Quantum algorithms for linear differential equations and eigenvalue transformations via linear combination of Hamiltonian simulation[a]

Dong An[b], Andrew Childs[c], Lin Lin[d], and Lexing Ying[e]

Introduction: Fault-tolerant quantum computers are expected to efficiently perform various unitary transformations of practical interest, notably including Hamiltonian dynamics. However, many tasks in scientific and engineering computation naturally involve non-unitary transformations. One of the most widely applicable tasks is solving a system of linear ordinary differential equations (ODEs) in the form

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = -A(t)u(t), \quad u(0) = u_0. \tag{1}$$

When $A(t) \equiv A$ is time-independent and non-normal, implementing Eq. (1) means performing the operation e^{-AT}, which can be viewed as an example of an eigenvalue transform. More generally, eigenvalue transformation aims to perform the matrix function $h(A) = \mathcal{V}h(D)\mathcal{V}^{-1}$ for a matrix diagonalized as $A = \mathcal{V}D\mathcal{V}^{-1}$. This task appears ubiquitously across various computational scientific domains. One example is the simulation of linear ODEs with a non-diagonal mass matrix in the form of

$$A\frac{\mathrm{d}u(t)}{\mathrm{d}t} = -u(t), \quad 0 \le t \le T, \tag{2}$$

which corresponds to performing the matrix function $e^{-TA^{-1}}$. Solving Eq. (2) is very different from solving Eq. (1) since the coefficient matrix A appearing on the left leads to an extra matrix inversion step. Besides solving linear ODEs in the form of Eq. (1) or Eq. (2), examples of eigenvalue transformation include non-Hermitian quantum dynamics, ODEs with inhomogeneous source terms, and powers of matrix inverses A^{-p} for $p > 0$, among others. Therefore, efficient quantum algorithms for ODEs[1] and eigenvalue transformations are desired to leverage the power of quantum computers for further computational speedup.

Contribution: We propose efficient, high-precision quantum algorithms for linear ODEs and performing a class of eigenvalue transformations that can be expressed as a certain type of matrix Laplace transformation. Our algorithms are a significant enhancement of the recent linear combination of Hamiltonian simulation (LCHS) method for ODEs [3] in two aspects.

1. We exponentially enhance the accuracy of the original LCHS method, enabling quantum algorithms for linear ODEs with near-optimal complexity for all parameters **for the first time**. [1]
2. By combining the Laplace transform and LCHS, we propose a new method (Lap-LCHS) that significantly extends the scope of LCHS for tasks beyond simulating ODEs for **a wide class** of eigenvalue transformations. [2]

Improved LCHS for ODEs: Our first main result is a relationship between the time-evolution operator of Eq. (1) and Hamiltonian simulation.

Theorem 1 (Improved LCHS for ODEs). *Let $A(t) = L(t) + iH(t)$, where $L(t) = \frac{A(t)+A^\dagger(t)}{2}$ and $H(t) = \frac{A(t)-A^\dagger(t)}{2i}$ are both Hermitian matrices. When $L(t) \succeq 0$, we have*

$$\mathcal{T}e^{-\int_0^t A(s)\,\mathrm{d}s} = \int_{\mathbb{R}} \frac{f(k)}{1-ik}\mathcal{T}e^{-i\int_0^t (kL(s)+H(s))\,\mathrm{d}s}\,\mathrm{d}k. \tag{3}$$

Here $f(k) = \frac{1}{2\pi e^{-2^\beta} e^{(1+iz)^\beta}}$, $\beta \in (0,1)$.

[a] This submission is based on two technical papers [1, 2].

[b] Beijing International Center for Mathematical Research, Peking University

[c] Joint Center for Quantum Information and Computer Science, Department of Computer Science, and Institute for Advanced Computer Studies, University of Maryland

[d] Department of Mathematics, University of California, Berkeley; Lawrence Berkeley National Laboratory

[e] Department of Mathematics and Institute for Computational and Mathematical Engineering, Stanford University

[1] In this work, quantum ODE algorithms only refer to those for Eq. (1). Algorithms for Eq. (2) will be regarded as an application of general quantum eigenvalue transformation algorithms.

Equation (3) represents the non-unitary time-evolution operator $\mathcal{T}e^{-\int_0^t A(s)\,\mathrm{d}s}$ of the ODE Eq. (1) as a continuous weighted linear combination of unitary operators $\mathcal{T}e^{-i\int_0^t (H(s)+kL(s))\,\mathrm{d}s}$ of Hamiltonian simulation problems with k-dependent Hamiltonians $kL(s)+H(s)$. The weight function $f(k)$ in Eq. (3) is called the *kernel function*. In the first technical paper [1], we show that there exists a large family of kernel functions that satisfy Eq. (3). The one we specify in Theorem 1 is near-optimal in the sense that it achieves almost the fastest possible asymptotic decay $e^{-c|k|^\beta}$ along the real axis. This leads to an algorithm whose complexity is nearly optimal as a function of the error ϵ, with an additional $\mathcal{O}((\log(1/\epsilon))^{1/\beta})$ factor (see Table I) that cannot be improved by modifying the kernel function within the current LCHS framework.

Eigenvalue transformation: Our second main result is the following Lap-LCHS representation for general eigenvalue transformations beyond ODEs.

Theorem 2 (Lap-LCHS). *Let $A = L + iH$, where $L = \frac{A+A^\dagger}{2}$ and $H = \frac{A-A^\dagger}{2i}$ are both Hermitian matrices. For a function $h(z)$, when $L \succeq 0$, we have*

$$h(A) = \int_0^\infty \int_{\mathbb{R}} \frac{f(k)g(t)}{1-ik} e^{-it(kL+H)}\,\mathrm{d}k\,\mathrm{d}t. \tag{4}$$

Here $f(k) = \frac{1}{2\pi e^{-2^\beta} e^{(1+iz)^\beta}}$, $\beta \in (0,1)$, and $g(t)$ is the inverse Laplace transform of $h(z)$ such that $h(z) = \int_0^\infty g(t)e^{-zt}\,\mathrm{d}t$, $\operatorname{Re} z \geq 0$.

Similarly to Theorem 1, Theorem 2 represents the matrix function $h(A)$ as a double integral of continuously parameterized Hamiltonian simulation problems $e^{-it(kL+H)}$. It can be derived by taking the matrix version of the Laplace transform $h(A) = \int_0^\infty g(t)e^{-At}\,\mathrm{d}t$ and then further expanding e^{-At} using Theorem 1. Notice that to implement a given function $h(z)$, we must find its inverse Laplace transform $g(t)$, which is not always well defined, and ensure that $g(t)$ is integrable. This restricts the functions of interest to a subspace of so-called Hardy functions on the right half of the complex plane.

Algorithms: Here, we discuss how to construct quantum algorithms for eigenvalue transformation based on Lap-LCHS (Theorem 2). Quantum algorithms for ODEs based on the improved LCHS (Theorem 1) can be constructed in the same way.

To implement Eq. (4), we first discretize its integrals and approximate the desired transformation as $h(A) \approx \sum_{j,l} c_{j,l} U_{k_j}(t_l)$, where the $c_{j,l}$ are weights and $U_{k_j}(t_l) = e^{-it_l(k_jL+H)}$. Then a quantum algorithm for eigenvalue transformation can be directly constructed in two steps. Step 1: We implement each unitary operator $U_{k_j}(t_l)$ by the state-of-the-art time-dependent Hamiltonian simulation algorithm based on quantum singular value transformation [4][2]. Step 2: We perform the linear combination $\sum_{j,l} c_{j,l} U_{k_j}(t_l)$ either coherently by the linear combination of unitaries (LCU) technique [7, 8] or stochastically by Monte Carlo sampling [9, 10].

Comparison of quantum ODE algorithms: During the past decade, several quantum linear ODE algorithms have been proposed. They can be roughly divided into two categories: linear-system-based approaches [11–15] (which discretize the ODE in time, converts the discretized ODE into a linear system of equations, and solve the resulting linear system with quantum linear system algorithms [8, 16–21]) and evolution-based approaches [3, 22**?** , 23] (which directly encode the time evolution operator using a unitary matrix that can be efficiently implemented on a quantum computer). In Table I, we compare our improved LCHS algorithm with the truncated Dyson series method [15] (the best linear-system-based approach) and the original LCHS method [3] (the best evolution-based approach prior to our work). We perform this comparison in terms of two primary factors: the number of queries to the matrix $A(t)$ and the number of queries to the initial state $u(0)$. Compared with the truncated Dyson series method, our improved LCHS achieves the lower state preparation cost of the original LCHS method while at least matching the complexity in queries to the matrix. This is because the improved LCHS approach avoids using quantum linear system algorithms, which typically require many queries to the state preparation oracle.

[2] To obtain a quantum ODE algorithm with time-dependent $A(t)$, we implement the Hamiltonian simulation $U_{k_j}(t)$ by the truncated Dyson series method [5, 6].

Method	**Queries to** $A(t)$	**Queries to** $\lvert u_0\rangle$
Truncated Dyson [15]	$\widetilde{\mathcal{O}}\left(\frac{\\|u_0\\|}{\\|u(T)\\|}\alpha_A T\left(\log\left(\frac{1}{\epsilon}\right)\right)^2\right)$	$\mathcal{O}\left(\frac{\\|u_0\\|}{\\|u(T)\\|}\alpha_A T\log\left(\frac{1}{\epsilon}\right)\right)$
Original LCHS [3]	$\widetilde{\mathcal{O}}\left(\left(\frac{\\|u_0\\|}{\\|u(T)\\|}\right)^2\alpha_A T/\epsilon\right)$	$\mathcal{O}\left(\frac{\\|u_0\\|}{\\|u(T)\\|}\right)$
Improved LCHS (our work)	$\widetilde{\mathcal{O}}\left(\frac{\\|u_0\\|}{\\|u(T)\\|}\alpha_A T\left(\log\left(\frac{1}{\epsilon}\right)\right)^{1+1/\beta}\right)$	$\mathcal{O}\left(\frac{\\|u_0\\|}{\\|u(T)\\|}\right)$

TABLE I. Comparison among improved LCHS and previous methods for ODEs. Here $\alpha_A \geq \|A(t)\|$, T is the evolution time, ϵ is the error and $\beta \in (0,1)$ is the parameter in the kernel function of the improved LCHS.

The original LCHS was proposed in [3] and represents the time evolution operator as

$$\mathcal{T}e^{-\int_0^t A(s)\,\mathrm{d}s} = \int_{\mathbb{R}} \frac{1}{\pi(1+k^2)}\mathcal{T}e^{-i\int_0^t (kL(s)+H(s))\,\mathrm{d}s}\,\mathrm{d}k. \tag{5}$$

Notice that Eq. (5) can be viewed as another version of Eq. (3) in Theorem 1 with a different kernel $f(k) = \frac{1}{1+ik}$. The drawback of this approach is the slow decay of its kernel function, resulting in non-optimal scaling with precision. Specifically, since $\frac{1}{\pi(1+k^2)}$ decays quadratically, we need to choose $\max|k_j| = \mathcal{O}(1/\epsilon)$ when numerically discretizating the integral to bound the error by $\mathcal{O}(\epsilon)$. This directly introduces a computational overhead of $\mathcal{O}(1/\epsilon)$ in the algorithm. Our improved LCHS in Theorem 1 uses a kernel function that decays as $e^{-c|k|^\beta}$, exponentially faster than in the original LCHS. Consequently, we can instead use the much smaller choice $\max|k_j| = \mathcal{O}((\log(1/\epsilon))^{1/\beta})$, leading to an exponential improvement in the computational cost with respect to ϵ.

Comparison of quantum transformation algorithms: The well-known quantum singular value transformation (QSVT) algorithm [4] implements matrix functions defined in terms of singular value transformations (SVTs), while most scientific and engineering computation tasks naturally require eigenvalue transformations. Certain eigenvalue transformation problems, such as the matrix inverse and functions of normal matrices, can be formulated as SVT problems. However, as eigenvalues and singular values can be very different for general non-normal matrices, most eigenvalue transformation problems cannot be reformulated as SVTs, and our Lap-LCHS approach can perform eigenvalue transformations that are beyond the scope of QSVT.

To our knowledge, there were two strategies for implementing general quantum eigenvalue transformations prior to our work. The first strategy is the contour integral-based formulation [24, 25], which represents a matrix function as an integral over matrix inverses and implements the integral by LCU. The second strategy is the recently developed quantum eigenvalue transformation (QEVT) with history state polynomial (HSP) [26], which implements eigenvalue transformations through the Chebyshev and Faber approximations.

The contour integral method, QEVT with HSP, and Lap-LCHS differ in their ranges of applicability, making it difficult to provide a succinct general comparison of these methods. As a rule of thumb, the contour integral method requires the entire contour to be within the stable region of the matrix function in the complex plane, imposing further restrictions on the range of matrix eigenvalues. QEVT with HSP requires solving a quantum linear system problem, which can yield near-optimal complexity with respect to the number of queries to A but is generally suboptimal in terms of the number of queries to the initial state preparation oracle. Our Lap-LCHS algorithm does not impose unnatural limitations on the range of matrix eigenvalues and can achieve a better state preparation cost compared to QEVT with HSP by avoiding the use of quantum linear system algorithms.

Further applications: In our second technical paper [2], we further discuss and analyze the complexity of our Lap-LCHS algorithm to various applications, including ODEs in the form of Eq. (2), inhomogeneous ODEs, powers of matrix inverses, and the decaying branch of a second-order differential equation. Remarkably, our Lap-LCHS can solve Eq. (2) (i.e., implement the transformation $e^{-TA^{-1}}$) without explicitly inverting the matrix A, and achieves the best known complexity both in terms of the number of queries to A and the initial state preparation oracle.

[1] D. An, A. M. Childs, and L. Lin, "Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters," (2023), arXiv:2312.03916 [quant-ph].

[2] D. An, A. M. Childs, L. Lin, and L. Ying, "Laplace eigenvalue transformation via linear combination of Hamiltonian simulation," In preparation.

[3] D. An, J.-P. Liu, and L. Lin, Phys. Rev. Lett. **131**, 150603 (2023), accepted as a talk at QIP 2024.

[4] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC '19 (ACM, 2019).

[5] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. Somma, Phys. Rev. Lett. **114**, 090502 (2015).

[6] G. H. Low and N. Wiebe, arXiv:1805.00675 (2019).

[7] A. M. Childs and N. Wiebe, Quantum Information and Computation **12** (2012), 10.26421/qic12.11-12.

[8] A. M. Childs, R. Kothari, and R. D. Somma, SIAM Journal on Computing **46**, 1920–1950 (2017).

[9] S. Wang, S. McArdle, and M. Berta, PRX Quantum **5**, 020324 (2024).

[10] S. Chakraborty, Quantum **8**, 1496 (2024).

[11] D. W. Berry, Journal of Physics A: Mathematical and Theoretical **47**, 105301 (2014).

[12] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, Communications in Mathematical Physics **356**, 1057 (2017).

[13] A. M. Childs and J.-P. Liu, Communications in Mathematical Physics **375**, 1427–1457 (2020).

[14] H. Krovi, Quantum **7**, 913 (2023).

[15] D. W. Berry and P. C. S. Costa, Quantum **8**, 1369 (2024).

[16] A. W. Harrow, A. Hassidim, and S. Lloyd, Physical Review Letters **103**, 150502 (2009).

[17] Y. Subaşı, R. D. Somma, and D. Orsucci, Physical Review Letters **122**, 060504 (2019).

[18] D. An and L. Lin, ACM Transactions on Quantum Computing **3**, 1–28 (2022).

[19] L. Lin and Y. Tong, Quantum **4**, 361 (2020).

[20] P. C. Costa, D. An, Y. R. Sanders, Y. Su, R. Babbush, and D. W. Berry, PRX Quantum **3**, 040303 (2022).

[21] A. M. Dalzell, "A shortcut to an optimal quantum linear system solver," (2024), arXiv:2406.12086 [quant-ph].

[22] S. Jin, N. Liu, and Y. Yu, Physical Review Letters **133**, 230602 (2024).

[23] J. Hu, S. Jin, N. Liu, and L. Zhang, "Dilation theorem via schrödingerisation, with applications to the quantum simulation of differential equations," (2023), arXiv:2309.16262 [quant-ph].

[24] S. Takahira, A. Ohashi, T. Sogabe, and T. S. Usuda, Quantum Information and Computation **20**, 14–36 (2020).

[25] Y. Tong, D. An, N. Wiebe, and L. Lin, Physical Review A **104**, 032422 (2021).

[26] G. H. Low and Y. Su, arXiv:2401.06240 (2024), accepted as a talk at QIP 2024.