

A quantum algorithm for linear autonomous differential equations using Padé approximation

Dekuan Dong¹, Yingzhou Li^{1,2}, and Jungong Xue¹

¹School of Mathematical Sciences, Fudan University

²Shanghai Key Laboratory for Contemporary Applied Mathematics

We propose a novel quantum algorithm for solving linear autonomous ordinary differential equations (ODEs) using the Padé approximation. For linear autonomous ODEs, the discretized solution can be represented by a product of matrix exponentials. The proposed algorithm approximates the matrix exponential by the diagonal Padé approximation, which is then encoded into a large, block-sparse linear system and solved via quantum linear system algorithms (QLSA). The detailed quantum circuit is given based on quantum oracle access to the matrix, the inhomogeneous term, and the initial state. The complexity of the proposed algorithm is analyzed. Compared to the method based on Taylor approximation, which approximates the matrix exponential using a k -th order Taylor series, the proposed algorithm improves the approximation order k from two perspectives: 1) the explicit complexity dependency on k is improved, and 2) a smaller k suffices for the same precision. Numerical experiments demonstrate the advantages of the proposed algorithm comparing to other related algorithms.

1 Introduction

Quantum computing, which fundamentally differs from classical computing, leverages quantum mechanics to perform computations using quantum states and quantum gates. For many problems, quantum computing offers a theoretical advantage over classical computing. In certain cases, quantum algorithms executed on quantum computers can achieve exponential speedup compared to their classical counterparts [15, 34, 37]. Solving large-scale ordinary differential equations (ODEs) is one such problem where quantum algorithms show significant potential.

Solving large-scale ODEs is a fundamental problem in science, with numerous real-world applications across various domains, including weather and climate prediction, chemical reaction modeling, fluid dynamics, and finance, such as option pricing models, etc. ODEs can generally be divided into two categories: linear ODEs and nonlinear ODEs. In each category, ODEs can further be divided into autonomous and non-autonomous ODEs. A

Dekuan Dong: dkdong21@m.fudan.edu.cn

Yingzhou Li: yingzhouli@fudan.edu.cn, The research was supported in part by NSFC under grant 12271109, STCSM under grant 22TQ017 and 24DP2600100, and SIMIS under grant SIMIS-ID-2024-(CN).

Jungong Xue: xuej@fudan.edu.cn, The research was supported in part by the National Science Foundation of China Grant 12171101 and the Laboratory of Mathematics for Nonlinear Science, Fudan University.

general first-order linear ODEs admit,

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{b}(t), & t \in [0, T], \\ \mathbf{x}(0) = \mathbf{x}_0, \end{cases} \quad (1)$$

where $\mathbf{A}(t) \in \mathbb{C}^{n \times n}$ is a matrix or discretized operator, $\mathbf{b}(t) \in \mathbb{C}^n$ is the inhomogeneous term, $\mathbf{x}_0 \in \mathbb{C}^n$ is the initial state, n is the dimension of the system, and T is a final time. When both $\mathbf{A}(t)$ and $\mathbf{b}(t)$ are time-independent, i.e., $\mathbf{A}(t) = \mathbf{A}$ and $\mathbf{b}(t) = \mathbf{b}$, the ODE system (1) is referred to as a first-order autonomous ODE system. Given suitable quantum oracles to access $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{x}_0$, the goal of this paper is to design an efficient quantum algorithm that produces a quantum state that is ϵ -close to the final state $|\mathbf{x}(T)\rangle = \frac{\mathbf{x}(T)}{\|\mathbf{x}(T)\|_2}$.

Many quantum algorithms for solving linear ODEs (1) have been proposed in the past decades. These algorithms can be grouped into three categories: a) discretized time steps with Quantum Linear System Algorithm (QLSA); b) time-ordered integral form with Linear Combination of Hamiltonian Simulation (LCHS); c) Schrödingerization with Hamiltonian simulation. The first category differs from the latter two categories in its use of Hamiltonian simulation. Methods in the first category rely on Hamiltonian simulation underlying the QLSA, whereas methods in the latter two categories apply Hamiltonian simulation directly. Mathematically, methods in the latter two categories yield similar final expressions, although their derivation procedures differ significantly.

Methods in the first category consist of three steps: discretizing the time variable, encoding the discretized linear differential equation into an enlarged linear system, and solving the resulting linear system using QLSA. This approach was first introduced by [4], which uses linear multi-step methods to discretize the time variable. However, due to the limitations of the discretization method, the query complexity in [4] is $\text{poly}(1/\epsilon)$ even if the most efficient QLSA is applied. To improve the dependence on the precision ϵ , Berry et al. [7] applies a truncated Taylor series to approximate the propagator $\exp(\mathbf{A}t)$ for autonomous ODEs and encodes the evaluation process of the truncated Taylor series into a linear system. The dependence on ϵ is improved to $\text{poly}(\log(1/\epsilon))$. The analysis in [7] requires $\mathbf{A}$ to be diagonalizable, i.e., there exists an invertible matrix $\mathbf{V}$ such that $\mathbf{V}^{-1}\mathbf{A}\mathbf{V}$ is a diagonal matrix, and the final complexity depends linearly on the condition number of $\mathbf{V}$, $\kappa_{\mathbf{V}} = \|\mathbf{V}\|_2 \cdot \|\mathbf{V}^{-1}\|_2$. To eliminate the dependence on $\kappa_{\mathbf{V}}$, Krovi [32] proposes a modified linear system that also encodes the evaluation process of the truncated Taylor series into a linear system, and improves the analysis in [7]. In [32], the resulting complexity depends on $C(\mathbf{A}) := \max_{0 \leq t \leq T} \|\exp(\mathbf{A}t)\|_2$ instead of $\kappa_{\mathbf{V}}$. More recently, Dong et al. [13] revisited the analysis in [7] and obtained a similar complexity result to [32] without modifying the linear system. Wu and Li [39] proposed structure-preserving quantum algorithms for both linear and nonlinear Hamiltonian systems. These algorithms maintain the symplectic properties of the underlying dynamics, thereby enhancing the robustness and stability of the numerical methods. Additionally, Childs and Liu [11] and Berry and C. S. Costa [5] generalize methods in this category to address non-autonomous linear ODEs, i.e., time-dependent $\mathbf{A}(t)$ and $\mathbf{b}(t)$, using spectral method and Dyson's series, respectively.

The second category of methods originates from [3]. These methods are based on a novel integral expression for the non-unitary evolution operator $\mathcal{T}e^{\int_0^t \mathbf{A}(s)ds}$, where $\mathcal{T}$ denotes the time-ordering operator. By decomposing $\mathbf{A}(t)$ into its Hermitian and anti-Hermitian parts, i.e., $\mathbf{A}(t) = \mathbf{L}(t) + i\mathbf{H}(t)$, where $\mathbf{L}(t) = \frac{\mathbf{A}(t) + \mathbf{A}^\dagger(t)}{2}$ and $\mathbf{H}(t) = \frac{\mathbf{A}(t) - \mathbf{A}^\dagger(t)}{2i}$, the non-unitary evolution operator can be expressed as an integral,

$$\mathcal{T}e^{\int_0^t \mathbf{A}(s)ds} = \int_{\mathbb{R}} \frac{1}{\pi(1+k^2)} \mathcal{T}e^{i \int_0^t (\mathbf{H}(s) + k\mathbf{L}(s))ds} dk. \quad (2)$$

This identity holds under the assumption that $\mathbf{L}(t)$ is negative semi-definite, i.e., $\mathbf{L}(t) \preceq 0$ for all $t \in [0, T]$. Later, in [2], a large family of identities similar to eq. (2) is unveiled, and the method in this category is improved to achieve near-optimal dependence on all parameters. The extension to the inhomogeneous case is also possible via Duhamel's principle. Compared to the methods in the first category, methods in this category consider the non-autonomous cases directly and achieve the optimal state preparation cost. However, methods in this category only give the terminal state, whereas methods in the first category provide information throughout the evolution.

The third category of methods is known as Schrödingerization [23, 25]. These methods use a warped phase transformation to map the linear differential equation into a higher-dimensional system, where it manifests as a Schrödinger equation in the Fourier space of the extra variable. In this way, the linear ODEs with non-unitary dynamics are converted into a system that evolves under unitary dynamics, i.e., a Schrödinger equation, which can be simulated on quantum computing via Hamiltonian simulation. Schrödingerization has been applied to a wide range of problems, including differential equations with various boundary conditions [10, 17, 19, 24, 26–28, 30], iterative methods in numerical linear algebra [20], and more. In [29], Schrödingerization is extended to solve linear differential equations with inhomogeneous terms, which is the same problem addressed in this paper. The resulting query complexity depends linearly on $1/\epsilon$. More recently, methods in this category have been adapted for analog quantum simulation [21, 22]. Explicit quantum circuit implementations corresponding to Schrödingerization have been explored in [18, 31].

We propose a novel quantum algorithm for solving linear autonomous ODEs (1) in this paper, which falls within the first category of existing quantum algorithms. Our key contributions and innovations can be summarized as follows.

- We replace the Taylor series approximation with the Padé approximation of the matrix exponential in related quantum algorithms [7, 13, 32]. The Padé approximation, a rational function commonly used in classical computing to approximate matrix exponentials [1, 16, 33], achieves similar accuracy to the Taylor approximation but with a lower order, though it requires a matrix inversion. In [39], the Padé approximation is also applied to preserve the symplectic structure. Their results suggest that our method also preserves the symplectic property when applied to the same Hamiltonian systems. However, our encoding of the Padé approximation differs from theirs.
- We propose a matrix encoding for the Padé approximation, denoted as $\mathbf{L}$. The encoding avoids explicit matrix inversion and maintains a comparable level of complexity to the matrix encoding used for Taylor approximation, as presented in [7, 13, 32]. In contrast to the Taylor approximation, whose corresponding matrix encoding uses a block lower bidiagonal form, the matrix encoding for the Padé relies on a block upper Hessenberg form.
- The condition number of the Padé approximation encoded matrix $\mathbf{L}$ is estimated, which plays a crucial role in the complexity analysis of the algorithm. Additionally, we propose the block-encoding of the Padé approximation encoded matrix, along with its detailed quantum circuit implementations.
- We analyze the overall complexities of the proposed quantum algorithm, including both the oracle query complexity and the quantum gate complexity. When $\mathbf{A}$ is Hermitian and negative semi-definite, our analysis relaxes the common restriction $\|\mathbf{A}h\|_2 \leq 1$ found in other related works [5, 7, 13, 32], where h represents the time

step. Compared to other methods in the first category, the proposed algorithm outperforms them both theoretically and numerically.

Overall, our proposed quantum algorithm and the corresponding analysis address linear autonomous ODEs. Extensions to the non-autonomous and/or mildly nonlinear ODEs are feasible using similar techniques as those in [5, 10, 11, 13]. The complexity analysis for the extended algorithm will need to be rederived accordingly, and the dependence on parameters requires further investigation.

The rest of the paper is organized as follows. In Section 2, we briefly review the method proposed in [7]. Section 3 introduces the linear system encoding via Padé approximation, analyzes the condition number of the linear system, and details the block-encoding of the linear system. Section 4 presents the complexity analysis of the proposed algorithm. In Section 5, we compare our method to the one based on Taylor approximation [7, 13, 32] and the LCHS-based method [2, 3] theoretically. Moreover, numerical experiments are conducted to illustrate the advantages of our method over the method based on Taylor approximation. Finally, Section 6 concludes the paper with a discussion on future directions.

2 Preliminaries

Consider a linear differential equation of the form:

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{b}, & t \in [0, T], \\ \mathbf{x}(0) = \mathbf{x}_0, \end{cases} \quad (3)$$

where $\mathbf{A} \in \mathbb{C}^{n \times n}$ and $\mathbf{b} \in \mathbb{C}^n$ are time-independent. When $\mathbf{A}$ is non-singular, the solution is given by

$$\mathbf{x}(t) = \exp(\mathbf{A}t)\mathbf{x}(0) + (\exp(\mathbf{A}t) - \mathbf{I}_n) \mathbf{A}^{-1}\mathbf{b}, \quad t \in [0, T].$$

In order to prepare the quantum state $|\mathbf{x}(T)\rangle = \frac{\mathbf{x}(T)}{\|\mathbf{x}(T)\|_2}$, reference [7] approximates the exponential function $\exp(\cdot)$ by its truncated Taylor series

$$T_k(z) := \sum_{j=0}^k \frac{z^j}{j!}.$$

Given a suitable number of time steps $m \in \mathbb{N}_+$ and the time step size $h = T/m$, the following procedure is used to approximate $\mathbf{x}(T)$:

$$\begin{aligned} \mathbf{x}(h) &\approx T_k(\mathbf{A}h)\mathbf{x}(0) + (T_k(\mathbf{A}h) - \mathbf{I}_n) \mathbf{A}^{-1}\mathbf{b} =: \hat{\mathbf{x}}(h), \\ \mathbf{x}(2h) &\approx T_k(\mathbf{A}h)\hat{\mathbf{x}}(h) + (T_k(\mathbf{A}h) - \mathbf{I}_n) \mathbf{A}^{-1}\mathbf{b} =: \hat{\mathbf{x}}(2h), \\ &\vdots \\ \mathbf{x}(mh) &\approx T_k(\mathbf{A}h)\hat{\mathbf{x}}((m-1)h) + (T_k(\mathbf{A}h) - \mathbf{I}_n) \mathbf{A}^{-1}\mathbf{b} =: \hat{\mathbf{x}}(T). \end{aligned} \quad (4)$$

Then, reference [7] shows that each step of the above procedure can be encoded into a linear system. The s -th step of the above procedure is encoded in the following linear system:

$$\begin{cases} \mathbf{M}_k(\mathbf{A}h)\mathbf{z}^{(s)} = \mathbf{y}_s, \\ \hat{\mathbf{x}}(sh) - (\mathbf{1}^T \otimes \mathbf{I}_n)\mathbf{z}^{(s)} = 0, \end{cases} \quad \forall s = 1, \dots, m, \quad (5)$$

where

$$M_k(Ah) := \begin{bmatrix} I_n & & & \\ -Ah & I_n & & \\ & \ddots & \ddots & \\ & & -\frac{Ah}{k} & I_n \end{bmatrix}, \quad y_s := \begin{bmatrix} \hat{x}((s-1)h) \\ hb \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \in \mathbb{C}^{n(k+1)}, \quad \mathbf{1} := \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{k+1}.$$

Combining the m steps and repeating $\hat{x}(T)$ for p times to boost the success probability, we get the linear system

$$\underbrace{\begin{bmatrix} M_k(Ah) & & & & \\ T & M_k(Ah) & & & \\ & \ddots & \ddots & & \\ & & T & M_k(Ah) & \\ & & & -\mathbf{1}^T \otimes I_n & I_n \\ & & & & -I_n & I_n \\ & & & & & \ddots & \ddots \\ & & & & & & -I_n & I_n \end{bmatrix}}_{=: C_{m,k,p}(Ah)} \begin{bmatrix} z^{(1)} \\ z^{(2)} \\ \vdots \\ z^{(m)} \\ \hat{x} \\ \hat{x} \\ \vdots \\ \hat{x} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_0 \\ \vdots \\ y_0 \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (6)$$

where

$$T := - \begin{bmatrix} I_n & \cdots & I_n \\ O_n & \cdots & O_n \\ \vdots & & \vdots \\ O_n & \cdots & O_n \end{bmatrix} \in \mathbb{R}^{n(k+1) \times n(k+1)}, \quad y_1 := \begin{bmatrix} x(0) \\ hb \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \in \mathbb{C}^{n(k+1)}, \quad y_0 := \begin{bmatrix} \mathbf{0} \\ hb \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \in \mathbb{C}^{n(k+1)}.$$

In the rest of this paper, we refer to the coefficient matrix in eq. (6) as $C_{m,k,p}(Ah)$. With appropriately chosen parameters m , k , and p , we can solve the linear system using QLSA [12], and then perform measurements on the resulting quantum state. This allows us to obtain a quantum state that approximates $|x(T)\rangle$ within the required precision, with constant positive probability. The most time-consuming step in this procedure is solving the linear system using QLSA. Assuming the block-encoding of $C_{m,k,p}(Ah)$ is available, the query complexity of QLSA depends at least linearly on the condition number of $C_{m,k,p}(Ah)$. Therefore, it is crucial to control the condition number of the resulting linear system.

3 Quantum algorithm based on Padé approximation

In this section, we present our quantum algorithm for solving linear differential equations using Padé approximation. We begin by outlining the Padé approximation to matrix exponential.

Definition 3.1. The (p, q) Padé approximation to e^A is defined by

$$R_{pq}(A) = [D_{pq}(A)]^{-1} N_{pq}(A) = N_{pq}(A) [D_{pq}(A)]^{-1}, \quad (7)$$

where

$$N_{pq}(\mathbf{A}) = \sum_{j=0}^p n_j \mathbf{A}^j, \quad n_j = \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!} \quad (8)$$

and

$$D_{pq}(\mathbf{A}) = \sum_{j=0}^q d_j (-\mathbf{A})^j, \quad d_j = \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!}. \quad (9)$$

The non-singularity of $D_{pq}(\mathbf{A})$ is assured if p and q are large enough or if the eigenvalues of $\mathbf{A}$ are all negative [33, 36].

Since Padé approximation is a rational function, the matrix encoding method from [7] cannot be applied directly. We propose a novel approach to encode rational polynomials into linear systems with a special structure. In the case where $\mathbf{A}$ is Hermitian and negative semi-definite, this structure allows us to bound the condition number of the resulting linear system without relying on the usual restriction $\|\mathbf{A}h\|_2 \leq 1$, which is typically required in related methods [4, 7, 13, 32].

3.1 Constructing the linear system

Replacing the Taylor approximation by Padé approximation, the procedure (4) becomes

$$\begin{aligned} \mathbf{x}(h) &\approx R_{pq}(\mathbf{A}h)\mathbf{x}(0) + (R_{pq}(\mathbf{A}h) - \mathbf{I}_n) \mathbf{A}^{-1}\mathbf{b} =: \hat{\mathbf{x}}(h), \\ \mathbf{x}(2h) &\approx R_{pq}(\mathbf{A}h)\hat{\mathbf{x}}(h) + (R_{pq}(\mathbf{A}h) - \mathbf{I}_n) \mathbf{A}^{-1}\mathbf{b} =: \hat{\mathbf{x}}(2h), \\ &\vdots \\ \mathbf{x}(mh) &\approx R_{pq}(\mathbf{A}h)\hat{\mathbf{x}}((m-1)h) + (R_{pq}(\mathbf{A}h) - \mathbf{I}_n) \mathbf{A}^{-1}\mathbf{b} =: \hat{\mathbf{x}}(T). \end{aligned} \quad (10)$$

We first consider the encoding in one step. Given $\hat{\mathbf{x}}((s-1)h)$, the encoding of the s -th step is given by,

$$\hat{\mathbf{x}}(sh) = R_{pq}(\mathbf{A}h)\hat{\mathbf{x}}((s-1)h) + (R_{pq}(\mathbf{A}h) - \mathbf{I}_n) \mathbf{A}^{-1}\mathbf{b}, \quad \forall s = 1, \dots, m. \quad (11)$$

Substituting $R_{pq}(\mathbf{A}h) = N_{pq}(\mathbf{A}h)D_{pq}^{-1}(\mathbf{A}h)$ into eq. (11), we obtain

$$\hat{\mathbf{x}}(sh) = N_{pq}(\mathbf{A}h) \left(D_{pq}^{-1}(\mathbf{A}h)\hat{\mathbf{x}}((s-1)h) + (D_{pq}^{-1}(\mathbf{A}h) - \mathbf{I}_n)\mathbf{A}^{-1}\mathbf{b} \right) + (N_{pq}(\mathbf{A}h) - \mathbf{I}_n)\mathbf{A}^{-1}\mathbf{b},$$

Introducing an auxiliary vector

$$\mathbf{v} := D_{pq}^{-1}(\mathbf{A}h)\hat{\mathbf{x}}((s-1)h) + (D_{pq}^{-1}(\mathbf{A}h) - \mathbf{I}_n)\mathbf{A}^{-1}\mathbf{b},$$

we can rewrite eq. (11) as a pair of equations:

$$\begin{cases} \hat{\mathbf{x}}((s-1)h) = D_{pq}(\mathbf{A}h)\mathbf{v} + (D_{pq}(\mathbf{A}h) - \mathbf{I}_n)\mathbf{A}^{-1}\mathbf{b}, \\ \hat{\mathbf{x}}(sh) = N_{pq}(\mathbf{A}h)\mathbf{v} + (N_{pq}(\mathbf{A}h) - \mathbf{I}_n)\mathbf{A}^{-1}\mathbf{b}. \end{cases} \quad (12)$$

Following the approach from [7], we can encode the pair of equations into two linear systems,

$$\left[\begin{array}{ccc|ccc} \mathbf{I}_n & & & & & \\ \hline \beta_1 \mathbf{A}h & \mathbf{I}_n & & & & \\ & & \ddots & & & \\ & & & \beta_q \mathbf{A}h & \mathbf{I}_n & \\ -\mathbf{I}_n & \cdots & -\mathbf{I}_n & -\mathbf{I}_n & \mathbf{I}_n & \end{array} \right] \begin{bmatrix} \mathbf{z}_0^{(s)} \\ \mathbf{z}_1^{(s)} \\ \vdots \\ \mathbf{z}_p^{(s)} \\ \hat{\mathbf{x}}((s-1)h) \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ -d_1 h \mathbf{b} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (13)$$

and

$$\begin{bmatrix} \mathbf{I}_n & & & \\ -\alpha_1 \mathbf{A}h & \mathbf{I}_n & & \\ & \ddots & \ddots & \\ & & -\alpha_p \mathbf{A}h & \mathbf{I}_n \\ -\mathbf{I}_n & \cdots & -\mathbf{I}_n & -\mathbf{I}_n & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{z}}_0^{(s)} \\ \tilde{\mathbf{z}}_1^{(s)} \\ \vdots \\ \tilde{\mathbf{z}}_p^{(s)} \\ \hat{\mathbf{x}}(sh) \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ n_1 h \mathbf{b} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \quad (14)$$

respectively, where we use the following notations for convenience,

$$\begin{cases} \alpha_0 := 1, \\ \alpha_{j+1} := \frac{n_{j+1}}{n_j} = \frac{p-j}{(j+1)(p+q-j)}, \quad j = 0, \dots, p-1, \\ \beta_0 := 1, \\ \beta_{j+1} := \frac{d_{j+1}}{d_j} = \frac{q-j}{(j+1)(p+q-j)}, \quad j = 0, \dots, q-1. \end{cases} \quad (15)$$

To combine the linear systems from eq. (13) and eq. (14), we begin by rewriting eq. (13) as follows

$$\begin{cases} \mathbf{z}_0^{(s)} = \mathbf{v}, \\ \begin{bmatrix} \beta_1 \mathbf{A}h & \mathbf{I}_n & & \\ & \ddots & \ddots & \\ & & \beta_q \mathbf{A}h & \mathbf{I}_n \\ -\mathbf{I}_n & \cdots & -\mathbf{I}_n & -\mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{z}_0^{(s)} \\ \mathbf{z}_1^{(s)} \\ \vdots \\ \mathbf{z}_q^{(s)} \end{bmatrix} = \begin{bmatrix} -d_1 h \mathbf{b} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\hat{\mathbf{x}}((s-1)h) \end{bmatrix}, \end{cases} \quad (16)$$

which symbolically treats $\mathbf{v}$ as an unknown and $\hat{\mathbf{x}}((s-1)h)$ as a known component. Next, combining eq. (14) with the second system from eq. (16), we obtain the following unified system

$$\begin{bmatrix} \mathbf{I}_n & \mathbf{I}_n & \cdots & \mathbf{I}_n \\ \mathbf{I}_n & \beta_q \mathbf{A}h & & \\ & \ddots & \ddots & \\ & & \mathbf{I}_n & \beta_1 \mathbf{A}h \\ & & & -\alpha_1 \mathbf{A}h & \mathbf{I}_n \\ & & & \ddots & \ddots \\ & & & & -\alpha_p \mathbf{A}h & \mathbf{I}_n \\ -\mathbf{I}_n & \cdots & -\mathbf{I}_n & -\mathbf{I}_n & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{z}_q^{(s)} \\ \mathbf{z}_{q-1}^{(s)} \\ \vdots \\ \mathbf{z}_0^{(s)} = \mathbf{v} \\ \tilde{\mathbf{z}}_1^{(s)} \\ \vdots \\ \tilde{\mathbf{z}}_p^{(s)} \\ \hat{\mathbf{x}}(sh) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}((s-1)h) \\ \mathbf{0} \\ \vdots \\ -d_1 h \mathbf{b} \\ n_1 h \mathbf{b} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}. \quad (17)$$

Since the diagonal Padé approximation ($p = q$) is usually preferred over the off-diagonal cases ($p \neq q$) [33], we consider the case $p = q = k$ in the rest of the paper. We present the following finding, whose proof is provided in Appendix A.1.

Lemma 3.2. *In the case when $p = q = k$, we have $\mathbf{z}_j^{(s)} = (-1)^j \tilde{\mathbf{z}}_j^{(s)}$ for all $j = 1, \dots, k$ and $s = 1, \dots, m$.*

Using Lemma 3.2, we can omit the computation of the terms $\tilde{\mathbf{z}}_j^{(s)}$, simplifying the linear

system in eq. (17) to

$$\begin{bmatrix} \mathbf{I}_n & \mathbf{I}_n & \cdots & \mathbf{I}_n \\ \mathbf{I}_n & \beta_k \mathbf{A}h & & \\ & \ddots & \ddots & \\ & & \mathbf{I}_n & \beta_1 \mathbf{A}h \\ (-\mathbf{I}_n)^{k+1} & (-\mathbf{I}_n)^k & \cdots & -\mathbf{I}_n & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{z}_k^{(s)} \\ \mathbf{z}_{k-1}^{(s)} \\ \vdots \\ \mathbf{z}_0^{(s)} \\ \widehat{\mathbf{x}}(sh) \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{x}}((s-1)h) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -d_1 h \mathbf{b} \\ \mathbf{0} \end{bmatrix}. \quad (18)$$

To ensure that the norm of the coefficient matrix in eq. (18) is bounded independent of k , we multiply both the first and the last block rows of the matrix by the factor $\frac{1}{\sqrt{k+1}}$. This modification also benefits the block-encoding and the condition number of the final linear system. With this modification, the s -th step eq. (11) of the procedure is encoded into the following linear system

$$\begin{cases} \mathbf{W}_k(\mathbf{A}h) \mathbf{z}^{(s)} = \widetilde{\mathbf{y}}_s, \\ \frac{\widehat{\mathbf{x}}(sh)}{\sqrt{k+1}} + \frac{\widetilde{\mathbf{1}}^T \otimes \mathbf{I}_n}{\sqrt{k+1}} \mathbf{z}^{(s)} = \mathbf{0}, \end{cases} \quad (19)$$

where

$$\mathbf{W}_k(\mathbf{A}h) := \begin{bmatrix} \frac{1}{\sqrt{k+1}} \mathbf{I}_n & \frac{1}{\sqrt{k+1}} \mathbf{I}_n & \cdots & \frac{1}{\sqrt{k+1}} \mathbf{I}_n \\ \mathbf{I}_n & \beta_k \mathbf{A}h & & \\ & \ddots & \ddots & \\ & & \mathbf{I}_n & \beta_1 \mathbf{A}h \end{bmatrix} \quad (20)$$

and

$$\mathbf{z}^{(s)} = \begin{bmatrix} \mathbf{z}_k^{(s)} \\ \mathbf{z}_{k-1}^{(s)} \\ \vdots \\ \mathbf{z}_0^{(s)} \end{bmatrix} \in \mathbb{C}^{n(k+1)}, \quad \widetilde{\mathbf{y}}_s := \begin{bmatrix} \frac{\widehat{\mathbf{x}}((s-1)h)}{\sqrt{k+1}} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -d_1 h \mathbf{b} \end{bmatrix} \in \mathbb{C}^{n(k+1)}, \quad \widetilde{\mathbf{1}} := \begin{bmatrix} (-1)^{k+1} \\ (-1)^k \\ \vdots \\ -1 \end{bmatrix} \in \mathbb{R}^{k+1}. \quad (21)$$

Finally, we combine m steps in the procedure eq. (10) together and obtain the entire linear system

$$\underbrace{\begin{bmatrix} \mathbf{W}_k(\mathbf{A}h) & & & \\ & \widetilde{\mathbf{T}} & & \mathbf{W}_k(\mathbf{A}h) \\ & & \ddots & \vdots \\ & & & \widetilde{\mathbf{T}} & \mathbf{W}_k(\mathbf{A}h) \\ & & & & \underbrace{\begin{bmatrix} \frac{\widetilde{\mathbf{1}}^T \otimes \mathbf{I}_n}{\sqrt{k+1}} & \frac{\mathbf{I}_n}{\sqrt{k+1}} & & \\ & -\mathbf{I}_n & \mathbf{I}_n & \\ & & \ddots & \ddots \\ & & & -\mathbf{I}_n & \mathbf{I}_n \end{bmatrix}}_{p \text{ blocks}} \end{bmatrix}}_{=: \mathbf{L}_{m,k,p}(\mathbf{A}h)} \begin{bmatrix} \mathbf{z}^{(1)} \\ \mathbf{z}^{(2)} \\ \vdots \\ \mathbf{z}^{(m)} \\ \widehat{\mathbf{x}} \\ \widehat{\mathbf{x}} \\ \vdots \\ \widehat{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{y}}_1 \\ \widetilde{\mathbf{y}}_0 \\ \vdots \\ \widetilde{\mathbf{y}}_0 \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (22)$$

where the vector $\hat{\mathbf{x}}$ is repeated p times to boost the success probability, and

$$\tilde{\mathbf{T}} := \frac{(\mathbf{e}_1 \tilde{\mathbf{1}}^T) \otimes \mathbf{I}_n}{\sqrt{k+1}}, \quad \mathbf{e}_1 := \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{k+1}, \quad \tilde{\mathbf{y}}_1 := \begin{bmatrix} \frac{x(0)}{\sqrt{k+1}} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -d_1 h \mathbf{b} \end{bmatrix} \in \mathbb{C}^{n(k+1)}, \quad \tilde{\mathbf{y}}_0 := \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -d_1 h \mathbf{b} \end{bmatrix} \in \mathbb{C}^{n(k+1)}. \quad (23)$$

In the following discussion, we use $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ to refer to the coefficient matrix in the linear system given by eq. (22). It is important to note that $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ and $\mathbf{C}_{m,k,p}(\mathbf{A}h)$, as defined in eq. (6), share the same block structure and block size.

3.2 Upper bounds of condition number

In this section, we provide upper bounds for the condition number of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$, under the assumption that either $\mathbf{A}$ is Hermitian and negative semi-definite, or $\|\mathbf{A}h\|_2 \leq 1$. The key step is to derive an upper bound for the inverse of $\mathbf{W}_k(\mathbf{A}h)$. The proofs of the relevant lemmas are provided in Appendices A.2 and A.3. To simplicity, we drop h in $\mathbf{W}_k(\mathbf{A}h)$ in these lemmas.

Lemma 3.3. *The matrix $\mathbf{W}_k(\mathbf{A})$ is non-singular if and only if $D_{kk}(\mathbf{A})$ is non-singular.*

Lemma 3.4. *Suppose $\mathbf{A} \in \mathbb{C}^{n \times n}$ is Hermitian and negative semi-definite, then*

$$\|\mathbf{W}_k(\mathbf{A})^{-1}\|_2 \leq \sqrt{(k+1)(4 \log(k+1) + 1)}. \quad (24)$$

Lemma 3.5. *Suppose $\mathbf{A} \in \mathbb{C}^{n \times n}$ satisfies $\|\mathbf{A}\|_2 \leq 1$, then*

$$\|\mathbf{W}_k(\mathbf{A})^{-1}\|_2 \leq \frac{2\sqrt{e}}{3-e} \sqrt{(k+1)(4 \log(k+1) + 1)}.$$

Using the above lemmas, we derive upper bounds for the inverse and the condition number of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ in the following theorem, whose proof is given in Appendix A.4.

Theorem 3.6. *Let $\mathbf{A} \in \mathbb{C}^{n \times n}$, $T > 0$, and the parameters m and $k \geq 3$ be chosen such that*

$$\|\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h)\|_2 \leq 1, \quad \forall i = 1, \dots, m, \quad (25)$$

where $h = T/m$. Denoting by κ the condition number of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$. Then

1. if $\mathbf{A}$ is Hermitian and negative semi-definite, then

$$\|\mathbf{L}_{m,k,p}(\mathbf{A}h)^{-1}\|_2 \leq 6(m+p)\sqrt{k \log k} \quad (26)$$

and

$$\kappa \leq 3(m+p)\sqrt{k \log k} (6 + \|\mathbf{A}h\|_2). \quad (27)$$

2. if $\|\mathbf{A}h\|_2 \leq 1$, then

$$\|\mathbf{L}_{m,k,p}(\mathbf{A}h)^{-1}\|_2 = \mathcal{O}\left(C(\mathbf{A})(m+p)\sqrt{k \log k}\right),$$

and

$$\kappa = \mathcal{O}\left(C(\mathbf{A})(m+p)\sqrt{k \log k}\right),$$

where $C(\mathbf{A}) := \max_{t \in [0, T]} \|\exp(\mathbf{A}t)\|_2$.

In summary, both of the two cases lead to

$$\kappa = \mathcal{O}\left(C(\mathbf{A})(m+p)\sqrt{k \log k} \|\mathbf{A}h\|_2\right).$$

Remark 3.7. One advantage of the Padé approximation over the Taylor approximation is that when $\mathbf{A}$ is Hermitian and negative semi-definite, the spectral norm of the inverse of $\mathbf{W}_k(\mathbf{A}h)$ and $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ are independent of $\|\mathbf{A}h\|_2$. In contrast, for their counterparts, $\mathbf{M}_k(\mathbf{A}h)$ and $\mathbf{C}_{m,k,p}(\mathbf{A}h)$, the spectral norms may grow exponentially with $\|\mathbf{A}h\|_2$. To illustrate this, observe that

$$\left\|\mathbf{C}_{m,k,p}^{-1}(\mathbf{A}h)\right\|_2 \geq \left\|\mathbf{M}_k^{-1}(\mathbf{A}h)\right\|_2,$$

and

$$\mathbf{M}_k^{-1}(\mathbf{A}h) \cdot (e_1 \otimes \mathbf{I}_n) = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{A}h \\ \frac{(\mathbf{A}h)^2}{2!} \\ \vdots \\ \frac{(\mathbf{A}h)^k}{k!} \end{bmatrix}.$$

When $\mathbf{A}$ is Hermitian, we then have

$$\left\|\mathbf{M}_k^{-1}(\mathbf{A}h)\right\|_2 \geq \sqrt{\sum_{j=0}^k \frac{\|\mathbf{A}h\|_2^{2j}}{(j!)^2}} \geq \frac{1}{\sqrt{k+1}} \sum_{j=0}^k \frac{\|\mathbf{A}h\|_2^j}{j!} \approx \frac{\exp(\|\mathbf{A}h\|_2)}{\sqrt{k+1}}. \quad (28)$$

3.3 Block-encoding of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$

In this section, we implement the block-encoding of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$. The definition of block-encoding is provided in Appendix B. We assume:

- The parameters n , m , $k+1$, and p are powers of two, specifically $n = 2^n$, $m = 2^m$, $k+1 = 2^k$, and $p = m \cdot (k+1) = 2^{m+k}$. The corresponding registers are denoted as $|\cdot\rangle_n$, $|\cdot\rangle_m$, and $|\cdot\rangle_k$, where the subscript indicates the number of qubits in this register. Additionally, a superscript a denotes that the register contains ancilla qubits.
- An $(\alpha, \mathfrak{d})$ -block-encoding of $\mathbf{A}$ is available, which is denoted by $U_{\mathbf{A}}$.

Before constructing the block-encoding, we summarize the total cost in the following theorem.

Theorem 3.8. Given an $(\alpha, \mathfrak{d})$ -block-encoding of $\mathbf{A}$, there exists a $(4 \cdot \max\{\alpha h, 1\}, \mathfrak{d} + 5)$ -block-encoding of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$. The gate complexity is

$$\mathcal{O}(k + \text{poly} \log(m \cdot k)), \quad (29)$$

and it requires one query to the block-encoding of $\mathbf{A}$.

Compared to the block-encoding presented in [32], ours is simpler, with both the normalization factor and the number of additional qubits being independent of k and m . With minor modifications, the block-encoding proposed in this paper can also be applied to $\mathbf{C}_{m,k,p}(\mathbf{A}h)$, which shares a similar block structure to $\mathbf{L}_{m,k,p}(\mathbf{A}h)$.

We begin by constructing the block-encoding for the case $h = 1$ and $\alpha = 1$, and will later generalize it for arbitrary $h > 0$ and $\alpha \geq \|\mathbf{A}\|_2$. To start, we rewrite the matrix as a sum of two matrices:

$$\begin{aligned} \mathbf{L}_{m,k,p}(\mathbf{A}) &= \begin{bmatrix} \mathbf{W} & & & & & \\ & \ddots & & & & \\ & & \mathbf{W} & & & \\ & & & \frac{\mathbf{I}_n}{\sqrt{k+1}} & & \\ & & & -\mathbf{I}_n & \mathbf{I}_n & \\ & & & & \ddots & \\ & & & & & -\mathbf{I}_n & \mathbf{I}_n \end{bmatrix} + \begin{bmatrix} \mathbf{O} & & & & & \\ \tilde{\mathbf{T}} & \mathbf{O} & & & & \\ & \ddots & \ddots & & & \\ & & \tilde{\mathbf{T}} & & & \\ & & & \mathbf{O} & & \\ & & & \frac{\tilde{\mathbf{T}}^T \otimes \mathbf{I}_n}{\sqrt{k+1}} & \mathbf{O}_n & \\ & & & \mathbf{O}_n & \mathbf{O}_n & \\ & & & & \ddots & \ddots \\ & & & & & \mathbf{O}_n & \mathbf{O}_n \end{bmatrix} \\ &=: \mathbf{L}_1 + \mathbf{L}_2, \end{aligned} \quad (30)$$

where we use $\mathbf{W}$ to represent $\mathbf{W}_k(\mathbf{A})$ for simplicity. Note that the nonzero components of $\tilde{\mathbf{T}}$ are confined to its first block row, which equals $\frac{\tilde{\mathbf{T}}^T \otimes \mathbf{I}_n}{\sqrt{k+1}}$, allowing us to express $\mathbf{L}_2$ as a tensor product. In the following two subsections, We will discuss the block-encoding of $\mathbf{L}_1$ and $\mathbf{L}_2$, and thus the block-encoding of $\mathbf{L}_{m,k,p}(\mathbf{A})$ can be implemented using LCU (Linear Combination of Unitaries).

3.3.1 Block-encoding of $\mathbf{L}_1$

The first term $\mathbf{L}_1$ can be written as

$$\mathbf{L}_1 = \begin{bmatrix} \mathcal{W} & \\ & \mathcal{B} \end{bmatrix},$$

where $\mathcal{W}, \mathcal{B} \in \mathbb{R}^{mn(k+1) \times mn(k+1)}$, and explicitly

$$\mathcal{W} = \mathbf{I}_m \otimes \mathbf{W}, \quad \mathcal{B} = \begin{bmatrix} \frac{1}{\sqrt{k+1}} & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \end{bmatrix} \otimes \mathbf{I}_n. \quad (31)$$

The matrix $\mathbf{W}$ can be decomposed as

$$\mathbf{W} = \mathbf{M}_1 \otimes \mathbf{I}_n + \mathbf{M}_2 \otimes \mathbf{I}_n + \mathbf{M}_3 \otimes \mathbf{A},$$

where

$$\mathbf{M}_1 = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}, \quad \mathbf{M}_2 = \begin{bmatrix} \frac{1}{\sqrt{k+1}} & \frac{1}{\sqrt{k+1}} & \cdots & \frac{1}{\sqrt{k+1}} \\ 0 & 0 & & \\ & \ddots & \ddots & \\ & & 0 & 0 \end{bmatrix}, \quad \mathbf{M}_3 = \begin{bmatrix} 0 & & & \\ & \beta_k & & \\ & & \ddots & \\ & & & \beta_1 \end{bmatrix}.$$

We now consider the block-encoding for $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}_3$:

- For matrix $\mathbf{M}_1$, its block-encoding $U_{\mathbf{M}_1}$ should satisfy

$$U_{\mathbf{M}_1} |0\rangle_1^a |j\rangle_{\mathfrak{k}} = \begin{cases} |0\rangle_1^a |(j+1) \bmod (k+1)\rangle_{\mathfrak{k}}, & j \neq k, \\ |1\rangle_1^a |(j+1) \bmod (k+1)\rangle_{\mathfrak{k}}, & j = k. \end{cases}$$

The corresponding quantum circuit is shown in Figure 1.

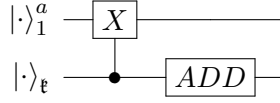


Figure 1: A quantum circuit implementing the block-encoding of $\mathbf{M}_1$, denoted by $U_{\mathbf{M}_1}$.

The addition module performs $(j + 1) \bmod (k + 1)$, which is a unitary operation. Thus, we can verify that

$$\langle 0|_1^a \langle i|_t U_{\mathbf{M}_1} |0\rangle_1^a |j\rangle_t = \begin{cases} 0 & i \neq (j + 1) \bmod (k + 1), \\ 0 & i = 0, j = k, \\ 1 & i = j + 1, j \neq k. \end{cases}$$

Therefore, $U_{\mathbf{M}_1}$ is a $(1, 1)$ -block-encoding of $\mathbf{M}_1$, and the gate complexity for implementing $U_{\mathbf{M}_1}$ is $O(\log(k))$.

- For matrix $\mathbf{M}_2$, we have

$$\mathbf{M}_2 = \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{k+1}} & \frac{1}{\sqrt{k+1}} & \cdots & \frac{1}{\sqrt{k+1}} \\ * & * & \cdots & * \\ \vdots & \vdots & \cdots & \vdots \\ * & * & \cdots & * \end{bmatrix}.$$

Since $k + 1 = 2^t$ is a power of 2, the second matrix can be chosen as $H^{\otimes t}$, where H is the Hadamard gate. The block-encoding of the first matrix is straightforward to implement in quantum circuits. Therefore, the full block-encoding $U_{\mathbf{M}_2}$ can be constructed using Lemma B.3, and the corresponding quantum circuit is shown in Figure 2. It is a $(1, 1)$ -block-encoding of $\mathbf{M}_2$, with gate complexity $O(\log(k))$.

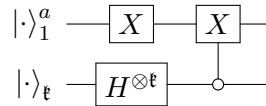


Figure 2: A quantum circuit that implements the block-encoding of $\mathbf{M}_2$, denoted by $U_{\mathbf{M}_2}$.

- For matrix $\mathbf{M}_3$, note that $|\beta_j| < 1$, and we can compute β_j explicitly. In fact,

$$\beta_j = \frac{k - j + 1}{j(2k - j + 1)}, \quad j = 1, \dots, k + 1.$$

Thus, we can construct a quantum circuit to perform

$$O_\beta |0\rangle_1^a |j\rangle_t = \left(\beta_{k+1-j} |0\rangle_1^a + \sqrt{1 - \beta_{k+1-j}^2} |1\rangle_1^a \right) |j\rangle_t, \forall j = 0, \dots, k,$$

using a uniformly controlled rotation. This operation can be further simplified using the method proposed in [9, 35]. Specifically, we need $k + 1$ single-qubit gates and $k + 1$ CNOT gates. It follows that O_β is indeed a block-encoding of $\mathbf{M}_3$, as

$$\langle 0|_1^a \langle i|_t O_\beta |0\rangle_1^a |j\rangle_t = \delta_{ij} \beta_{k+1-j}.$$

Setting $U_{M_3} = O_\beta$, we conclude that U_{M_3} is a $(1, 1)$ -block-encoding of $\mathbf{M}_3$ with gate complexity $O(k)$.

Using the unitaries U_{M_1} , U_{M_2} and U_{M_3} , we obtain a $(3, \mathfrak{d} + 3)$ -block-encoding of $\mathbf{W}$ through LCU, as illustrated in Figure 3, where $\zeta = 2 \arccos \frac{\sqrt{6}}{3}$. The gate complexity of this block-encoding is $O(k)$, except for the implementation of the controlled U_A .

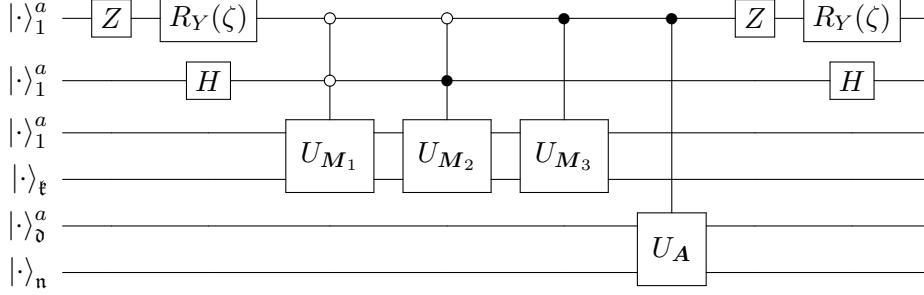


Figure 3: A quantum circuit that implements the block-encoding of $\mathbf{W}$, denoted by $U_{\mathbf{W}}$.

Next, we consider the block-encoding of $\mathcal{B}$ in eq. (31). The bi-diagonal matrix can be decomposed as

$$\begin{bmatrix} \frac{1}{\sqrt{k+1}} & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \end{bmatrix} = \begin{bmatrix} 0 & & & & \\ -1 & 0 & & & \\ & \ddots & \ddots & & \\ & & -1 & 0 & \end{bmatrix} + \begin{bmatrix} \frac{1}{\sqrt{k+1}} & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \end{bmatrix} =: \mathbf{M}_4 + \mathbf{M}_5,$$

where $\mathbf{M}_4 \in \mathbb{R}^{p \times p}$ is similar to $\mathbf{M}_1$ but with different size and opposite signs. To introduce the negative sign, we use a negative Pauli gate Z . The quantum circuit for U_{M_4} is shown in Figure 4

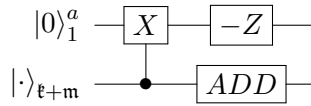


Figure 4: A quantum circuit that implements the block-encoding of $\mathbf{M}_4$, denoted by U_{M_4} .

Thus, U_{M_4} is a $(1, 1)$ -block-encoding of matrix $\mathbf{M}_4$ with gate complexity $O(\log(m \cdot k))$.

The second matrix is diagonal and can be implemented using the circuit in Figure 5.

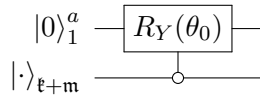


Figure 5: A quantum circuit that implements the block-encoding of $\mathbf{M}_5$, denoted by U_{M_5} .

The parameter $\theta_0 = 2 \arccos \frac{1}{\sqrt{k+1}}$, and U_{M_5} is a $(1, 1)$ -block-encoding of $\mathbf{M}_5$ with gate complexity $O(\log(m \cdot k))$. Next, using LCU, we obtain the block-encoding of $U_{M_4} + U_{M_5}$,

which provides a $(2, 2)$ -block-encoding of $\mathcal{B}$. To match the $(3, \mathfrak{d} + 3)$ -block-encoding of $\mathcal{W}$, we add a qubit, yielding a $(3, 3)$ -block-encoding of $\mathcal{B}$ using Lemma B.5, as shown in Figure 6, where $\theta_1 = 2 \arccos \frac{2}{3}$ such that $\cos \frac{\theta_1}{2} = \frac{2}{3}$. The gate complexity for the block-encoding of $\mathcal{B}$ is also $O(\log(m \cdot k))$.

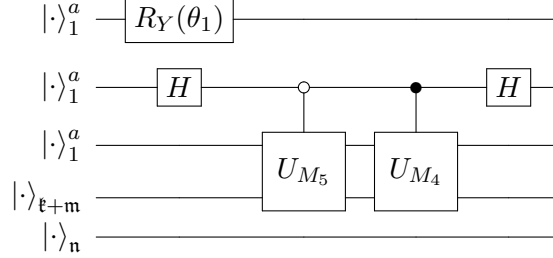


Figure 6: A quantum circuit that implements the block-encoding of $\mathcal{B}$, denoted by $U_{\mathcal{B}}$.

Combining the block-encoding of $\mathcal{W}$ and $\mathcal{B}$, we obtain a $(3, \mathfrak{d} + 3)$ -block-encoding of $\mathbf{L}_1$ with gate complexity $O(k + \log(m \cdot k))$. The corresponding quantum circuit is shown in Figure 7.

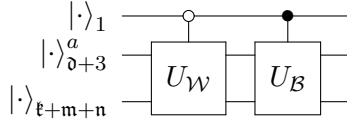


Figure 7: A quantum circuit that implements the block-encoding of $\mathbf{L}_1$, denoted by $U_{\mathbf{L}_1}$.

3.3.2 Block-encoding of $\mathbf{L}_2$

The second term $\mathbf{L}_2$ in eq. (30) can be written as

$$\mathbf{L}_2 = \left(\begin{bmatrix} \mathbf{I}_m & \mathbf{O}_m \end{bmatrix} \otimes \tilde{\mathbf{T}} \right) \cdot \left(\begin{bmatrix} 0 & & & 1 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \otimes \mathbf{I}_{n(k+1)} \right) =: \mathbf{L}_2^{(1)} \mathbf{L}_2^{(2)},$$

where

$$\tilde{\mathbf{T}} = \frac{1}{\sqrt{k+1}} \begin{bmatrix} (-1)^{k+1} & (-1)^k & \dots & -1 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \otimes \mathbf{I}_n =: \mathbf{M}_6 \otimes \mathbf{I}_n, \quad \text{and} \quad \mathbf{M}_7 := \begin{bmatrix} \mathbf{I}_m & \\ & \mathbf{O}_m \end{bmatrix}.$$

Similar to $\mathbf{M}_2$, we have

$$\mathbf{M}_6 = \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} \begin{bmatrix} \frac{(-1)^{k+1}}{\sqrt{k+1}} & \frac{(-1)^k}{\sqrt{k+1}} & \dots & \frac{-1}{\sqrt{k+1}} \\ * & * & \dots & * \\ \vdots & \vdots & \dots & \vdots \\ * & * & \dots & * \end{bmatrix}$$

Since $k + 1 = 2^{\mathfrak{k}}$, we have $(-1)^{k+1} = 1$, and the following observation.

Remark 3.9. The second row of $H^{\otimes \mathfrak{k}}$ is of the form

$$\left[\frac{1}{\sqrt{k+1}} \quad \frac{-1}{\sqrt{k+1}} \quad \cdots \quad \frac{1}{\sqrt{k+1}} \quad \frac{-1}{\sqrt{k+1}} \right].$$

Using this observation, we can express $\mathbf{M}_6$ as

$$\mathbf{M}_6 = \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} \left(H^{\otimes (\mathfrak{k}-1)} \otimes (XH) \right).$$

Define $\tilde{H}_{\mathfrak{k}} := \left(H^{\otimes (\mathfrak{k}-1)} \otimes (XH) \right)$, the quantum circuit in Figure 8 implements a $(1, 1)$ -block-encoding of $\mathbf{M}_6$ with gate complexity $O(\log(k))$.

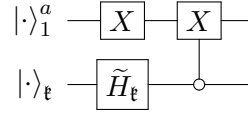


Figure 8: A quantum circuit that implements the block-encoding of $\mathbf{M}_6$, denoted by $U_{\mathbf{M}_6}$.

Moreover, the matrix $\mathbf{M}_7$ can be viewed as the top-left block of the following matrix

$$\left[\begin{array}{c|c} \mathbf{I}_m & \mathbf{O}_m \\ \hline \mathbf{O}_m & \mathbf{I}_m \end{array} \right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \otimes \mathbf{I}_m,$$

which can be implemented using a CNOT gate. The corresponding quantum circuit is shown in Figure 9.

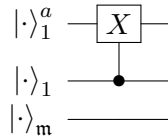


Figure 9: A quantum circuit that implements the block-encoding of $\mathbf{M}_7$, denoted by $U_{\mathbf{M}_7}$.

With the block-encodings of $\mathbf{M}_6$ and $\mathbf{M}_7$, we can obtain the block-encoding of matrix $\mathbf{L}_2^{(1)}$ using Lemma B.2. Noting that $\mathbf{L}_2^{(2)}$ can be implemented by an addition module on the register $|\cdot\rangle_{m+1}$, and applying Lemma B.3, the matrix $\mathbf{L}_2$ can be implemented using the quantum circuit shown in Figure 10. This circuit provides a $(1, 2)$ -block-encoding with gate complexity $O(\log(m \cdot k))$.

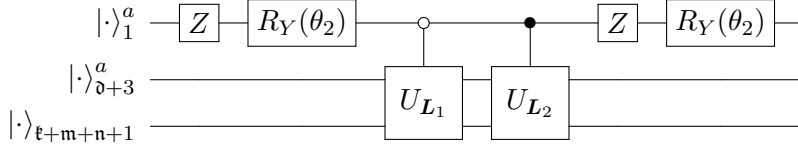


Figure 11: A quantum circuit that implements the block-encoding of $\mathbf{L}_{m,k,p}(\mathbf{A})$.

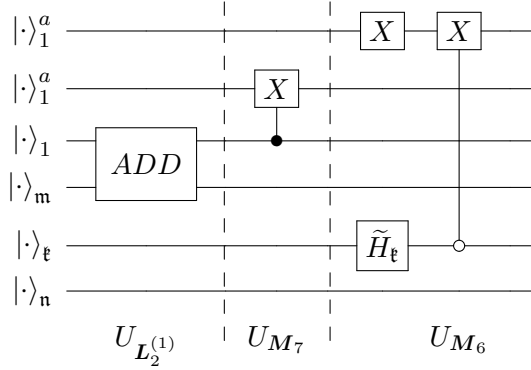


Figure 10: A quantum circuit that implements the block-encoding of $\mathbf{L}_2$, denoted by U_{L_2} .

3.3.3 Block-encoding of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$

In summary, we now have a $(3, \mathfrak{d} + 3)$ -block-encoding of $\mathbf{L}_1$ and a $(1, 2)$ -block-encoding of $\mathbf{L}_2$. Using the following circuit, we can obtain a $(4, \mathfrak{d} + 4)$ -block-encoding of $\mathbf{L}_{m,k,p}(\mathbf{A}) = \mathbf{L}_1 + \mathbf{L}_2$ with gate complexity $O(k + \log(m \cdot k))$. We choose $\theta_2 = \pi/3$ such that

$$R_Y(\theta_2)Z = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 1 & \\ & -1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}.$$

Currently, we have implemented the block-encoding of $\mathbf{L}_{m,k,p}(\mathbf{A})$ based on a given $(1, \mathfrak{d})$ -block-encoding of $\mathbf{A}$. However, in our application, we need to implement the block-encoding of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$, and we may only have an $(\alpha, \mathfrak{d})$ -block-encoding of $\mathbf{A}$. If $h = \frac{1}{\alpha}$, we can use this block-encoding as before. However, if $h \neq 1/\alpha$, some modifications are needed:

- If $h < \frac{1}{\alpha}$, using Lemma B.5, we obtain a $(\frac{1}{h}, \mathfrak{d} + 1)$ -block-encoding of $\mathbf{A}$, which can be directly applied in the framework discussed above. As a result, we will achieve a $(4, \mathfrak{d} + 5)$ -block-encoding of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$.
- If $h > \frac{1}{\alpha}$, we apply Lemma B.5 to all of the block-encodings in the above construction, except $U_{\mathbf{A}}$, scaling them by a factor of $\frac{1}{\alpha h}$. As a result, we will obtain a $(4\alpha h, \mathfrak{d} + 5)$ -block-encoding of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$.

Hence, we have completed the constructive proof of Theorem 3.8.

4 Complexity analysis

With the help of the upper bound of the condition number of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$, we proceed with the complexity analysis of the designed algorithm. We begin by discussing how to choose

the parameters m and k to control the absolute error of the resulting approximation. Then, given fixed values of m and k , we explore how to select p such that the algorithm succeeds with constant positive probability.

4.1 Approximation accuracy

Given m and k , we aim to bound the errors

$$\|\hat{\mathbf{x}}(ih) - \mathbf{x}(ih)\|_2, \quad \forall i = 1, \dots, m, \quad (32)$$

where $\mathbf{x}(ih)$ is the exact solution to eq. (3) at time ih , and $\hat{\mathbf{x}}(ih)$ is the approximation defined in eq. (10). Throughout this section, we assume that $\|\mathbf{A}T\|_2 \geq 1$. We first bound these errors in a manner similar to the analysis in [7, 13, 32], for the case where $m = \lceil \|\mathbf{A}T\|_2 \rceil$, implying that $\|\mathbf{A}h\|_2 \leq 1$. The result is summarized in the following lemma, with the proof provided in Appendix C.1.

Lemma 4.1. *Suppose $\|\mathbf{A}T\|_2 \geq 1$. Let $m = \lceil \|\mathbf{A}T\|_2 \rceil$ and $\delta \in (0, \frac{1}{m})$. If k satisfies*

$$\frac{k!k!}{(2k)!(2k+1)!} \leq \frac{\delta}{100}, \quad (33)$$

then we have

$$\|\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h)\|_2 \leq \delta \|\mathbf{A}T\|_2, \quad \forall i = 1, \dots, m, \quad (34)$$

and

$$\|\hat{\mathbf{x}}(ih) - \mathbf{x}(ih)\|_2 \leq \delta T \cdot (\|\mathbf{A}\|_2 \|\mathbf{x}(ih)\|_2 + \|\mathbf{b}\|_2), \quad \forall i = 1, \dots, m. \quad (35)$$

Next, we bound these errors from another perspective. Given k and the desired approximation accuracy, we aim to determine how large the step size h can be. This may relax the condition that $\|\mathbf{A}h\|_2 \leq 1$. To achieve this, we draw a tighter bound on the remainder of Padé approximation. As pointed out in [16], the remainder can be expressed as the following series,

$$\rho_k(x) := e^{-x} R_{kk}(x) - 1 = \sum_{j=2k+1}^{\infty} c_j x^j, \quad (36)$$

where the series converges absolutely for $|x| < \min\{|t| : D_{kk}(t) = 0\} =: \nu_k$, which is the smallest absolute value among the zero points of $D_{kk}(\cdot)$. Therefore, let

$$\mathbf{G} := e^{-\mathbf{A}h} R_{kk}(\mathbf{A}h) - \mathbf{I},$$

and we obtain the following bound

$$\|\mathbf{G}\|_2 = \|\rho_k(\mathbf{A}h)\|_2 \leq \sum_{j=2k+1}^{\infty} |c_j| \theta^j =: f_k(\theta), \quad (37)$$

where $\theta := \|\mathbf{A}h\|_2 < \nu_k$. It is clear that if $\mathbf{A}$ is a general matrix and only $\|\mathbf{A}\|$ is known, then eq. (37) provides the tightest bound for $\|\mathbf{G}\|_2$. We state the following lemma, whose proof is provided in Appendix C.1.

Lemma 4.2. *Let $\delta \in (0, \frac{1}{\|\mathbf{A}T\|_2})$, with $\|\mathbf{A}T\|_2 \geq 1$, and let m be chosen such that*

$$\frac{f_k(\theta)}{\theta} \leq \frac{\delta}{e-1}, \quad (38)$$

k	5	6	7	8	9	10	11	12	13	14	15	16	17	18
θ_k	1.49	2.36	3.34	4.40	5.53	6.69	7.89	9.11	10.35	11.61	12.88	14.16	15.45	16.74

Table 1: Maximal values θ_k of $\|\mathbf{A}h\|_2$ such that the condition eq. (38) is satisfied with $\delta = 10^{-8}$.

where $\theta := \|\mathbf{A}h\|_2$ and $f_k(\cdot)$ is defined in eq. (37). Then, we have

$$\|\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h)\|_2 \leq \delta \|\mathbf{A}T\|_2, \quad \forall i = 1, \dots, m, \quad (39)$$

and

$$\|\hat{\mathbf{x}}(ih) - \mathbf{x}(ih)\|_2 \leq \delta T \cdot (\|\mathbf{A}\|_2 \|\mathbf{x}(ih)\|_2 + \|\mathbf{b}\|_2), \quad \forall i = 1, \dots, m. \quad (40)$$

In practice, we can evaluate $f_k(\theta)$ to high precision using symbolic computation, as demonstrated in [16]. Let $\delta = 10^{-8}$. We apply the bisection method to find the largest θ that satisfies eq. (38), denoted by θ_k . The results, shown in Table 1, indicate that $\|\mathbf{A}h\|_2$ can be significantly larger than 1. Therefore, if the approximation order k is predetermined, Lemma 4.2 allows us to choose h without the restriction $\|\mathbf{A}h\|_2 \leq 1$. This is especially advantageous when $\mathbf{A}$ is Hermitian and negative semi-definite, as the condition number of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ can be bounded without the $\|\mathbf{A}h\|_2 \leq 1$ condition.

4.2 Success probability

In quantum computation, we are concerned with the success probability of the algorithm. In the case of solving eq. (22), the resulting vector is stored in a quantum superposition state. After measuring this superposition, we obtain the desired quantum state with a certain probability. The equation below illustrates how the resulting vector is stored in a quantum state,

$$\frac{1}{C} \begin{bmatrix} \mathbf{z}^{(1)} \\ \mathbf{z}^{(2)} \\ \vdots \\ \mathbf{z}^{(m)} \\ \hat{\mathbf{x}} \\ \hat{\mathbf{x}} \\ \vdots \\ \hat{\mathbf{x}} \end{bmatrix} = \sum_{i=0}^{m-1} \sum_{j=0}^k \frac{\|\mathbf{z}_j^{(i)}\|_2}{C} |i(k+1) + j\rangle |\mathbf{z}_j^{(i)}\rangle + \sum_{a=0}^{p-1} \frac{\|\hat{\mathbf{x}}\|_2}{C} |m(k+1) + a\rangle |\hat{\mathbf{x}}\rangle, \quad (41)$$

where $C = \sqrt{\sum_{i=1}^m \|\mathbf{z}^{(i)}\|_2^2 + p\|\hat{\mathbf{x}}\|_2^2}$ is a normalization factor. In the rest of this paper, we refer to the qubits in the first ket as the index register, while the qubits in the second ket as the value register. We measure the index register and obtain a result greater than $m(k+1) - 1$ with probability

$$\mathbb{P}_{succ} := \frac{p\|\hat{\mathbf{x}}\|_2^2}{\sum_{i=1}^m \|\mathbf{z}^{(i)}\|_2^2 + p\|\hat{\mathbf{x}}\|_2^2}.$$

Conditioned on the measuring result of the index register being greater than $m(k+1) - 1$, the quantum state of the value register is $|\hat{\mathbf{x}}\rangle$, and the algorithm successfully provides the desired result.

Theorem 4.3. Let $\mathbf{A} \in \mathbb{C}^{n \times n}$, $T > 0$, and $\delta > 0$ satisfy the condition

$$\delta' := \delta T \cdot \left(\|\mathbf{A}\|_2 + \frac{\|\mathbf{b}\|_2}{\|\mathbf{x}(T)\|_2} \right) < \frac{1}{8}.$$

Then, the following results hold:

1. if $\mathbf{A}$ is Hermitian and negative semi-definite, and the conditions of Lemma 4.2 are satisfied, we have

$$\mathbb{P}_{\text{succ}} \geq \frac{1}{2} \cdot \frac{p}{6mg^2(h^2 + 1) + p}, \quad (42)$$

where

$$g := \frac{\max \{ \max_{0 \leq t \leq T} \|\mathbf{x}(t)\|_2, \|\mathbf{b}\|_2 \}}{\|\mathbf{x}(T)\|_2}.$$

2. if $\mathbf{A}$ is an arbitrary matrix, m is chosen such that $\|\mathbf{A}h\|_2 \leq 1$, and the conditions of Lemma 4.1 are satisfied, we have

$$\mathbb{P}_{\text{succ}} \geq \frac{1}{2} \cdot \frac{p}{204mg^2(h^2 + 1) + p}. \quad (43)$$

In summary, for both scenarios, let $p = \lceil 6m(1 + h^2) \rceil$, then we have

$$\mathbb{P}_{\text{succ}} \geq \frac{1}{2} \cdot \frac{6}{204g^2 + 6} \geq \frac{1}{70g^2}.$$

The proof of Theorem 4.3 is provided in Appendix C.2.

4.3 Main Result

Before stating our main result, we first fill a small gap between the QLSA proposed in [12] and our application. The QLSA, as presented in [12], is as follows.

Theorem 4.4 ([12], Theorem 19). Let $\mathbf{L}$ be a matrix such that $\|\mathbf{L}\|_2 = 1$ and $\|\mathbf{L}^{-1}\|_2 = \kappa$. Given an oracle block-encoding of $\mathbf{L}$ and an oracle for implementing $|\mathbf{b}\rangle$, there exists a quantum algorithm that produces the normalized state $|\mathbf{L}^{-1}\mathbf{b}\rangle$ within an error ϵ , using $\mathcal{O}(\kappa \log \frac{1}{\epsilon})$ calls to the oracles.

Theorem 4.4 requires $\|\mathbf{L}\|_2 = 1$ and a $(1, *)$ -block-encoding of $\mathbf{L}$, but the matrix $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ does not satisfy these conditions. Specifically, its norm can exceed one, and its block-encoding, as given by Theorem 3.8, is not a $(1, *)$ -block-encoding. To address this, suppose we have only an $(\alpha, *)$ -block-encoding of $\mathbf{L}$, where $\alpha \geq \|\mathbf{L}\|_2$. We illustrate the impact of this modification in the case where $\mathbf{L}$ is Hermitian and positive definite. To perform the adiabatic quantum simulation, which is the key step of the QLSA, the initial Hamiltonian is chosen as $H_0 = \begin{bmatrix} 0 & Q_{\mathbf{b}} \\ Q_{\mathbf{b}} & 0 \end{bmatrix}$, the same as in [12], where $Q_{\mathbf{b}} = I - |\mathbf{b}\rangle\langle\mathbf{b}|$.

However, the terminal Hamiltonian must be $H_1 = \begin{bmatrix} 0 & \frac{1}{\alpha}\mathbf{L}Q_{\mathbf{b}} \\ \frac{1}{\alpha}Q_{\mathbf{b}}\mathbf{L} & 0 \end{bmatrix}$ to match the block-encoding of $\mathbf{L}$. The Hamiltonian used in the adiabatic quantum computation is then of the form:

$$H(s) = (1 - f(s))H_0 + f(s)H_1,$$

where $f(s) : [0, 1] \rightarrow [0, 1]$ is the schedule function. Then the quantity $\Delta_0(s)$ (Definition 2 in [12]), which indicates the eigenvalues' gap of $H(s)$, is given by

$$\Delta_0(s) = 1 - f(s) + \frac{f(s)}{\alpha \|\mathbf{L}^{-1}\|_2},$$

instead of $\Delta_0(s) = 1 - f(s) + \frac{f(s)}{\kappa}$ in [12]. This shows that $\alpha \|\mathbf{L}^{-1}\|_2$ plays the role of κ in Theorem 4.4. Therefore, under these settings, the query complexity for preparing the state $|\mathbf{L}^{-1}\mathbf{b}\rangle$ within error ϵ becomes $\mathcal{O}(\alpha \|\mathbf{L}^{-1}\|_2 \log \frac{1}{\epsilon})$. In the ideal case, where the block-encoding satisfies $\alpha = \|\mathbf{L}\|_2$, this simplifies to $\alpha \|\mathbf{L}^{-1}\|_2 = \kappa$, and the query complexity matches the result of Theorem 4.4.

Theorem 4.5. Suppose $\mathbf{A} \in \mathbb{C}^{n \times n}$ and an $(\alpha, \mathfrak{d})$ -block-encoding of $\mathbf{A}$, denoted by $U_{\mathbf{A}}$, is available. Let $\mathbf{x}_0$ and $\mathbf{b}$ be n -dimensional vectors with known norms, and assume access to two controlled oracles, $O_{\mathbf{x}}$ and $O_{\mathbf{b}}$, which prepare quantum states proportional to $\mathbf{x}_0$ and $\mathbf{b}$, respectively. Let $\mathbf{x}(t)$ evolve according to the differential equation

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{b}, \quad t \in [0, T], \quad (44)$$

with the initial condition $\mathbf{x}(0) = \mathbf{x}_0$. Without loss of generality, assume that $\|\mathbf{A}\mathbf{T}\|_2 \geq 1$, and choose $m = \lceil \|\mathbf{A}\mathbf{T}\|_2 \rceil$. Under these conditions, the proposed algorithm produces a quantum state that is ϵ -close to $\mathbf{x}(T)/\|\mathbf{x}(T)\|_2$ in l^2 norm, with a success probability of $\Omega(1)$, and includes a flag indicating success. The algorithm makes

$$\mathcal{O} \left((h^2 + 1) \cdot \alpha T \cdot C(\mathbf{A}) \cdot g \cdot \sqrt{k \log k} \cdot \log \frac{\sqrt{m(h^2 + 1)}g}{\epsilon} \right) \quad (45)$$

queries to $U_{\mathbf{A}}$, $O_{\mathbf{x}}$, and $O_{\mathbf{b}}$, where

$$C(\mathbf{A}) := \max_{0 \leq t \leq T} \|\exp(\mathbf{A}t)\|_2, \quad g := \frac{\max \{ \max_{0 \leq t \leq T} \|\mathbf{x}(t)\|_2, \|\mathbf{b}\|_2 \}}{\|\mathbf{x}(T)\|_2},$$

$$k = \left\lceil \frac{\log M}{\log \log M} \right\rceil, \quad M = \frac{401T}{\epsilon} \left(\|\mathbf{A}\|_2 + \frac{\|\mathbf{b}\|_2}{\|\mathbf{x}(T)\|_2} \right), \quad \text{and} \quad \epsilon < \frac{1}{2}.$$

The gate complexity of this algorithm is larger than its query complexity by a factor of

$$\mathcal{O}(k + \text{poly log}(m \cdot k)). \quad (46)$$

Proof. Note that the choice of k ensures that

$$\frac{k!k!}{(2k)!(2k+1)!} \leq \frac{1}{M}, \quad (47)$$

and $k \log k = \mathcal{O}(\log M)$. Next, let δ satisfy

$$\frac{100}{M} \leq \delta \leq \frac{\epsilon}{4T \left(\|\mathbf{A}\|_2 + \frac{\|\mathbf{b}\|_2}{\|\mathbf{x}(T)\|_2} \right)} < \frac{1}{\|\mathbf{A}\mathbf{T}\|_2},$$

such that condition eq. (33) is satisfied. With the choice of δ , we have

$$\delta T \cdot \left(\|\mathbf{A}\|_2 + \frac{\|\mathbf{b}\|_2}{\|\mathbf{x}(T)\|_2} \right) \leq \frac{\epsilon}{4} < \frac{1}{8}.$$

Using Lemma 4.1, we have

$$\|\hat{\mathbf{x}}(ih) - \mathbf{x}(ih)\|_2 \leq \delta T (\|\mathbf{A}\|_2 \|\mathbf{x}(ih)\|_2 + \|\mathbf{b}\|_2), \quad \forall i = 1, \dots, m, \quad (48)$$

and

$$\|\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h)\|_2 \leq \delta \|\mathbf{A}T\|_2 < 1, \quad \forall i = 1, \dots, m. \quad (49)$$

Therefore, the condition (25) of Theorem 3.6 is satisfied. We then set $p = \lceil 6m(1 + h^2) \rceil$ and construct the linear system eq. (22). Choosing $i = m$ in (48), we obtain

$$\frac{\|\hat{\mathbf{x}}(T) - \mathbf{x}(T)\|_2}{\|\mathbf{x}(T)\|_2} < \frac{\epsilon}{4},$$

where $\hat{\mathbf{x}}(T)$ is the vectors corresponding to $\hat{\mathbf{x}}$ in eq. (22). Using Lemma D.1, we then have

$$\|\mathbf{x}(T)\rangle - |\hat{\mathbf{x}}(T)\rangle\|_2 \leq \frac{\epsilon}{2}. \quad (50)$$

We use the QLSA proposed in [12] to solve the linear system, which may introduce errors. This method requires the block-encoding of the linear operator $\mathbf{L}_{m,k,p}(\mathbf{A}h)$, and in Section 3.3, we propose a way to block-encode it using a single query to $U_{\mathbf{A}}$. Additionally, the right-hand-side vector in the linear system eq. (22) can be formed with a constant number of calls to $O_{\mathbf{x}}$ and $O_{\mathbf{b}}$, as shown in [7, 32]. According to Theorem 3.6, Theorem 3.8, and Theorem 4.4, to obtain an ϵ' -close solution to the linear system eq. (22), the query complexity of the QLSA is given by

$$\mathcal{O}\left(\alpha h \cdot C(\mathbf{A}) \cdot (m + p) \cdot \sqrt{k \log k} \cdot \log \frac{1}{\epsilon'}\right) = \mathcal{O}\left((h^2 + 1) \cdot \alpha T \cdot C(\mathbf{A}) \cdot \sqrt{k \log k} \cdot \log \frac{1}{\epsilon'}\right), \quad (51)$$

where we set $p = \lceil 6m(1 + h^2) \rceil$ and $h = T/m$ to arrive at the right-hand side. After solving the linear system, we measure the index register of the quantum state eq. (41) in the standard basis. Conditioned on the outcome being in

$$S := \{m(k + 1), m(k + 1) + 1, \dots, m(k + 1) + p - 1\},$$

we output the state of the value register. We will show that it is sufficient to choose $\epsilon' = \mathcal{O}\left(\frac{\epsilon}{\sqrt{m(h^2 + 1)g}}\right)$, such that the probability of this event occurring is $\Omega(1/g^2)$, and the output state is $\epsilon/2$ -close to the state $|\hat{\mathbf{x}}(T)\rangle$.

Equation (41) gives the normalized exact solution to the linear system eq. (22). Let $d = m(k + 1) + p$, and define $\mathbf{x}_l = \mathbf{z}_j^{(i)}$ for $l = i(k + 1) + j$. The normalized exact solution can then be expressed as

$$|\mathbf{x}\rangle = \sum_{l=0}^d \gamma_l |l\rangle |\mathbf{x}_l\rangle,$$

where $\gamma_l = \frac{\|\mathbf{x}_l\|_2}{\|\mathbf{x}\|_2}$. Note that $|\mathbf{x}_l\rangle = |\hat{\mathbf{x}}(T)\rangle$ for any $l \in S$, and using Theorem 4.3, we have

$$\gamma_l = \frac{\|\hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \geq \frac{1}{c\sqrt{mg}}, \quad \forall l \in S,$$

where $c = \sqrt{420(h^2 + 1)}$. Now suppose the QLSA outputs the state

$$|\mathbf{x}'\rangle = \sum_{l=0}^d \gamma'_l |l\rangle |\mathbf{x}'_l\rangle$$

which satisfies

$$\| |\mathbf{x}\rangle - |\mathbf{x}'\rangle \|_2 \leq \epsilon'.$$

Then, for any $l \in S$, by Lemma D.2, we have

$$\| |\mathbf{x}_l\rangle - |\mathbf{x}'_l\rangle \|_2 \leq \frac{2\epsilon'}{\gamma_l - \epsilon'}.$$

Choosing $\epsilon' = \frac{\epsilon}{5c\sqrt{mg}}$, we get

$$\frac{2\epsilon'}{\gamma_l - \epsilon'} \leq \frac{\frac{2\epsilon}{5c\sqrt{mg}}}{\frac{1}{c\sqrt{mg}} - \frac{\epsilon}{5c\sqrt{mg}}} = \frac{2\epsilon}{5 - \epsilon} < \frac{\epsilon}{2}.$$

Thus, we have

$$\| |\mathbf{x}'_l\rangle - |\mathbf{x}(T)\rangle \|_2 \leq \| |\mathbf{x}(T)\rangle - |\mathbf{x}_l\rangle \|_2 + \| |\mathbf{x}_l\rangle - |\mathbf{x}'_l\rangle \|_2 < \epsilon, \quad \forall l \in S.$$

Furthermore, by Lemma D.3, we have

$$\gamma'_l \geq \gamma_l - \epsilon' \geq \frac{9}{10c\sqrt{mg}}.$$

Therefore, if we measure the index register of $|\mathbf{x}'\rangle$ in the standard basis, the probability of getting outcome $l \in S$ is

$$\sum_{l \in S} |\gamma'_l|^2 \geq \frac{81p}{100c^2mg^2} = \frac{81}{7000g^2},$$

and when this occurs, the state of the value register becomes $|\mathbf{x}'_l\rangle$, which is ϵ -close to the desired state $|\mathbf{x}(T)\rangle$ in l^2 norm. Using amplitude amplification [8], we can raise this probability to $\Omega(1)$ with $\mathcal{O}(g)$ repetitions of the above procedure.

In summary, the query complexity is

$$\mathcal{O} \left((h^2 + 1) \cdot \alpha T \cdot C(\mathbf{A}) \cdot g \cdot \sqrt{k \log k} \cdot \log \frac{\sqrt{m(h^2 + 1)g}}{\epsilon} \right).$$

Finally, the total gate complexity is multiplied by the query complexity along with the gate complexity for block-encoding, as given in Theorem 3.8. $\square$

Corollary 4.6. *Suppose $\mathbf{A} \in \mathbb{C}^{n \times n}$ is Hermitian and negative semi-definite, and an $(\alpha, \mathfrak{d})$ -block-encoding of $\mathbf{A}$, denoted by $U_{\mathbf{A}}$, is available. Let $\mathbf{x}_0$ and $\mathbf{b}$ be n -dimensional vectors with known norms, and assume access to two controlled oracles, $O_{\mathbf{x}}$ and $O_{\mathbf{b}}$, which prepare quantum states proportional to $\mathbf{x}_0$ and $\mathbf{b}$, respectively. Let $\mathbf{x}(t)$ evolve according to the differential equation*

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{b}, \quad t \in [0, T], \quad (52)$$

with the initial condition $\mathbf{x}(0) = \mathbf{x}_0$. Without loss of generality, assume that $\|\mathbf{A}\mathbf{T}\|_2 \geq 1$, and let the parameter m be chosen to satisfy the condition (38) in Lemma 4.2. Under these conditions, the proposed algorithm produces a quantum state that is ϵ -close to $\mathbf{x}(T)/\|\mathbf{x}(T)\|_2$ in l^2 norm, with a success probability of $\Omega(1)$, and includes a flag indicating success. The algorithm makes

$$\mathcal{O} \left((h^2 + 1) \cdot \alpha T \cdot g \cdot \sqrt{k \log k} \cdot \log \frac{\sqrt{m(h^2 + 1)g}}{\epsilon} \right) \quad (53)$$

queries to $U_{\mathbf{A}}$, $O_{\mathbf{x}}$, and $O_{\mathbf{b}}$, where the parameters g and k are defined as in Theorem 4.5. The gate complexity is larger than its query complexity by a factor of

$$\mathcal{O}(k + \text{poly} \log(m \cdot k)). \quad (54)$$

Remark 4.7. The difference between Theorem 4.5 and Corollary 4.6 lies in the assumption on $\mathbf{A}$. In Theorem 4.5, $\mathbf{A}$ is a general matrix where as in Corollary 4.6, $\mathbf{A}$ is a Hermitian and negative semi-definite matrix. When $\mathbf{A}$ is a general matrix, we choose $m = \lceil \|\mathbf{A}\mathbf{T}\|_2 \rceil$ such that $\|\mathbf{A}h\|_2 = \left\| \mathbf{A} \frac{\mathbf{T}}{m} \right\|_2 \leq 1$. When $\mathbf{A}$ is a Hermitian and negative semi-definite matrix, we apply Lemma 4.2 to choose $m \leq \lceil \|\mathbf{A}\mathbf{T}\|_2 \rceil$ such that the bounds in (48) and (49) remain valid. By proceeding with the same construction and applying the results tailored to this special case, we obtain the same asymptotic complexity as in the general case, with a potentially smaller constant prefactor. Notably, in this setting $C(\mathbf{A}) = 1$, and thus it does not appear in the final query complexity.

5 Comparison with previous methods

In this section, we compare our method with two other approaches: the method based on Taylor approximation [7, 13, 32] and the method based on linear combination of Hamiltonian simulation (LCHS) [2], which achieves near-optimal dependence on all parameters. First, we compare the theoretical query complexity of the three methods. Then, we conduct numerical comparisons between our method and the Taylor approximation-based method in two distinct scenarios.

- **Scenario I:** For a given desired precision ϵ and a fixed approximation order k , we compare the smallest values of m required by both methods to achieve the condition $\frac{\|\hat{\mathbf{x}} - \mathbf{x}(T)\|_2}{\|\mathbf{x}(T)\|_2} < \epsilon$.
- **Scenario II:** For a given desired precision ϵ and $m = \lceil \|\mathbf{A}\mathbf{T}\|_2 \rceil$, we compare the smallest approximation order k required by both methods to achieve the condition $\frac{\|\hat{\mathbf{x}} - \mathbf{x}(T)\|_2}{\|\mathbf{x}(T)\|_2} < \epsilon$.

5.1 Theoretical comparison

We use the result from [13] as the theoretical query complexity for the method based on Taylor approximation. In [13], an oracle $O_{\mathbf{A}}$ is used to compute the non-zero entries of $\mathbf{A}$. However, in our setting, we assume the availability of a $(\alpha, \mathfrak{d})$ -block-encoding of $\mathbf{A}$, denoted as $U_{\mathbf{A}}$. To make their results and ours comparable, we modify their setting accordingly. In this modified setting, their algorithm requires

$$\mathcal{O}\left(\alpha T \cdot C(\mathbf{A}) \cdot g \cdot k \cdot \log \frac{1}{\delta}\right), \quad (55)$$

queries to $U_{\mathbf{A}}$, $O_{\mathbf{x}}$ and $O_{\mathbf{b}}$, where the parameters are defined as

$$\delta = \frac{\epsilon}{25\sqrt{mg}}, \quad k = \left\lfloor \frac{2 \log \Omega}{\log \log \Omega} \right\rfloor, \quad \Omega = \frac{2e^3 T}{\delta} \left(\|\mathbf{A}\|_2 + \frac{e^2 \|\mathbf{A}\mathbf{T}\|_2 \|\mathbf{b}\|_2}{\|\mathbf{x}(T)\|_2} \right), \quad m = \lceil \|\mathbf{A}\mathbf{T}\|_2 \rceil,$$

and $g = \frac{\max\{\max_{0 \leq t \leq T} \|\mathbf{x}(t)\|_2, \|\mathbf{b}\|_2\}}{\|\mathbf{x}(T)\|_2}$.¹ By comparing the query complexity in eq. (55) with that in eq. (45), our approach provides the following improvements.

- The dependence on the approximation order k is improved. This improvement arises from the factor $\frac{1}{\sqrt{k+1}}$ introduced in the linear system eq. (22). Due to the special structure of the linear system, incorporating this factor reduces the spectral norm of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ by a factor of $\frac{1}{\sqrt{k+1}}$, while only increases the spectral norm of $\mathbf{L}_{m,k,p}(\mathbf{A}h)^{-1}$ by a constant factor. Since $k = \tilde{\mathcal{O}}\left(\log\left(\frac{1}{\epsilon}\right)\right)$, this improvement leads to a better dependence on precision, which will be later shown in Table 2.
- The parameter Ω is larger than the corresponding parameter M in our analysis by a factor of $\sqrt{mg}$, as we use eq. (50) instead of the condition $\|\mathbf{x}(T)\rangle - |\hat{\mathbf{x}}\rangle\|_2 \leq \delta$ (equation (118) in [7]), where $\delta = \mathcal{O}\left(\frac{\epsilon}{\sqrt{mg}}\right)$. Additionally, Ω includes an extra term $\|\mathbf{A}T\|_2$ compared to M . We eliminate this term by employing an optimized upper bound on $\left\|\left(\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h)\right) \mathbf{A}^{-1}\right\|_2$, which is applied in Lemma 4.2 and Lemma 4.1. The corresponding result in [32] is Lemma 10.
- Even if we assume $M = \Omega$, the choice of k in the original method is twice as large as our approach. This difference arises because the remainder of the Padé approximation includes a factor of $\frac{k!k!}{(2k)!(2k+1)!}$, whereas the remainder of Taylor approximation only has a factor of $\frac{1}{(k+1)!}$. This improvement directly reduces the gate complexity, as the gate complexity for block-encoding $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ (or $\mathbf{C}_{m,k,p}(\mathbf{A}h)$) scales linearly with k .
- If matrix $\mathbf{A}$ is Hermitian and negative semi-definite, we can choose $m < \lceil \|\mathbf{A}T\|_2 \rceil$ while still maintaining a theoretical guarantee for our algorithm. This advantage arises from the special structure of $\mathbf{W}_k(\mathbf{A}h)$. Lemma 3.4 provides an upper bound on the spectral norm of $\mathbf{W}_k(\mathbf{A}h)^{-1}$ that is independent of $\|\mathbf{A}h\|_2$, whereas the spectral norm of $\mathbf{M}_k(\mathbf{A}h)$ may depend exponentially on $\|\mathbf{A}h\|_2$. A smaller m directly reduces the total query complexity and gate complexity. More importantly, the numerical results indicate that a smaller m is also associated with a lower condition number and a higher success probability $\mathbb{P}_{succ}$, which can significantly decrease the overall query complexity.

Table 2 compares our method with the other two methods in terms of query complexity, with logarithm terms omitted except for $1/\epsilon$. The method based on LCHS achieves the best query complexity to $O_{\mathbf{x}}$ and $O_{\mathbf{b}}$. Regarding the query complexity to $\mathbf{U}_{\mathbf{A}}$, our method has a fixed dependence on precision, whereas the LCHS method's dependence is determined by the parameter β . Additionally, the query complexity of our method depends on the quantities $\max_{0 \leq t \leq T} \|\mathbf{x}(t)\|_2$ and $C(\mathbf{A})$, while the LCHS method depends on $\|\mathbf{b}\|_2 T$. When $\beta = 2/3$, both methods exhibit the same dependence on precision, but the overall complexity is determined by the specific choice of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{x}_0$.

¹Note that the definition of g in this paper differs slightly from that in [13] and [32]. Specifically, equation (4.22) of [32] should include an additional term, $\sum_{i=0}^m |i, 1, h\mathbf{b}\rangle$, and equation (5.87) should have an extra term, $mh^2\|\mathbf{b}\|_2^2$, in the denominator. These necessary adjustments lead to the definition of g adopted in this work.

Method	Queries to U_A	Queries to O_x and O_b
Taylor approximation [7, 13, 32]	$\tilde{\mathcal{O}}\left(\frac{\max\{\max_{0 \leq t \leq T} \ \mathbf{x}(t)\ _2, \ \mathbf{b}\ _2\}}{\ \mathbf{x}(T)\ _2} C(\mathbf{A}) \alpha T \left(\log\left(\frac{1}{\epsilon}\right)\right)^2\right)$	
LCHS [2, 3]	$\tilde{\mathcal{O}}\left(\frac{\ \mathbf{x}(0)\ _2 + \ \mathbf{b}\ _2 T}{\ \mathbf{x}(T)\ _2} \alpha T \left(\log\left(\frac{1}{\epsilon}\right)\right)^{1/\beta}\right)$	$\mathcal{O}\left(\frac{\ \mathbf{x}(0)\ _2 + \ \mathbf{b}\ _2 T}{\ \mathbf{x}(T)\ _2}\right)$
Padé approximation	$\tilde{\mathcal{O}}\left(\frac{\max\{\max_{0 \leq t \leq T} \ \mathbf{x}(t)\ _2, \ \mathbf{b}\ _2\}}{\ \mathbf{x}(T)\ _2} C(\mathbf{A}) \alpha T \left(\log\left(\frac{1}{\epsilon}\right)\right)^{1.5}\right)$	

Table 2: Comparison of the query complexity. These complexities apply to ODEs with time-independent $\mathbf{A}$ and $\mathbf{b}$, with logarithm terms omitted except for $1/\epsilon$. Here, $\alpha \geq \|\mathbf{A}\|_2$, T is the evolution time, ϵ is the desired accuracy, $\beta \in (0, 1)$ is a chosen constant, and $C(\mathbf{A}) = \max_{0 \leq t \leq T} \|\exp(\mathbf{A}t)\|_2$.

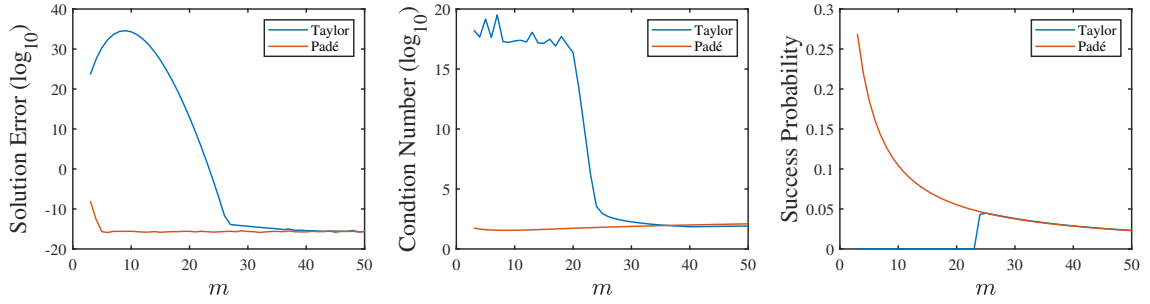


Figure 12: Comparison of the relative solution error, condition number, and success probability $\mathbb{P}_{succ}$ as m increases, between the method based on Padé approximation and the method based on Taylor approximation.

5.2 Numerical comparison

In this section, we present numerical experiments to compare our method with the method based on Taylor approximation.

5.2.1 Numerical result of scenario I

Experiment 1: Let $T = 30$, $k = 9$, $p = 1$,

$$\mathbf{A} = \begin{bmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}, \quad \text{and} \quad \mathbf{x}_0 = \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

We construct $\mathbf{C}_{m,k,1}(\mathbf{A}h)$ and $\mathbf{L}_{m,k,1}(\mathbf{A}h)$, and solve the corresponding linear systems for different values of m . The results, including relative solution error, condition number, and the success probability $\mathbb{P}_{succ}$, are shown in Figure 12.

The left figure demonstrates that the method based on Padé approximation can achieve high relative accuracy with a relatively small m , while the method based on Taylor approx-

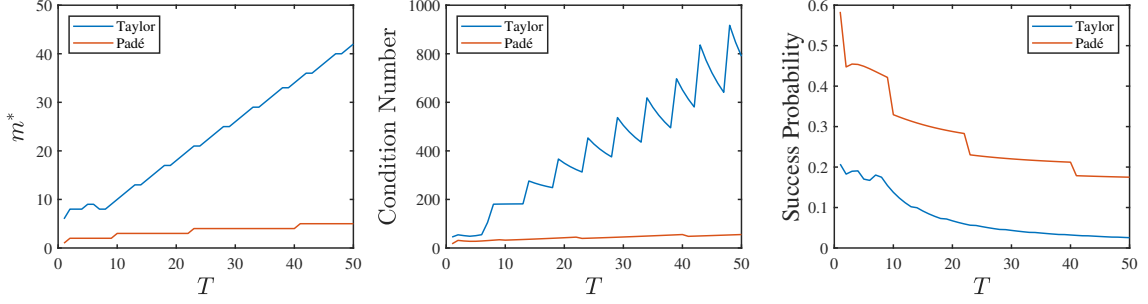


Figure 13: Comparison of the minimum choice of m for both methods under the same accuracy requirement, with varying time interval length T . The condition number and success probability $\mathbb{P}_{succ}$ of the corresponding linear system at these values of m are also compared.

imation requires a much larger m to achieve similar accuracy. The middle figure reveals that the condition number of our newly proposed method is at a low level for all values of m , while the condition number of the Taylor approximation-based method can become extremely large if m is not large enough. This observation aligns with our theoretical analysis. The right figure compares the success probability $\mathbb{P}_{succ}$. The method based on Padé approximation achieves high success probabilities for smaller values of m , while the success probabilities of both methods converge to nearly the same value as m increases sufficiently.

In summary, the first experiment demonstrates that the method based on Taylor approximation becomes unstable when m is not large enough, while the method based on Padé approximation provides satisfactory results even for relatively small m .

Experiment 2: The advantage of our newly proposed method becomes more pronounced as the time interval length T increases. To demonstrate this, we fix the accuracy $\epsilon = 10^{-10}$ and the approximation order $k = 9$, and investigate how the optimal choice m^* (the smallest m required to achieve the desired accuracy) changes with T for both methods. With the parameters $k = 9, p = 1$, and $m = m^*$, we construct and solve the corresponding linear systems. We also compute the condition number and success probability $\mathbb{P}_{succ}$ for these linear systems. The results are shown in Figure 13.

From the results, we observe that the method based on Padé approximation consistently outperforms the method based on Taylor approximation, with the gap growing more significant as T increases. A smaller m^* directly benefits the block-encoding of the linear operator, since the gate complexity in eq. (29) depends on m . Moreover, the query complexity of the QLSA is linearly dependent on the condition number of the corresponding linear system, so a smaller condition number significantly improves the solving process. In this experiment, we set $p = 1$. However, to further boost the success probability $\mathbb{P}_{succ}$, a suitable choice of $p > 1$ could be employed for both methods. A larger success probability in the $p = 1$ case implies that the corresponding method requires a smaller p to achieve the desired success probability. Furthermore, the condition number of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ and $\mathbf{C}_{m,k,p}(\mathbf{A}h)$ depends linearly on the parameter p , and thus the efficiency of QLSA is thus also linearly dependent on p .

The first two experiments compare the two methods using a specific choice of matrix $\mathbf{A}$. However, since all Hermitian matrices are unitarily diagonalizable, the results from these experiments capture the behavior of all Hermitian and negative semi-definite matrices. These results support our theoretical analysis for this class of matrices: the condition number and success probability of the proposed method remain well-controlled, even when

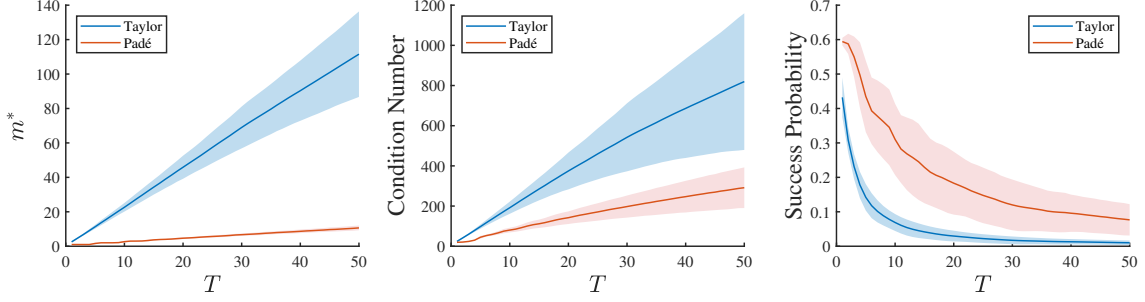


Figure 14: For varying time interval length T , and with the same accuracy requirement ϵ and approximation order k , the leftmost plot compares the average minimum choice of m for the two methods. The middle plot compares the average condition number, and the rightmost plot compares the average success probability. The shaded area represents the mean value plus or minus the standard deviation.

$$m \ll \|AT\|_2.$$

Experiment 3: The set of Hermitian and negative semi-definite matrices is a subset of the broader class of matrices whose eigenvalues have negative real parts. Within this broader class, the method based on Padé approximation is also expected to outperform the method based on Taylor approximation. However, providing a theoretical analysis for this more general case is challenging. To investigate this further, we conduct a numerical comparison between the two methods on matrices from this broader class.

Let $\epsilon = 10^{-10}$, $k = 9$, $p = 1$ and $\mathbf{x}_0 = \mathbf{b} = [1, 1, 1, 1]^T$. We generate one hundred matrices $\mathbf{A} \in \mathbb{C}^{5 \times 5}$ randomly, where each matrix's eigenvalues have negative real parts. For each matrix $\mathbf{A}$ and each time interval $T = 1, \dots, 50$, we determine the smallest value of m that achieves the desired accuracy ϵ , denoted by m^* . With the parameters $k = 9$, $p = 1$, and $m = m^*$, we construct the corresponding linear systems and compute the condition number and success probability for each linear system. Finally, we calculate the mean value and standard deviation over the one hundred samples. The results are plotted in Figure 14.

The results in Figure 14 demonstrate that the Padé approximation-based method consistently outperforms the Taylor approximation-based method across all three compared metrics, on average. Notably, the gap between the two methods widens as T increases. This experiment highlights that the proposed method remains more efficient for a broader class of matrices.

5.2.2 Numerical result of scenario II

Experiment 4: In scenario II, we always choose $m = \lceil \|AT\|_2 \rceil$. Therefore, it suffices to compare the two methods for the case where $T = 1$ and $\|\mathbf{A}\|_2 = 1$. In this experiment, we set $m = p = 1$, $\mathbf{x}_0 = \mathbf{b} = [1, 1, 1, 1]^T$, and generate one hundred matrices $\mathbf{A} \in \mathbb{C}^{5 \times 5}$ with eigenvalues having negative real parts and spectral norm equal 1. For each $\mathbf{A}$ and a required precision ϵ , we determine the smallest k that achieves the required precision, denoted by k^* . With parameters $m = p = 1$ and $k = k^*$, we construct the corresponding linear systems. We then evaluate the condition number of each linear system and calculate the mean value and standard deviation across the one hundred samples. The results are shown in Figure 15.

From the left plot of Figure 15, we observe that the average magnitude of k^* required by the Padé approximation-based method is roughly half that of the Taylor approximation-based method. This aligns with the third point in our theoretical comparison. The smaller

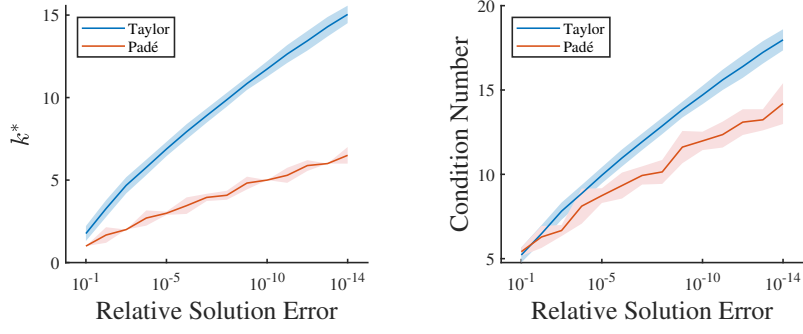


Figure 15: Comparison of the two methods for different precision requirements. The left plot compares the smallest k needed to achieve the required precision for both methods, on average. The right plot compares the average condition number of the linear systems derived from both methods with the smallest choice of k that achieves the required precision. The shaded area represents the mean value plus or minus the standard deviation.

k benefits both the total query complexity (see eq. (45)) and the gate complexity (as shown in eq. (46)). Additionally, the average condition number of the linear system based on the Padé approximation is smaller, further supporting the efficiency of the Padé approximation-based method.

6 Conclusion

We propose a quantum algorithm to solve linear autonomous ordinary differential equations (ODEs) of the form (3) using the Padé approximation. By discretizing the time step-by-step with step size h , the solution to (3) can be expressed in terms of $e^{\mathbf{A}h}$. The proposed method then uses the Padé approximation, a rational function of $\mathbf{A}h$, to approximate $e^{\mathbf{A}h}$. The matrix rational function is encoded into a large linear system $\mathbf{L}_{m,k,p}(\mathbf{A}h)$, as shown in (22), which does not explicitly involve matrix inversion. The matrix $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ has a sparse block structure, allowing it to be efficiently block encoded using an oracle query to $\mathbf{A}$. Details of the quantum circuit for encoding $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ can be found in Section 3.3. By applying this block-encoding in QLSA, we obtain the overall quantum algorithm for solving the linear ODEs.

The complexity analysis of the proposed quantum algorithm is conducted in detail. The core of the analysis focuses on deriving asymptotic upper bounds for the condition number of $\mathbf{L}_{m,k,p}(\mathbf{A}h)$. Unlike existing work, when the matrix $\mathbf{A}$ is negative semi-definite, our upper bound for the condition number is independent of $\|\mathbf{A}h\|_2$, which relaxes the constraint $\|\mathbf{A}h\|_2 < 1$ in the final complexity analysis. Combining the condition number analysis with the complexity analysis of QLSA and the success probability analysis, we obtain the overall complexity analysis, which includes both the oracle query complexity and the gate complexity. Compared to other algorithms of a similar nature [7, 13, 32], the proposed algorithm shows improved dependence on the approximation order k . Furthermore, thanks to the Padé approximation, the approximating order k can be chosen much smaller than that in the case of using Taylor approximation. Numerical experiments are conducted to validate the theoretical results and illustrate the complexity comparison of the proposed method with other algorithms of the same type.

Several straightforward future directions emerge from this work. First, we could combine the proposed Padé approximation-based algorithm with ODE linearization techniques to tackle nonlinear ODEs. Additionally, by carefully selecting the time step h , the proposed quantum algorithm could be extended to handle non-autonomous ODEs; however,

the corresponding complexity analysis would need to be revisited. Furthermore, Padé approximation could be applied to other quantum ODE algorithms to enhance their performance, potentially achieving optimal dependency on all parameters and improving the prefactors.

A Construction and condition number analysis of the linear system

A.1 Proof of Lemma 3.2

Proof. From eq. (17), we can observe that

$$\begin{cases} \mathbf{z}_1^{(s)} + \beta_1 \mathbf{A} h \mathbf{z}_0^{(s)} = -d_1 h \mathbf{b} = -\frac{1}{2} h \mathbf{b}, \\ \mathbf{z}_{j+1}^{(s)} + \beta_{j+1} \mathbf{A} h \mathbf{z}_j^{(s)} = \mathbf{0}, \quad j = 1, \dots, k-1. \end{cases}$$

Again by eq. (17), we obtain

$$\begin{cases} \tilde{\mathbf{z}}_1^{(s)} - \alpha_1 \mathbf{A} h \mathbf{z}_0^{(s)} = n_1 h \mathbf{b} = \frac{1}{2} h \mathbf{b}, \\ \tilde{\mathbf{z}}_{j+1}^{(s)} - \alpha_{j+1} \mathbf{A} h \tilde{\mathbf{z}}_j^{(s)} = \mathbf{0}, \quad j = 1, \dots, k-1. \end{cases}$$

Since $\alpha_j = \beta_j$ for all $j = 1, \dots, k$, we conclude that

$$\mathbf{z}_j^{(s)} = (-1)^j \tilde{\mathbf{z}}_j^{(s)}, \forall j = 1, \dots, k.$$

□

A.2 Proof of Lemma 3.3

Proof. For matrix $\mathbf{W}_k(\mathbf{A})$, we have the following decomposition:

$$\begin{bmatrix} \mathbf{I}_n & \mathbf{I}_n & \mathbf{I}_n & \cdots & \mathbf{I}_n \\ \mathbf{I}_n & \beta_k \mathbf{A} & & & \\ & \mathbf{I}_n & \beta_{k-1} \mathbf{A} & & \\ & & \ddots & \ddots & \\ & & & \mathbf{I}_n & \beta_1 \mathbf{A} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & \mathbf{I}_n & \mathbf{I}_n - \beta_k \mathbf{A} & \cdots & \mathbf{I}_n + \sum_{j=2}^k \beta_2 \cdots \beta_j (-\mathbf{A})^{j-1} \\ & \mathbf{I}_n & & & \\ & & \mathbf{I}_n & & \\ & & & \ddots & \\ & & & & \mathbf{I}_n \end{bmatrix} \times \begin{bmatrix} 0 & 0 & \cdots & 0 & \mathbf{I}_n + \sum_{j=1}^k \beta_1 \cdots \beta_j (-\mathbf{A})^j \\ \mathbf{I}_n & \beta_k \mathbf{A} & & & \\ & \mathbf{I}_n & \beta_{k-1} \mathbf{A} & & \\ & & \ddots & \ddots & \\ & & & \mathbf{I}_n & \beta_1 \mathbf{A} \end{bmatrix}, \quad (56)$$

where

$$\mathbf{I}_n + \sum_{j=1}^k \beta_1 \cdots \beta_j (-\mathbf{A})^j = \sum_{j=0}^k d_j (-\mathbf{A})^j = D_{kk}(\mathbf{A}).$$

Therefore, matrix $\mathbf{W}_k(\mathbf{A})$ is non-singular if and only if $D_{kk}(\mathbf{A})$ is non-singular. □

A.3 Proofs of Lemma 3.4 and Lemma 3.5

Lemma A.1. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be an invertible matrix and partitioned as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix},$$

where $\mathbf{A}_{11} \in \mathbb{R}^{n_1 \times n_1}$, $\mathbf{A}_{22} \in \mathbb{R}^{n_2 \times n_2}$, and $\mathbf{A}_{11}$ is assumed to be invertible. In this case, define $\mathbf{S} = \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$. Then, $\mathbf{S}$ is also invertible, and the inverse of $\mathbf{A}$ is given by

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{S}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{S}^{-1} \\ -\mathbf{S}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{S}^{-1} \end{bmatrix}.$$

Lemma A.2. Let $\mathbf{M}$ be a matrix of the form

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \cdots & \mathbf{M}_{1n} \\ \vdots & & \vdots \\ \mathbf{M}_{m1} & \cdots & \mathbf{M}_{mn} \end{bmatrix},$$

where each $\mathbf{M}_{ij}$ is a matrix block. Define the matrix $\widetilde{\mathbf{M}}$ such that $\widetilde{\mathbf{M}}_{ij} \geq \|\mathbf{M}_{ij}\|_2$ for each i, j . Then, we have the inequality

$$\|\mathbf{M}\|_2 \leq \|\widetilde{\mathbf{M}}\|_2.$$

Proof. Let $\mathbf{x}^T = [\mathbf{x}_1^T, \dots, \mathbf{x}_n^T]$ be an arbitrary unit vector. Then, we have

$$\begin{aligned} \|\mathbf{M}\mathbf{x}\|_2 &= \left\| \begin{bmatrix} \sum_{j=1}^n \mathbf{M}_{1j}\mathbf{x}_j \\ \vdots \\ \sum_{j=1}^n \mathbf{M}_{mj}\mathbf{x}_j \end{bmatrix} \right\|_2 = \sqrt{\sum_{i=1}^m \left\| \sum_{j=1}^n \mathbf{M}_{ij}\mathbf{x}_j \right\|_2^2} \\ &\leq \sqrt{\sum_{i=1}^m \left(\sum_{j=1}^n \|\mathbf{M}_{ij}\|_2 \|\mathbf{x}_j\|_2 \right)^2} \\ &\leq \left\| \widetilde{\mathbf{M}} \begin{bmatrix} \|\mathbf{x}_1\|_2 \\ \vdots \\ \|\mathbf{x}_n\|_2 \end{bmatrix} \right\|_2 \leq \|\widetilde{\mathbf{M}}\|_2. \end{aligned}$$

By the definition of the matrix 2-norm, we have $\|\mathbf{M}\|_2 \leq \|\widetilde{\mathbf{M}}\|_2$. □

Lemma A.3. Let $D_{kk}(\cdot)$ be the denominator of the (k, k) Padé approximation given by (7). Then, we have

$$D_{kk}(-1) \leq \sqrt{e} \quad \text{for all } k \in \mathbb{N}_+.$$

Proof. Using the fact that

$$\frac{(2k-j)!k!}{(2k)!(k-j)!} \leq \left(\frac{1}{2}\right)^j,$$

we obtain

$$D_{kk}(-1) = \sum_{j=0}^k \frac{(2k-j)!k!}{(2k)!(k-j)!} \frac{1}{j!} \leq \sum_{j=0}^k \left(\frac{1}{2}\right)^j \frac{1}{j!} \leq \sqrt{e}.$$

□

Lemma A.4. Let $\lambda > 0$. Then, we have the following inequality

$$\left\| \mathbf{W}_k(-\lambda)^{-1} \right\|_2 \leq \sqrt{4(k+1) \log(k+1) + 1}.$$

Proof. There is a permutation matrix P such that

$$\widetilde{\mathbf{W}} := P\mathbf{W}_k(-\lambda) = \left[\begin{array}{ccc|c} 1 & -\beta_k \lambda & & \\ & \ddots & \ddots & \\ & & 1 & -\beta_2 \lambda \\ \hline & & & 1 \\ \hline \frac{1}{\sqrt{k+1}} & \cdots & \frac{1}{\sqrt{k+1}} & \frac{1}{\sqrt{k+1}} \end{array} \right] =: \left[\begin{array}{c|c} B & -\beta_1 \lambda e_k \\ \hline \frac{1}{\sqrt{k+1}} \mathbf{1}^T & \frac{1}{\sqrt{k+1}} \end{array} \right].$$

It suffices to prove that

$$\left\| \widetilde{\mathbf{W}}^{-1} \right\|_2 \leq \sqrt{4(k+1) \log(k+1) + 1}.$$

By Lemma A.1, the matrix $\widetilde{\mathbf{W}}^{-1}$ is given by

$$\widetilde{\mathbf{W}}^{-1} = \begin{bmatrix} B^{-1} - \frac{\beta_1 \lambda B^{-1} e_k \mathbf{1}^T B^{-1}}{1 + \beta_1 \lambda \mathbf{1}^T B^{-1} e_k} & \frac{\sqrt{k+1} \beta_1 \lambda B^{-1} e_k}{1 + \beta_1 \lambda \mathbf{1}^T B^{-1} e_k} \\ \frac{-\mathbf{1}^T B^{-1}}{1 + \beta_1 \lambda \mathbf{1}^T B^{-1} e_k} & \frac{\sqrt{k+1}}{1 + \beta_1 \lambda \mathbf{1}^T B^{-1} e_k} \end{bmatrix}, \quad \text{with } B^{-1} = \begin{bmatrix} 1 & \beta_k \lambda & \cdots & (\beta_2 \cdots \beta_k) \lambda^{k-1} \\ & 1 & \ddots & \vdots \\ & & \ddots & \beta_2 \lambda \\ & & & 1 \end{bmatrix}. \quad (57)$$

It is easy to check that

$$\begin{aligned} \mathbf{1}^T B^{-1} &= [1, \frac{\sum_{s=k-1}^k \beta_{k-1} \cdots \beta_s \lambda^{s-k+2}}{\beta_{k-1} \lambda}, \dots, \frac{\sum_{s=1}^k \beta_1 \cdots \beta_s \lambda^s}{\beta_1 \lambda}] \\ B^{-1} e_k &= [(\beta_2 \cdots \beta_k) \lambda^{k-1}, \dots, \beta_2 \lambda, 1]^T, \\ 1 + \beta_1 \lambda \mathbf{1}^T B^{-1} e_k &= 1 + \sum_{s=1}^k \beta_1 \cdots \beta_s \lambda^s = D_{kk}(-\lambda). \end{aligned}$$

Therefore,

$$\widetilde{\mathbf{W}}^{-1} = \frac{1}{D_{kk}(-\lambda)} \begin{bmatrix} D_{kk}(-\lambda) B^{-1} - \beta_1 \lambda B^{-1} e_k \mathbf{1}^T B^{-1} & \sqrt{k+1} \beta_1 \lambda B^{-1} e_k \\ -\mathbf{1}^T B^{-1} & \sqrt{k+1} \end{bmatrix}.$$

Next, we compute the (i, j) -elements of $\widetilde{\mathbf{W}}^{-1}$ for $1 \leq i, j \leq k$.

$$\widetilde{\mathbf{W}}_{ij}^{-1} = e_i^T B^{-1} e_j - \beta_1 \lambda \frac{e_i^T B^{-1} e_k \mathbf{1}^T B^{-1} e_j}{D_{kk}(-\lambda)}.$$

- If $i > j$, the first term $e_i^T B^{-1} e_j = 0$, so we have

$$\begin{aligned} \widetilde{\mathbf{W}}_{ij}^{-1} &= -\beta_1 \lambda \frac{e_i^T B^{-1} e_k \mathbf{1}^T B^{-1} e_j}{D_{kk}(-\lambda)} \\ &= -\frac{1}{D_{kk}(-\lambda)} (\beta_1 \cdots \beta_{k-i+1}) \lambda^{k-i+1} \cdot \frac{\sum_{s=k-j+1}^k \beta_{k-j+1} \cdots \beta_s \lambda^{s+j-k}}{\beta_{k-j+1} \lambda} \\ &= -\frac{\beta_1 \cdots \beta_{k-i+1}}{\beta_1 \cdots \beta_{k-j+1}} \frac{\sum_{s=k-j+1}^k \beta_1 \cdots \beta_s \lambda^{s+j-i}}{D_{kk}(-\lambda)}. \end{aligned}$$

Here, we use the fact that

$$\frac{\beta_1 \cdots \beta_{k-i+1}}{\beta_1 \cdots \beta_{k-j+1}} \leq \frac{\beta_1 \cdots \beta_{k-i+1} \beta_{k-i+2} \cdots \beta_{s+j-i}}{\beta_1 \cdots \beta_{k-j+1} \beta_{k-j+2} \cdots \beta_s} = \frac{\beta_1 \cdots \beta_{s+j-i}}{\beta_1 \cdots \beta_s},$$

which follows from the monotonicity of the β_i 's. Therefore, we obtain the inequality

$$|\widetilde{\mathbf{W}}_{ij}^{-1}| \leq \frac{\sum_{s=k-j+1}^k \beta_1 \cdots \beta_{s+j-i} \lambda^{s+j-i}}{D_{kk}(-\lambda)} \leq 1.$$

- If $i \leq j$, the first term $e_i^T B^{-1} e_j = \beta_{k-j+2} \cdots \beta_{k-i+1} \lambda^{j-i}$, and thus we have

$$\begin{aligned} \widetilde{\mathbf{W}}_{ij}^{-1} &= \beta_{k-j+2} \cdots \beta_{k-i+1} \lambda^{j-i} - \frac{\beta_1 \cdots \beta_{k-i+1}}{\beta_1 \cdots \beta_{k-j+1}} \frac{\sum_{s=k-j+1}^k \beta_1 \cdots \beta_s \lambda^{s+j-i}}{D_{kk}(-\lambda)} \\ &= \frac{\beta_1 \cdots \beta_{k-i+1}}{\beta_1 \cdots \beta_{k-j+1}} \cdot \left(\lambda^{j-i} - \frac{\sum_{s=k-j+1}^k \beta_1 \cdots \beta_s \lambda^{s+j-i}}{D_{kk}(-\lambda)} \right) \\ &= \frac{\beta_1 \cdots \beta_{k-i+1}}{\beta_1 \cdots \beta_{k-j+1}} \cdot \frac{\sum_{s=0}^k \beta_1 \cdots \beta_s \lambda^{s+j-i} - \sum_{s=k-j+1}^k \beta_1 \cdots \beta_s \lambda^{s+j-i}}{D_{kk}(-\lambda)} \\ &= \frac{\beta_1 \cdots \beta_{k-i+1}}{\beta_1 \cdots \beta_{k-j+1}} \cdot \frac{\sum_{s=0}^{k-j} \beta_1 \cdots \beta_s \lambda^{s+j-i}}{D_{kk}(-\lambda)}. \end{aligned}$$

Here,

$$\frac{\beta_1 \cdots \beta_{k-i+1}}{\beta_1 \cdots \beta_{k-j+1}} = \frac{\beta_1 \cdots \beta_{s+j-i} \beta_{s+j-i+1} \cdots \beta_{k-i+1}}{\beta_1 \cdots \beta_s \beta_{s+1} \cdots \beta_{k-j+1}} \leq \frac{\beta_1 \cdots \beta_{s+j-i}}{\beta_1 \cdots \beta_s},$$

which follows from the monotonicity of the β_i 's. Therefore, we get

$$|\widetilde{\mathbf{W}}_{ij}^{-1}| \leq \frac{\sum_{s=0}^{k-j} \beta_1 \cdots \beta_{s+j-i} \lambda^{s+j-i}}{D_{kk}(-\lambda)} \leq 1.$$

Moreover, it is easy to verify that $|\widetilde{\mathbf{W}}_{ij}^{-1}| \leq 1$ for $i = k+1$ and $j \leq k$. Next, we consider an upper bound for the summation

$$\sum_{j=1}^k \sum_{i=1}^{k+1} |\widetilde{\mathbf{W}}_{ij}^{-1}|. \quad (58)$$

From the previous discussion, we know that $\widetilde{\mathbf{W}}_{ij}^{-1} < 0$ if and only if $i > j$, and we also have $\mathbf{1}^T \widetilde{\mathbf{W}}^{-1} e_j = e_{k+1}^T \widetilde{\mathbf{W}} \widetilde{\mathbf{W}}^{-1} e_j = 0$ for all $j = 1, \dots, k$. This implies that

$$\begin{aligned} \sum_{i=1}^{k+1} |\widetilde{\mathbf{W}}_{ij}^{-1}| &= 2 \sum_{i=1}^j \widetilde{\mathbf{W}}_{ij}^{-1} = \frac{2}{D_{kk}(-\lambda)} \sum_{i=1}^j \sum_{s=0}^{k-j} \frac{\beta_1 \cdots \beta_{k-i+1} \cdot \beta_1 \cdots \beta_s}{\beta_1 \cdots \beta_{k-j+1}} \lambda^{s+j-i} \\ &= \frac{2}{D_{kk}(-\lambda)} \sum_{l=0}^{j-1} \sum_{s=0}^{k-j} \frac{\beta_1 \cdots \beta_{k-j+l+1} \cdot \beta_1 \cdots \beta_s}{\beta_1 \cdots \beta_{k-j+1}} \lambda^{s+l} \\ &= \frac{2}{D_{kk}(-\lambda)} \sum_{l=0}^{j-1} \sum_{s=0}^{k-j} \frac{\beta_{s+l+1} \cdots \beta_{k-j+l+1}}{\beta_{s+1} \cdots \beta_{k-j+1}} \beta_1 \cdots \beta_{s+l} \lambda^{s+l} \\ &= \frac{2}{D_{kk}(-\lambda)} \sum_{r=0}^{k-1} \sum_{l=\max\{0, r+j-k\}}^{\min\{j-1, r\}} \frac{\beta_{r+1} \cdots \beta_{k-j+l+1}}{\beta_{r-l+1} \cdots \beta_{k-j+1}} \beta_1 \cdots \beta_r \lambda^r, \end{aligned} \quad (59)$$

where we make the variable substitution $l = j - i$ in the third equality, and $r = s + l$ in the last equality. Thus, we have

$$\sum_{j=1}^k \sum_{i=1}^{k+1} |\widetilde{\mathbf{W}}_{ij}^{-1}| = \frac{2}{D_{kk}(-\lambda)} \sum_{r=0}^{k-1} \left(\sum_{j=1}^k \sum_{l=\max\{0, r+j-k\}}^{\min\{j-1, r\}} \frac{\beta_{r+1} \cdots \beta_{k-j+l+1}}{\beta_{r-l+1} \cdots \beta_{k-j+1}} \right) \beta_1 \cdots \beta_r \lambda^r \quad (60)$$

Next, we give an upper bound for the inner summation, which can be written as

$$\sum_{j=1}^{k-r} \sum_{l=0}^{\min\{j-1, r\}} \frac{\beta_{r+1} \cdots \beta_{k-j+l+1}}{\beta_{r-l+1} \cdots \beta_{k-j+1}} + \sum_{j=k-r+1}^k \sum_{l=r+j-k}^{\min\{j-1, r\}} \frac{\beta_{r+1} \cdots \beta_{k-j+l+1}}{\beta_{r-l+1} \cdots \beta_{k-j+1}}. \quad (61)$$

For the first term in (61), we have $r \leq k - j$, and thus

$$\begin{aligned} \frac{\beta_{r+1} \cdots \beta_{k-j+l+1}}{\beta_{r-l+1} \cdots \beta_{k-j+1}} &= \frac{\beta_{r+1} \cdots \beta_{k-j+1} \cdots \beta_{k-j+l+1}}{\beta_{r-l+1} \cdots \beta_{r+1} \cdots \beta_{k-j+1}} \leq \frac{\beta_{k-j+2} \cdots \beta_{k-j+l+1}}{\beta_{r-l+1} \cdots \beta_r} \\ &\leq \left(\frac{\beta_{k-j+2}}{\beta_r} \right)^l \leq \left(\frac{r}{k-j+2} \right)^l, \quad \forall l \geq 1, \end{aligned}$$

where we use the assumption that $\beta_j/\beta_i \leq i/j \leq 1$ for all $1 \leq i \leq j \leq k$ in the last two inequalities. Note that the inequality also holds for $l = 0$. Therefore, we obtain

$$\sum_{l=0}^{\min\{j-1, r\}} \frac{\beta_{r+1} \cdots \beta_{k-j+l+1}}{\beta_{r-l+1} \cdots \beta_{k-j+1}} \leq \sum_{l=0}^{\min\{j-1, r\}} \left(\frac{r}{k-j+2} \right)^l \leq \sum_{l=0}^{\infty} \left(\frac{r}{k-j+2} \right)^l = \frac{k-j+2}{k-j-r+2}.$$

For the second term in (61), we have $r > k - j$. Using the assumption that $\beta_j/\beta_i \leq i/j \leq 1$ for all $1 \leq i \leq j \leq k$ again, we get

$$\frac{\beta_{r+1} \cdots \beta_{k-j+l+1}}{\beta_{r-l+1} \cdots \beta_{k-j+1}} \leq \left(\frac{\beta_{r+1}}{\beta_{k-j+1}} \right)^{k-j+l-r+1} \leq \left(\frac{k-j+1}{r+1} \right)^{k-j+l-r+1}.$$

Thus, we have

$$\begin{aligned} \sum_{l=r+j-k}^{\min\{j-1, r\}} \frac{\beta_{r+1} \cdots \beta_{k-j+l+1}}{\beta_{r-l+1} \cdots \beta_{k-j+1}} &\leq \sum_{l=r+j-k}^{\min\{j-1, r\}} \left(\frac{k-j+1}{r+1} \right)^{k-j+l-r+1} = \sum_{l=0}^{\min\{j-1, r\}+k-j-r} \left(\frac{k-j+1}{r+1} \right)^{l+1} \\ &\leq \sum_{l=0}^{\infty} \left(\frac{k-j+1}{r+1} \right)^{l+1} = \frac{k-j+1}{r+j-k}. \end{aligned}$$

Then, the summation in (61) is upper bounded by

$$\begin{aligned} \sum_{j=1}^{k-r} \frac{k-j+2}{k-j-r+2} + \sum_{j=k-r+1}^k \frac{k-j+1}{r+j-k} &\leq (k+1) \sum_{j=1}^{k-r} \frac{1}{k-j-r+2} + r \sum_{j=k-r+1}^k \frac{1}{r+j-k} \\ &= (k+1) \left(\frac{1}{2} + \cdots + \frac{1}{k-r+1} \right) + r \left(1 + \cdots + \frac{1}{r} \right) \\ &\leq (k+1) \log(k-r+1) + r(1 + \log r) \\ &\leq (k+1) (\log(r(k-r+1)) + 1) \\ &\leq (k+1) \left(\log \left(\frac{k+1}{2} \right)^2 + 1 \right) \\ &\leq 2(k+1) \log(k+1). \end{aligned} \quad (62)$$

Substituting this bound into (60), we obtain

$$\sum_{j=1}^k \sum_{i=1}^{k+1} |\widetilde{\mathbf{W}}_{ij}^{-1}| \leq 4(k+1) \log(k+1) \frac{\sum_{r=0}^{k-1} d_r \lambda^r}{D_{kk}(-\lambda)} \leq 4(k+1) \log(k+1).$$

Finally, using the fact that

$$\sum_{i=1}^{k+1} |\widetilde{\mathbf{W}}_{i,k+1}^{-1}| = \sum_{i=1}^{k+1} \widetilde{\mathbf{W}}_{i,k+1}^{-1} = \mathbf{1}^T \widetilde{\mathbf{W}}^{-1} e_{k+1} = \sqrt{k+1} e_{k+1}^T \widetilde{\mathbf{W}} \widetilde{\mathbf{W}}^{-1} e_{k+1} = \sqrt{k+1},$$

we get

$$\begin{aligned} \|\widetilde{\mathbf{W}}^{-1}\|_2 &\leq \|\widetilde{\mathbf{W}}^{-1}\|_F = \sqrt{\sum_{j=1}^{k+1} \sum_{i=1}^{k+1} |\widetilde{\mathbf{W}}_{ij}^{-1}|^2} = \sqrt{\sum_{j=1}^k \sum_{i=1}^{k+1} |\widetilde{\mathbf{W}}_{ij}^{-1}|^2 + \sum_{i=1}^{k+1} |\widetilde{\mathbf{W}}_{i,k+1}^{-1}|^2} \\ &\leq \sqrt{\sum_{j=1}^k \sum_{i=1}^{k+1} |\widetilde{\mathbf{W}}_{ij}^{-1}| + \left(\sum_{i=1}^{k+1} |\widetilde{\mathbf{W}}_{i,k+1}^{-1}| \right)^2} \\ &\leq \sqrt{4(k+1) \log(k+1) + k+1} = \sqrt{(k+1)(4 \log(k+1) + 1)}. \end{aligned}$$

Here, we use the fact that $|\widetilde{\mathbf{W}}_{ij}^{-1}| \leq 1$ for all $1 \leq i \leq k+1$ and $1 \leq j \leq k$ in the second inequality. $\square$

Proof of Lemma 3.4. Since $\mathbf{A}$ is Hermitian and negative semi-definite, by unitary similarity transformation, $\mathbf{W}_k(\mathbf{A})$ is unitarily similar to a block diagonal matrix. Each diagonal block has the same structure as $\mathbf{W}_k(\mathbf{A})$ but with scalar blocks. Therefore, without loss of generality, we assume $\mathbf{A} = -\lambda$, where $\lambda \geq 0$. Next, applying Lemma A.4, we obtain the bound

$$\|\mathbf{W}_k(\mathbf{A})^{-1}\|_2 \leq \sqrt{(k+1)(4 \log(k+1) + 1)}.$$

$\square$

Lemma A.5. For any matrix $\mathbf{A} \in \mathbb{C}^{n \times n}$, if the matrix $\mathbf{W}_k(\mathbf{A})$ defined by eq. (20) is invertible, then the inverse is given by

$$\mathbf{W}_k(\mathbf{A})^{-1} = \left[\mathbf{I}_{k+1} \otimes (D_{kk}(\mathbf{A}))^{-1} \right] \times \begin{bmatrix} d_k \sqrt{k+1} (-\mathbf{A})^k & \frac{d_k}{d_k} \sum_{j=0}^{k-1} d_j (-\mathbf{A})^j & \cdots & \frac{d_k}{d_2} \sum_{j=0}^1 d_j (-\mathbf{A})^{j+k-2} & \frac{d_k}{d_1} d_0 (-\mathbf{A})^{k-1} \\ d_{k-1} \sqrt{k+1} (-\mathbf{A})^{k-1} & -\frac{d_{k-1}}{d_k} \sum_{j=k}^k d_j (-\mathbf{A})^{j-1} & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \frac{d_2}{d_2} \sum_{j=0}^1 d_j (-\mathbf{A})^j & \frac{d_2}{d_1} d_0 (-\mathbf{A}) \\ d_1 \sqrt{k+1} (-\mathbf{A}) & -\frac{d_1}{d_k} \sum_{j=k}^k d_j (-\mathbf{A})^{j-k+1} & \cdots & -\frac{d_1}{d_2} \sum_{j=2}^k d_j (-\mathbf{A})^{j-1} & \frac{d_1}{d_1} d_0 \mathbf{I}_n \\ d_0 \sqrt{k+1} \mathbf{I}_n & -\frac{d_0}{d_k} \sum_{j=k}^k d_j (-\mathbf{A})^{j-k} & \cdots & -\frac{d_0}{d_2} \sum_{j=2}^k d_j (-\mathbf{A})^{j-2} & -\frac{d_0}{d_1} \sum_{j=1}^k d_j (-\mathbf{A})^{j-1} \end{bmatrix}.$$

Proof. Applying the discussion in Lemma A.4, we can directly write each block of $\mathbf{W}_k(\mathbf{A})^{-1}$. $\square$

Proof of Lemma 3.5. First, using Lemma C.1, we have

$$\|D_{kk}(\mathbf{A})^{-1}\|_2 \leq \frac{2}{3-e}.$$

Next, we construct a $(k+1) \times (k+1)$ matrix $\widehat{\mathbf{W}}$ satisfying

$$\widehat{\mathbf{W}}_{ls} = \begin{cases} \frac{d_{l-1}}{d_t} \sum_{j=0}^{t-1} d_j, & s \geq 2, t+1 \leq l \leq k+1, \\ \frac{d_{l-1}}{d_t} \sum_{j=t}^k d_j, & s \geq 2, 1 \leq l \leq t, \\ \sqrt{k+1} d_{k+1-l}, & s = 1, \end{cases}$$

where we define $t = k+2-s$ for simplicity. Then, by Lemma A.5 and Lemma A.2, we have

$$\|\mathbf{W}_k(\mathbf{A})^{-1}\|_2 \leq \|D_{kk}(\mathbf{A})^{-1}\|_2 \cdot \|\widehat{\mathbf{W}}\|_2 \leq \frac{2}{3-e} \|\widehat{\mathbf{W}}\|_2.$$

Now, we focus on bounding the spectral norm of matrix $\widehat{\mathbf{W}}$, which is indeed a special case of the discussion in Lemma A.4. Setting $\lambda = 1$ in Lemma A.4, and get

$$\|\widehat{\mathbf{W}}\|_2 \leq D_{kk}(-1) \cdot \sqrt{(k+1)(4\log(k+1)+1)},$$

where $D_{kk}(-1) \leq \sqrt{e}$ by Lemma A.3. Thus, we obtain

$$\|\mathbf{W}_k(\mathbf{A})^{-1}\|_2 \leq \frac{2\sqrt{e}}{3-e} \sqrt{(k+1)(4\log(k+1)+1)}.$$

□

A.4 Proof of Theorem 3.6

Lemma A.6. Suppose that α and β are defined as in eq. (15). For indexes $i > j$, it holds that $\alpha_i < \alpha_j$ and $\beta_i < \beta_j$.

Proof. We examine the quantity α_j/α_{j+1}

$$\frac{\alpha_j}{\alpha_{j+1}} = \frac{(j+1)(p+q-j)(p-j+1)}{j(p-j)(p+q-j+1)} = \left(1 + \frac{1}{j}\right) \left(1 + \frac{q}{(p-j)(p+q-j+1)}\right) > 1.$$

By induction, we conclude that $\alpha_i < \alpha_j$ for $i > j$. Similarly, we can show that $\beta_i < \beta_j$ for $i > j$. □

Lemma A.7. Suppose $\mathbf{A} \in \mathbb{C}^{n \times n}$ is Hermitian and negative semi-definite, and $\mathbf{W}_k(\mathbf{A})$ is defined by eq. (20). Then, we have

$$\|\widetilde{\mathbf{E}}^T \mathbf{W}_k(\mathbf{A})^{-1}\|_2 \leq \sqrt{5k+1}, \quad (63)$$

where $\widetilde{\mathbf{E}} = \widetilde{\mathbf{1}} \otimes \mathbf{I}_n$, and $\widetilde{\mathbf{1}}$ is defined in eq. (21).

Proof. For simplicity, we write $\mathbf{W} = \mathbf{W}_k(\mathbf{A})$ in this proof. Similar to the proof of Lemma 3.4, we only need to consider the case $\mathbf{A} = -\lambda \leq 0$. At first, we consider solving the linear system

$$\mathbf{W}^T \mathbf{x} = \widetilde{\mathbf{1}}, \quad (64)$$

where $\mathbf{x} = (x_{k+1}, \dots, x_1)^T$. We aim to show that the solution to this linear system eq. (64) is given by

$$\begin{cases} x_{k+1} = -\sqrt{k+1} \frac{D(\lambda)}{D(-\lambda)}, \\ x_j = \frac{(-1)^{j+1}}{D(-\lambda)} \sum_{i=0}^{k-j} \frac{d_{j+i}}{d_j} (-\lambda)^i P_{j+i+1}(\lambda), 1 \leq j \leq k. \end{cases} \quad (65)$$

Here, the notations are defined as follows

$$\begin{cases} d_0 = 1, \\ d_j = d_0 \beta_1 \cdots \beta_j, \quad 1 \leq j \leq k, \end{cases} \quad D(\lambda) = \sum_{j=0}^k d_j (-\lambda)^j, \quad P_j(\lambda) = D(-\lambda) + (-1)^j D(\lambda).$$

Using the explicit expression for $\mathbf{W}^{-1}$ from Lemma A.5, we can verify that $x_{k+1} = -\sqrt{k+1} \frac{D(\lambda)}{D(-\lambda)}$. Moreover, by the first row of the linear system (64), we have

$$x_k = (-1)^{k+1} - \frac{1}{\sqrt{k+1}} x_{k+1} = \frac{(-1)^{k+1}}{D(-\lambda)} P_{k+1}(\lambda)$$

which satisfies the expression (65).

Next, we verify for general j by induction. Using the recurrence relation

$$\frac{1}{\sqrt{k+1}} x_{k+1} - \beta_{j+1} \lambda x_{j+1} + x_j = (-1)^{j+1}, \quad \forall j = 0, \dots, k-1, \quad (66)$$

we obtain

$$\begin{aligned} x_j &= (-1)^{j+1} - \frac{1}{\sqrt{k+1}} x_{k+1} + \beta_{j+1} \lambda x_{j+1} \\ &= \frac{(-1)^{j+1}}{D(-\lambda)} \left(P_{j+1}(\lambda) + (-1)^{j+1} D(-\lambda) \beta_{j+1} \lambda x_{j+1} \right) \\ &= \frac{(-1)^{j+1}}{D(-\lambda)} \left(P_{j+1}(\lambda) + (-\beta_{j+1} \lambda) \sum_{i=0}^{k-j-1} \frac{d_{j+i+1}}{d_{j+1}} (-\lambda)^i P_{j+i+2}(\lambda) \right) \\ &= \frac{(-1)^{j+1}}{D(-\lambda)} \sum_{i=0}^{k-j} \frac{d_{j+i}}{d_j} (-\lambda)^i P_{j+i+1}(\lambda). \end{aligned}$$

This shows that x_j also satisfies the expression in (65).

The next step is to show that $(-1)^{j+1} x_j \geq 0$. For the summation in the expression of x_j , we have

$$\begin{aligned} &\sum_{i=0}^{k-j} \frac{d_{j+i}}{d_j} (-\lambda)^i \left(\sum_{l=0}^k d_l \lambda^l \left(1 + (-1)^{j+i+1+l} \right) \right) \\ &= \frac{1}{d_j} \sum_{i=0}^{k-j} \sum_{l=0}^k \left((-1)^i + (-1)^{j+l+1} \right) d_{j+i} d_l \lambda^{l+i} \\ &= \frac{1}{d_j} \sum_{r=0}^{2k-j} \sum_{i=\max\{0, r-k\}}^{\min\{k-j, r\}} \left((-1)^i + (-1)^{j+r-i+1} \right) d_{j+i} d_{r-i} \lambda^r \\ &= \frac{1}{d_j} \sum_{r=0}^{2k-j} \left(1 + (-1)^{j+r+1} \right) \left(\sum_{i=\max\{0, r-k\}}^{\min\{k-j, r\}} (-1)^i d_{j+i} d_{r-i} \right) \lambda^r, \end{aligned} \quad (67)$$

where we performed the variable substitution $r = l + i$ in the second equality. In the last line of (67), all terms are obviously non-negative except for the inner summation term:

$$\sum_{i=\bar{i}}^{\bar{i}} (-1)^i d_{j+i} d_{r-i}, \quad (68)$$

where $\underline{i} = \max\{0, r - k\}$ and $\bar{i} = \min\{k - j, r\}$.

We now show that this summation is also non-negative under the condition that $j + r + 1$ is an even number. This condition is reasonable because the coefficient $1 + (-1)^{j+r+1}$ vanishes when $j + r + 1$ is odd. First, we consider the summation

$$\sum_{i=0}^s (-1)^i d_{j+i} d_{r-i}. \quad (69)$$

- If $j > r$, we have

$$\begin{aligned} \sum_{i=0}^s (-1)^i d_{j+i} d_{r-i} &= \sum_{i=0}^{\lceil s/2 \rceil - 1} (d_{j+2i} d_{r-2i} - d_{j+2i+1} d_{r-2i-1}) + d_{j+s} d_{r-s} \delta_{\{s \text{ is even}\}} \\ &\geq \sum_{i=0}^{\lceil s/2 \rceil - 1} d_{j+2i} d_{r-2i} \left(1 - \frac{\beta_{j+2i+1}}{\beta_{r-2i}} \right) \geq 0, \end{aligned}$$

where we use the decreasing property of the sequence β_i .

- If $j < r$ but $j + s \geq r$, we can find $i^* \in [0, s]$ such that $j + i^* = r$. The integer i^* is odd, and we have

$$\sum_{i=0}^{i^*} (-1)^i d_{j+i} d_{r-i} = \sum_{l=0}^{i^*} (-1)^{i^*-l} d_{j+i^*-l} d_{r-i^*+l} = - \sum_{l=0}^{i^*} (-1)^l d_{r-l} d_{j+l} = 0,$$

where we reverse the summation order and make the variable substitution $l = i^* - i$ in the first equality. Taking the common factor $(-1)^{i^*} = -1$ outside the summation, we see that the sum equals zero. Therefore, we obtain

$$\sum_{i=0}^s (-1)^i d_{j+i} d_{r-i} = \sum_{i=i^*+1}^s (-1)^i d_{j+i} d_{r-i} = \sum_{i=0}^{s-i^*-1} (-1)^{i+i^*+1} d_{j+i^*+1+i} d_{r-i^*-1-i}.$$

Let $s' = s - i^* - 1$, $j' = j + i^* + 1$, and $r' = r - i^* - 1$. Then, we have $r' + j' + 1$ as an even number, with $j' > r'$, and thus the above summation is non-negative as in the first case.

In summary, the summation (69) is non-negative if $j + s \geq r$. Specifically, it equals zero when $j + s = r$. We now rewrite the summation (68) as

$$\sum_{i=\underline{i}}^{\bar{i}} (-1)^i d_{j+i} d_{r-i} = (-1)^{\underline{i}} \sum_{i=0}^{\bar{i}-\underline{i}} (-1)^i d_{j+\underline{i}+i} d_{r-\underline{i}-i}. \quad (70)$$

Defining $s' = \bar{i} - \underline{i}$, $j' = j + \underline{i}$, and $r' = r - \underline{i}$, we observe $r' + j' + 1$ is an even number. Moreover, we compute

$$j' + s' - r' = j + \min\{k - j, r\} + \max\{0, r - k\} - r = \min\{k, r + j\} - \min\{k, r\} \geq 0.$$

Thus, the sign of (68) is determined by

$$(-1)^{\underline{i}} = (-1)^{\max\{0, r-k\}}.$$

Notably, if $r \geq k$, we have $j' + s' = r'$, leading to the summation equaling zero. Consequently, we conclude that the summation (68) is always non-negative.

Hence, we have established that $(-1)^{j+1}x_j \geq 0$. Moreover, from (66), we obtain

$$0 \leq (-1)^{j+1}x_j = 1 - (-1)^{j+1} \frac{x_{k+1}}{\sqrt{k+1}} + (-1)^{j+1}\beta_{j+1}\lambda x_{j+1}.$$

Using the fact that $|x_{k+1}| < \sqrt{k+1}$, we conclude that

$$1 - (-1)^{j+1} \frac{x_{k+1}}{\sqrt{k+1}} \geq 0, \quad \text{while} \quad (-1)^{j+1}\beta_{j+1}\lambda x_{j+1} \leq 0.$$

Thus, we obtain

$$|x_j| = (-1)^{j+1}x_j \leq 1 - (-1)^{j+1} \frac{x_{k+1}}{\sqrt{k+1}} \leq 2.$$

This implies the final bound

$$\|\tilde{\mathbf{1}}^T \mathbf{W}^{-1}\|_2 \leq \sqrt{5k+1}.$$

□

Lemma A.8. Suppose $\mathbf{A} \in \mathbb{C}^{n \times n}$ satisfies $\|\mathbf{A}\|_2 \leq 1$, and let $\mathbf{W}_k(\mathbf{A})$ be defined as in eq. (20). Then, we have

$$\|\tilde{\mathbf{E}}^T \mathbf{W}_k(\mathbf{A})^{-1}\|_2 \leq \left(\sqrt{e} + \frac{2e}{3-e} \right) \sqrt{2k+1}.$$

Here, $\tilde{\mathbf{E}} = \tilde{\mathbf{1}} \otimes \mathbf{I}_n$, where $\tilde{\mathbf{1}}$ is define in eq. (21).

Proof. As in Lemma A.7, we consider solving the linear system

$$\begin{bmatrix} \frac{\mathbf{I}_n}{\sqrt{k+1}} & \mathbf{I}_n & & & \\ \frac{\mathbf{I}_n}{\sqrt{k+1}} & \beta_k \mathbf{A} & \mathbf{I}_n & & \\ \vdots & & \ddots & \ddots & \\ \frac{\mathbf{I}_n}{\sqrt{k+1}} & & & \beta_2 \mathbf{A} & \mathbf{I}_n \\ \frac{\mathbf{I}_n}{\sqrt{k+1}} & & & & \beta_1 \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{k+1} \\ \mathbf{X}_k \\ \vdots \\ \mathbf{X}_2 \\ \mathbf{X}_1 \end{bmatrix} = \begin{bmatrix} (-\mathbf{I}_n)^{k+1} \\ (-\mathbf{I}_n)^k \\ \vdots \\ (-\mathbf{I}_n)^2 \\ -\mathbf{I}_n \end{bmatrix}.$$

Using the notation in Lemma A.7, we obtain

$$\begin{cases} \mathbf{X}_{k+1} = -\sqrt{k+1} D_{kk}(\mathbf{A})^{-1} D_{kk}(-\mathbf{A}), \\ \mathbf{X}_j = (-1)^{j+1} \sum_{i=0}^{k-j} \frac{d_{j+i}}{d_j} \mathbf{A}^i (\mathbf{I}_n + (-1)^{j+i+1} D_{kk}(\mathbf{A})^{-1} D_{kk}(-\mathbf{A})), \quad 1 \leq j \leq k. \end{cases}$$

Given the assumption $\|\mathbf{A}\|_2 \leq 1$, along with Lemma A.3 and Lemma C.1, we obtain

$$\|D_{kk}(\mathbf{A})^{-1} D_{kk}(-\mathbf{A})\|_2 \leq \frac{2}{3-e} D_{kk}(-1) \leq \frac{2\sqrt{e}}{3-e}.$$

Thus, we have $\|\mathbf{X}_{k+1}\|_2 \leq \frac{2\sqrt{e}}{3-e} \sqrt{k+1}$, and

$$\begin{aligned} \|\mathbf{X}_j\|_2 &\leq \left(1 + \frac{2\sqrt{e}}{3-e} \right) \sum_{i=0}^{k-j} \frac{d_{j+i}}{d_j} \leq \left(1 + \frac{2\sqrt{e}}{3-e} \right) \sum_{i=0}^k d_i \\ &\leq \sqrt{e} + \frac{2e}{3-e}. \end{aligned}$$

Finally, applying Lemma A.2, we obtain

$$\|\tilde{\mathbf{E}}^T \mathbf{W}_k(\mathbf{A})^{-1}\|_2 \leq \left(\sqrt{e} + \frac{2e}{3-e} \right) \sqrt{2k+1}.$$

□

Proof of Theorem 3.6. For simplicity, we use $\mathbf{L}$ and $\mathbf{W}$ to denote the matrices $\mathbf{L}_{m,k,p}(\mathbf{A}h)$ and $\mathbf{W}_k(\mathbf{A}h)$, respectively. By decomposing $\mathbf{L}$ into the sum of its block diagonal and block sub-diagonal parts, we obtain

$$\|\mathbf{L}\|_2 \leq \max\{\|\mathbf{W}\|_2, 1\} + \|\tilde{\mathbf{T}}\|_2,$$

where $\|\tilde{\mathbf{T}}\|_2 = 1$. Furthermore, we decompose $\mathbf{W}$ as follows

$$\mathbf{W} = \begin{bmatrix} \frac{1}{\sqrt{k+1}}\mathbf{I}_n & \frac{1}{\sqrt{k+1}}\mathbf{I}_n & \cdots & \frac{1}{\sqrt{k+1}}\mathbf{I}_n \\ \mathbf{O} & \mathbf{O} & & \\ & \ddots & \ddots & \\ & & \mathbf{O} & \mathbf{O} \end{bmatrix} + \begin{bmatrix} \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{I}_n & \mathbf{O} & & \\ & \ddots & \ddots & \\ & & \mathbf{I}_n & \mathbf{O} \end{bmatrix} + \begin{bmatrix} \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \beta_k \mathbf{A}h & & \\ & \ddots & \ddots & \\ & & \mathbf{O} & \beta_1 \mathbf{A}h \end{bmatrix}.$$

From this decomposition, a reasonable bound on $\|\mathbf{W}\|_2$ is

$$\|\mathbf{W}\|_2 \leq \beta_1 h \|\mathbf{A}\|_2 + 2.$$

Thus, we obtain

$$\|\mathbf{L}\|_2 \leq \beta_1 h \|\mathbf{A}\|_2 + 3.$$

To bound the spectral norm of $\mathbf{L}^{-1}$, we decompose it as

$$\mathbf{L} = \begin{bmatrix} \mathbf{W} & & & \\ & \ddots & & \\ & & \mathbf{W} & \\ & & \frac{1}{\sqrt{k+1}}\mathbf{I}_n & \\ & & & \mathbf{I}_n & \\ & & & & \ddots \\ & & & & & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{I} & & & & & & \\ \mathbf{W}^{-1}\tilde{\mathbf{T}} & \mathbf{I} & & & & & \\ & \ddots & \ddots & & & & \\ & & \mathbf{W}^{-1}\tilde{\mathbf{T}} & \mathbf{I} & & & \\ & & \tilde{\mathbf{1}}^T \otimes \mathbf{I}_n & -\mathbf{I}_n & \mathbf{I}_n & & \\ & & & -\mathbf{I}_n & \mathbf{I}_n & \ddots & \\ & & & & & -\mathbf{I}_n & \mathbf{I}_n \end{bmatrix} \\ =: \mathbf{D}(\mathbf{I} - \mathbf{N}),$$

where $\mathbf{N}$ is a nilpotent matrix. Moreover, defining $\tilde{\mathbf{V}} = -\frac{1}{\sqrt{k+1}}\mathbf{W}^{-1}(\mathbf{e}_1 \otimes \mathbf{I}_n)$, and $\tilde{\mathbf{E}} = \tilde{\mathbf{1}} \otimes \mathbf{I}_n$, we obtain

$$-\mathbf{W}^{-1}\tilde{\mathbf{T}} = \left(-\frac{1}{\sqrt{k+1}}\mathbf{W}^{-1}(\mathbf{e}_1 \otimes \mathbf{I}_n)\right)(\tilde{\mathbf{1}}^T \otimes \mathbf{I}_n) = \tilde{\mathbf{V}}\tilde{\mathbf{E}}^T.$$

With these notations, we can express the inverse of $\mathbf{L}$ as

$$\mathbf{L}^{-1} = \sum_{j=0}^{m+p} \mathbf{N}^j \mathbf{D}^{-1},$$

where $\mathbf{N}^j$ for all $j \geq 1$ is a j -th block sub-diagonal matrix. The blocks in this sub-diagonal are given by

$$\begin{cases} (\tilde{\mathbf{V}}\tilde{\mathbf{E}}^T)^j, \dots, (\tilde{\mathbf{V}}\tilde{\mathbf{E}}^T)^j, -\tilde{\mathbf{E}}^T(\tilde{\mathbf{V}}\tilde{\mathbf{E}}^T)^{j-1}, \dots, -\tilde{\mathbf{E}}^T, \mathbf{I}_n, \dots, \mathbf{I}_n, & 1 \leq j \leq m \\ -\tilde{\mathbf{E}}^T(\tilde{\mathbf{V}}\tilde{\mathbf{E}}^T)^m, \dots, -\tilde{\mathbf{E}}^T, \mathbf{I}_n, \dots, \mathbf{I}_n, & j > m. \end{cases}$$

The spectral norm of $\mathbf{L}^{-1}$ can be bounded as follows:

$$\|\mathbf{L}^{-1}\|_2 \leq \sum_{j=0}^{m+p} \|\mathbf{N}^j \mathbf{D}^{-1}\|_2,$$

where

$$\|\mathbf{N}^j \mathbf{D}^{-1}\|_2 \leq \max \left\{ \left\| \left(\widetilde{\mathbf{V}} \widetilde{\mathbf{E}}^T \right)^l \mathbf{W}^{-1} \right\|_2, \left\| \widetilde{\mathbf{E}}^T \left(\widetilde{\mathbf{V}} \widetilde{\mathbf{E}}^T \right)^l \mathbf{W}^{-1} \right\|_2, \sqrt{k+1} \mid 0 \leq l \leq m \right\},$$

for all $0 \leq j \leq m+p$. Since $\widetilde{\mathbf{V}}$ is proportional to the first block column of $\mathbf{W}^{-1}$, we obtain

$$\widetilde{\mathbf{E}}^T \widetilde{\mathbf{V}} = [D_{kk}(\mathbf{A}h)]^{-1} \left(\sum_{j=0}^k d_j (\mathbf{A}h)^j \right) = [D_{kk}(\mathbf{A}h)]^{-1} N_{kk}(\mathbf{A}h) = R_{kk}(\mathbf{A}h).$$

Substituting this into $\mathbf{N}^j$, we derive

$$\left(\widetilde{\mathbf{V}} \widetilde{\mathbf{E}}^T \right)^j = \widetilde{\mathbf{V}} \left(\widetilde{\mathbf{E}}^T \widetilde{\mathbf{V}} \right)^{j-1} \widetilde{\mathbf{E}}^T = \left(\mathbf{I}_{k+1} \otimes [R_{kk}(\mathbf{A}h)]^{j-1} \right) \widetilde{\mathbf{V}} \widetilde{\mathbf{E}}^T,$$

where, in the second equality, we interchange the multiplication order of $R_{kk}(\mathbf{A})$ and $\widetilde{\mathbf{V}}$. For $l \geq 1$, we obtain the following bounds

$$\begin{aligned} \left\| \left(\widetilde{\mathbf{V}} \widetilde{\mathbf{E}}^T \right)^l \mathbf{W}^{-1} \right\|_2 &\leq \left\| [R_{kk}(\mathbf{A}h)]^{l-1} \right\|_2 \left\| \widetilde{\mathbf{V}} \widetilde{\mathbf{E}}^T \mathbf{W}^{-1} \right\|_2, \\ \left\| \widetilde{\mathbf{E}}^T \left(\widetilde{\mathbf{V}} \widetilde{\mathbf{E}}^T \right)^{l-1} \mathbf{W}^{-1} \right\|_2 &\leq \left\| [R_{kk}(\mathbf{A}h)]^{l-1} \right\|_2 \left\| \widetilde{\mathbf{E}}^T \mathbf{W}^{-1} \right\|_2. \end{aligned}$$

Since m and k satisfy the condition $\left\| \mathbf{I} - R_{kk}^l(\mathbf{A}h) \exp(-l\mathbf{A}h) \right\|_2 \leq 1$ for all $1 \leq l \leq m$, there is a uniform bound

$$\left\| [R_{kk}(\mathbf{A}h)]^l \right\|_2 \leq 2 \cdot C(\mathbf{A}), \quad \forall l = 1, \dots, m,$$

where $C(\mathbf{A}) = \sup_{t \in [0, T]} \|\exp(\mathbf{A}t)\|_2$. Thus, we derive the bound for $\|\mathbf{L}^{-1}\|_2$:

$$\|\mathbf{L}^{-1}\|_2 \leq 2 \cdot C(\mathbf{A})(m+p) \max \left\{ \|\mathbf{W}^{-1}\|_2, \left\| \widetilde{\mathbf{V}} \widetilde{\mathbf{E}}^T \mathbf{W}^{-1} \right\|_2, \left\| \widetilde{\mathbf{E}}^T \mathbf{W}^{-1} \right\|_2, \sqrt{k+1} \right\}. \quad (71)$$

Up to this point, no special properties of $\mathbf{A}$ have been assumed.

- If we assume $\mathbf{A}$ is Hermitian and negative semi-definite, we can apply Lemma 3.4 and Lemma A.7 to obtain a quite tight bound on the terms within the $\max\{\cdot\}$ operator. We have

$$\begin{aligned} \left\| \mathbf{W}^{-1} \right\|_2 &\leq \sqrt{(k+1)(4 \log(k+1) + 1)}, \\ \left\| \widetilde{\mathbf{E}}^T \mathbf{W}^{-1} \right\|_2 &\leq \sqrt{5k+1}. \end{aligned}$$

Since $\|\widetilde{\mathbf{V}}\|_2 \leq 1$, we can further bound

$$\left\| \widetilde{\mathbf{V}} \widetilde{\mathbf{E}}^T \mathbf{W}^{-1} \right\|_2 \leq \left\| \widetilde{\mathbf{V}} \right\|_2 \left\| \widetilde{\mathbf{E}}^T \mathbf{W}^{-1} \right\|_2 \leq \left\| \widetilde{\mathbf{E}}^T \mathbf{W}^{-1} \right\|_2.$$

For $k \geq 3$, we obtain

$$\max \left\{ \|\mathbf{W}^{-1}\|_2, \left\| \widetilde{\mathbf{V}} \widetilde{\mathbf{E}}^T \mathbf{W}^{-1} \right\|_2, \left\| \widetilde{\mathbf{E}}^T \mathbf{W}^{-1} \right\|_2, \sqrt{k+1} \right\} \leq 3\sqrt{k \log k}.$$

Moreover, in the case where $\mathbf{A}$ is Hermitian and negative semi-definite, we have $C(\mathbf{A}) = 1$, leading to

$$\left\| \mathbf{L}^{-1} \right\|_2 \leq 6(m+p)\sqrt{k \log k}.$$

Finally, the condition number satisfies

$$\kappa \leq 3(m+p)\sqrt{k \log k} (6 + \|\mathbf{A}h\|_2).$$

- If $\mathbf{A}$ is an arbitrary matrix, but we choose $m = \lceil \|\mathbf{A}T\|_2 \rceil$ such that $\|\mathbf{A}h\|_2 \leq 1$, we can apply Lemma 3.5 and Lemma A.8 to obtain the following bounds

$$\begin{aligned}\|\mathbf{W}^{-1}\|_2 &\leq \frac{2\sqrt{e}}{3-e} \sqrt{(k+1)(4\log(k+1)+1)} \\ \|\tilde{\mathbf{E}}^T \mathbf{W}^{-1}\|_2 &\leq \left(\sqrt{e} + \frac{2e}{3-e} \right) \sqrt{2k+1}.\end{aligned}$$

Since $\|\tilde{\mathbf{V}}\|_2$ can also be bounded by a constant, we conclude

$$\|\mathbf{L}^{-1}\|_2 = \mathcal{O}\left(C(\mathbf{A})(m+p)\sqrt{k\log k}\right).$$

Finally, the condition number satisfies

$$\kappa = \mathcal{O}\left(C(\mathbf{A})(m+p)\sqrt{k\log k}\right).$$

□

B Definition and Lemmas for block-encoding

Definition B.1 (block-encoding). *Given an n -qubit matrix $\mathbf{A} \in \mathbb{C}^{N \times N}$ with $N = 2^n$, if we can find $\alpha, \epsilon \in \mathbb{R}_+$, and an $(m+n)$ -qubit unitary matrix $U_{\mathbf{A}}$ such that*

$$\|\mathbf{A} - \alpha(|0^m\rangle \otimes I_N) U_{\mathbf{A}} (|0^m\rangle \otimes I_N)\|_2 \leq \epsilon,$$

then $U_{\mathbf{A}}$ is called an (α, m, ϵ) -block-encoding of $\mathbf{A}$. In particular, if the block-encoding is exact, i.e., $\epsilon = 0$, then $U_{\mathbf{A}}$ is referred to as an (α, m) -block-encoding of $\mathbf{A}$.

Lemma B.2. *Suppose $\mathbf{A} \in \mathbb{C}^{N_1 \times N_1}$, and $\mathbf{B} \in \mathbb{C}^{N_2 \times N_2}$, where $U_{\mathbf{A}}$ is an (α_1, m_1) -block-encoding of $\mathbf{A}$ and $U_{\mathbf{B}}$ is an (α_2, m_2) -block-encoding of $\mathbf{B}$. Then, the tensor product $U_{\mathbf{A}} \otimes U_{\mathbf{B}}$ is an $(\alpha_1\alpha_2, m_1 + m_2)$ -block-encoding of $\mathbf{A} \otimes \mathbf{B}$.*

Proof. By definition of block-encoding, we have

$$\begin{aligned}\mathbf{A} &= \alpha_1 (|0^{m_1}\rangle \otimes I_{N_1}) U_{\mathbf{A}} (|0^{m_1}\rangle \otimes I_{N_1}), \\ \mathbf{B} &= \alpha_2 (|0^{m_2}\rangle \otimes I_{N_2}) U_{\mathbf{B}} (|0^{m_2}\rangle \otimes I_{N_2}).\end{aligned}$$

Thus, their tensor product satisfies

$$\begin{aligned}\mathbf{A} \otimes \mathbf{B} &= \alpha_1\alpha_2 [(|0^{m_1}\rangle \otimes I_{N_1}) U_{\mathbf{A}} (|0^{m_1}\rangle \otimes I_{N_1})] \otimes [(|0^{m_2}\rangle \otimes I_{N_2}) U_{\mathbf{B}} (|0^{m_2}\rangle \otimes I_{N_2})] \\ &= \alpha_1\alpha_2 [|0^{m_1}\rangle \otimes I_{N_1} \otimes |0^{m_2}\rangle \otimes I_{N_2}] [U_{\mathbf{A}} \otimes U_{\mathbf{B}}] [|0^{m_1}\rangle \otimes I_{N_1} \otimes |0^{m_2}\rangle \otimes I_{N_2}].\end{aligned}$$

This confirms that $U_{\mathbf{A}} \otimes U_{\mathbf{B}}$ is an $(\alpha_1\alpha_2, m_1 + m_2)$ -block-encoding of $\mathbf{A} \otimes \mathbf{B}$. □

Lemma B.3 (Lemma 53 in [14]). *If $U_{\mathbf{A}}$ is an (α, a, δ) -block-encoding of an s -qubit operator $\mathbf{A}$, and $U_{\mathbf{B}}$ is a (β, b, ϵ) -block-encoding of an s -qubit operator $\mathbf{B}$, then $(I_b \otimes U_{\mathbf{A}})(I_a \otimes U_{\mathbf{B}})$ is an $(\alpha\beta, a + b, a\epsilon + \beta\delta)$ -block-encoding of $\mathbf{A}\mathbf{B}$.*

Lemma B.4 (Linear Combination of Unitaries [6]). *Let $M = \sum_i \alpha_i U_i$ be a linear combination of unitaries U_i with $\alpha_i > 0$ for all i . Define V as an operator satisfying $V|0^m\rangle := \frac{1}{\sqrt{\alpha}} \sum_i \sqrt{\alpha_i} |i\rangle$, where $\alpha := \sum_i \alpha_i$. Then, the operator $W := V^\dagger U V$ satisfies*

$$W|0^m\rangle|\psi\rangle = \frac{1}{\alpha} |0^m\rangle M|\psi\rangle + |\perp\rangle$$

for all states $|\psi\rangle$, where $U := \sum_i |i\rangle\langle i| \otimes U_i$ and $(|0^m\rangle\langle 0^m| \otimes I)|\perp\rangle = 0$.

Lemma B.5 (Adjust the block-encoding parameters). *Suppose $U_{\mathbf{A}}$ is an (α, m, ϵ) -block-encoding of $\mathbf{A}$. Then, there exist a θ such that $R_Y(\theta) \otimes U_{\mathbf{A}}$ is a $(\beta, m+1, \epsilon)$ -block-encoding of $\mathbf{A}$, where $\beta > \alpha$ and*

$$R_Y(\theta) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}.$$

Proof. Let $U'_{\mathbf{A}} = R_Y(\theta) \otimes U_{\mathbf{A}}$, where $\theta = 2 \arccos \frac{\alpha}{\beta}$. Then, we have

$$\begin{aligned} & \left\| \mathbf{A} - \beta \left(\langle 0^{m+1} | \otimes I_N \right) U'_{\mathbf{A}} \left(|0^{m+1}\rangle \otimes I_N \right) \right\|_2 \\ &= \left\| \mathbf{A} - \beta \cos \frac{\theta}{2} \left(\langle 0^m | \otimes I_N \right) U_{\mathbf{A}} \left(|0^m\rangle \otimes I_N \right) \right\|_2 \\ &= \left\| \mathbf{A} - \alpha \left(\langle 0^m | \otimes I_N \right) U_{\mathbf{A}} \left(|0^m\rangle \otimes I_N \right) \right\|_2 \leq \epsilon. \end{aligned}$$

□

C Analysis of the approximation accuracy and success probability

C.1 Proofs of Lemma 4.1 and Lemma 4.2

As a preliminary step for proving Lemma 4.1 and Lemma 4.2, we introduce two lemmas that are slight modifications of the result found in Appendix A of [33].

Lemma C.1. *If $\|\mathbf{A}\|_2 \leq 1$ and $k \geq 1$, then $\|D_{kk}^{-1}(\mathbf{A})\|_2 \leq \frac{2}{3-e}$.*

Proof. From the definition of D_{pq} as in eq. (9), we have $D_{kk}(\mathbf{A}) = \mathbf{I} + \mathbf{F}$, where

$$\mathbf{F} = \sum_{j=1}^k \frac{(2k-j)!k!}{(2k)!(k-j)!} \frac{(-\mathbf{A})^j}{j!}.$$

Using the fact that

$$\frac{(2k-j)!k!}{(2k)!(k-j)!} \leq \left(\frac{1}{2}\right)^j,$$

we find that

$$\|\mathbf{F}\|_2 \leq \sum_{j=1}^k \left[\frac{1}{2} \|\mathbf{A}\|_2 \right]^j \frac{1}{j!} \leq \frac{1}{2} \|\mathbf{A}\|_2 (e-1) \leq \frac{e-1}{2}.$$

Therefore,

$$\|D_{kk}(\mathbf{A})^{-1}\|_2 = \|(\mathbf{I} + \mathbf{F})^{-1}\|_2 \leq 1/(1 - \|\mathbf{F}\|_2) \leq \frac{2}{3-e}.$$

□

Lemma C.2. *If $\|\mathbf{A}\|_2 \leq 1$ and $k \geq 1$. Define*

$$\mathbf{G} := e^{-\mathbf{A}} R_{kk}(\mathbf{A}) - \mathbf{I}.$$

Then, the following bounds hold

$$\|\mathbf{G}\|_2 \leq 20 \cdot \frac{k!k!}{(2k)!(2k+1)!}, \quad \text{and} \quad \|\mathbf{G}\mathbf{A}^{-1}\|_2 \leq 20 \cdot \frac{k!k!}{(2k)!(2k+1)!}.$$

Proof. Using the remainder theorem for Padé approximation [33, 38], we have

$$R_{kk}(\mathbf{A}) = e^{\mathbf{A}} - \frac{(-1)^k}{(2k)!} \mathbf{A}^{2k+1} D_{kk}(\mathbf{A})^{-1} \int_0^1 e^{(1-u)\mathbf{A}} u^k (1-u)^k du. \quad (72)$$

Therefore,

$$\mathbf{G} := e^{-\mathbf{A}} R_{kk}(\mathbf{A}) - \mathbf{I} = \frac{(-1)^k}{(2k)!} \mathbf{A}^{2k+1} D_{kk}(\mathbf{A})^{-1} \int_0^1 e^{-u\mathbf{A}} u^k (1-u)^k du.$$

Using Lemma C.1, we obtain

$$\|\mathbf{G}\|_2 \leq \frac{2e}{3-e} \frac{1}{(2k)!} \int_0^1 u^k (1-u)^k du \leq 20 \frac{k!k!}{(2k)!(2k+1)!}.$$

Similarly, the inequality for $\|\mathbf{G}\mathbf{A}^{-1}\|_2$ can be derived in the same manner. $\square$

Proof of Lemma 4.1. It is straightforward to verify that

$$\hat{\mathbf{x}}(ih) = R_{kk}^i(\mathbf{A}h)\mathbf{x}_0 + \left(R_{kk}^i(\mathbf{A}h) - \mathbf{I}\right) \mathbf{A}^{-1}\mathbf{b}, \quad \forall i = 1, \dots, m.$$

Since

$$\mathbf{x}(ih) = e^{i\mathbf{A}h} \mathbf{x}_0 + \left(e^{i\mathbf{A}h} - \mathbf{I}\right) \mathbf{A}^{-1}\mathbf{b}, \quad \forall i = 1, \dots, m,$$

we have

$$\mathbf{x}(ih) - \hat{\mathbf{x}}(ih) = \left(\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h)\right) \mathbf{x}(ih) + \left(\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h)\right) \mathbf{A}^{-1}\mathbf{b}.$$

Thus, we obtain the following inequality

$$\|\mathbf{x}(ih) - \hat{\mathbf{x}}(ih)\|_2 \leq \left\| \mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h) \right\|_2 \|\mathbf{x}(ih)\|_2 + \left\| \left(\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h) \right) \mathbf{A}^{-1} \right\|_2 \|\mathbf{b}\|_2. \quad (73)$$

Let

$$\mathbf{G} = e^{-\mathbf{A}h} R_{kk}(\mathbf{A}h) - \mathbf{I},$$

then by Lemma C.2 and condition eq. (33), we have $\|\mathbf{G}\|_2 \leq \frac{\delta}{5}$. Using the conditions $\delta \in \left(0, \frac{1}{m}\right)$ and $\|\mathbf{A}T\|_2 \geq 1$, we deduce that

$$m = \lceil \|\mathbf{A}T\|_2 \rceil \leq 2\|\mathbf{A}T\|_2 \quad \text{and} \quad m\|\mathbf{G}\|_2 \leq \frac{m\delta}{5} < 1.$$

Thus, for all $i = 1, \dots, m$,

$$\begin{aligned} \left\| \mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h) \right\|_2 &= \left\| \mathbf{I} - (\mathbf{I} + \mathbf{G})^i \right\|_2 \\ &\leq \|\mathbf{G}\|_2 \cdot \left(1 + (1 + \|\mathbf{G}\|_2) + \dots + (1 + \|\mathbf{G}\|_2)^{m-1}\right) \\ &= (1 + \|\mathbf{G}\|_2)^m - 1 \\ &\leq e^{m\|\mathbf{G}\|_2} - 1 \leq (e - 1)m\|\mathbf{G}\|_2 \\ &\leq \frac{2(e-1)\delta}{5} \|\mathbf{A}T\|_2 \leq \delta \|\mathbf{A}T\|. \end{aligned} \quad (74)$$

For the second term in eq. (73), we have

$$\begin{aligned} \left(\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h) \right) \mathbf{A}^{-1} &= \left(\mathbf{I} - e^{-\mathbf{A}h} R_{kk}(\mathbf{A}h) \right) \mathbf{A}^{-1} \cdot \sum_{l=0}^{i-1} e^{-l\mathbf{A}h} R_{kk}^l(\mathbf{A}h) \\ &= \mathbf{G} \mathbf{A}^{-1} \cdot \sum_{l=0}^{i-1} (\mathbf{I} + \mathbf{G})^l. \end{aligned}$$

Thus,

$$\begin{aligned} \left\| \left(\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h) \right) \mathbf{A}^{-1} \right\|_2 &\leq h \cdot \left\| \mathbf{G} \cdot (\mathbf{A}h)^{-1} \right\|_2 \cdot \left(1 + (1 + \|\mathbf{G}\|_2) + \cdots (1 + \|\mathbf{G}\|_2)^{m-1} \right) \\ &\leq h \left(\left(1 + \frac{\delta}{5} \right)^m - 1 \right) \leq \delta h \|\mathbf{A}T\| \leq \delta T, \quad \forall i = 1, \dots, m, \end{aligned}$$

where we use Lemma C.2 again, yielding $\|\mathbf{G} \cdot (\mathbf{A}h)^{-1}\|_2 \leq \frac{\delta}{5}$. Finally, we obtain

$$\|\mathbf{x}(ih) - \widehat{\mathbf{x}}(ih)\|_2 \leq \delta T (\|\mathbf{A}\|_2 \|\mathbf{x}(ih)\|_2 + \|\mathbf{b}\|_2), \quad \forall i = 1, \dots, m.$$

□

proof of Lemma 4.2. As in the proof of Lemma 4.1, we need to bound the terms $\left\| \mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h) \right\|_2$ and $\left\| \left(\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h) \right) \mathbf{A}^{-1} \right\|_2$ for all $i = 1, \dots, m$. Let $\mathbf{G} = e^{-\mathbf{A}h} R_{kk}(\mathbf{A}h) - \mathbf{I}$. Then, we have $\|\mathbf{G}\| \leq f_k(\theta)$, where $\theta := \|\mathbf{A}h\|$.

For the first term, as shown in eq. (74), we have

$$\left\| \mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h) \right\|_2 \leq e^{m\|\mathbf{G}\|_2} - 1 \leq e^{mf_k(\theta)} - 1 \leq (e - 1)mf_k(\theta) \leq \delta \|\mathbf{A}T\|_2, \quad \forall i = 1, \dots, m$$

where we use the condition $\delta \in \left(0, \frac{1}{\|\mathbf{A}T\|_2} \right)$ and $f_k(\theta) \leq \frac{\theta\delta}{e-1}$.

For the second term, we have

$$\left\| \mathbf{G} \cdot (\mathbf{A}h)^{-1} \right\|_2 \leq \frac{f_k(\theta)}{\theta} \leq \frac{\delta}{e-1},$$

and thus

$$\begin{aligned} \left\| \left(\mathbf{I} - e^{-i\mathbf{A}h} R_{kk}^i(\mathbf{A}h) \right) \mathbf{A}^{-1} \right\|_2 &\leq h \cdot \left\| \mathbf{G} \cdot (\mathbf{A}h)^{-1} \right\|_2 \cdot \left(1 + (1 + \|\mathbf{G}\|_2) + \cdots (1 + \|\mathbf{G}\|_2)^{m-1} \right) \\ &\leq \frac{h}{\theta} \left(\left(1 + \frac{\delta\theta}{e-1} \right)^m - 1 \right) \\ &\leq \frac{h}{\theta} (e-1)m \frac{\delta\theta}{e-1} = \delta T, \quad \forall i = 1, \dots, m. \end{aligned}$$

In conclusion, we obtain the same error bound as in Lemma 4.1. □

C.2 Proof of Theorem 4.3

Lemma C.3. *Let $D_{kk}(\cdot)$ be the denominator of the (k, k) Padé approximation, as defined in eq. (7). For any $\lambda \geq 0$, we have the following inequalities:*

$$D_{kk}(-\lambda) \geq 1, \quad \text{and} \quad 0 \leq \frac{D_{kk}(-\lambda) - 1}{\lambda D_{kk}(-\lambda)} \leq 1.$$

Proof. The first inequality, $D_{kk}(-\lambda) \geq 1$, is straightforward to verify. For the second inequality, we observe that

$$\frac{D_{kk}(-\lambda) - 1}{\lambda D_{kk}(-\lambda)} = \frac{1}{\lambda + \frac{\lambda}{D_{kk}(-\lambda) - 1}} \leq \min \left\{ \frac{1}{\lambda}, \frac{D_{kk}(-\lambda) - 1}{\lambda} \right\}.$$

Now, consider the function

$$h(\lambda) := \frac{D_{kk}(-\lambda) - 1}{\lambda} = \sum_{j=1}^k d_j \lambda^{j-1}.$$

This function is increasing on $[0, +\infty)$, while $1/\lambda$ is decreasing. Furthermore, we have the limits

$$\lim_{\lambda \rightarrow +\infty} h(\lambda) = +\infty, \quad \lim_{\lambda \rightarrow +\infty} \frac{1}{\lambda} = 0.$$

Additionally, evaluating at $\lambda = 1$,

$$h(1) = D_{kk}(-1) - 1 \leq \sqrt{e} - 1 < 1 = \frac{1}{\lambda} \Big|_{\lambda=1},$$

where the first inequality follows from Lemma A.3. Since $h(\lambda)$ is increasing and $1/\lambda$ is decreasing, there exists a unique $\lambda^* \in (1, +\infty)$ such that $h(\lambda^*) = 1/\lambda^*$. Thus, for all $\lambda \geq 0$, we conclude that

$$\frac{D_{kk}(-\lambda) - 1}{\lambda D_{kk}(-\lambda)} \leq \frac{1}{\lambda^*} < 1.$$

□

Proof of Theorem 4.3. At first, we consider the norm of $\mathbf{z}^{(1)}$. It is easy to verify that

$$\mathbf{z}_0^{(1)} = D_{kk}^{-1}(\mathbf{A}h) \mathbf{x}_0 + \left(D_{kk}^{-1}(\mathbf{A}h) - \mathbf{I} \right) \mathbf{A}^{-1} \mathbf{b}, \quad (75)$$

since $\mathbf{z}_0^{(1)}$ corresponds to the auxiliary variable $\mathbf{v}$ in eq. (12).

- If $\mathbf{A}$ is Hermitian and negative semi-definite, it admits a spectral decomposition $\mathbf{A} = \mathbf{U} \Lambda \mathbf{U}^\dagger$, where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ with $\lambda_1 \leq \dots \leq \lambda_n \leq 0$. Then, we have

$$\mathbf{U}^\dagger \mathbf{z}_0^{(1)} = D_{kk}^{-1}(\Lambda h) \left(\mathbf{U}^\dagger \mathbf{x}_0 \right) + \left(D_{kk}^{-1}(\Lambda h) - \mathbf{I} \right) \Lambda^{-1} \left(\mathbf{U}^\dagger \mathbf{b} \right),$$

and

$$\begin{aligned} \left\| \mathbf{z}_0^{(1)} \right\|_2 &\leq \left\| D_{kk}^{-1}(\Lambda h) \right\|_2 \left\| \mathbf{x}_0 \right\|_2 + \left\| \left(D_{kk}^{-1}(\Lambda h) - \mathbf{I} \right) (\Lambda h)^{-1} \right\|_2 \left\| h \mathbf{b} \right\|_2 \\ &\leq \left\| \mathbf{x}_0 \right\|_2 + h \left\| \mathbf{b} \right\|_2, \end{aligned}$$

where the second inequality follows from Lemma C.3. For $j = 1, \dots, k$, we have

$$\mathbf{z}_j^{(1)} = -d_j D_{kk}^{-1}(\mathbf{A}h) \cdot (-\mathbf{A}h)^{j-1} (\mathbf{A}h \mathbf{x}_0 + h \mathbf{b}), \quad j = 1, \dots, k.$$

This follows from the explicit expression of $\mathbf{W}_k(\mathbf{A}h)^{-1}$. Similarly,

$$\hat{\mathbf{z}}_j^{(1)} = -h d_j D_{kk}^{-1}(\Lambda h) \cdot (-\Lambda h)^{j-1} \left(\Lambda \hat{\mathbf{x}}_0 + \hat{\mathbf{b}} \right), \quad j = 1, \dots, k,$$

where we use the notations

$$\hat{\mathbf{z}}_j^{(1)} = \mathbf{U}^\dagger \mathbf{z}_j^{(1)}, \quad \hat{\mathbf{x}}_0 = \mathbf{U}^\dagger \mathbf{x}_0, \quad \hat{\mathbf{b}} = \mathbf{U}^\dagger \mathbf{b}.$$

Therefore, we obtain

$$\begin{aligned}
\sum_{j=1}^k \left\| \mathbf{z}_j^{(1)} \right\|_2^2 &= \sum_{j=1}^k \left\| \widehat{\mathbf{z}}_j^{(1)} \right\|_2^2 = \sum_{j=1}^k \sum_{l=1}^n \left| \widehat{\mathbf{z}}_{j,l}^{(1)} \right|^2 \\
&= h^2 \sum_{j=1}^k \sum_{l=1}^n d_j^2 \frac{(-\lambda_l h)^{2j-2}}{D_{kk}^2(\lambda_l h)} \left| \lambda_l \widehat{\mathbf{x}}_{0,l} + \widehat{\mathbf{b}}_l \right|^2 \\
&= h^2 \sum_{l=1}^n \left| \lambda_l \widehat{\mathbf{x}}_{0,l} + \widehat{\mathbf{b}}_l \right|^2 \frac{\sum_{j=1}^k d_j^2 (-\lambda_l h)^{2j-2}}{D_{kk}^2(\lambda_l h)} \\
&= h^2 \sum_{|\lambda_l| \leq 1} \left| \lambda_l \widehat{\mathbf{x}}_{0,l} + \widehat{\mathbf{b}}_l \right|^2 \frac{\sum_{j=1}^k d_j^2 (-\lambda_l h)^{2j-2}}{D_{kk}^2(\lambda_l h)} + \sum_{|\lambda_l| > 1} \left| \widehat{\mathbf{x}}_{0,l} + \lambda_l^{-1} \widehat{\mathbf{b}}_l \right|^2 \frac{\sum_{j=1}^k d_j^2 (-\lambda_l h)^{2j}}{D_{kk}^2(\lambda_l h)} \\
&\leq 2h^2 \sum_{|\lambda_l| \leq 1} \left(\left| \widehat{\mathbf{x}}_{0,l} \right|^2 + \left| \widehat{\mathbf{b}}_l \right|^2 \right) \frac{\left(\sum_{j=1}^k d_j (-\lambda_l h)^{j-1} \right)^2}{D_{kk}^2(\lambda_l h)} + 2 \sum_{|\lambda_l| > 1} \left(\left| \widehat{\mathbf{x}}_{0,l} \right|^2 + \left| \widehat{\mathbf{b}}_l \right|^2 \right) \frac{\left(\sum_{j=1}^k d_j (-\lambda_l h)^j \right)^2}{D_{kk}^2(\lambda_l h)} \\
&\leq 2(h^2 + 1) \left(\left\| \widehat{\mathbf{x}}_0 \right\|_2^2 + \left\| \widehat{\mathbf{b}} \right\|_2^2 \right) = 2(h^2 + 1) \left(\left\| \mathbf{x}_0 \right\|_2^2 + \left\| \mathbf{b} \right\|_2^2 \right),
\end{aligned}$$

where the first inequality follows from the facts that $\lambda_l \leq 0, \forall l$ and $|a + b|^2 \leq 2|a|^2 + 2|b|^2, \forall a, b \in \mathbb{C}$, while the second equation follows from Lemma C.3. Combining these results, we obtain

$$\begin{aligned}
\left\| \mathbf{z}^{(1)} \right\|_2^2 &= \sum_{j=0}^k \left\| \mathbf{z}_j^{(1)} \right\|_2^2 \leq (\left\| \mathbf{x}_0 \right\|_2 + h \left\| \mathbf{b} \right\|_2)^2 + 2(h^2 + 1) \left(\left\| \mathbf{x}_0 \right\|_2^2 + \left\| \mathbf{b} \right\|_2^2 \right) \\
&\leq 2 \left((h^2 + 2) \left\| \mathbf{x}_0 \right\|_2^2 + (2h^2 + 1) \left\| \mathbf{b} \right\|_2^2 \right).
\end{aligned} \tag{76}$$

For $i = 1, \dots, m - 1$, we have

$$\left\| \mathbf{z}^{(i+1)} \right\|_2^2 \leq 2 \left((h^2 + 2) \left\| \widehat{\mathbf{x}}(ih) \right\|_2^2 + (2h^2 + 1) \left\| \mathbf{b} \right\|_2^2 \right).$$

Under the assumptions in Lemma 4.2, we obtain inequality eq. (40). Consequently, we derive

$$\begin{aligned}
\left\| \widehat{\mathbf{x}}(ih) - \mathbf{x}(ih) \right\|_2 &\leq \delta T \cdot (\left\| \mathbf{A} \right\|_2 \left\| \mathbf{x}(ih) \right\|_2 + \left\| \mathbf{b} \right\|_2) \\
&\leq \delta T \cdot \left(\left\| \mathbf{A} \right\|_2 \max_{0 \leq t \leq T} \left\| \mathbf{x}(t) \right\|_2 + \left\| \mathbf{b} \right\|_2 \right) \\
&\leq \delta' \cdot \max_{0 \leq t \leq T} \left\| \mathbf{x}(t) \right\|_2,
\end{aligned}$$

for all $i = 1, \dots, m$, where

$$\delta' = \delta T \cdot \left(\left\| \mathbf{A} \right\|_2 + \frac{\left\| \mathbf{b} \right\|_2}{\left\| \mathbf{x}(T) \right\|_2} \right) < \frac{1}{8}.$$

In particular, we obtain $\left\| \widehat{\mathbf{x}}(T) - \mathbf{x}(T) \right\|_2 \leq \delta' \left\| \mathbf{x}(T) \right\|_2$. Thus, we have

$$\frac{\left\| \widehat{\mathbf{x}}(ih) \right\|_2}{\left\| \widehat{\mathbf{x}}(T) \right\|_2} \leq \frac{\left\| \mathbf{x}(ih) \right\|_2 + \left\| \mathbf{x}(ih) - \widehat{\mathbf{x}}(ih) \right\|_2}{\left\| \mathbf{x}(T) \right\|_2 - \left\| \mathbf{x}(T) - \widehat{\mathbf{x}}(T) \right\|_2} \leq \frac{(1 + \delta') \max_{0 \leq t \leq T} \left\| \mathbf{x}(t) \right\|_2}{(1 - \delta') \left\| \mathbf{x}(T) \right\|_2}.$$

Finally, we derive

$$\begin{aligned}
\mathbb{P}_{succ} &\geq \frac{p \|\widehat{\mathbf{x}}(T)\|_2^2}{2 \sum_{i=0}^{m-1} \left((h^2 + 2) \|\widehat{\mathbf{x}}(ih)\|_2^2 + (2h^2 + 1) \|\mathbf{b}\|_2^2 \right) + p \|\widehat{\mathbf{x}}(T)\|_2^2} \\
&= \frac{p}{2 \sum_{i=0}^{m-1} \left((h^2 + 2) \frac{\|\widehat{\mathbf{x}}(ih)\|_2^2}{\|\widehat{\mathbf{x}}(T)\|_2^2} + (2h^2 + 1) \frac{\|\mathbf{b}\|_2^2}{\|\widehat{\mathbf{x}}(T)\|_2^2} \right) + p} \\
&\geq \frac{p}{2 \sum_{i=0}^{m-1} \left((h^2 + 2) \frac{(1+\delta')^2 \max_{0 \leq t \leq T} \|\mathbf{x}(t)\|_2^2}{(1-\delta')^2 \|\mathbf{x}(T)\|_2^2} + (2h^2 + 1) \frac{\|\mathbf{b}\|_2^2}{(1-\delta')^2 \|\mathbf{x}(T)\|_2^2} \right) + p} \\
&\geq \left(\frac{1-\delta'}{1+\delta'} \right)^2 \cdot \frac{p}{6mg^2(h^2 + 1) + p} \\
&\geq \frac{1}{2} \cdot \frac{p}{6mg^2(h^2 + 1) + p}
\end{aligned}$$

where

$$g := \frac{\max \{ \max_{0 \leq t \leq T} \|\mathbf{x}(t)\|_2, \|\mathbf{b}\|_2 \}}{\|\mathbf{x}(T)\|_2}.$$

- If $\mathbf{A}$ is an arbitrary matrix satisfying $\|\mathbf{A}h\|_2 \leq 1$, then we have

$$\begin{aligned}
\|\mathbf{z}_0^{(1)}\|_2 &\leq \|D_{kk}(\mathbf{A}h)^{-1}\|_2 \left(\|\mathbf{x}_0\|_2 + \left\| \sum_{j=1}^k d_j (\mathbf{A}h)^{j-1} \right\|_2 h \|\mathbf{b}\|_2 \right) \\
&\leq \frac{2}{3-e} (\|\mathbf{x}_0\|_2 + (D_{kk}(-1) - 1) h \|\mathbf{b}\|_2) \\
&\leq \frac{2}{3-e} (\|\mathbf{x}_0\|_2 + h \|\mathbf{b}\|_2).
\end{aligned}$$

The second inequality follows from Lemma C.1 and the bound

$$\left\| \sum_{j=1}^k d_j (\mathbf{A}h)^{j-1} \right\|_2 \leq \sum_{j=1}^k d_j \|\mathbf{A}h\|_2^{j-1} \leq \sum_{j=1}^k d_j = D_{kk}(-1) - 1.$$

The last inequality is deriving using Lemma A.3, which ensures $D_{kk}(-1) - 1 \leq 1$. Moreover, applying Lemma C.1 again, we obtain

$$\|\mathbf{z}_j^{(1)}\|_2 = d_j \|D_{kk}^{-1}(\mathbf{A}h)\|_2 \cdot \|\mathbf{A}h\|_2^{j-1} h \|\mathbf{A}\mathbf{x}_0 + \mathbf{b}\|_2 \leq \frac{2hd_j}{3-e} \|\mathbf{A}\mathbf{x}_0 + \mathbf{b}\|_2,$$

for all $j = 1, \dots, k$. Summing over all j , we get

$$\begin{aligned}
\sum_{j=1}^k \|\mathbf{z}_j^{(1)}\|_2^2 &= \left(\frac{2h}{3-e} \|\mathbf{A}\mathbf{x}_0 + \mathbf{b}\|_2 \right)^2 \sum_{j=1}^k d_j^2 \leq \left(\frac{2h}{3-e} \right)^2 \|\mathbf{A}\mathbf{x}_0 + \mathbf{b}\|_2^2 \\
&\leq 2 \left(\frac{2}{3-e} \right)^2 (\|\mathbf{A}h\|_2^2 \|\mathbf{x}_0\|_2^2 + h^2 \|\mathbf{b}\|_2^2) \leq 2 \left(\frac{2}{3-e} \right)^2 (\|\mathbf{x}_0\|_2^2 + h^2 \|\mathbf{b}\|_2^2).
\end{aligned}$$

More generally, we obtain

$$\sum_{j=0}^k \|\mathbf{z}_j^{(i+1)}\|_2^2 \leq 4 \left(\frac{2}{3-e} \right)^2 (\|\widehat{\mathbf{x}}(ih)\|_2^2 + h^2 \|\mathbf{b}\|_2^2), \quad \forall i = 0, \dots, m-1,$$

where the constant factor satisfies $\left(\frac{2}{3-e}\right)^2 \leq 51$. Following a similar argument as in the previous case, we establish

$$\begin{aligned}\mathbb{P}_{succ} &\geq \frac{p\|\hat{\mathbf{x}}(T)\|_2^2}{204\sum_{i=0}^{m-1}\left(\|\hat{\mathbf{x}}(ih)\|_2^2 + h^2\|\mathbf{b}\|_2^2\right) + p\|\hat{\mathbf{x}}(T)\|_2^2} \\ &\geq \frac{1}{2} \cdot \frac{p}{204mg^2(h^2 + 1) + p},\end{aligned}$$

with the aid of Lemma 4.1, where

$$g := \frac{\max\{\max_{0 \leq t \leq T} \|\mathbf{x}(t)\|_2, \|\mathbf{b}\|_2\}}{\|\mathbf{x}(T)\|_2}.$$

□

D Lemmas for quantum states

The following three lemmas from [7] are used to bound the distance between quantum states.

Lemma D.1 (Lemma 13 in [7]). *Let ψ and φ be two vectors such that $\|\psi\|_2 \geq \alpha > 0$ and $\|\psi - \varphi\|_2 \leq \beta$. Then*

$$\left\| \frac{\psi}{\|\psi\|_2} - \frac{\varphi}{\|\varphi\|_2} \right\|_2 \leq \frac{2\beta}{\alpha}.$$

Lemma D.2 (Lemma 14 in [7]). *Let $|\psi\rangle = \alpha|0\rangle|\psi_0\rangle + \sqrt{1-\alpha^2}|1\rangle|\psi_1\rangle$ and $|\varphi\rangle = \beta|0\rangle|\varphi_0\rangle + \sqrt{1-\beta^2}|1\rangle|\varphi_1\rangle$, where $|\psi_0\rangle, |\psi_1\rangle, |\varphi_0\rangle, |\varphi_1\rangle$ are unit vectors, and $\alpha, \beta \in [0, 1]$. Suppose $\| |\psi\rangle - |\varphi\rangle \|_2 \leq \delta < \alpha$. Then $\| |\psi_0\rangle - |\varphi_0\rangle \|_2 \leq \frac{2\delta}{\alpha-\delta}$.*

Lemma D.3 (Lemma 15 in [7]). *Let $|\psi\rangle = \alpha|0\rangle|\psi_0\rangle + \sqrt{1-\alpha^2}|1\rangle|\psi_1\rangle$ and $|\varphi\rangle = \beta|0\rangle|\varphi_0\rangle + \sqrt{1-\beta^2}|1\rangle|\varphi_1\rangle$, where $|\psi_0\rangle, |\psi_1\rangle, |\varphi_0\rangle, |\varphi_1\rangle$ are unit vectors, and $\alpha, \beta \in [0, 1]$. Suppose $\| |\psi\rangle - |\varphi\rangle \|_2 \leq \delta < \alpha$. Then $\beta \geq \alpha - \delta$.*

References

- [1] Awad H. Al-Mohy and Nicholas J. Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM Journal on Matrix Analysis and Applications*, 31(3):970–989, 2010. DOI: [10.1137/09074721X](https://doi.org/10.1137/09074721X). URL <https://doi.org/10.1137/09074721X>.
- [2] Dong An, Andrew M. Childs, and Lin Lin. Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters. *ArXiv*, abs/2312.03916, 2023. URL <https://doi.org/10.48550/arXiv.2312.03916>.
- [3] Dong An, Jin-Peng Liu, and Lin Lin. Linear combination of hamiltonian simulation for nonunitary dynamics with optimal state preparation cost. *Phys. Rev. Lett.*, 131: 150603, Oct 2023. DOI: [10.1103/PhysRevLett.131.150603](https://doi.org/10.1103/PhysRevLett.131.150603). URL <https://doi.org/10.1103/PhysRevLett.131.150603>.
- [4] Dominic W Berry. High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical*, 47(10):105301, feb 2014. DOI: [10.1088/1751-8113/47/10/105301](https://doi.org/10.1088/1751-8113/47/10/105301). URL <https://dx.doi.org/10.1088/1751-8113/47/10/105301>.

- [5] Dominic W. Berry and Pedro C. S. Costa. Quantum algorithm for time-dependent differential equations using Dyson series. *Quantum*, 8:1369, June 2024. ISSN 2521-327X. DOI: [10.22331/q-2024-06-13-1369](https://doi.org/10.22331/q-2024-06-13-1369). URL <https://doi.org/10.22331/q-2024-06-13-1369>.
- [6] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 792–809, 2015. DOI: [10.1109/FOCS.2015.54](https://doi.org/10.1109/FOCS.2015.54). URL <https://doi.org/10.1109/FOCS.2015.54>.
- [7] 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*, 356:1057–1081, 2017. DOI: [10.1007/s00220-017-3002-y](https://doi.org/10.1007/s00220-017-3002-y). URL <https://doi.org/10.1007/s00220-017-3002-y>.
- [8] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation, 2002. ISSN 0271-4132. URL <https://doi.org/10.48550/arXiv.quant-ph/0005055>.
- [9] Daan Camps and Roel Van Beeumen. FABLE: Fast Approximate Quantum Circuits for Block-Encodings. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 104–113, Los Alamitos, CA, USA, September 2022. IEEE Computer Society. DOI: [10.1109/QCE53715.2022.00029](https://doi.org/10.1109/QCE53715.2022.00029). URL <https://doi.org/10.1109/QCE53715.2022.00029>.
- [10] Yu Cao, Shi Jin, and Nana Liu. Quantum simulation for time-dependent hamiltonians – with applications to non-autonomous ordinary and partial differential equations, 2023. URL <https://doi.org/10.48550/arXiv.2312.02817>.
- [11] Andrew M. Childs and Jin-Peng Liu. Quantum spectral methods for differential equations. *Communications in Mathematical Physics*, 375:1427 – 1457, 2019. URL <https://doi.org/10.1007/s00220-020-03699-z>.
- [12] Pedro C.S. Costa, Dong An, Yuval R. Sanders, Yuan Su, Ryan Babbush, and Dominic W. Berry. Optimal scaling quantum linear-systems solver via discrete adiabatic theorem. *PRX Quantum*, 3:040303, Oct 2022. DOI: [10.1103/PRXQuantum.3.040303](https://doi.org/10.1103/PRXQuantum.3.040303). URL <https://doi.org/10.1103/PRXQuantum.3.040303>.
- [13] Xiaojing Dong, Yizhe Peng, Qili Tang, Yin Yang, and Yue Yu. Investigation on a quantum algorithm for linear differential equations, 2024. URL <https://doi.org/10.48550/arXiv.2408.01762>.
- [14] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 193–204, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367059. DOI: [10.1145/3313276.3316366](https://doi.org/10.1145/3313276.3316366). URL <https://doi.org/10.1145/3313276.3316366>.
- [15] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009. DOI: [10.1103/PhysRevLett.103.150502](https://doi.org/10.1103/PhysRevLett.103.150502). URL <https://doi.org/10.1103/PhysRevLett.103.150502>.
- [16] Nicholas J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, 26(4):1179–1193, 2005. DOI: [10.1137/04061101X](https://doi.org/10.1137/04061101X). URL <https://doi.org/10.1137/04061101X>.
- [17] Junpeng Hu, Shi Jin, Nana Liu, and Lei Zhang. Dilation theorem via schrödingerisation, with applications to the quantum simulation of differential equations, 2023. URL <https://doi.org/10.48550/arXiv.2309.16262>.

- [18] Junpeng Hu, Shi Jin, Nana Liu, and Lei Zhang. Quantum Circuits for partial differential equations via Schrödingerisation. *Quantum*, 8:1563, December 2024. ISSN 2521-327X. DOI: [10.22331/q-2024-12-12-1563](https://doi.org/10.22331/q-2024-12-12-1563). URL <https://doi.org/10.22331/q-2024-12-12-1563>.
- [19] Junpeng Hu, Shi Jin, and Lei Zhang. Quantum algorithms for multiscale partial differential equations. *Multiscale Modeling & Simulation*, 22(3):1030–1067, 2024. DOI: [10.1137/23M1566340](https://doi.org/10.1137/23M1566340). URL <https://doi.org/10.1137/23M1566340>.
- [20] Shi Jin and Nana Liu. Quantum simulation of discrete linear dynamical systems and simple iterative methods in linear algebra via schrodingerisation, 2023. URL <https://doi.org/10.48550/arXiv.2304.02865>.
- [21] Shi Jin and Nana Liu. Analog quantum simulation of partial differential equations. *Quantum Science and Technology*, 9(3):035047, jun 2024. DOI: [10.1088/2058-9565/ad49cf](https://dx.doi.org/10.1088/2058-9565/ad49cf). URL <https://dx.doi.org/10.1088/2058-9565/ad49cf>.
- [22] Shi Jin and Nana Liu. Analog quantum simulation of parabolic partial differential equations using jaynes-cummings-like models, 2024. URL <https://doi.org/10.48550/arXiv.2407.01913>.
- [23] Shi Jin, Nana Liu, and Yue Yu. Quantum simulation of partial differential equations via schrodingerisation, 2022. URL <https://doi.org/10.48550/arXiv.2212.13969>.
- [24] Shi Jin, Nana Liu, and Chuwen Ma. Quantum simulation of maxwell’s equations via schrödingersation, 2023. URL <https://doi.org/10.48550/arXiv.2308.08408>.
- [25] Shi Jin, Nana Liu, and Yue Yu. Quantum simulation of partial differential equations: Applications and detailed analysis. *Phys. Rev. A*, 108:032603, Sep 2023. DOI: [10.1103/PhysRevA.108.032603](https://doi.org/10.1103/PhysRevA.108.032603). URL <https://doi.org/10.1103/PhysRevA.108.032603>.
- [26] Shi Jin, Xiantao Li, Nana Liu, and Yue Yu. Quantum simulation for partial differential equations with physical boundary or interface conditions. *Journal of Computational Physics*, 498:112707, 2024. ISSN 0021-9991. DOI: [10.1016/j.jcp.2023.112707](https://doi.org/10.1016/j.jcp.2023.112707). URL <https://doi.org/10.1016/j.jcp.2023.112707>.
- [27] Shi Jin, Xiantao Li, Nana Liu, and Yue Yu. Quantum simulation for quantum dynamics with artificial boundary conditions. *SIAM Journal on Scientific Computing*, 46(4):B403–B421, 2024. DOI: [10.1137/23M1563451](https://doi.org/10.1137/23M1563451). URL <https://doi.org/10.1137/23M1563451>.
- [28] Shi Jin, Nana Liu, and Chuwen Ma. Schrödingerisation based computationally stable algorithms for ill-posed problems in partial differential equations, 2024. URL <https://doi.org/10.48550/arXiv.2403.19123>.
- [29] Shi Jin, Nana Liu, and Chuwen Ma. On schrödingerization based quantum algorithms for linear dynamical systems with inhomogeneous terms, 2024. URL <https://doi.org/10.48550/arXiv.2402.14696>.
- [30] Shi Jin, Nana Liu, and Yue Yu. Quantum simulation of the fokker-planck equation via schrodingerization, 2024. URL <https://doi.org/10.48550/arXiv.2404.13585>.
- [31] Shi Jin, Nana Liu, and Yue Yu. Quantum circuits for the heat equation with physical boundary conditions via schrodingerisation, 2025. URL <https://doi.org/10.48550/arXiv.2407.15895>.
- [32] Hari Krovi. Improved quantum algorithms for linear and nonlinear differential equations. *Quantum*, 7:913, February 2023. ISSN 2521-327X. DOI: [10.22331/q-2023-02-02-913](https://doi.org/10.22331/q-2023-02-02-913). URL <https://doi.org/10.22331/q-2023-02-02-913>.
- [33] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003. DOI: [10.1137/S00361445024180](https://doi.org/10.1137/S00361445024180). URL <https://doi.org/10.1137/S00361445024180>.

- [34] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2(1):15023, Jan 2016. ISSN 2056-6387. DOI: [10.1038/npjqi.2015.23](https://doi.org/10.1038/npjqi.2015.23). URL <https://doi.org/10.1038/npjqi.2015.23>.
- [35] Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa. Quantum circuits for general multiqubit gates. *Phys. Rev. Lett.*, 93:130502, Sep 2004. DOI: [10.1103/PhysRevLett.93.130502](https://doi.org/10.1103/PhysRevLett.93.130502). URL <https://doi.org/10.1103/PhysRevLett.93.130502>.
- [36] Edward B Saff and Richard S Varga. On the zeros and poles of padé approximants to ez. *Numerische Mathematik*, 25(1):1–14, 1975. DOI: [10.1007/BF01411842](https://doi.org/10.1007/BF01411842). URL <https://doi.org/10.1007/BF01411842>.
- [37] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. DOI: [10.1137/S0097539795293172](https://doi.org/10.1137/S0097539795293172). URL <https://doi.org/10.1137/S0097539795293172>.
- [38] Richard S. Varga. On higher order stable implicit methods for solving parabolic partial differential equations. *Journal of Mathematics and Physics*, 40(1-4):220–231, 1961. DOI: <https://doi.org/10.1002/sapm1961401220>.
- [39] Hsuan-Cheng Wu and Xiantao Li. Structure-preserving quantum algorithms for linear and nonlinear hamiltonian systems, 2025. URL <https://doi.org/10.48550/arXiv.2411.03599>.