac{g}{2}. \quad (4.138)$$

Since $y_1^0 = (y_{\text{in}})_1 = u_{\text{in}}$, using (4.138), we have

$$\frac{\|y_1^m\|}{\|y_1^0\|} = \frac{\|y_1^m\|}{\|u_{\text{in}}\|} \geq \frac{\|u(T)\| - \|u(T) - y_1^m\|}{\|u_{\text{in}}\|} = \frac{g - \|u(T) - y_1^m\|}{\|u_{\text{in}}\|} \geq \frac{g}{2\|u_{\text{in}}\|} = \frac{1}{2q}. \quad (4.139)$$

Second, we upper bound $\|y^k\|^2$ by

$$\|y^k\|^2 = \|\hat{y}(kh) - [y(kh) - y^k]\|^2 \leq 2(\|\hat{y}(kh)\|^2 + \|y(kh) - y^k\|^2). \quad (4.140)$$

Using $\|\hat{y}(t)\|^2 < 4N\|u_{\text{in}}\|^2$ by (4.99), and $\|y(kh) - y^k\| \leq \|\hat{y}(T) - y^m\| \leq g/4 < \|u_{\text{in}}\|/4$ by (4.131), and $R < 1$, we have

$$\|y^k\|^2 \leq 2\left(4N\|u_{\text{in}}\|^2 + \frac{\|u_{\text{in}}\|^2}{16}\right) < 9N\|u_{\text{in}}\|^2. \quad (4.141)$$

Therefore

$$\frac{\|y_1^0\|^2}{\|Y\|^2} = \frac{\|y_1^0\|^2}{\sum_{k=0}^{m+p} \|y^k\|^2} \geq \frac{\|u_{\text{in}}\|^2}{9N(m+p+1)\|u_{\text{in}}\|^2} = \frac{1}{9N(m+p+1)}. \quad (4.142)$$

Finally, using (4.139) and (4.142) in (4.137) and (4.136), we have

$$P_{\text{measure}} \geq \frac{p+1}{9(m+p+1)Nq^2} \quad (4.143)$$

as claimed. □

Choosing $m = p$, we have $P_{\text{measure}} = \Omega(1/Nq^2)$. Using amplitude amplification, $O(\sqrt{N}q)$ iterations suffice to succeed with constant probability.

4.4.5 Proof of [Theorem 4.1](#)

Proof. We first present the quantum Carleman linearization (QCL) algorithm and then analyze its complexity.

The QCL algorithm. We start by rescaling the system to satisfy (4.3) and (4.4). Given a quadratic ODE (4.1) satisfying $R < 1$ (where R is defined in (4.2)), we define a scaling factor $\gamma > 0$, and rescale u to

$$\bar{u} := \gamma u. \tag{4.144}$$

Replacing u by \bar{u} in (4.1), we have

$$\frac{d\bar{u}}{dt} = \frac{1}{\gamma} F_2 \bar{u}^2 + F_1 \bar{u} + \gamma F_0(t), \tag{4.145}$$

$$\bar{u}(0) = \bar{u}_{\text{in}} := \gamma u_{\text{in}}.$$

We let $\bar{F}_2 := \frac{1}{\gamma} F_2$, $\bar{F}_1 := F_1$, and $\bar{F}_0(t) := \gamma F_0(t)$ so that

$$\frac{d\bar{u}}{dt} = \bar{F}_2 \bar{u}^2 + \bar{F}_1 \bar{u} + \bar{F}_0(t), \tag{4.146}$$

$$\bar{u}(0) = \bar{u}_{\text{in}}.$$

Note that R is invariant under this rescaling, so $R < 1$ still holds for the rescaled equation.

Concretely, we take³

$$\gamma = \frac{1}{\sqrt{\|u_{\text{in}}\|r_+}}. \quad (4.147)$$

After rescaling, the new quadratic ODE satisfies $\|\bar{u}_{\text{in}}\|\bar{r}_+ = \gamma^2\|u_{\text{in}}\|r_+ = 1$. Since $\|u_{\text{in}}\| < r_+$ by [Lemma 4.1](#), we have $\bar{r} < \|\bar{u}_{\text{in}}\| < 1 < \bar{r}_+$, so [\(4.4\)](#) holds. Furthermore, 1 is located between the two roots \bar{r} and \bar{r}_+ , which implies $\|\bar{F}_2\| \cdot 1^2 + |\text{Re}(\bar{\lambda}_1)| \cdot 1 + \|\bar{F}_0\| < 0$ as shown in [Lemma 4.1](#), so [\(4.3\)](#) holds for the rescaled problem.

Having performed this rescaling, we henceforth assume that [\(4.3\)](#) and [\(4.4\)](#) are satisfied. We then introduce the choice of parameters as follows. Given g and an error bound $\epsilon \leq 1$, we define

$$\delta := \frac{g\epsilon}{1 + \epsilon} \leq \frac{g}{2}. \quad (4.148)$$

Given $\|u_{\text{in}}\|$, $\|F_2\|$, and $\text{Re}(\lambda_1) < 0$, we choose

$$N = \left\lceil \frac{\log(2T\|F_2\|/\delta)}{\log(1/\|u_{\text{in}}\|)} \right\rceil = \left\lceil \frac{\log(2T\|F_2\|/\delta)}{\log(r_+)} \right\rceil. \quad (4.149)$$

Since $\|u_{\text{in}}\|/\delta > 1$ by [\(4.148\)](#) and $g < \|u_{\text{in}}\|$, [Lemma 4.2](#) gives

$$\|u(T) - \hat{y}_1(T)\| \leq \|\eta(T)\| \leq TN\|F_2\|\|u_{\text{in}}\|^{N+1} = TN\|F_2\|\left(\frac{1}{r_+}\right)^{N+1} \leq \frac{\delta}{2}. \quad (4.150)$$

Thus, [\(4.67\)](#) holds since $\delta \leq g/2$.

Now we discuss the choice of h . On the one hand, h must satisfy [\(4.65\)](#) to satisfy the conditions of [Lemma 4.3](#) and [Lemma 4.4](#). On the other hand, [Lemma 4.3](#) gives the

³In fact, one can show that any $\gamma \in \left(\frac{1}{r_+}, \frac{1}{\kappa u_{\text{in}}k}\right)$ suffices to satisfy [\(4.3\)](#) and [\(4.4\)](#).

upper bound

$$\|\hat{y}_1(T) - y_1^m\| \leq 3N^{2.5}Th[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|] \leq \frac{g\epsilon}{4} \leq \frac{g\epsilon}{2(1+\epsilon)} = \frac{\delta}{2}. \quad (4.151)$$

This also ensures that (4.131) holds since $\delta \leq g/2$. Thus, we choose

$$h \leq \min \left\{ \frac{g\epsilon}{12N^{2.5}T[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}, \frac{1}{N\|F_1\|}, \frac{2(|\operatorname{Re}(\lambda_1)| - \|F_2\| - \|F_0\|)}{N(|\operatorname{Re}(\lambda_1)|^2 - (\|F_2\| + \|F_0\|)^2 + \|F_1\|^2)} \right\} \quad (4.152)$$

to satisfy (4.65) and (4.151).

Combining (4.150) with (4.151), we have

$$\|u(T) - y_1^m\| \leq \|u(T) - \hat{y}_1(T)\| + \|\hat{y}_1(T) - y_1^m\| \leq \delta. \quad (4.153)$$

Thus, (4.138) holds since $\delta \leq g/2$. Using

$$\left\| \frac{u(T)}{\|u(T)\|} - \frac{y_1^m}{\|y_1^m\|} \right\| \leq \frac{\|u(T) - y_1^m\|}{\min\{\|u(T)\|, \|y_1^m\|\}} \leq \frac{\|u(T) - y_1^m\|}{g - \|u(T) - y_1^m\|} \quad (4.154)$$

and (4.153), we obtain

$$\left\| \frac{u(T)}{\|u(T)\|} - \frac{y_1^m}{\|y_1^m\|} \right\| \leq \frac{\delta}{g - \delta} = \epsilon, \quad (4.155)$$

i.e., the normalized output state is ϵ -close to $\frac{u(T)}{\|u(T)\|}$.

We follow the procedure in Lemma 4.5 to prepare the initial state $|\hat{y}_{\text{in}}\rangle$. We apply

the QLSA [46] to the linear system (4.12) with $m = p = \lceil T/h \rceil$, giving a solution $|Y\rangle$. We then perform a measurement to obtain a normalized state of $|y_j^k\rangle$ for some $k \in [m + p + 1]_0$ and $j \in [N]$. By Lemma 4.6, the probability of obtaining a state $|y_1^k\rangle$ for some $k \in [m + p + 1]_0 \setminus [m + 1]_0$, giving the normalized vector $y_1^m / \|y_1^m\|$, is

$$P_{\text{measure}} \geq \frac{p + 1}{9(m + p + 1)Nq^2} \geq \frac{1}{18Nq^2}. \quad (4.156)$$

By amplitude amplification, we can achieve success probability $\Omega(1)$ with $O(\sqrt{N}q)$ repetitions of the above procedure.

Analysis of the complexity. By Lemma 4.5, the right-hand side $|B\rangle$ in (4.12) can be prepared with $O(N)$ queries to O_x and $O(m)$ queries to O_{F_0} , with gate complexity larger by a $\text{poly}(\log N, \log n)$ factor. The matrix L in (4.12) is an $(m + p + 1)\Delta \times (m + p + 1)\Delta$ matrix with $O(Ns)$ nonzero entries in any row or column. By Lemma 4.4 and our choice of parameters, the condition number of L is at most

$$\begin{aligned} & 3(m + p + 1) \\ &= O\left(\frac{N^{2.5}T^2[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{\delta} + NT\|F_1\| \right. \\ & \quad \left. + \frac{NT[|\text{Re}(\lambda_1)|^2 - (\|F_2\| + \|F_0\|)^2 + \|F_1\|^2]}{2(|\text{Re}(\lambda_1)| - \|F_2\| - \|F_0\|)}\right) \\ &= O\left(\frac{N^{2.5}T^2[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{g\epsilon} + \frac{1}{(1 - \|u_{\text{in}}\|)^2} \cdot \frac{NT\|F_1\|^2}{\|F_2\| + \|F_0\|}\right) \\ &= O\left(\frac{N^{2.5}T^2[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{(1 - \|u_{\text{in}}\|)^2(\|F_2\| + \|F_0\|)g\epsilon}\right). \end{aligned} \quad (4.157)$$

Here we use $\|F_2\| + \|F_0\| < |\operatorname{Re}(\lambda_1)| \leq \|F_1\|$ and

$$\begin{aligned} 2(|\operatorname{Re}(\lambda_1)| - \|F_2\| - \|F_0\|) &> (\|u_{\text{in}}\| + \frac{1}{\|u_{\text{in}}\|} - 2)(\|F_2\| + \|F_0\|) \\ &= \frac{1}{\|u_{\text{in}}\|}(1 - \|u_{\text{in}}\|)^2(\|F_2\| + \|F_0\|) > (1 - \|u_{\text{in}}\|)^2(\|F_2\| + \|F_0\|). \end{aligned} \quad (4.158)$$

The first inequality above is from the sum of $|\operatorname{Re}(\lambda_1)| > \|F_2\|\|u_{\text{in}}\| + \|F_0\|/\|u_{\text{in}}\|$ and $|\operatorname{Re}(\lambda_1)| = \|F_2\|r_+ + \|F_0\|/r_+$, where $r_+ = 1/\|u_{\text{in}}\|$. Consequently, by Theorem 5 of [46], the QLSA produces the state $|Y\rangle$ with

$$\begin{aligned} &\frac{N^{3.5}sT^2[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{(1 - \|u_{\text{in}}\|)^2(\|F_2\| + \|F_0\|)g\epsilon} \operatorname{poly}\left(\log \frac{NsT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|}{(1 - \|u_{\text{in}}\|)g\epsilon}\right) \\ &= \frac{sT^2[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{(1 - \|u_{\text{in}}\|)^2(\|F_2\| + \|F_0\|)g\epsilon} \operatorname{poly}\left(\log \frac{sT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|}{(1 - \|u_{\text{in}}\|)g\epsilon} / \log(1/\|u_{\text{in}}\|)\right) \end{aligned} \quad (4.159)$$

queries to the oracles O_{F_2} , O_{F_1} , O_{F_0} , and O_x . Using $O(\sqrt{N}q)$ steps of amplitude amplification to achieve success probability $\Omega(1)$, the overall query complexity of our algorithm is

$$\begin{aligned} &\frac{N^4sT^2q[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{(1 - \|u_{\text{in}}\|)^2(\|F_2\| + \|F_0\|)g\epsilon} \operatorname{poly}\left(\log \frac{NsT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|}{(1 - \|u_{\text{in}}\|)g\epsilon}\right) \\ &= \frac{sT^2q[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{(1 - \|u_{\text{in}}\|)^2(\|F_2\| + \|F_0\|)g\epsilon} \operatorname{poly}\left(\log \frac{sT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|}{(1 - \|u_{\text{in}}\|)g\epsilon} / \log(1/\|u_{\text{in}}\|)\right) \end{aligned} \quad (4.160)$$

and the gate complexity exceeds this by a factor of $\operatorname{poly}(\log(nsT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|/(1 - R)g\epsilon)/\log(1/\|u_{\text{in}}\|))$.

If the eigenvalues λ_j of F_1 are all real, by (4.66), the condition number of L is at

most

$$\begin{aligned}
3(m + p + 1) &= O\left(\frac{N^{2.5}T^2[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{\delta} + NT\|F_1\|\right) \\
&= O\left(\frac{N^{2.5}T^2[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{g\epsilon}\right).
\end{aligned} \tag{4.161}$$

Similarly, the QLSA produces the state $|Y\rangle$ with

$$\frac{sT^2[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{g\epsilon} \text{poly}\left(\log \frac{sT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|}{g\epsilon} / \log(1/\|u_{\text{in}}\|)\right) \tag{4.162}$$

queries to the oracles O_{F_2} , O_{F_1} , O_{F_0} , and O_x . Using amplitude amplification to achieve success probability $\Omega(1)$, the overall query complexity of the algorithm is

$$\frac{sT^2q[(\|F_2\| + \|F_1\| + \|F_0\|)^2 + \|F_0^\theta\|]}{g\epsilon} \text{poly}\left(\log \frac{sT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|}{g\epsilon} / \log(1/\|u_{\text{in}}\|)\right) \tag{4.163}$$

and the gate complexity is larger by $\text{poly}(\log(nsT\|F_2\|\|F_1\|\|F_0\|\|F_0^\theta\|/g\epsilon) / \log(1/\|u_{\text{in}}\|))$ as claimed. \square

4.5 Lower bound

In this section, we establish a limitation on the ability of quantum computers to solve the quadratic ODE problem when the nonlinearity is sufficiently strong. We quantify the strength of the nonlinearity in terms of the quantity R defined in (4.2). Whereas there is an efficient quantum algorithm for $R < 1$ (as shown in [Theorem 4.1](#)), we show here that the problem is intractable for $R \geq \sqrt{2}$.

Theorem 4.2. *Assume $R \geq \sqrt{2}$. Then there is an instance of the quantum quadratic ODE problem defined in [Problem 4.1](#) such that any quantum algorithm for producing a quantum state approximating $u(T)/\|u(T)\|$ with bounded error must have worst-case time complexity exponential in T .*

We establish this result by showing how the nonlinear dynamics can be used to distinguish nonorthogonal quantum states, a task that requires many copies of the given state. Note that since our algorithm only approximates the quantum state corresponding to the solution, we must lower bound the query complexity of *approximating* the solution of a quadratic ODE.

4.5.1 Hardness of state discrimination

Previous work on the computational power of nonlinear quantum mechanics shows that the ability to distinguish non-orthogonal states can be applied to solve unstructured search (and other hard computational problems) [78, 79, 80]. Here we show a similar limitation using an information-theoretic argument.

Lemma 4.7. *Let $|\psi\rangle, |\phi\rangle$ be states of a qubit with $|\langle\psi|\phi\rangle| = 1 - \epsilon$. Suppose we are either given a black box that prepares $|\psi\rangle$ or a black box that prepares $|\phi\rangle$. Then any bounded-error protocol for determining whether the state is $|\psi\rangle$ or $|\phi\rangle$ must use $\Omega(1/\epsilon)$ queries.*

Proof. Using the black box k times, we prepare states with overlap $(1 - \epsilon)^k$. By the well-known relationship between fidelity and trace distance, these states have trace distance at most $\sqrt{1 - (1 - \epsilon)^{2k}} \leq \sqrt{2k\epsilon}$. Therefore, by the Helstrom bound (which states that the

advantage over random guessing for the best measurement to distinguish two quantum states is given by their trace distance [121]), we need $k = \Omega(1/\epsilon)$ to distinguish the states with bounded error. \square

4.5.2 State discrimination with nonlinear dynamics

Lemma 4.7 can be used to establish limitations on the ability of quantum computers to simulate nonlinear dynamics, since nonlinear dynamics can be used to distinguish nonorthogonal states. Whereas previous work considers models of nonlinear quantum dynamics (such as the Weinberg model [79, 80] and the Gross-Pitaevskii equation [78]), here we aim to show the difficulty of efficiently simulating more general nonlinear ODEs—in particular, quadratic ODEs with dissipation—using quantum algorithms.

Lemma 4.8. *There exists an instance of the quantum quadratic ODE problem as defined in Problem 4.1 with $R \geq \sqrt{2}$, and two states of a qubit with overlap $1 - \epsilon$ (for $0 < \epsilon < 1 - 3/\sqrt{10}$) as possible initial conditions, such that the two final states after evolution time $T = O(\log(1/\epsilon))$ have an overlap no larger than $3/\sqrt{10}$.*

Proof. Consider a 2-dimensional system of the form

$$\begin{aligned}\frac{du_1}{dt} &= -u_1 + ru_1^2, \\ \frac{du_2}{dt} &= -u_2 + ru_2^2,\end{aligned}\tag{4.164}$$

for some $r > 0$, with an initial condition $u(0) = [u_1(0); u_2(0)] = u_{\text{in}}$ satisfying $\|u_{\text{in}}\| = 1$.

According to the definition of R in (4.2), we have $R = r$, so henceforth we write this

parameter as R . The analytic solution of (4.164) is

$$\begin{aligned} u_1(t) &= \frac{1}{R - e^t(R - 1/u_1(0))}, \\ u_2(t) &= \frac{1}{R - e^t(R - 1/u_2(0))}. \end{aligned} \tag{4.165}$$

When $u_2(0) > 1/R$, $u_2(t)$ is finite within the domain

$$0 \leq t < t := \log\left(\frac{R}{R - 1/u_2(0)}\right); \tag{4.166}$$

when $u_2(0) = 1/R$, we have $u_2(t) = 1/R$ for all t ; and when $u_2(0) < 1/R$, $u_2(t)$ goes to 0 as $t \rightarrow \infty$. The behavior of $u_1(t)$ depends similarly on $u_1(0)$.

Without loss of generality, we assume $u_1(0) \leq u_2(0)$. For $u_2(0) \geq u_1(0) > 1/R$, both $u_1(t)$ and $u_2(t)$ are finite within the domain (4.166), which we consider as the domain of $u(t)$.

Now we consider 1-qubit states that provide inputs to (4.164). Given a sufficiently small $\epsilon > 0$, we first define $\theta \in (0, \pi/4)$ by

$$2 \sin^2 \frac{\theta}{2} = \epsilon. \tag{4.167}$$

We then construct two 1-qubit states with overlap $1 - \epsilon$, namely

$$|\phi(0)\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{4.168}$$

and

$$|\psi(0)\rangle = \cos\left(\theta + \frac{\pi}{4}\right)|0\rangle + \sin\left(\theta + \frac{\pi}{4}\right)|1\rangle. \quad (4.169)$$

Then the overlap between the two initial states is

$$\langle\phi(0)|\psi(0)\rangle = \cos\theta = 1 - \epsilon. \quad (4.170)$$

The initial overlap (4.170) is larger than the target overlap $3/\sqrt{10}$ in Lemma 4.8 provided $\epsilon < 1 - 3/\sqrt{10}$. For simplicity, we denote

$$\begin{aligned} v_0 &:= \cos\left(\theta + \frac{\pi}{4}\right), \\ w_0 &:= \sin\left(\theta + \frac{\pi}{4}\right), \end{aligned} \quad (4.171)$$

and let $v(t)$ and $w(t)$ denote solutions of (4.164) with initial conditions $v(0) = v_0$ and $w(0) = w_0$, respectively. Since $w_0 > 1/R$, we see that $w(t)$ increases with t , satisfying

$$\frac{1}{R} \leq \frac{1}{\sqrt{2}} < w_0 < w(t), \quad (4.172)$$

and

$$v(t) < w(t) \quad (4.173)$$

for any time $0 < t < t$, whatever the behavior of $v(t)$.

We now study the outputs of our problem. For the state $|\phi(0)\rangle$, the initial condition

for (4.164) is $[1/\sqrt{2}; 1/\sqrt{2}]$. Thus, the output for any $t \geq 0$ is

$$|\phi(t)\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \quad (4.174)$$

For the state $|\psi(0)\rangle$, the initial condition for (4.164) is $[v_0; w_0]$. We now discuss how to select a terminal time T to give a useful output state $|\psi(T)\rangle$. For simplicity, we denote the ratio of $w(t)$ and $v(t)$ by

$$K(t) := \frac{w(t)}{v(t)}. \quad (4.175)$$

Noticing that $w(t)$ goes to infinity as t approaches t^* , while $v(t)$ remains finite within (4.166), there exists a terminal time T such that⁴

$$K(T) \geq 2. \quad (4.176)$$

The normalized output state at this time T is

$$|\psi(T)\rangle = \frac{1}{\sqrt{K(T)^2 + 1}}(|0\rangle + K(T)|1\rangle). \quad (4.177)$$

Combining (4.174) with (4.177), the overlap of $|\phi(T)\rangle$ and $|\psi(T)\rangle$ is

$$\langle \phi(T) | \psi(T) \rangle = \frac{K(T) + 1}{\sqrt{2K(T)^2 + 2}} \leq \frac{3}{\sqrt{10}} \quad (4.178)$$

⁴More concretely, we take $v_{\max} = \max\{v_0, v(t^*)\}$ that upper bounds $v(t)$ on the domain $[0, t^*)$, in which $v(t)$ is a finite value since $v_0 < w_0$. Then there exists a terminal time T such that $w(T) = 2v_{\max}$, and hence $K(T) = w(T)/v(T) \geq 2$.

using (4.176).

Finally, we estimate the evolution time T , which is implicitly defined by (4.176).

We can upper bound its value by t . According to (4.166), we have

$$T < t = \log\left(\frac{R}{R - \frac{1}{w_0}}\right) < \log\left(\frac{\sqrt{2}}{\sqrt{2} - \frac{1}{w_0}}\right) \quad (4.179)$$

since the function $\log(x/(x - c))$ decreases monotonically with x for $x > c > 0$. Using (4.170) to rewrite this expression in terms of ϵ , we have

$$T < t < \log\left(\frac{\sqrt{2}}{\sqrt{2} - \frac{1}{\sin(\theta + \frac{\pi}{4})}}\right) = \log\left(1 + \frac{1}{\sqrt{2\epsilon - \epsilon^2} - \epsilon}\right), \quad (4.180)$$

which scales like $\frac{1}{2} \log(1/2\epsilon)$ as $\epsilon \rightarrow 0$. Therefore $T = O(\log(1/\epsilon))$ as claimed. \square

4.5.3 Proof of Theorem 4.2

We now establish our main lower bound result.

Proof. As introduced in the proof of Lemma 4.8, consider the quadratic ODE (4.164); the two initial states of a qubit $|\phi(0)\rangle$ and $|\psi(0)\rangle$ defined in (4.168) and (4.169), respectively; and the terminal time T defined in (4.176).

Suppose we have a quantum algorithm that, given a black box to prepare a state that is either $|\phi(0)\rangle$ or $|\psi(0)\rangle$, can produce quantum states $|\phi^\theta(T)\rangle$ or $|\psi^\theta(T)\rangle$ that are within distance δ of $|\phi(T)\rangle$ and $|\psi(T)\rangle$, respectively. Since by Lemma 4.8, $|\phi(T)\rangle$ and $|\psi(T)\rangle$ have constant overlap, the overlap between $|\phi^\theta(T)\rangle$ and $|\psi^\theta(T)\rangle$ is also constant

for sufficiently small δ . More precisely, we have

$$\langle \phi(T) | \psi(T) \rangle \leq \frac{3}{\sqrt{10}} \quad (4.181)$$

by (4.178), which implies

$$\| |\phi(T)\rangle - |\psi(T)\rangle \| \geq \sqrt{2 \left(1 - \frac{3}{\sqrt{10}} \right)} > 0.32. \quad (4.182)$$

We also have

$$\| |\phi(T)\rangle - |\phi^\theta(T)\rangle \| \leq \delta, \quad (4.183)$$

and similarly for $\psi(T)$. These three inequalities give us

$$\begin{aligned} \| |\phi^\theta(T)\rangle - |\psi^\theta(T)\rangle \| &= \| (|\phi(T)\rangle - |\psi(T)\rangle) - (|\phi(T)\rangle - |\phi^\theta(T)\rangle) - (|\psi^\theta(T)\rangle - |\psi(T)\rangle) \| \\ &\geq \| (|\phi(T)\rangle - |\psi(T)\rangle) \| - \| (|\phi(T)\rangle - |\phi^\theta(T)\rangle) \| - \| (|\psi^\theta(T)\rangle - |\psi(T)\rangle) \| \\ &> 0.32 - 2\delta, \end{aligned} \quad (4.184)$$

which is at least a constant for (say) $\delta < 0.15$.

Lemma 4.7 therefore shows that preparing the states $|\phi^\theta(T)\rangle$ and $|\psi^\theta(T)\rangle$ requires time $\Omega(1/\epsilon)$, as these states can be used to distinguish the two possibilities with bounded error. By **Lemma 4.8**, this time is $2^{\Omega(T)}$. This shows that we need at least exponential simulation time to approximate the solution of arbitrary quadratic ODEs to within sufficiently small bounded error when $R \geq \sqrt{2}$. \square

Note that exponential time is achievable since our QCL algorithm can solve the

problem by taking N to be exponential in T , where N is the truncation level of Carleman linearization. (The algorithm of Leyton and Osborne also solves quadratic differential equations with complexity exponential in T , but requires the additional assumptions that the quadratic polynomial is measure-preserving and Lipschitz continuous [74].)

4.6 Applications

Due to the assumptions of our analysis, our quantum Carleman linearization algorithm can only be applied to problems with certain properties. First, there are two requirements to guarantee convergence of the inhomogeneous Carleman linearization: the system must have linear dissipation, manifested by $\text{Re}(\lambda_1) < 0$; and the dissipation must be sufficiently stronger than both the nonlinear and the forcing terms, so that $R < 1$. Dissipation typically leads to an exponentially decaying solution, but for the dependency on g and q in (4.160) to allow an efficient algorithm, the solution cannot exponentially approach zero.

However, this issue does not arise if the forcing term F_0 resists the exponential decay towards zero, instead causing the solution to decay towards some non-zero (possibly time-dependent) state. The norm of the state that is exponentially approached can possibly decay towards zero, but this decay itself must happen slower than exponentially for the algorithm to be efficient.⁵

We now investigate possible applications that satisfy these conditions. First we present an application governed by ordinary differential equations, and then we present

⁵Also note that the QCL algorithm might provide an advantage over classical computation for homogeneous equations in cases where only evolution for a short time is of interest.

possible applications in partial differential equations.

Several physical systems can be represented in terms of quadratic ODEs. Examples include models of interacting populations of predators and prey [122], dynamics of chemical reactions [123, 124], and the spread of an epidemic [125]. We now give an example of the latter, based on the epidemiological model used in [126] to describe the early spread of the COVID-19 virus.

The so-called SEIR model divides a population of P individuals into four components: susceptible (P_S), exposed (P_E), infected (P_I), and recovered (P_R). We denote the rate of transmission from an infected to a susceptible person by r_{tra} , the typical time until an exposed person becomes infectious by the latent time T_{lat} , and the typical time an infectious person can infect others by the infectious time T_{inf} . Furthermore we assume that there is a flux Λ of individuals constantly refreshing the population. This flux corresponds to susceptible individuals moving into, and individuals of all components moving out of, the population, in such a way that the total population remains constant.

To ensure that there is sufficiently strong linear decay to guarantee Carleman convergence, we also add a vaccination term to the P_S component. We choose an approach similar to that of [127], but denote the vaccination rate, which is approximately equal to the fraction

of susceptible individuals vaccinated each day, by r_{vac} . The model is then

$$\frac{dP_S}{dt} = -\Lambda \frac{P_S}{P} - r_{\text{vac}}P_S + \Lambda - r_{\text{tra}}P_S \frac{P_I}{P} \quad (4.185)$$

$$\frac{dP_E}{dt} = -\Lambda \frac{P_E}{P} - \frac{P_E}{T_{\text{lat}}} + r_{\text{tra}}P_S \frac{P_I}{P} \quad (4.186)$$

$$\frac{dP_I}{dt} = -\Lambda \frac{P_I}{P} + \frac{P_E}{T_{\text{lat}}} - \frac{P_I}{T_{\text{inf}}} \quad (4.187)$$

$$\frac{dP_R}{dt} = -\Lambda \frac{P_R}{P} + r_{\text{vac}}P_S + \frac{P_I}{T_{\text{inf}}}. \quad (4.188)$$

The sum of equations (4.185)–(4.188) shows that $P = P_S + P_E + P_I + P_R$ is a constant. Hence we do not need to include the equation for P_R in our analysis, which is crucial since the P_R component would have introduced positive eigenvalues. The maangces corresponding to (4.1) are then

$$F_0 = \begin{pmatrix} \Lambda \\ 0 \\ 0 \end{pmatrix}, \quad F_1 = \begin{pmatrix} -\frac{\Lambda}{P} - r_{\text{vac}} & 0 & 0 \\ 0 & -\frac{\Lambda}{P} - \frac{1}{T_{\text{lat}}} & 0 \\ 0 & \frac{1}{T_{\text{lat}}} & -\frac{\Lambda}{P} - \frac{1}{T_{\text{inf}}} \end{pmatrix}, \quad (4.189)$$

$$F_2 = \begin{pmatrix} 0 & 0 & -\frac{r_{\text{tra}}}{P} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{r_{\text{tra}}}{P} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (4.190)$$

Since F_1 is a triangular matrix, its eigenvalues are located on its diagonal, so $\text{Re}(\lambda_1) = -\Lambda/P - \min\{r_{\text{vac}}, 1/T_{\text{lat}}, 1/T_{\text{inf}}\}$. Furthermore we can bound $P/\sqrt{3} \leq$

$\|u_{\text{in}}\| \leq P$, $\|F_0\| = \Lambda$, and $\|F_2\| = \sqrt{2}r_{\text{tra}}/P$, so

$$R \leq \frac{\sqrt{2}r_{\text{tra}} + \sqrt{3}\Lambda/P}{\min\{r_{\text{vac}}, 1/T_{\text{lat}}, 1/T_{\text{inf}}\} + \Lambda/P}. \quad (4.191)$$

We see that the condition for guaranteed convergence of Carleman linearization is $\sqrt{2}r_{\text{tra}} < \min\{r_{\text{vac}}, 1/T_{\text{lat}}, 1/T_{\text{inf}}\} - (\sqrt{3} - 1)$. Essentially, the Carleman method only converges if the (nonlinear) transmission is sufficiently slower than the (linear) decay.

To assess how restrictive this assumption is, we consider the SEIR parameters used in [126]. Note that they also included separate components for asymptomatic and hospitalized persons, but to simplify the analysis we include both of these components in the P_I component. In their work, they considered a city with approximately $P = 10^7$ inhabitants. In a specific period, they estimated a travel flux⁶ of $\Lambda \approx 0$ individuals per day, latent time $T_{\text{lat}} = 5.2$ days, infectious time $T_{\text{inf}} = 2.3$ days, and transmission rate $r_{\text{tra}} \approx 0.13$ days⁻¹. We let the initial condition be dominated by the susceptible component so that $\|u_{\text{in}}\| \approx P$, and we assume⁷ that $r_{\text{vac}} > 1/T_{\text{lat}} \approx 0.19$ days⁻¹. With the stated parameters, a direct calculation gives $R = 0.956$, showing that the assumptions of our algorithm can correspond to some real-world problems that are only moderately nonlinear.

While the example discussed above has only a constant number of variables, this example can be generalized to a high-dimensional system of ODEs that models the early

⁶Since we require that $u(t)$ does not approach 0 exponentially, we can assume that the travel flux is some non-zero, but small, value, e.g., $\Lambda = 1$ individual per day.

⁷This example arguably corresponds to quite rapid vaccination, and is chosen here such that R remains smaller than one, as required to formally guarantee convergence of the Carleman method. However, as shown in the upcoming example of the Burgers equation, larger values of R might still allow for convergence in practice, suggesting that our algorithm might handle lower values of the vaccination rate.

spread over a large number of cities with interaction, similar to what is done in [128] and [129].

Other examples of high-dimensional ODEs arise from the discretization of certain PDEs. Consider, for example, equations for $\mathbf{u}(\mathbf{r}, t)$ of the type

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \beta \mathbf{u} = \nu \nabla^2 \mathbf{u} + \mathbf{f}. \quad (4.192)$$

with the forcing term \mathbf{f} being a function of both space and time. This equation can be cast in the form of (4.1) by using standard discretizations of space and time. Equations of the form (4.192) can represent Navier–Stokes-type equations, which are ubiquitous in fluid mechanics [118], and related models such as those studied in [130, 131, 132] to describe the formation of large-scale structure in the universe. Similar equations also appear in models of magnetohydrodynamics (e.g., [133]), or the motion of free particles that stick to each other upon collision [134]. In the inviscid case, $\nu = 0$, the resulting Euler-type equations with linear damping are also of interest, both for modeling micromechanical devices [135] and for their intimate connection with viscous models [136].

As a specific example, consider the one-dimensional forced viscous Burgers equation

$$\partial_t u + u \partial_x u = \nu \partial_x^2 u + f, \quad (4.193)$$

which is the one-dimensional case of equation (4.192) with $\beta = 0$. Equation (4.193) is often used as a simple model of convective flow [117]. For concreteness, let the initial condition be $u(x, 0) = U_0 \sin(2\pi x/L_0)$ on the domain $x \in [-L_0/2, L_0/2]$, and use

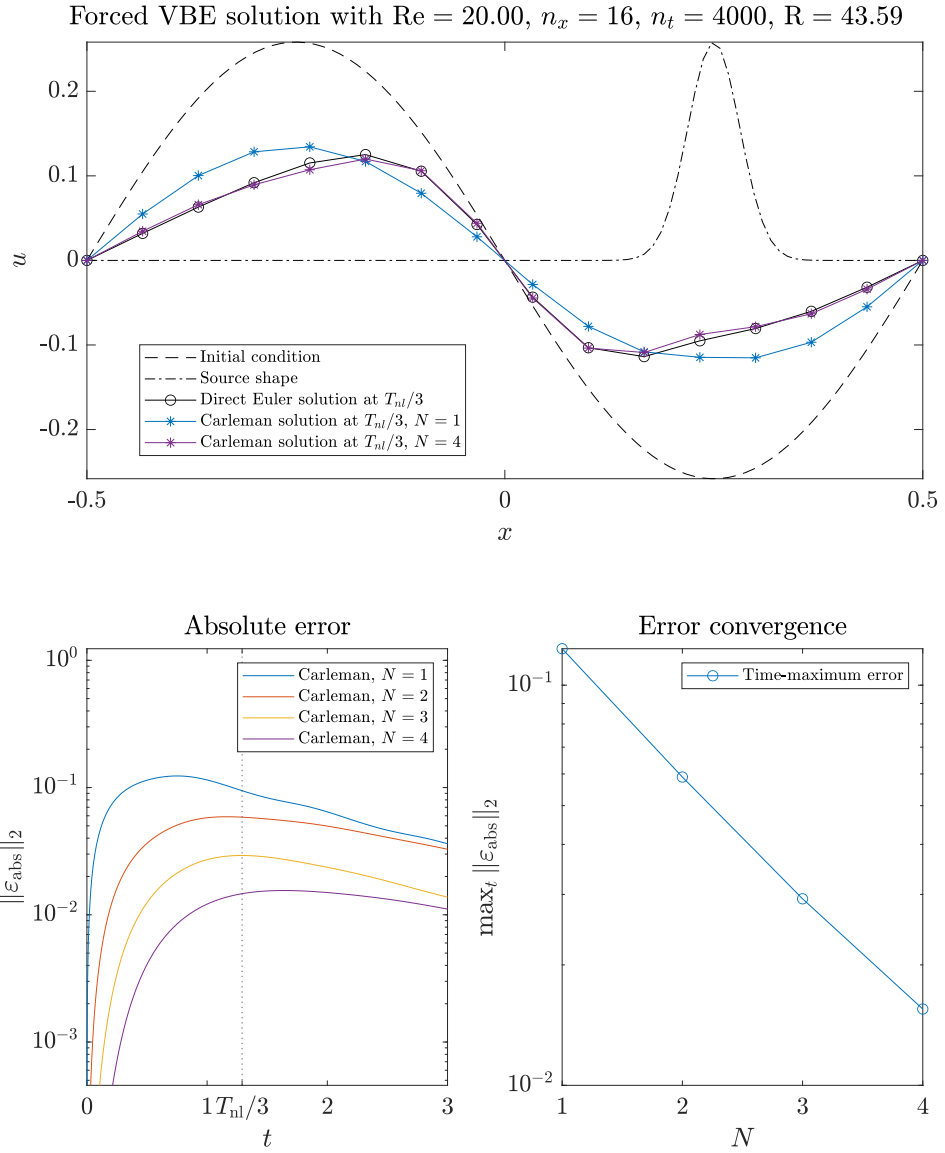


Figure 4.1: Integration of the forced viscous Burgers equation using Carleman linearization on a classical computer (source code available at <https://github.com/hermankol-den/CarlemanBurgers>). The viscosity is set so that the Reynolds number $Re = U_0 L_0 / \nu = 20$. The parameters $n_x = 16$ and $n_t = 4000$ are the number of spatial and temporal discretization intervals, respectively. The corresponding Carleman convergence parameter is $R = 43.59$. Top: Initial condition and solution plotted at a third of the nonlinear time $\frac{1}{3}T_{nl} = \frac{L_0}{3U_0}$. Bottom: l_2 norm of the absolute error between the Carleman solutions at various truncation levels N (left), and the convergence of the corresponding time-maximum error (right).

Dirichlet boundary conditions $u(-L_0/2, 0) = u(L_0/2, 0) = 0$. We force this equation using a localized off-center Gaussian with a sinusoidal time dependence,⁸ given by $f(x, t) = U_0 \exp\left(-\frac{(x - L_0/4)^2}{2(L_0/32)^2}\right) \cos(2\pi t)$. To solve this equation using the Carleman method, we discretize the spatial domain into n_x points and use central differences for the derivatives to get

$$\partial_t u_i = \nu \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} - \frac{u_{i+1}^2 - u_{i-1}^2}{4\Delta x} + f_i \quad (4.194)$$

with $\Delta x = L_0/(n_x - 1)$. This equation is of the form (4.1) and can thus generate the Carleman system (4.6). The resulting linear ODE can then be integrated using the forward Euler method, as shown in Figure 4.1. In this example, the viscosity ν is defined such that the Reynolds number $\text{Re} := U_0 L_0 / \nu = 20$, and $n_x = 16$ spatial discretization points were sufficient to resolve the solution. The figure compares the Carleman solution with the solution obtained via direct integration of (4.194) with the forward Euler method (i.e., without Carleman linearization). By inserting the matrices F_0 , F_1 , and F_2 corresponding to equation (4.194) into the definition of R (4.2), we find that $\text{Re}(\lambda_1)$ is indeed negative as required, given Dirichlet boundary conditions, but the parameters used in this example result in $R \approx 44$. Even though this does not satisfy the requirement $R < 1$ of the QCL algorithm, we see from the absolute error plot in Figure 4.1 that the maximum absolute error over time decreases exponentially as the truncation level N is incremented (in this example, the maximum Carleman truncation level considered is $N = 4$). Surprisingly, this suggests that in this example, the error of the classical Carleman method converges exponentially with N , even though $R > 1$.

⁸Note that this forcing does not satisfy the general conditions for efficient implementation of our algorithm since it is not sparse. However, we expect that the algorithm can still be implemented efficiently for a structured non-sparse forcing term such as in this example.

4.7 Discussion

In this chapter we have presented a quantum Carleman linearization (QCL) algorithm for a class of quadratic nonlinear differential equations. Compared to the previous approach of [74], our algorithm improves the complexity from an exponential dependence on T to a nearly quadratic dependence, under the condition $R < 1$ as defined in (4.2). Qualitatively, this means that the system must be dissipative and that the nonlinear and inhomogeneous effects must be small relative to the linear effects. We have also provided numerical results suggesting the classical Carleman method may work on certain PDEs that do not strictly satisfy the assumption $R < 1$. Furthermore, we established a lower bound showing that for general quadratic differential equations with $R \geq \sqrt{2}$, quantum algorithms must have worst-case complexity exponential in T . We also discussed several potential applications arising in biology and fluid and plasma dynamics.

It is natural to ask whether the result of [Theorem 4.1](#) can be achieved with a classical algorithm, i.e., whether the assumption $R < 1$ makes differential equations classically tractable. Clearly a naive integration of the truncated Carleman system (4.6) is not efficient on a classical computer since the system size is $\Theta(n^N)$. But furthermore, it is unlikely that *any* classical algorithm for this problem can run in time polylogarithmic in n . If we consider [Problem 4.1](#) with dissipation that is small compared to the total evolution time, but let the nonlinearity and forcing be even smaller such that $R < 1$, then in the asymptotic limit we have a linear differential equation with no dissipation. Hence any classical algorithm that could solve [Problem 4.1](#) could also solve non-dissipative linear differential equations, which is a BQP-hard problem even when the dynamics are

unitary [137]. In other words, an efficient classical algorithm for this problem would imply efficient classical algorithms for any problem that can be solved efficiently by a quantum computer, which is considered unlikely.

Our upper and lower bounds leave a gap in the range $1 \leq R < \sqrt{2}$, for which we do not know the complexity of the quantum quadratic ODE problem. We hope that future work will close this gap and determine for which R the problem can be solved efficiently by quantum computers in the worst case.

Furthermore, the complexity of our algorithm has nearly quadratic dependence on T , namely $T^2 \text{poly}(\log T)$. It is unknown whether the complexity for quadratic ODEs must be at least linear or quadratic in T . Note that sublinear complexity is impossible in general because of the no-fast-forwarding theorem [33]. However, it should be possible to fast-forward the dynamics in special cases, and it would be interesting to understand the extent to which dissipation enables this.

The complexity of our algorithm depends on the parameter q defined in [Theorem 4.1](#), which characterizes the decay of the final solution relative to the initial condition. This restricts the utility of our result, since we must have a suitable initial condition and terminal time such that the final state is not exponentially smaller than the initial state. However, it is unlikely that such a dependence can be significantly improved, since renormalization of the state can be used to implement postselection, which would imply the unlikely consequence $\text{BQP} = \text{PP}$ (see Section 8 of [53] for further discussion). As discussed in the introduction, the solution of a homogeneous dissipative equation necessarily decays exponentially in time, so our method is not asymptotically efficient. However, for inhomogeneous equations the final state need not be exponentially smaller

than the initial state even in a long-time simulation, suggesting that our algorithm could be especially suitable for models with forcing terms.

It is possible that variations of the Carleman linearization procedure could increase the accuracy of the result. For instance, instead of using just tensor powers of u as auxiliary variables, one could use other nonlinear functions. Several previous papers on Carleman linearization have suggested using multidimensional orthogonal polynomials [73, 138]. They also discuss approximating higher-order terms with lower-order ones in (4.6) instead of simply dropping them, possibly improving accuracy. Such changes would however alter the structure of the resulting linear ODE, which could affect the quantum implementation.

The quantum part of the algorithm might also be improved. In this chapter we limit ourselves to the first-order Euler method to discretize the linearized ODEs in time. This is crucial for the analysis in Lemma 4.3, which states the global error increases at most linearly with T . To establish this result for the Euler method, it suffices to choose the time step (4.65) to ensure $\|I + Ah\| \leq 1$, and then estimate the growth of global error by (4.92). However, it is unclear how to give a similar bound for higher-order numerical schemes. If this obstacle could be overcome, the error dependence of the complexity might be improved.

It is also natural to ask whether our approach can be improved by taking features of particular systems into account. Since the Carleman method has only received limited attention and has generally been used for purposes other than numerical integration, it seems likely that such improvements are possible. In fact, the numerical results discussed in Section 4.6 (see in particular Figure 4.1) suggest that the condition $R < 1$ is not a

strict requirement for the viscous Burgers equation, since we observe convergence even though $R \approx 44$. This suggests that some property of equation (4.193) makes it more amenable to Carleman linearization than our current analysis predicts. We leave a detailed investigation of this for future work.

A related question is whether our algorithm can efficiently simulate systems exhibiting dynamical chaos. The condition $R < 1$ might preclude chaos, but we do not have a proof of this. More generally, the presence or absence of chaos might provide a more fine-grained picture of the algorithm's efficiency.

When contemplating applications, it should be emphasized that our approach produces a state vector that encodes the solution without specifying how information is to be extracted from that state. Simply producing a state vector is not enough for an end-to-end application since the full quantum state cannot be read out efficiently. In some cases it may be possible to extract useful information by sampling a simple observable, whereas in other cases, more sophisticated postprocessing may be required to infer a desired property of the solution. Our method does not address this issue, but can be considered as a subroutine whose output will be parsed by subsequent quantum algorithms. We hope that future work will address this issue and develop end-to-end applications of these methods.

Finally, the algorithm presented in this chapter might be extended to solve related mathematical problems on quantum computers. Obvious candidates include initial value problems with time-dependent coefficients and boundary value problems. Carleman methods for such problems are explored in [83], but it is not obvious how to implement those methods in a quantum algorithm. It is also possible that suitable formulations of problems in nonlinear optimization or control could be solvable using related techniques.

Chapter 5: Conclusion and future work

In this dissertation, we developed an understanding of quantum algorithms for differential equations concerning their design, analysis, and applications.

In [Chapter 2](#), we have presented high-precision quantum algorithms to solve linear, time-dependent, d -dimensional systems of ordinary differential equations. Specifically, we showed how to employ a global approximation based on the spectral method as an alternative to the more straightforward finite difference method. Compared to the previous work [\[52\]](#), our algorithm improves the complexity of solving time-dependent linear differential equations from $\text{poly}(1/\epsilon)$ to $\text{poly}(\log(1/\epsilon))$.

In [Chapter 3](#), we have presented high-precision quantum algorithms for d -dimensional partial differential equations. We developed quantum adaptive finite difference methods for Poisson's equation, and generalized the quantum spectral method for general second-order elliptic equations. Whereas previous algorithms scaled as $\text{poly}(d, 1/\epsilon)$, our algorithms scale as $\text{poly}(d, \log(1/\epsilon))$.

In [Chapter 4](#), we have presented the first efficient quantum algorithm for a class of nonlinear differential equations, exponentially improving the dependence of the evolution time over previous quantum algorithms. The key ingredient we adopted and modified is the Carleman linearization, which provides a long-time stable linear approximation to the

nonlinear system. We developed quantum Carleman linearization (QCL) for dissipative nonlinear differential equations provided the condition $R < 1$. Here the quantity R is used to quantify the ℓ_2 relative strength of the nonlinearity and forcing to the linear dissipation. The complexity of this approach is $\tilde{\mathcal{O}}(T^2 q/\epsilon)$, where T is the evolution time, ϵ is the allowed error, and q measures the decay of the solution. Moreover, We also provided a quantum lower bound for the worst-case exponential time complexity of simulating general nonlinear dynamics given $R \geq \sqrt{2}$. Beyond the worst-case complexity analysis, the numerical experiments for the Burgers equation revealed that Carleman linearization could be practicable for certain nonlinear problems even for larger R .

Our work has raised enthusiasm from the communities in quantum physics, computer science, applied mathematics, fluid and plasma dynamics, and other fields. Detailed open questions and future directions followed by each topic can be found in the discussion sections in [Chapter 2](#), [Chapter 3](#), and [Chapter 4](#). In general, it is great interest for us to find answers to the following questions:

From the aspect of theories, can we offer exponential quantum speedups for more general differential equations with provable guarantees? It is possible that the assumptions on the models can be relaxed. For linear ODEs and PDEs, we wonder if we can develop high-precision quantum algorithms under weaker smoothness assumptions, and if we can relax the elliptic condition for second-order linear PDEs. And for nonlinear differential equations, we wonder whether the weak nonlinearity condition $R < 1$ can be relaxed. Numerical results in [Chapter 4](#) suggest that $R < 1$ is not a strict requirement for the viscous Burgers equation, since we observe convergence even though $R \approx 44$. This suggests that some properties of nonlinear systems make them more amenable to Carleman

linearization than our current analysis predicts. We also wonder whether the complexity of quantum algorithms for quadratic ODEs ($\tilde{T}^2 q/\epsilon$) can be improved. It is implausible to significantly reduce the dependence on q , since renormalization of the state can implement postselection, which could imply the unlikely consequence $\text{BQP} = \text{PP}$. The error dependence might be improved if the quantum adaptive FDM or the quantum spectral method could be applied to the nonlinear case, as a rigorous bound on the condition number of the resulting quantum linear system is required. It is unclear whether a linear or even sublinear dependence on T is achievable. While there is a no-fast-forwarding theorem for Hamiltonian simulations, it might be feasible to fast-forward certain initial and boundary value problems of linear and nonlinear differential equations. Despite that previous studies focus on the performance in terms of the worst-case error, we may find further improvements when considering the average-case error with random initial inputs of typical differential equations. Finally, it is also likely that the techniques presented in this dissertation could inspire new quantum algorithms for related problems in optimization, control, and machine learning.

From the aspect of applications, can we investigate end-to-end quantum applications for differential equations and related real-world problems? For the preprocessing procedure, we are concerned with how to efficiently prepare the initial states by loading classical data. The ability to prepare such quantum states requires the implementation of quantum random access memory (qRAM), which is widely thought to be impractical. But for some particular initial conditions such as the Gaussian distribution, it is promising to generate the initial states by a certain efficient routine instead of involving qRAM. For the postprocessing procedure, we are of great interest in extracting meaningful classical

observables of practical interest. Typical instances include the scattering cross section from a wave propagation, and the free energy from a physical or biological system. Our algorithms only produce a state vector that encodes the solution, which is not enough for an end-to-end application since the full quantum state cannot be read out efficiently. In some circumstances, it may be able to extract useful information by sampling a simple observable, whereas in other cases, more sophisticated postprocessing may be required to infer the desired property of the solution. For producing such classical readouts by quantum computers, we can investigate the complexity and figure out whether an exponential quantum speedup is possible. In addition, we are interested in the smallest system that could be used to demonstrate proof-of-principle versions of these algorithms. It is of great interest to estimate the implementation cost (e.g. the one- or two-qubit gate counts) for early fault-tolerant quantum computers. We hope that future work will address these issues toward end-to-end applications.

Bibliography

- [1] Michael A. Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.
- [2] John Watrous. *The theory of quantum information*. Cambridge university press, 2018.
- [3] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1982.
- [4] Seth Lloyd. Universal quantum simulators. *Science*, pages 1073–1078, 1996.
- [5] Ivan Kassal, Stephen P. Jordan, Peter J. Love, Masoud Mohseni, and Alán Aspuru-Guzik. Polynomial-time quantum algorithm for the simulation of chemical dynamics. *Proceedings of the National Academy of Sciences*, 105(48):18681–18686, 2008. [arXiv:0801.2986](#).
- [6] Ian D. Kivlichan, Nathan Wiebe, Ryan Babbush, and Alán Aspuru-Guzik. Bounding the costs of quantum simulation of many-body physics in real space. *Journal of Physics A: Mathematical and Theoretical*, 50(30):305301, 2017.
- [7] Borzu Toloui and Peter J. Love. Quantum algorithms for quantum chemistry based on the sparsity of the CI-matrix, 2013. [arXiv:1312.2579](#).
- [8] Ryan Babbush, Dominic W. Berry, Yuval R. Sanders, Ian D. Kivlichan, Artur Scherer, Annie Y. Wei, Peter J. Love, and Alán Aspuru-Guzik. Exponentially more precise quantum simulation of fermions in the configuration interaction representation. *Quantum Science and Technology*, 3(1):015006, 2017.
- [9] Ryan Babbush, Dominic W. Berry, Jarrod R. McClean, and Hartmut Neven. Quantum simulation of chemistry with sublinear scaling in basis size. *Npj Quantum Information*, 5(1):1–7, 2019.
- [10] Yuan Su, Dominic W Berry, Nathan Wiebe, Nicholas Rubin, and Ryan Babbush. Fault-tolerant quantum simulations of chemistry in first quantization. *PRX Quantum*, 2(4):040332, 2021. [arXiv:2105.12767](#).

- [11] Alán Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, 2005. [arXiv:quant-ph/0604193](#).
- [12] James D. Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. Simulation of electronic structure Hamiltonians using quantum computers. *Molecular Physics*, 109(5):735–750, 2011. [arXiv:1001.3855](#).
- [13] Jacob T. Seeley, Martin J. Richard, and Peter J. Love. The Bravyi-Kitaev transformation for quantum computation of electronic structure. *The Journal of Chemical Physics*, 137(22):224109, 2012. [arXiv:1208.5986](#).
- [14] Dave Wecker, Bela Bauer, Bryan K. Clark, Matthew B. Hastings, and Matthias Troyer. Gate-count estimates for performing quantum chemistry on small quantum computers. *Physical Review A*, 90(2):022305, 2014. [arXiv:1312.1695](#).
- [15] Matthew B. Hastings, Dave Wecker, Bela Bauer, and Matthias Troyer. Improving quantum algorithms for quantum chemistry. *Quantum Information & Computation*, 15(1-2):1–21, 2015. [arXiv:1403.1539](#).
- [16] David Poulin, Matthew B. Hastings, David Wecker, Nathan Wiebe, Andrew C. Doherty, and Matthias Troyer. The Trotter step size required for accurate quantum simulation of quantum chemistry. *Quantum Information & Computation*, 15(5-6):361–384, 2015. [arXiv:1406.4920](#).
- [17] Jarrod R. McClean, Ryan Babbush, Peter J. Love, and Alán Aspuru-Guzik. Exploiting locality in quantum computation for quantum chemistry. *The Journal of Physical Chemistry Letters*, 5(24):4368–4380, 2014.
- [18] Ryan Babbush, Jarrod McClean, Dave Wecker, Alán Aspuru-Guzik, and Nathan Wiebe. Chemical basis of Trotter-Suzuki errors in quantum chemistry simulation. *Physical Review A*, 91(2):022311, 2015. [arXiv:1410.8159](#).
- [19] Markus Reiher, Nathan Wiebe, Krysta M. Svore, Dave Wecker, and Matthias Troyer. Elucidating reaction mechanisms on quantum computers. *Proceedings of the National Academy of Sciences*, 114(29):7555–7560, 2017. [arXiv:1605.03590](#).
- [20] Ryan Babbush, Dominic W. Berry, Ian D. Kivlichan, Annie Y. Wei, Peter J. Love, and Alán Aspuru-Guzik. Exponentially more precise quantum simulation of fermions in second quantization. *New Journal of Physics*, 18(3):033032, 2016.
- [21] Mario Motta, Erika Ye, Jarrod R. McClean, Zhendong Li, Austin J. Minnich, Ryan Babbush, and Garnet Kin-Lic Chan. Low rank representations for quantum simulation of electronic structure. *npj Quantum Information*, 7(1):1–7, 2021.
- [22] Earl Campbell. Random compiler for fast Hamiltonian simulation. *Physical Review Letters*, 123(7):070503, 2019. [arXiv:1811.08017](#).

- [23] Dominic W. Berry, Craig Gidney, Mario Motta, Jarrod R. McClean, and Ryan Babbush. Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization. *Quantum*, 3:208, 2019. [arXiv:1902.02134](#).
- [24] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of Trotter error with commutator scaling. *Physical Review X*, 11(1):011020, 2021.
- [25] Vera von Burg, Guang Hao Low, Thomas Häner, Damian S. Steiger, Markus Reiher, Martin Roetteler, and Matthias Troyer. Quantum computing enhanced computational catalysis. *Physical Review Research*, 3(3):033055, 2021.
- [26] Joonho Lee, Dominic Berry, Craig Gidney, William J. Huggins, Jarrod R. McClean, Nathan Wiebe, and Ryan Babbush. Even more efficient quantum computations of chemistry through tensor hypercontraction. *PRX Quantum*, 2(3):030305, 2021. [arXiv:2011.03494](#).
- [27] Yuan Su, Hsin-Yuan Huang, and Earl T. Campbell. Nearly tight Trotterization of interacting electrons. *Quantum*, 5:495, 2021. [arXiv:2012.09194](#).
- [28] Stephen P. Jordan, Keith S.M. Lee, and John Preskill. Quantum algorithms for quantum field theories. *Science*, 336(6085):1130–1133, 2012.
- [29] John Preskill. Simulating quantum field theory with a quantum computer. In *The 36th Annual International Symposium on Lattice Field Theory*, volume 334, page 024. SISSA Medialab, 2019. [arXiv:1811.10085](#).
- [30] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews*, 120(22):12685–12717, 2020. [arXiv:2001.03685](#).
- [31] Yudong Cao, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, et al. Quantum chemistry in the age of quantum computing. *Chemical Reviews*, 119(19):10856–10915, 2019. [arXiv:1812.09976](#).
- [32] Ryan Babbush, Nathan Wiebe, Jarrod McClean, James McClain, Hartmut Neven, and Garnet Kin-Lic Chan. Low-depth quantum simulation of materials. *Physical Review X*, 8(1):011044, 2018.
- [33] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. Efficient quantum algorithms for simulating sparse Hamiltonians. *Communications in Mathematical Physics*, 270(2):359–371, 2007. [arXiv:quant-ph/0508139](#).
- [34] David Poulin, Angie Qarry, Rolando D. Somma, and Frank Verstraete. Quantum simulation of time-dependent Hamiltonians and the convenient illusion of Hilbert space. *Physical Review Letters*, 106(17):170501, 2011. [arXiv:1102.1360](#).

- [35] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse Hamiltonians. *Forum of Mathematics, Sigma*, 5:e8, 2017. [arXiv:1312.1414](#).
- [36] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Physical Review Letters*, 114(9):090502, 2015. [arXiv:1412.4687](#).
- [37] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 792–809. IEEE, 2015. [arXiv:1501.01715](#).
- [38] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002. [arXiv:quant-ph/0005055](#).
- [39] Dominic W. Berry and Leonardo Novo. Corrected quantum walk for optimal hamiltonian simulation. [arXiv:1606.03443](#), 2016.
- [40] Leonardo Novo and Dominic W Berry. Improved hamiltonian simulation via a truncated Taylor series and corrections. [arXiv:1611.10033](#), 2016.
- [41] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019. [arXiv:1610.06546](#).
- [42] Guang Hao Low and Isaac L. Chuang. Optimal hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118(1):010501, 2017. [arXiv:1606.02685](#).
- [43] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. [arXiv:0811.3171](#).
- [44] Andris Ambainis. Variable time amplitude amplification and quantum algorithms for linear algebra problems. In *29th Symposium on Theoretical Aspects of Computer Science*, volume 14, pages 636–647. LIPIcs, 2012. [arXiv:1010.4458](#).
- [45] Dong An and Lin Lin. Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm, 2019. [arXiv:1909.05500](#).
- [46] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6):1920–1950, 2017. [arXiv:1511.02306](#).

- [47] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019. [arXiv:1806.01838](#).
- [48] Lin Lin and Yu Tong. Optimal quantum eigenstate filtering with application to solving quantum linear systems. *Quantum*, 4:361, 2020. [arXiv:1910.14596](#).
- [49] Yiğit Subaşı, Rolando D. Somma, and Davide Orsucci. Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing. *Physical Review Letters*, 122(6):060504, 2019. [arXiv:1805.10549](#).
- [50] Yu Tong, Dong An, Nathan Wiebe, and Lin Lin. Fast inversion, preconditioned quantum linear system solvers, and fast evaluation of matrix functions, 2020. [arXiv:2008.13295](#).
- [51] Pedro Costa, Dong An, Yuval R. Sanders, Yuan Su, Ryan Babbush, and Dominic W. Berry. Optimal scaling quantum linear systems solver via discrete adiabatic theorem, 2021. [arXiv:2111.08152](#).
- [52] Dominic W. Berry. High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical*, 47(10):105301, 2014. [arXiv:1010.2745](#).
- [53] Dominic W. Berry, Andrew M. Childs, Aaron Ostrander, and Guoming Wang. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Communications in Mathematical Physics*, 356(3):1057–1081, 2017. [arXiv:1701.03684](#).
- [54] Andrew M. Childs and Jin-Peng Liu. Quantum spectral methods for differential equations. [arXiv:1901.00961](#), 2019.
- [55] B. David Clader, Bryan C. Jacobs, and Chad R. Sprouse. Preconditioned quantum linear system algorithm. *Physical Review Letters*, 110(25):250504, 2013. [arXiv:1301.2340](#).
- [56] Yudong Cao, Anargyros Papageorgiou, Iasonas Petras, Joseph Traub, and Sabre Kais. Quantum algorithm and circuit design solving the Poisson equation. *New Journal of Physics*, 15(1):013021, 2013. [arXiv:1207.2485](#).
- [57] Ashley Montanaro and Sam Pallister. Quantum algorithms and the finite element method. *Physical Review A*, 93(3):032324, 2016. [arXiv:1512.05903](#).
- [58] Pedro C. S. Costa, Stephen Jordan, and Aaron Ostrander. Quantum algorithm for simulating the wave equation. *Physical Review A*, 99(1):012323, 2019. [arXiv:1711.05394](#).
- [59] Andrew M. Childs, Jin-Peng Liu, and Aaron Ostrander. High-precision quantum algorithms for partial differential equations, 2020. [arXiv:2002.07868](#).

- [60] Alexander Engel, Graeme Smith, and Scott E. Parker. Quantum algorithm for the Vlasov equation. *Physical Review A*, 100(6):062315, 2019. [arXiv:1907.09418](#).
- [61] Noah Linden, Ashley Montanaro, and Changpeng Shao. Quantum vs. classical algorithms for solving the heat equation. [arXiv:2004.06516](#).
- [62] Alexei Y. Kitaev. Quantum measurements and the abelian stabilizer problem, 1995. [arXiv:quant-ph/9511026](#).
- [63] David Kahaner, Cleve Moler, and Stephen Nash. *Numerical methods and software*. Prentice-Hall, Inc., 1989.
- [64] Rainer Kress. *Numerical Analysis: V. 181*. Springer Verlag, 1998.
- [65] Esmail Babolian and Mohammad Mahdi Hosseini. A modified spectral method for numerical solution of ordinary differential equations with non-analytic solution. *Applied Mathematics and Computation*, 132(2-3):341–351, 2002.
- [66] Mohammad Mahdi Hosseini. A modified pseudospectral method for numerical solution of ordinary differential equations systems. *Applied mathematics and computation*, 176(2):470–475, 2006.
- [67] Călin Ioan Gheorghiu. *Spectral methods for differential problems*. Casa Cărții de Știință Cluj-Napoca, 2007.
- [68] Jie Shen, Tao Tang, and Li-Lian Wang. *Spectral methods: algorithms, analysis and applications*, volume 41. Springer Science & Business Media, 2011.
- [69] Yudong Cao, Anargyros Papageorgiou, Iasonas Petras, Joseph Traub, and Sabre Kais. Quantum algorithm and circuit design solving the Poisson equation. *New Journal of Physics*, 15(1):013021, 2013. [arXiv:1207.2485](#).
- [70] Pedro C. S. Costa, Stephen Jordan, and Aaron Ostrander. Quantum algorithm for simulating the wave equation. *Physical Review A*, 99(1):012323, 2019. [arXiv:1711.05394](#).
- [71] Ivo Babuška and Manil Suri. The $h - p$ version of the finite element method with quasiuniform meshes. *Mathematical Modelling and Numerical Analysis*, 21(2):199–238, 1987.
- [72] Edward H. Kerner. Universal formats for nonlinear ordinary differential systems. *Journal of Mathematical Physics*, 22(7):1366–1371, 1981.
- [73] Marcelo Forets and Amaury Pouly. Explicit error bounds for Carleman linearization, 2017. [arXiv:1711.02552](#).
- [74] Sarah K. Leyton and Tobias J. Osborne. A quantum algorithm to solve nonlinear differential equations. [arXiv:0812.4423](#), 2008.

- [75] Ilon Joseph. Koopman-von Neumann approach to quantum simulation of nonlinear classical dynamics, 2020. [arXiv:2003.09980](#).
- [76] Ilya Y. Dodin and Edward A. Startsev. On applications of quantum computing to plasma simulations, 2020. [arXiv:2005.14369](#).
- [77] Seth Lloyd, Giacomo De Palma, Can Gokler, Bobak Kiani, Zi-Wen Liu, Milad Marvian, Felix Tennie, and Tim Palmer. Quantum algorithm for nonlinear differential equations, 2020. [arXiv:2011.06571](#).
- [78] Andrew M. Childs and Joshua Young. Optimal state discrimination and unstructured search in nonlinear quantum mechanics. *Physical Review A*, 93(2):022314, 2016. [arXiv:1507.06334](#).
- [79] Daniel S. Abrams and Seth Lloyd. Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and #P problems. *Physical Review Letters*, 81(18):3992, 1998. [arXiv:quant-ph/9801041](#).
- [80] Scott Aaronson. NP-complete problems and physical reality. *ACM SIGACT News*, 36(1):30–52, 2005. [arXiv:quant-ph/0502072](#).
- [81] Jin-Peng Liu, Herman Øie Kolden, Hari K. Krovi, Nuno F. Loureiro, Konstantina Trivisa, and Andrew M. Childs. Efficient quantum algorithm for dissipative nonlinear differential equations, 2020. [arXiv:2011.03185](#).
- [82] Torsten Carleman. Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires. *Acta Mathematica*, 59(1):63–87, 1932.
- [83] Krzysztof Kowalski and Willi-Hans Steeb. *Nonlinear Dynamical Systems and Carleman Linearization*. World Scientific, 1991.
- [84] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. [arXiv:0811.3171](#).
- [85] Tom M. Apostol. Calculus, volume ii: multi-variable calculus and linear algebra, with applications to differential equations and probability, 1969.
- [86] Sangtae Kim and Seppo J. Karrila. *Microhydrodynamics: principles and selected applications*. Courier Corporation, 2013.
- [87] R. K. Michael Thambynayagam. *The diffusion handbook: applied solutions for engineers*. McGraw Hill Professional, 2011.
- [88] Stewart Harris. *An introduction to the theory of the Boltzmann equation*. Courier Corporation, 2004.
- [89] John David Jackson. Classical electrodynamics, 1999.

- [90] Bo Thidé. *Electromagnetic field theory*. Upsilon books Uppsala, 2004.
- [91] R. Byron Bird. Transport phenomena. *Appl. Mech. Rev.*, 55(1):R1–R4, 2002.
- [92] Vivek V. Shende, Stephen S. Bullock, and Igor L. Markov. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, 2006. [arXiv:quant-ph/0406176](https://arxiv.org/abs/quant-ph/0406176).
- [93] Andrew M Childs. On the relationship between continuous-and discrete-time quantum walk. *Communications in Mathematical Physics*, 294(2):581–603, 2010. [arXiv:0810.0312](https://arxiv.org/abs/0810.0312).
- [94] François Fillion-Gourdeau and Emmanuel Lorin. Simple digital quantum algorithm for symmetric first-order linear hyperbolic systems. *Numerical Algorithms*, 82(3):1009–1045, 2019. [arXiv:1705.09361](https://arxiv.org/abs/1705.09361).
- [95] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.
- [96] Christoph Zenger. Sparse grids, 1991. <https://www5.in.tum.de/pub/zenger91sg.pdf>.
- [97] Jie Shen and Haijun Yu. Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems. *SIAM Journal on Scientific Computing*, 32(6):3228–3250, 2010.
- [98] Jie Shen and Haijun Yu. Efficient spectral sparse grid methods and applications to high-dimensional elliptic equations II. Unbounded domains. *SIAM Journal on Scientific Computing*, 34(2):A1141–A1164, 2012.
- [99] Lawrence C. Evans. *Partial differential equations (2nd ed.)*. American Mathematical Society, 2010.
- [100] Jianping Li. General explicit difference formulas for numerical differentiation. *Journal of Computational and Applied Mathematics*, 183(1):29–52, 2005.
- [101] Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge University Press, 2012.
- [102] Norbert Schuch and Jens Siewert. Programmable networks for quantum algorithms. *Physical Review Letters*, 91(2):027902, 2003. [arXiv:quant-ph/0303063](https://arxiv.org/abs/quant-ph/0303063).
- [103] Jonathan Welch, Daniel Greenbaum, Sarah Mostame, and Alán Aspuru-Guzik. Efficient quantum circuits for diagonal unitaries without ancillas. *New Journal of Physics*, 16(3):033040, 2014. [arXiv:1306.3991](https://arxiv.org/abs/1306.3991).
- [104] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Physical Review Letters*, 114(9):090502, 2015. [arXiv:1412.4687](https://arxiv.org/abs/1412.4687).

- [105] Daniel T. Colbert and William H. Miller. A novel discrete variable representation for quantum mechanical reactive scattering via the S-matrix Kohn method. *The Journal of Chemical Physics*, 96(3):1982–1991, 1992.
- [106] Daniel Spielman. Rings, paths, and Cayley graphs (course notes), 2014. <http://www.cs.yale.edu/homes/spielman/561/lect05-15.pdf>.
- [107] Tao Tang. *Spectral and high-order methods with applications*. Science Press Beijing, 2006.
- [108] Andreas Klappenecker and Martin Rötteler. Discrete cosine transforms on quantum computers. In *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis*, pages 464–468, 2001. [arXiv:quant-ph/0111038](https://arxiv.org/abs/quant-ph/0111038).
- [109] Martin Rötteler, Markus Püschel, and Thomas Beth. Fast signal transforms for quantum computers. In *Proceedings of the Workshop on Physics and Computer Science*, pages 31–43, 1999.
- [110] Markus Püschel, Martin Rötteler, and Thomas Beth. Fast quantum Fourier transforms for a class of non-abelian groups. In *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, pages 148–159, 1999. [arXiv:quant-ph/9807064](https://arxiv.org/abs/quant-ph/9807064).
- [111] Willi-Hans Steeb and F. Wilhelm. Non-linear autonomous systems of differential equations and Carleman linearization procedure. *Journal of Mathematical Analysis and Applications*, 77(2):601–611, 1980.
- [112] Roberto F. S. Andrade. Carleman embedding and Lyapunov exponents. *Journal of Mathematical Physics*, 23(12):2271–2275, 1982.
- [113] Gerasimos Lyberatos and Christos A. Tsiliogiannis. A linear algebraic method for analysing Hopf bifurcation of chemical reaction systems. *Chemical Engineering Science*, 42(5):1242–1244, 1987.
- [114] Roger Brockett. The early days of geometric nonlinear control. *Automatica*, 50(9):2203–2224, 2014.
- [115] Andreas Rauh, Johanna Minisini, and Harald Aschemann. Carleman linearization for control and for state and disturbance estimation of nonlinear dynamical processes. *IFAC Proceedings Volumes*, 42(13):455–460, 2009.
- [116] Alfredo Germani, Costanzo Manes, and Pasquale Palumbo. Filtering of differential nonlinear systems via a Carleman approximation approach. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 5917–5922, 2005.
- [117] Johannes M. Burgers. A mathematical model illustrating the theory of turbulence. In *Advances in Applied Mechanics*, volume 1, pages 171–199. Elsevier, 1948.

- [118] Pierre Gilles Lemarié-Rieusset. *The Navier-Stokes problem in the 21st century*. CRC Press, 2018.
- [119] Germund G. Dahlquist. A special stability problem for linear multistep methods. *BIT Numerical Mathematics*, 3(1):27–43, 1963.
- [120] Kendall E. Atkinson. *An introduction to numerical analysis*. John Wiley & Sons, 2008.
- [121] Carl W. Helstrom. Quantum detection and estimation theory. *Journal of Statistical Physics*, 1:231–252, 1969.
- [122] Paul Waltman. *Competition models in population biology*. SIAM, 1983.
- [123] Elliott W. Montroll. On coupled rate equations with quadratic nonlinearities. *Proceedings of the National Academy of Sciences*, 69(9):2532–2536, 1972.
- [124] Alessandro Ceccato, Paolo Nicolini, and Diego Frezzato. Recasting the mass-action rate equations of open chemical reaction networks into a universal quadratic format. *Journal of Mathematical Chemistry*, 57:1001–1018, 2019.
- [125] Fred Brauer and Carlos Castillo-Chavez. *Mathematical models in population biology and epidemiology*, volume 2. Springer, 2012.
- [126] Chaolong Wang, Li Liu, Xingjie Hao, Huan Guo, Qi Wang, Jiao Huang, Na He, Hongjie Yu, Xihong Lin, An Pan, Sheng Wei, and Tangchun Wu. Evolving epidemiology and impact of non-pharmaceutical interventions on the outbreak of coronavirus disease 2019 in Wuhan, China. *Journal of the American Medical Association*, 323(19):1915–1923, 2020.
- [127] Gul Zaman, Yong Han Kang, and Il Hyo Jung. Stability analysis and optimal vaccination of an SIR epidemic model. *Biosystems*, 93(3):240–249, 2008.
- [128] Derdei Bichara, Yun Kang, Carlos Castillo-Chávez, Richard Horan, and Charles Perrings. SIS and SIR epidemic models under virtual dispersal, 03 2015.
- [129] Rany Qurratu Aini, Deden Aldila, and Kiki Sugeng. Basic reproduction number of a multi-patch SVI model represented as a star graph topology, 10 2018.
- [130] Lev Kofman, Dmitri Pogosian, and Sergei Shandarin. Structure of the universe in the two-dimensional model of adhesion. *Monthly Notices of the Royal Astronomical Society*, 242(2):200–208, 1990.
- [131] Sergei F. Shandarin and Yakov B. Zeldovich. The large-scale structure of the universe: turbulence, intermittency, structures in a self-gravitating medium. *Reviews of Modern Physics*, 61(2):185–220, 1989.
- [132] Massimo Vergassola, Bérengère Dubrulle, Uriel Frisch, and Alain Noullez. Burgers’ equation, devil’s staircases and the mass distribution for large-scale structures. *Astronomy and Astrophysics*, 289:325–356, 1994.

- [133] P. A. Davidson. *An Introduction to Magnetohydrodynamics*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2001.
- [134] Yann Brenier and Emmanuel Grenier. Sticky particles and scalar conservation laws. *SIAM Journal on Numerical Analysis*, 35(6):2317–2328, 1998.
- [135] Min-Hang Bao. *Micro mechanical transducers: pressure sensors, accelerometers and gyroscopes*. Elsevier, 2000.
- [136] Constantine M. Dafermos. *Hyperbolic conservation laws in continuum physics*, volume 3. Springer, 2005.
- [137] Richard P. Feynman. Quantum mechanical computers. *Optics News*, 11:11–20, 1985.
- [138] Richard Bellman and John M. Richardson. On some questions arising in the approximate solution of nonlinear differential equations. *Quarterly of Applied Mathematics*, 20:333–339, 1963.