

Fast-forwarding quantum algorithms for linear dissipative differential equations

Dong An^{1,2}, Akwum Onwunta³, and Gengzhi Yang^{2,4}

¹Beijing International Center for Mathematical Research (BICMR), Peking University, Beijing, China

²Joint Center for Quantum Information and Computer Science (QulCS), University of Maryland, College Park, MD, USA

³Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

⁴Department of Mathematics, University of Maryland, College Park, MD, USA

We establish improved complexity estimates of quantum algorithms for linear dissipative ordinary differential equations (ODEs) and show that the time dependence can be fast-forwarded to be sub-linear. Specifically, we show that a quantum algorithm based on truncated Dyson series can prepare history states of dissipative ODEs up to time T with cost $\tilde{O}(\log(T)(\log(1/\epsilon))^2)$, which is an exponential speedup over the best previous result. For final state preparation at time T , we show that its complexity is $\tilde{O}(\sqrt{T}(\log(1/\epsilon))^2)$, achieving a polynomial speedup in T . We also analyze the complexity of simpler lower-order quantum algorithms, such as the forward Euler method and the trapezoidal rule, and find that even lower-order methods can still achieve $\tilde{O}(\sqrt{T})$ cost with respect to time T for preparing final states of dissipative ODEs. As applications, we show that quantum algorithms can simulate dissipative non-Hermitian quantum dynamics and heat processes with fast-forwarded complexity sub-linear in time.

Contents

1	Introduction	2
1.1	Main results	3
1.2	Additional results	5
1.3	Related works and comparison	6
1.4	Applications	7
1.5	Discussions and open questions	7
1.6	Organization	8
2	Dissipative ODEs and stability	8
3	Quantum ODE algorithm	9
3.1	Time discretization	9
3.2	Algorithms	10
3.2.1	History state preparation	10
3.2.2	Final state preparation	10
4	Fast-forwarded complexity	11
4.1	Condition number	11
4.2	Complexity of history state preparation	13
4.2.1	General inhomogeneous case	13
4.2.2	Homogeneous case	17
4.3	Complexity of final state preparation	17
5	Specific numerical methods	20

5.1	Truncated Dyson series method	20
5.2	Lower-order methods	23
5.2.1	Forward Euler method	23
5.2.2	Trapezoidal rule	26
6	Applications	28
6.1	Quantum dynamics with non-Hermitian Hamiltonian	28
6.2	Generalized heat equation	29
A	Proof of Lemma 5	33
B	Proof of Lemma 6	33
C	Proof of Theorem 9	34
D	Proof of Corollary 13	36
E	Constructing block-encodings in lower-order methods	37
E.1	Forward Euler method	37
E.2	Trapezoidal rule	39
F	Error analysis of lower-order numerical schemes	41
F.1	Forward Euler	41
F.1.1	Evolution operator	41
F.1.2	Inhomogeneous term	41
F.2	Trapezoidal rule	42
F.2.1	Evolution operator	42
F.2.2	Inhomogeneous term	43

1 Introduction

Differential equations are fundamental in modeling the dynamics of a system and have been widely used in physics, chemistry, engineering, economics, and other disciplines. In this paper, we consider the solution of a system of ordinary differential equations (ODEs) of the form

$$\frac{du(t)}{dt} = A(t)u(t) + b(t), \quad t \in [0, T], \quad (1)$$

$$u(0) = u_0. \quad (2)$$

Here, $A(t) \in \mathbb{C}^{N \times N}$ and $b(t) \in \mathbb{C}^N$. We are interested in approximating the final solution $u(t)$ at time $t = T$, or computing a history state encoding the trajectory of the dynamics, i.e., $\mathbf{u} = [u_0; u_1; \dots; u_M]^\top$ such that $u_j \approx u(jT/M)$ where M is the number of time steps, and we let $h = T/M$ denote the time step size.

In practice, the dimension N of Eq. (1) can be huge, especially in semi-discretized partial differential equations with high accuracy or systems that model the motion and interaction of a large number of objects. This high dimensionality poses significant computational challenges for efficient classical simulation of Eq. (1), and has evoked emerging interest in exploring quantum algorithms for differential equations with substantially improved efficiency.

The first efficient quantum algorithm for general ODEs can be dated back to [1], which leverages the power of quantum linear system algorithms. Specifically, the algorithm in [1] first discretizes Eq. (1) by a multi-step method, formulates the discretized ODE as a linear system of equations, and solves the linear system by the Harrow–Hassidim–Lloyd (HHL) algorithm [2]. Since then, there have been several subsequent works on improved quantum ODE algorithms based on the same idea of converting the ODE to a linear system of equations but with more advanced time discretization methods and quantum linear system algorithms [3, 4, 5, 6]. There have also

been attempts on designing quantum ODE algorithms based on ideas other than the linear-system-based approach, such as time-marching [7], quantum eigenvalue processing [8], and reduction to Hamiltonian simulation problems [9, 10, 11, 12].

Since simulating ODEs for a long time period appears ubiquitously in scientific and engineering computation, an important aspect among others of quantum ODE algorithms is how the complexity depends on the evolution time T . The state-of-the-art linear-system-based algorithm is the truncated Dyson series method [6], which uses the truncated Dyson series for time discretization, applies optimal quantum linear system algorithm [13], and achieves $\tilde{\mathcal{O}}(T)$ query complexity. Similarly, the best available method without solving a linear system of equations also has query complexity $\tilde{\mathcal{O}}(T)$ [10]. Both of the two strategies scale almost linearly in the evolution time T , which can still be expensive for long time simulation.

In this work, we explore the possibility of further fast-forwarding quantum ODE algorithms in time dependence and focus on the following question:

Can we solve (a subset of) differential equations on a quantum computer with a sub-linear cost in the evolution time?

Notice that for Hamiltonian simulation problems where $A(t)$ is anti-Hermitian and $b(t) \equiv 0$, [14] establishes a lower bound with linear time dependence $\Omega(T)$, so we do not expect such fast-forwarding to be generically possible, though fast-forwarded Hamiltonian simulation of specific systems has been proposed in [15, 16]. Instead, in this work we focus on dissipative ODEs where $A(t)$ has a uniformly negative logarithmic norm.

For dissipative ODEs, we establish an improved complexity estimate for all linear-system-based algorithms regardless of time discretization scheme. When applied to the truncated Dyson series method, our analysis yields an $\tilde{\mathcal{O}}(\text{poly} \log(T/\epsilon))$ query complexity for history state preparation. This is an exponential improvement in evolution time T compared to existing results. For final state preparation, we show that the truncated Dyson series method can scale only $\tilde{\mathcal{O}}(\sqrt{T} \text{poly} \log(1/\epsilon))$, achieving a quadratic speedup over the best existing results. We also analyze the complexity of linear-system-based algorithms with simpler lower-order time discretization schemes, including the forward Euler method and trapezoidal rule. Interestingly, we find that with lower-order methods we can still achieve $\tilde{\mathcal{O}}(\sqrt{T})$ for final state preparation in terms of T , and lower convergence order only worsens the error dependence. Our algorithms and analysis can be applied to simulating dissipative quantum dynamics with non-Hermitian Hamiltonian and heat process.

1.1 Main results

We study fast-forwarding quantum algorithms for linear dissipative ODEs. In Eq. (1), we assume the logarithmic norm of $A(t)$ is uniformly negative, i.e., there exists a constant $\eta > 0$ such that

$$A(t) + A(t)^\dagger \leq -2\eta < 0. \quad (3)$$

We are interested in computing a final state proportional to $u(T)$, or a history state proportional to

$$\mathbf{u}_{\text{exact}} = [u(0); u(T/M); u(2T/M); \cdots; u(T)], \quad (4)$$

where M is the number of time steps. Our main mathematical result is a general quantum complexity estimate for dissipative ODEs.

Result 1 (Informal version of Theorem 8 and Corollary 11). *Consider Eq. (1) with $A(t) + A(t)^\dagger \leq -2\eta < 0$ for a constant $\eta > 0$ and we use M steps for time discretization. Then, we can prepare an ϵ -approximation of*

1. the history state $\mathbf{u}_{\text{exact}}/\|\mathbf{u}_{\text{exact}}\|$ up to time T with query complexity

$$\mathcal{O}\left(\frac{M}{T} \log\left(\frac{1}{\epsilon}\right)\right), \quad (5)$$

2. the final state $u(T)/\|u(T)\|$ at time T with query complexity

$$\tilde{\mathcal{O}}\left(\frac{\max_t \|u(t)\|}{\|u(T)\|} \frac{M}{\sqrt{T}} \log\left(\frac{1}{\epsilon}\right)\right). \quad (6)$$

The algorithm used in [Result 1](#) is a slightly modified linear-system-based approach from [1, 3, 4, 5, 6]. Specifically, for history state preparation, we use M many equidistant time steps to discretize [Eq. \(1\)](#). Since [Eq. \(1\)](#) is linear, its discretized version is naturally a linear system of equations with solution $\mathbf{u} = [u_0; u_1; u_2; \dots; u_M]$ where each u_j aims to approximate $u(jT/M)$. Then we solve this linear system of equations by quantum algorithms with optimal asymptotic scaling (e.g., [13, 17, 18]) to prepare the desired history state. Notice that a final state $u(T)/\|u(T)\|$ can be directly obtained from the history state by post-selecting the register encoding the index of the time step onto M . However, the success probability can be small as most blocks in the history state are intermediate steps. To boost the success probability, we may use the padding trick introduced in [1] by adding trivial lines at the end of the linear system of equations to make multiple copies of the final state. A difference from [1] is that we can use much fewer padding lines for dissipative ODEs, and we will discuss the reason later. A detailed discussion on the quantum algorithms is presented in [Section 3](#).

For history state preparation, our [Result 1](#) shows that the complexity is only $\mathcal{O}(\frac{M}{T} \log(\frac{1}{\epsilon}))$, where the factor $\mathcal{O}(M/T)$ is the condition number of the linear system, and the $\mathcal{O}(\log(1/\epsilon))$ is due to the optimal quantum linear system algorithm. Compared to the previous analysis, our improvement lies in a better estimate of the condition number. Specifically, previous analysis [1, 3, 4, 5, 6] only bounds the condition number by $\mathcal{O}(M)$, so we gain a multiplicative factor of $\mathcal{O}(1/T)$. Such an improvement is crucially dependent on the additional dissipation in the dynamics. Recall that the condition number of the linear system represents how a small perturbation in the right-hand side is amplified in the solution. For a stable but non-dissipative dynamics¹, a small perturbation in the initial condition u_0 or the inhomogeneous term $b(t)$ can be carried out to the entire dynamics with the same amplitude, so the condition number becomes linear in terms of the number of steps. However, in a dissipative ODE, the effect of a small perturbation in u_0 or $b(t)$ decays exponentially in time, so such perturbation will only introduce non-trivial perturbations on the dynamics for a fixed time period instead of the entire time period up to T . This is why in the condition number we can obtain a $\mathcal{O}(1/T)$ multiplicative factor improvement.

For final state preparation, our [Result 1](#) gives an $\tilde{\mathcal{O}}\left(\frac{\max_t \|u(t)\|}{\|u(T)\|} \frac{M}{\sqrt{T}} \log(\frac{1}{\epsilon})\right)$ query complexity, while previous analysis in [1, 3, 4, 5, 6] suggests the query complexity to be $\tilde{\mathcal{O}}\left(\frac{\max_t \|u(t)\|}{\|u(T)\|} M \log(\frac{1}{\epsilon})\right)$.

Unlike the history state preparation case, here we only obtain an $\mathcal{O}(1/\sqrt{T})$ factor of improvement. This is because of the fewer padding lines we use in the linear system. Specifically, notice that if we add M_p many padding lines, then the condition number will be increased by an additive factor $\mathcal{O}(M_p)$. In the general stable but non-dissipative case, the condition number of the linear system without padding is $\mathcal{O}(M)$, so we can add as many as $M_p = M$ padding lines while not increasing the condition number asymptotically. Then the number of copies of the final state $u(T)$ is the same as the number of intermediate steps $u(jT/M)$ for $j < M$, so the probability of extracting $u(T)$ from the padded history state only depends on the norm decay rate $\max_t \|u(t)\|/\|u(T)\|$ and the overall complexity is still linear in $\mathcal{O}(M)$. However, for the dissipative ODEs, we have shown that the condition number without padding is just $\mathcal{O}(M/T)$, so the maximum number of extra padding lines allowed without increasing the condition number asymptotically is just $M_p \sim M/T$, which is asymptotically smaller than the number of time steps. Then we need more repeats to successfully extract the final state $u(T)$ from the padded history state, and with amplitude amplification, this cancels an $\mathcal{O}(\sqrt{T})$ factor. We remark that the choice $M_p \sim M/T$ is indeed optimal: we show in [Theorem 10](#) the query complexity of final state preparation for any choice of M_p and find the optimal choice to be $\mathcal{O}(M/T)$ in [Corollary 11](#).

So far we have not specified a concrete time discretization method in the algorithm. In fact, our [Result 1](#) works for any single step time discretization method, including the simplest forward Euler method, Runge-Kutta method, and the truncated Dyson series method, while the choice of M to achieve a certain level of accuracy varies among different methods. For the truncated Dyson series method [6], our analysis yields the following complexity for dissipative ODEs.

¹This refers to the dynamics [Eq. \(1\)](#) where $A(t)$ has only non-positive logarithmic norm, i.e., $A(t) + A(t)^\dagger \leq 0$, instead of the dissipative ODE which requires the logarithmic norm to be strictly smaller than 0. Notice that the stable condition $A(t) + A(t)^\dagger \leq 0$ is also assumed in previous works for quantum ODE algorithms [6, 9, 10] and includes the case of Hamiltonian simulation.

Result 2 (Informal version of [Corollary 12](#)). Consider the truncated Dyson series method for [Eq. \(1\)](#) with $A(t) + A(t)^\dagger \leq -2\eta < 0$. Then, we can prepare an ϵ -approximation of

1. the history state $\mathbf{u}_{\text{exact}}/\|\mathbf{u}_{\text{exact}}\|$ up to time T with query complexity

$$\mathcal{O}\left(\log(T)\left(\log\left(\frac{1}{\epsilon}\right)\right)^2\right), \quad (7)$$

2. the final state $u(T)/\|u(T)\|$ at time T with query complexity

$$\tilde{\mathcal{O}}\left(\frac{\max_t \|u(t)\|}{\|u(T)\|} \sqrt{T} \left(\log\left(\frac{1}{\epsilon}\right)\right)^2\right). \quad (8)$$

The proof of [Result 2](#) is discussed in [Section 5.1](#), which we can obtain from our general [Result 1](#) by noticing that $M = \mathcal{O}(T)$ is sufficient in the truncated Dyson series method. The extra $\log(T)$ factor in the history state preparation cost is due to the truncated order of the Dyson series, which typically depends on the evolution time and requires more queries to the matrix $A(t)$ for constructing the linear system. The highlight of [Result 2](#) is the time dependence: in both history and final state preparation cases, we obtain sub-linear scalings in time T and this is better than previous results. We will make a careful comparison between [Result 2](#) and existing works later in [Section 1.3](#).

1.2 Additional results

Lower-order methods: In [Result 2](#), we have analyzed the query complexity of the truncated Dyson series method for dissipative ODEs. Although the truncated Dyson series method can achieve high-order convergence, its quantum implementation requires complicated quantum control logic. In [Section 5.2](#), we also discuss the complexity of two lower-order methods that can be implemented in a much simpler way: the first-order forward Euler method and the second-order trapezoidal rule.

Result 3. Consider the forward Euler method and the trapezoidal rule for [Eq. \(1\)](#) with $A(t) + A(t)^\dagger \leq -2\eta < 0$. Then, we can prepare an ϵ -approximation of

1. the history state $\mathbf{u}_{\text{exact}}/\|\mathbf{u}_{\text{exact}}\|$ up to time T with query complexity²

$$\mathcal{O}\left(\frac{T^{1/(2p)}}{\epsilon^{1/p}} \log\left(\frac{1}{\epsilon}\right)\right), \quad (9)$$

where $p = 1$ for the forward Euler, and $p = 2$ for the trapezoidal rule,

2. the final state $u(T)/\|u(T)\|$ at time T with query complexity

$$\tilde{\mathcal{O}}\left(\frac{\sqrt{T}}{\epsilon^{1/p}} \log\left(\frac{1}{\epsilon}\right)\right), \quad (10)$$

where $p = 1$ for the forward Euler, and $p = 2$ for the trapezoidal rule.

From [Result 3](#), we can see that the error dependence is standard, i.e., $\mathcal{O}(\epsilon^{-1/p})$ for a p -th order method, but the time dependence is always sub-linear. For history state preparation, higher-order method can provide a further improved time scaling. Interestingly, for final state preparation, both methods scale $\mathcal{O}(\sqrt{T})$, which is exactly the same as the truncated Dyson series method. Therefore, if we only care about complexity in terms of time T , the simplest Euler method suffices to achieve the best available scaling. The reason why we can achieve the same time scaling is, although the condition number of the linear system becomes larger for lower-order methods as suggested by the complexity of history state preparation, we may correspondingly add more padding lines to increase the probability of measuring onto the final state without further increasing the condition number, so the final state preparation complexity can still remain asymptotically the same. We also expect [Result 3](#) to hold for general p -th order time discretization method.

²For technical simplicity, here we only focus on the explicit dependence of T and ϵ , and assume all the other parameters to be constants (e.g., the decay rate $\max_t \|u(t)\|/\|u(T)\|$, the norms $\max_t \|A(t)\|$ and $\|b(t)\|$, and parameters related to the stability condition). The same also applies for the final state preparation.

Methods	Assumptions		Query complexities	
	Time	Stability	History state	Final state
p -th order multi-step [1]	TI	$\text{Re}(\text{eig}(A)) \leq 0$	$\tilde{\mathcal{O}}(T^{2+2/p}/\epsilon^{1+2/p})$	$\tilde{\mathcal{O}}(T^{2+2/p}/\epsilon^2)$
Taylor [3]	TI	$\text{Re}(\text{eig}(A)) \leq 0$	$\tilde{\mathcal{O}}(T \text{ poly log}(1/\epsilon))$	$\tilde{\mathcal{O}}(T \text{ poly log}(1/\epsilon))$
Improved Taylor [5]	TI	$\max_t \ e^{At}\ = \mathcal{O}(1)$	$\tilde{\mathcal{O}}(T \text{ poly log}(1/\epsilon))$	$\tilde{\mathcal{O}}(T \text{ poly log}(1/\epsilon))$
Spectral [4]	TD	$\text{Re}(\text{eig}(A(t))) \leq 0$	$\tilde{\mathcal{O}}(T \text{ poly log}(1/\epsilon))$	$\tilde{\mathcal{O}}(T \text{ poly log}(1/\epsilon))$
Dyson [6]	TD	$A(t) + A(t)^\dagger \leq 0$	$\tilde{\mathcal{O}}(T(\log(1/\epsilon))^2)$	$\tilde{\mathcal{O}}(T(\log(1/\epsilon))^2)$
Eigenvalue processing [8]	TI	$A + A^\dagger \leq 0$	/	$\mathcal{O}((T + \log(1/\epsilon)) \log(1/\epsilon))$
Time-marching [7]	TD	$A(t) + A(t)^\dagger \leq 0$	/	$\tilde{\mathcal{O}}(T^2 \log(1/\epsilon))$
Schrödingerization [11]	TD	$A(t) + A(t)^\dagger \leq 0$	/	$\tilde{\mathcal{O}}(T/\epsilon)$
LCHS [10]	TD	$A(t) + A(t)^\dagger \leq 0$	/	$\tilde{\mathcal{O}}(T(\log(1/\epsilon))^{2+o(1)})$
Fast-forwarded Taylor [20]	TI	$PA + A^\dagger P < 0$	$\tilde{\mathcal{O}}(T^{1/2}(\log(1/\epsilon))^2)$	$\tilde{\mathcal{O}}(T^{3/4}(\log(1/\epsilon))^2)$
Fast-forwarded Euler (this work)	TD	$A(t) + A(t)^\dagger < 0$	$\mathcal{O}(T^{1/2}/\epsilon)$	$\mathcal{O}(T^{1/2}/\epsilon)$
Fast-forwarded Dyson (this work)	TD	$A(t) + A(t)^\dagger < 0$	$\mathcal{O}(\log(T)(\log(1/\epsilon))^2)$	$\tilde{\mathcal{O}}(T^{1/2}(\log(1/\epsilon))^2)$

Table 1: A comparison among our fast-forwarded results and previous works. Here in the “Assumptions: Time” column, “TI” means “time-independent” (the algorithm only applies to ODEs with time-independent matrix $A(t) \equiv A$), and “TD” means “time-dependent” (the algorithm works for ODEs with possibly time-dependent matrix $A(t)$). The “Assumptions: Stability” column indicates further stability condition on $A(t)$ for the algorithm to be effective. In the query complexities, we only present explicit scalings in time T and error ϵ , and assume all the other parameters to be constants (e.g., the decay rate $\max_t \|u(t)\|/\|u(T)\|$, the norm $\max_t \|A(t)\|$, and parameters related to the stability condition).

Homogeneous ODEs: So far, all the presented results hold for possibly inhomogeneous ODEs. Here we consider the special homogeneous case where $b(t) \equiv 0$. Notice that the final state preparation problem for homogeneous dissipative ODEs always has exponential dependence on the evolution time T , since the solution $u(t)$ decays exponentially and extracting an exponentially small component from a quantum state requires exponential cost [19]. So we only consider the history state preparation and have the following result.

Result 4. Consider Eq. (1) with $A(t) + A(t)^\dagger \leq -2\eta < 0$ and $b(t) \equiv 0$. Then, for preparing the history state $\mathbf{u}_{\text{exact}}/\|\mathbf{u}_{\text{exact}}\|$ up to time T with error at most ϵ ,

1. the truncated Dyson series method has query complexity $\mathcal{O}\left((\log(\frac{1}{\epsilon}))^2\right)$,
2. the forward Euler method has query complexity $\mathcal{O}\left(\frac{1}{\epsilon} \log(\frac{1}{\epsilon})\right)$, and the trapezoidal rule has query complexity $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}} \log(\frac{1}{\epsilon})\right)$.

Result 4 suggests that we can completely get rid of any explicit time dependence for homogeneous dissipative ODEs, no matter which time discretization method we use. Intuitively, this is because the solution of a homogeneous dissipative ODE decays exponentially in time, so the non-trivial part in a history state is always within a constant time period.

1.3 Related works and comparison

As we discussed earlier, several generic quantum ODE algorithms have been proposed during the past decade. They can be roughly divided into two categories. The first category is the linear-system-based approach [1, 3, 4, 5, 6], which discretizes the dynamics in time, formulates the discretized ODE as a linear system of equations, and solves the linear system by quantum linear system solvers. The second category includes those that attempt to directly implement the time

evolution operator of Eq. (1) [7, 8, 9, 10, 11], and we call them evolution-based approach. Our algorithm in this work is almost the same as the linear-system-based approach only with a slight difference in the number of padding lines for final state preparation, and our contribution mainly lies in the improved complexity analysis.

In Table 1, we compare query complexities of our fast-forwarded Dyson and Euler methods with these generic quantum ODE algorithms. The first group in Table 1 is the linear-system-based approach for ODEs with time-independent coefficient matrix $A(t) \equiv A$, the second group is the linear-system-based approach for time-dependent $A(t)$, and the third group is the evolution-based approach. In previous works, it was not clearly stated how the evolution-based approach can efficiently prepare a history state, so we exclude them from the comparison of history state preparation. Notice that all the previous generic quantum ODE algorithms require additional stability conditions, specified in the third column of Table 1. Though technically different, these ensure the dynamics to be stable but not necessarily strictly dissipative, i.e., the homogeneous time evolution operator $\mathcal{T}e^{\int_0^t A(s)ds}$ does not increase asymptotically for long time. Our results require stronger stability condition that the dynamics needs to be strictly dissipative, but we can obtain improved time dependence. Compared with the best previous method, for history state preparation, our fast-forwarded results achieve an exponential improvement ($\mathcal{O}(\log(T))$ versus $\tilde{\mathcal{O}}(T)$), and for final state preparation, our fast-forwarded results achieve a quadratic speedup ($\tilde{\mathcal{O}}(\sqrt{T})$ versus $\tilde{\mathcal{O}}(T)$).

Other than generic algorithms, there have been a few attempts at fast-forwarding algorithms for special types of ODEs. An earlier work [20] also fast-forwards linear-system-based quantum ODE algorithms for dissipative ODEs by analyzing the condition number of the linear system. We compare our results with [20] in the last group of Table 1, and there are two main differences. First, the applicable regimes are different. [20] considers the time-independent matrix A and imposes the Lyapunov stability condition that there exists an invertible matrix P such that $PA + A^\dagger P < 0$. Our results generalize the fast-forwarding results to the time-dependent matrix $A(t)$ but require a more restrictive stability on the negative logarithmic norm. Second, within the commonly applicable regime, our results give a tighter upper bound of the linear system condition number and yield better complexity estimates in terms of T . Specifically, for history state preparation, we can achieve $\mathcal{O}(\log(T))$ complexity while [20] only has $\mathcal{O}(\sqrt{T})$. For final state preparation, our results also improve the complexity from $\tilde{\mathcal{O}}(T^{3/4})$ to $\tilde{\mathcal{O}}(T^{1/2})$. Besides [20], some other previous works also consider fast-forwarding quantum ODE algorithms but under different scenarios, such as [15, 16] for Hamiltonian simulation and [19] for imaginary time evolution.

1.4 Applications

We have discussed quantum algorithms with fast-forwarded complexity for linear dissipative ODEs. Such linear dissipative ODEs have many applications including quantum dynamics with non-Hermitian Hamiltonian, heat processes, damped systems, and linearized dissipative nonlinear differential equations.

In Section 6, we discuss two applications in more details. The first application is the dissipative quantum dynamics with non-Hermitian Hamiltonian, which is described by a time-dependent Schrödinger equation with a non-Hermitian matrix as the Hamiltonian and has appeared ubiquitously in various branches of physics and chemistry, such as open quantum system, quantum resonances, quantum transport, field theories, quantum many-body system, to name a few [21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]. The second application is the generalized heat equation with absorbing boundary, which is not only a fundamental model for heat processes but also useful for modeling hydrodynamical shocks in fluid dynamics, edge detection in image analysis, and option prices in finance [33, 34, 35, 36]. For both examples, we show that quantum algorithms can solve them with fast-forwarded complexity sub-linear in time.

1.5 Discussions and open questions

In this work, we show that quantum algorithms can prepare history states of dissipative ODEs with cost $\mathcal{O}(\log(T)(\log(1/\epsilon))^2)$, exponentially better than previous results, and can prepare final

states of dissipative ODEs with cost $\tilde{\mathcal{O}}\left(\sqrt{T}(\log(1/\epsilon))^2\right)$, achieving a quadratic speedup in T . A natural open question is whether one can achieve an even better time scaling for final state preparation or an $\Omega(\sqrt{T})$ lower bound can be proved. Another open question is to generalize our fast-forwarding results to more types of ODEs beyond dissipative ones. One possibility might be ODEs with a time-dependent version of Lyapunov stability condition $P(t)A(t) + A(t)^\dagger P(t) < 0$ for an invertible time-dependent matrix $P(t)$. However, notice that the solution of such an ODE might still grow linearly in the worst case as we may construct a piece-wise constant $A(t)$ by gluing time-independent Lyapunov stable matrices up to their transient growing periods. So we expect that it is only possible for fast-forwarding under further decay assumption on $P(t)$ to rule out such growth.

In our work, we focus on analyzing the linear-system-based approach for ODEs. It is interesting to further explore the possibility of fast-forwarding evolution-based approach as well, especially the best linear combination of Hamiltonian simulation (LCHS) algorithm in [10]. We refer interested readers to our follow-up work [37] for quantum dissipative ODE algorithms based on LCHS and the time-marching method [7] with fast-forwarded complexity.

Finally, we would like to remark that all the complexities in this work are query complexities, which are measured by the number of queries to the input model of the matrices (i.e., block-encoding) and the vectors (i.e., state preparation oracle). Although query complexities of all the algorithms do not explicitly depend on the dimension of the ODE, how the gate complexities scale in terms of the dimension depends on the efficiency of constructing the block-encoding and the state preparation oracle. For certain cases, these oracles can be efficiently constructed with only poly-logarithmic dimension dependence by known block-encoding techniques [38, 39, 40, 41, 42] and state preparation techniques [43, 44], and a further study on more general scenarios would be a practically important direction to fully unlock the power of our quantum dissipative ODE algorithms in a gate-based implementation.

1.6 Organization

The rest of the paper is organized as follows. We first start with discussing the stability condition for the dissipative ODEs in Section 2. Then, we state the quantum ODE algorithms for history state preparation and final state preparation in Section 3, and establish the framework of obtaining fast-forwarded complexity estimates in Section 4. In Section 5, we show in a more explicit way the complexity of truncated Dyson series methods and two lower-order methods for stable dynamics. We discuss applications of our algorithms in Section 6.

2 Dissipative ODEs and stability

This work focuses on fast-forwarding quantum algorithms for dissipative ODEs. The dissipative ODEs in this work refers to Eq. (1) where the logarithmic norm of $A(t)$ is uniformly negative. Specifically, we assume that there exists a positive number $\eta > 0$ such that

$$A(t) + A(t)^\dagger \leq -2\eta < 0. \quad (11)$$

An important feature of dissipative ODEs is that the homogeneous time evolution operator always decays exponentially in time. We show it in the following result and give its proof in Appendix A.

Lemma 5. *For any $0 \leq t_0 \leq t_1 \leq T$,*

1. *we have*

$$\left\| \mathcal{T}e^{\int_{t_0}^{t_1} A(s)ds} \right\| \leq e^{-\eta(t_1-t_0)}. \quad (12)$$

2. *for any quantum state $|v\rangle$, we have*

$$\left\| \mathcal{T}e^{\int_{t_0}^{t_1} A(s)ds} |v\rangle \right\| \geq e^{-(t_1-t_0) \max_t \|A(t)\|}. \quad (13)$$

Notice that here the “dissipative” condition is only imposed on the homogeneous time evolution operator. In the homogeneous case where $b(t) \equiv 0$, the solution of Eq. (1) can be written as $u(t) = \mathcal{T}e^{\int_0^t A(s)ds}u_0$, so Lemma 5 implies an exponential decay of $u(t)$ in time t . However, for general inhomogeneous case with non-trivial $b(t)$, the solution of Eq. (1) does not necessarily decay. In fact, according to Duhamel’s principle, the solution of Eq. (1) can be written as

$$u(t) = \mathcal{T}e^{\int_0^t A(s)ds}u_0 + \int_0^t \mathcal{T}e^{\int_\tau^t A(s)ds}b(\tau)d\tau, \quad (14)$$

and the second term on the right hand side does not necessarily decay. Nevertheless, Eq. (14) suggests that the inhomogeneous term $b(\tau)$ at a fixed time τ only affects the dynamics locally, since the evolution operator $\mathcal{T}e^{\int_\tau^t A(s)ds}$ has an exponential decay in time.

3 Quantum ODE algorithm

In this section, we discuss the main steps of the quantum ODE algorithm. Our algorithm is a unified framework of existing linear-system-based approach in [1, 3, 4, 5], which can be incorporated with any single-step time discretization scheme. We first present a general single-step time discretization approach for the ODE in Eq. (1), and then discuss our algorithms for history state preparation and final state preparation.

3.1 Time discretization

For time discretization, let us consider a single-step scheme of the form

$$L(j, h)u_{j+1} = R(j, h)u_j + v(j, h). \quad (15)$$

Here each u_j aims at approximating the solution $u(jh)$. L and R are two matrix-valued functions that might depend on the index j and the step size h . $v(j, h)$ is a vector-valued function for approximating the inhomogeneous term. For notational convenience, we will omit the explicit h dependence in the matrices and vectors and denote

$$L_j = L(j, h), \quad R_j = R(j, h), \quad v_j = v(j, h). \quad (16)$$

The simplest example of Eq. (15) is the forward Euler method, which is given by

$$\frac{u_{j+1} - u_j}{h} = A(jh)u_j + b(jh). \quad (17)$$

Observe that Eq. (17) corresponds to the form Eq. (15) with $L_j = I$, $R_j = I + hA(jh)$ and $v_j = hb(jh)$. Another example is the trapezoidal rule

$$\frac{u_{j+1} - u_j}{h} = \frac{1}{2}(A(jh)u_j + b(jh) + A((j+1)h)u_{j+1} + b((j+1)h)), \quad (18)$$

i.e.,

$$\left(I - \frac{h}{2}A((j+1)h)\right)u_{j+1} = \left(I + \frac{h}{2}A(jh)\right)u_j + \frac{h}{2}(b(jh) + b((j+1)h)). \quad (19)$$

which corresponds to the form Eq. (15) with

$$L_j = I - \frac{h}{2}A((j+1)h), \quad R_j = I + \frac{h}{2}A(jh), \quad v_j = \frac{h}{2}(b(jh) + b((j+1)h)). \quad (20)$$

The Dyson series method [6] truncated at order K can also be represented in the form of Eq. (15) with $L_j = I$,

$$R_j = \sum_{k=0}^K \int_{jh}^{(j+1)h} dt_1 \int_{jh}^{t_1} dt_2 \cdots \int_{jh}^{t_{k-1}} dt_k A(t_1)A(t_2) \cdots A(t_k), \quad (21)$$

and

$$v_j = \sum_{k=1}^K \int_{jh}^{(j+1)h} dt_1 \int_{jh}^{t_1} dt_2 \cdots \int_{jh}^{t_{k-1}} dt_k A(t_1) A(t_2) \cdots A(t_{k-1}) b(t_k). \quad (22)$$

Notice that even for a specific time discretization scheme, the choices of L , R , and v are not unique, as we may always multiply both sides of Eq. (15) by any matrix. We will choose proper L , R and v for ease of algorithmic implementation.

3.2 Algorithms

3.2.1 History state preparation

We first aim to obtain a history state proportional to

$$\mathbf{u}_{\text{exact}} = [u(0); u(h); u(2h); \cdots; u(Mh)] \quad (23)$$

encoding the information on all dynamics at $M + 1$ equidistant time steps. The algorithm is designed based on so-called the all-at-once system, which is a dilated linear system of equations

$$\mathbf{A}_{M,0} \mathbf{u} = \mathbf{b}_{M,0}. \quad (24)$$

Here

$$\mathbf{A}_{M,0} = \begin{pmatrix} I & & & & \\ -R_0 & L_0 & & & \\ & -R_1 & L_1 & & \\ & & \ddots & \ddots & \\ & & & -R_{M-1} & L_{M-1} \end{pmatrix}, \quad \mathbf{b}_{M,0} = \begin{pmatrix} u_0 \\ v_0 \\ v_1 \\ \vdots \\ v_{M-1} \end{pmatrix}. \quad (25)$$

We add two subscripts for $\mathbf{A}$ and $\mathbf{b}$. The first subscript indicates the overall number of time steps in the discrete evolution. The second subscript denotes the number of padding rows, whose meaning will be clear later. In the case of history state preparation, the second subscript is always 0.

A quantum algorithm for estimating a history state proportional to $\mathbf{u}_{\text{exact}} = [u(0); u(h); u(2h); \cdots; u(T)]$ can then be constructed as follows. We first construct a state preparation oracle of $\mathbf{b}_{M,0}$ and a block-encoding of $\mathbf{A}_{M,0}$, defined in Eq. (25). The state preparation oracle of $\mathbf{b}_{M,0}$ requires access to $|u_0\rangle$ and $b(s)$, and the block-encoding of $\mathbf{A}_{M,0}$ requires access to $A(t)$. Constructions of both oracles typically depend on specific numerical methods, and we will discuss some examples in Section 5. Then, with these input models of $\mathbf{A}_{M,0}$ and $\mathbf{b}_{M,0}$, we can apply the optimal quantum linear system algorithm [13, 18] to approximate $|\mathbf{u}\rangle = \mathbf{A}_{M,0}^{-1} \mathbf{b}_{M,0} / \|\mathbf{A}_{M,0}^{-1} \mathbf{b}_{M,0}\|$, which is the desired approximation of the history state.

3.2.2 Final state preparation

Now we discuss quantum algorithms for preparing the state $u(T)/\|u(T)\|$ only encoding the solution at the final time. A common strategy to obtain $u(T)/\|u(T)\|$ is to post select the history state $[u(0); u(h); \cdots; u(T)]$ obtained by the history state preparation algorithm. Notice that the history state can be written as (up to a normalization factor) $\sum_{j=0}^M \|u(jh)\| |j\rangle |u(jh)\rangle$, so if we measure the first ancilla register onto $|M\rangle$, then the system register will encode the desired final solution. However, the probability of getting M can be small as we typically use many time steps for accurate simulation. To boost such probability, we may use the padding trick, introduced in [1], to make multiple copies of the final solution in the history state.

Specifically, for a single step method, instead of solving the linear system defined by Eq. (25),

now we consider $\mathbf{A}_{M,M_p-1}\mathbf{u} = \mathbf{b}_{M,M_p-1}$ where

$$\mathbf{A}_{M,M_p-1} = \begin{pmatrix} I & & & & & & & & \\ -R_0 & L_0 & & & & & & & \\ & -R_1 & L_1 & & & & & & \\ & & \ddots & \ddots & & & & & \\ & & & -R_{M-1} & L_{M-1} & & & & \\ & & & & -I & I & & & \\ & & & & & -I & I & & \\ & & & & & & \ddots & \ddots & \\ & & & & & & & -I & I \end{pmatrix}, \quad (26)$$

$$\mathbf{b}_{M,M_p-1} = [u_0; v_0; v_1; \cdots; v_{M-1}; 0; 0; \cdots; 0]. \quad (27)$$

Here $\mathbf{A}_{M,M_p-1}$ is a dilation of that in Eq. (25) by having $(M_p - 1)$ many additional rows, of which each has an identity on the diagonal block and a negative identity on the sub-diagonal block. $\mathbf{b}_{M,M_p-1}$ is correspondingly dilated by appending zeros. The second subscript $M_p - 1$ denotes the number of the additional rows, where M_p is a positive integer. Notice that the case $M_p = 1$ implies no padding row and becomes the matrix Eq. (25) used in history state preparation. Then, the exact solution of the linear system $\mathbf{A}_{M,M_p-1}\mathbf{u} = \mathbf{b}_{M,M_p-1}$ becomes $\mathbf{u} = [u_0; u_1; u_2; \cdots; u_{M-1}; u_M; u_M; \cdots; u_M]$, which is the history solution (up to step $M - 1$) followed by M_p copies of the final solution u_M .

Our quantum algorithm for preparing the final solution first solves the linear system defined by Eq. (26) and Eq. (27) using optimal quantum linear system algorithm [13, 18], then measures the index register until success. The algorithm succeeds if the measurement outcome is larger than or equal to M . To further reduce the complexity quadratically, we may combine it with the amplitude amplification technique [45].

The choice of M_p heavily affects the complexity of the algorithm. If M_p is too small, then there will be too few copies of the final solution and thus small success probability. If M_p is too large, then the condition number of $\mathbf{A}_{M,M_p-1}$ will increase, and solving corresponding linear system of equations becomes expensive. Later we will present a complexity estimate for any value of M_p and any single-step method, and discuss optimal choices of M_p in specific schemes.

4 Fast-forwarded complexity

Here we establish our fast-forwarded complexity analysis for dissipative ODEs. First, for any value of M and M_p , we bound the condition number of the matrix $\mathbf{A}_{M,M_p-1}$ in the linear system Eq. (26), which is the dominant factor in quantum linear system algorithms. Then we estimate the complexity of history state preparation, and final state preparation separately. For history state preparation, we give a complexity estimate for general inhomogeneous ODEs. In the special case of homogeneous ODEs, although the general theorem also applies, by more careful analysis we may obtain a better complexity, which we will present separately. For final state preparation, we will only focus on the general inhomogeneous case, as the final solution of the homogeneous dissipative ODE always decays exponentially and becomes trivial.

4.1 Condition number

We estimate the condition number of $\mathbf{A}_{M,M_p-1}$ by bounding $\|\mathbf{A}_{M,M_p-1}\|$ and $\|\mathbf{A}_{M,M_p-1}^{-1}\|$. Our main technique is a bound of the spectral norm of a block matrix through the spectral norms of its blocks, which we state here and prove in Appendix B.

Lemma 6. *Let $\mathbf{M}$ be the block matrix as*

$$\mathbf{M} = \begin{pmatrix} M_{0,0} & \cdots & M_{0,n-1} \\ \vdots & & \vdots \\ M_{n-1,0} & \cdots & M_{n-1,n-1} \end{pmatrix} \quad (28)$$

where M_{ij} 's are d dimensional square matrices. Then

$$\|\mathbf{M}\| \leq \sqrt{\left(\max_j \sum_{k=0}^{n-1} \|M_{k,j}\|\right) \left(\max_i \sum_{k=0}^{n-1} \|M_{i,k}\|\right)}. \quad (29)$$

The spectral norm of $\mathbf{A}_{M,M_p-1}$ can be bounded straightforwardly from [Lemma 6](#). For $\mathbf{A}_{M,M_p-1}^{-1}$, we may first perform block version of Gaussian elimination to have an explicit form of $\mathbf{A}_{M,M_p-1}^{-1}$, then apply [Lemma 6](#) to bound its spectral norm.

Lemma 7. Let $A(t)$ be a matrix such that $A(t) + A(t)^\dagger \leq -2\eta < 0$, and $\mathbf{A}_{M,M_p-1}$ be the matrix defined in [Eq. \(26\)](#). Suppose that $\eta h \leq 1$ and the local truncation error is bounded as

$$\left\| L_j^{-1} R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \frac{1}{2} \eta h e^{-\eta h}, \quad \forall 0 \leq j \leq M-1. \quad (30)$$

Then, for any $M > T$, we have

$$\|\mathbf{A}_{M,M_p-1}\| \leq 2 + \max_j \|L_j\| + \max_j \|R_j\|, \quad (31)$$

$$\|\mathbf{A}_{M,M_p-1}^{-1}\| \leq \left(\frac{2eM}{\eta T} + M_p \right) \left(1 + \max_j \|L_j^{-1}\| \right). \quad (32)$$

Proof. The upper bound of $\|\mathbf{A}_{M,M_p-1}\|$ can be directly obtained by applying [Lemma 6](#) and noticing that the block row sums and column sums of $\|\mathbf{A}_{M,M_p-1}\|$ are both bounded by $2 + \max_j \|L_j\| + \max_j \|R_j\|$.

We focus on bounding $\|\mathbf{A}_{M,M_p-1}^{-1}\|$. For notation simplicity, let us denote

$$L_j = I, \quad j = -1 \text{ or } M \leq j \leq M + M_p - 2, \quad (33)$$

and

$$R_j = I, \quad M \leq j \leq M + M_p - 2. \quad (34)$$

Then we can simply write

$$\mathbf{A}_{M,M_p-1} = \begin{pmatrix} L_{-1} & & & & & \\ -R_0 & L_0 & & & & \\ & -R_1 & L_1 & & & \\ & & \ddots & \ddots & & \\ & & & -R_{M+M_p-2} & L_{M+M_p-2} \end{pmatrix}. \quad (35)$$

By Gaussian elimination, we have $\mathbf{A}_{M,M_p-1}^{-1} = (B_{i,j})$, where the block

$$B_{i,j} = \mathbf{1}_{i \geq j} \left(\prod_{l=j}^{i-1} L_l^{-1} R_l \right) L_{j-1}^{-1}. \quad (36)$$

To apply [Lemma 6](#), we first bound the spectral norm of the block $B_{i,j}$. Notice that, for any $0 \leq j \leq M-1$, according to [Eq. \(30\)](#) and [Lemma 5](#), we have

$$\|L_j^{-1} R_j\| \leq \left\| L_j^{-1} R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| + \left\| \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \frac{1}{2} \eta h e^{-\eta h} + e^{-\eta h} \leq e^{-\eta h/2}, \quad (37)$$

where the last inequality is because $1 + x \leq e^x$ for any $x \geq 0$. In other words, we may write

$$\|L_j^{-1} R_j\| \leq c^h, \quad 0 \leq j \leq M-1, \quad (38)$$

for a constant

$$c = e^{-\eta/2}. \quad (39)$$

Notice that $L_j^{-1} = R_j = I$ for $j \geq M$. Then

$$\|B_{i,j}\| \leq \mathbf{1}_{i \geq j} \left(\prod_{l=j}^{i-1} \|L_l^{-1} R_l\| \right) \left(1 + \max_l \|L_l^{-1}\| \right) \leq \begin{cases} c^{(i-j)h} (1 + \max_l \|L_l^{-1}\|), & M-1 \geq i \geq j, \\ c^{(M-j)h} (1 + \max_l \|L_l^{-1}\|), & i \geq M > j, \\ 1 + \max_l \|L_l^{-1}\|, & i \geq j \geq M, \\ 0, & i < j. \end{cases} \quad (40)$$

Using this upper bound of $\|B_{i,j}\|$, we may bound the block row sums of $\mathbf{A}_{M,M_p-1}^{-1}$ as

$$\sum_{k=0}^{M+M_p-1} \|B_{i,k}\| \leq \begin{cases} \sum_{k=0}^i c^{(i-k)h} (1 + \max_l \|L_l^{-1}\|), & i \leq M-1, \\ \left(\sum_{k=0}^M c^{(M-k)h} + i - M \right) (1 + \max_l \|L_l^{-1}\|), & i \geq M \end{cases} \quad (41)$$

$$\leq \begin{cases} \frac{1}{1-c^h} (1 + \max_l \|L_l^{-1}\|), & i \leq M-1, \\ \left(\frac{1}{1-c^h} + i - M \right) (1 + \max_l \|L_l^{-1}\|), & i \geq M \end{cases} \quad (42)$$

$$\leq \left(\frac{1}{1-c^h} + M_p \right) \left(1 + \max_l \|L_l^{-1}\| \right), \quad (43)$$

and the block column sums as

$$\sum_{k=0}^{M+M_p-1} \|B_{k,j}\| \leq \begin{cases} \left(\sum_{k=j}^{M-1} c^{(k-j)h} + M_p c^{(M-j)h} \right) (1 + \max_l \|L_l^{-1}\|), & j \leq M-1, \\ (M + M_p - j) (1 + \max_l \|L_l^{-1}\|), & j \geq M \end{cases} \quad (44)$$

$$\leq \begin{cases} \left(\frac{1}{1-c^h} + M_p \right) (1 + \max_l \|L_l^{-1}\|), & j \leq M-1, \\ (M + M_p - j) (1 + \max_l \|L_l^{-1}\|), & j \geq M \end{cases} \quad (45)$$

$$\leq \left(\frac{1}{1-c^h} + M_p \right) \left(1 + \max_l \|L_l^{-1}\| \right). \quad (46)$$

According to [Lemma 6](#), we have

$$\|\mathbf{A}_{M,M_p-1}^{-1}\| \leq \left(\frac{1}{1-c^h} + M_p \right) \left(1 + \max_l \|L_l^{-1}\| \right). \quad (47)$$

Notice that $1 - c^h = 1 - e^{-\eta h/2} \geq \eta h/(2e)$ since the inequality $1 - e^x \geq -x/e$ holds for $x \in [-1, 0]$. Thus,

$$\|\mathbf{A}_{M,M_p-1}^{-1}\| \leq \left(\frac{2e}{\eta h} + M_p \right) \left(1 + \max_l \|L_l^{-1}\| \right) = \left(\frac{2e}{\eta} \frac{M}{T} + M_p \right) \left(1 + \max_l \|L_l^{-1}\| \right). \quad (48)$$

□

4.2 Complexity of history state preparation

In the history state preparation, we choose $M_p = 1$. We first derive quantum complexity in the general inhomogeneous case, then present a simpler version only applicable to the homogeneous case.

4.2.1 General inhomogeneous case

Theorem 8. Consider the ODE in [Eq. \(1\)](#) such that $A(t) + A(t)^\dagger \leq -2\eta < 0$. Let $\epsilon > 0$ be the target error, $T \geq (\max \|A(t)\|)^{-1}$ be the evolution time, and M be the number of time steps. Suppose that we use the single-step method in [Eq. \(15\)](#) for its time discretization, the time step size $h = T/M$ is chosen such that $\eta h \leq 1$ and for all $0 \leq j \leq T/h - 1$,

$$\left\| L_j^{-1} R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \min \left\{ \frac{1}{2} \eta h e^{-\eta h}, \frac{\eta^{3/2} h \epsilon}{144 \sqrt{2} \sqrt{1 + \frac{T \max \|b(t)\|^2}{\eta \|u_0\|^2}} \sqrt{\max \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}} \right\}, \quad (49)$$

and

$$\left\| L_j^{-1} v_j - \int_{jh}^{(j+1)h} \mathcal{T} e^{\int_s^{(j+1)h} A(\tau) d\tau} b(s) ds \right\| \leq \frac{\|u_0\| \eta h \epsilon}{72 \sqrt{2} \sqrt{T} \sqrt{\max \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}}. \quad (50)$$

Then, we can obtain an ϵ -approximation of the history state $\mathbf{u}_{exact}/\|\mathbf{u}_{exact}\|$, using

$$\mathcal{O}\left(\left(1 + \max_j \|L_j\| + \max_j \|R_j\|\right) \left(1 + \max_j \|L_j^{-1}\|\right) \frac{M}{\eta T} \log\left(\frac{1}{\epsilon}\right)\right) \quad (51)$$

queries to the block-encoding of $\mathbf{A}_{M,0}$ and the state preparation oracle of $|\mathbf{b}_{M,0}\rangle$.

Proof. We first show that, under the assumptions in the statement of the theorem, the exact normalized solution of the linear system $\mathbf{A}_{M,0} \mathbf{u} = \mathbf{b}_{M,0}$ is an $\epsilon/2$ -approximation of the ideal history state.

Let $M = T/h$, $\mathbf{u} = \mathbf{A}_{M,0}^{-1} \mathbf{b}_{M,0}$ and $|\mathbf{u}\rangle = \mathbf{u}/\|\mathbf{u}\|$. We denote $L_{-1} = I$. Let

$$\epsilon_1 = \frac{\eta^{3/2} h \epsilon}{144 \sqrt{2} \sqrt{1 + \frac{T \max \|b\|^2}{\eta \|u_0\|^2}} \sqrt{\max \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}}, \quad \epsilon_2 = \frac{\|u_0\| \eta h \epsilon}{72 \sqrt{2} \sqrt{T} \sqrt{\max \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}}. \quad (52)$$

Then the assumptions on the local operators become

$$\left\| L_j^{-1} R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \epsilon_1, \quad (53)$$

and

$$\left\| L_j^{-1} v_j - \int_{jh}^{(j+1)h} \mathcal{T} e^{\int_s^{(j+1)h} A(\tau) d\tau} b(s) ds \right\| \leq \epsilon_2. \quad (54)$$

The numerical scheme becomes

$$u_{j+1} = L_j^{-1} R_j u_j + L_j^{-1} v_j = P_j u_j + L_j^{-1} v_j, \quad (55)$$

where we denote

$$P_j = L_j^{-1} R_j. \quad (56)$$

Thus, applying Eq. (55) iteratively, we have

$$u_k = \left(\prod_{j=0}^{k-1} P_j \right) u_0 + \sum_{j=0}^{k-1} \left(\prod_{l=j+1}^{k-1} P_l \right) L_j^{-1} v_j. \quad (57)$$

Similarly, for the exact solution $u(t)$, by Duhamel's principle we have

$$u((j+1)h) = \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} u(jh) + \int_{jh}^{(j+1)h} \mathcal{T} e^{\int_s^{(j+1)h} A(\tau) d\tau} b(s) ds = Q_j u(jh) + w_j, \quad (58)$$

where we denote

$$Q_j = \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds}, \quad w_j = \int_{jh}^{(j+1)h} \mathcal{T} e^{\int_s^{(j+1)h} A(\tau) d\tau} b(s) ds. \quad (59)$$

Hence, we have

$$u(kh) = \left(\prod_{j=0}^{k-1} Q_j \right) u_0 + \sum_{j=0}^{k-1} \left(\prod_{l=j+1}^{k-1} Q_l \right) w_j. \quad (60)$$

Subtracting Eq. (60) from Eq. (57) and using triangle inequality, we have

$$\|u_k - u(kh)\| \leq \left\| \prod_{j=0}^{k-1} P_j - \prod_{j=0}^{k-1} Q_j \right\| \|u_0\| + \sum_{j=0}^{k-1} \left\| \left(\prod_{l=j+1}^{k-1} P_l \right) L_j^{-1} v_j - \left(\prod_{l=j+1}^{k-1} Q_l \right) w_j \right\|. \quad (61)$$

For the first term in the right-hand side of Eq. (60), we use the triangle inequality to write

$$\left\| \prod_{j=0}^{k-1} P_j - \prod_{j=0}^{k-1} Q_j \right\| \leq \left\| \sum_{l=0}^{k-1} \left[\left(\prod_{j=l+1}^{k-1} P_j \right) (P_l - Q_l) \left(\prod_{j=0}^{l-1} Q_j \right) \right] \right\| \quad (62)$$

$$\leq \sum_{l=0}^{k-1} \left[\left(\prod_{j=l+1}^{k-1} \|P_j\| \right) \|P_l - Q_l\| \left(\prod_{j=0}^{l-1} \|Q_j\| \right) \right] \quad (63)$$

$$\leq \epsilon_1 \sum_{l=0}^{k-1} \left(\prod_{j=l+1}^{k-1} \|P_j\| \right) \left(\prod_{j=0}^{l-1} \|Q_j\| \right). \quad (64)$$

By Lemma 5 and the proof of Lemma 7, we have $\|Q_j\| \leq e^{-\eta h} \leq e^{-\eta h/2}$ and $\|P_j\| \leq e^{-\eta h/2}$, and thus

$$\left\| \prod_{j=0}^{k-1} P_j - \prod_{j=0}^{k-1} Q_j \right\| \leq k\epsilon_1 e^{-\eta(k-1)h/2}. \quad (65)$$

For the second term in the right-hand side of Eq. (61), we again use the triangle inequality to obtain (the following derivation assumes $j \leq k-2$, but the final conclusion also holds true for $j = k-1$)

$$\left\| \left(\prod_{l=j+1}^{k-1} P_l \right) L_j^{-1} v_j - \left(\prod_{l=j+1}^{k-1} Q_l \right) w_j \right\| \quad (66)$$

$$\leq \left\| \left(\prod_{l=j+1}^{k-1} P_l - \prod_{l=j+1}^{k-1} Q_l \right) w_j \right\| + \left\| \left(\prod_{l=j+1}^{k-1} P_l \right) (L_j^{-1} v_j - w_j) \right\| \quad (67)$$

$$\leq \left\| \prod_{l=j+1}^{k-1} P_l - \prod_{l=j+1}^{k-1} Q_l \right\| \|w_j\| + \left(\prod_{l=j+1}^{k-1} \|P_l\| \right) \|L_j^{-1} v_j - w_j\|. \quad (68)$$

Using $\|P_l\| \leq e^{-\eta h/2}$, $\|w_j\| \leq h \max \|b(t)\|$ from the definition of w_j , and $\left\| \prod_{l=j+1}^{k-1} P_l - \prod_{l=j+1}^{k-1} Q_l \right\| \leq (k-j-1)\epsilon_1 e^{-\eta(k-j-2)h/2}$ which can be obtained by the same proof of Eq. (65), we have

$$\left\| \left(\prod_{l=j+1}^{k-1} P_l \right) L_j^{-1} v_j - \left(\prod_{l=j+1}^{k-1} Q_l \right) w_j \right\| \leq (k-j-1)\epsilon_1 e^{-\eta(k-j-2)h/2} h \max \|b(t)\| + e^{-\eta(k-j-1)h/2} \epsilon_2. \quad (69)$$

Plugging Eqs. (65) and (69) back to Eq. (61), we have

$$\|u_k - u(kh)\| \quad (70)$$

$$\leq k\epsilon_1 e^{-\eta(k-1)h/2} \|u_0\| + \sum_{j=0}^{k-1} (k-j-1)\epsilon_1 e^{-\eta(k-j-2)h/2} h \max \|b(t)\| + \sum_{j=0}^{k-1} e^{-\eta(k-j-1)h/2} \epsilon_2 \quad (71)$$

$$= k\epsilon_1 e^{-\eta(k-1)h/2} \|u_0\| + \epsilon_1 h \max \|b(t)\| \frac{1 - e^{-\eta kh/2} - k e^{-\eta(k-1)h/2} (1 - e^{-\eta h/2})}{(1 - e^{-\eta h/2})^2} + \epsilon_2 \frac{1 - e^{-\eta kh/2}}{1 - e^{-\eta h/2}} \quad (72)$$

$$\leq k\epsilon_1 e^{-\eta(k-1)h/2} \|u_0\| + \frac{\epsilon_1 h \max \|b(t)\|}{(1 - e^{-\eta h/2})^2} + \frac{\epsilon_2}{1 - e^{-\eta h/2}}. \quad (73)$$

Then the error in the unnormalized history vector can be bounded as

$$\|\mathbf{u} - \mathbf{u}_{\text{exact}}\| = \sqrt{\sum_{k=1}^M \|u_k - u(kh)\|^2} \quad (74)$$

$$\leq \sqrt{3} \sqrt{\sum_{k=1}^M k^2 \epsilon_1^2 e^{-\eta(k-1)h} \|u_0\|^2 + \sum_{k=1}^M \frac{\epsilon_1^2 h^2 \max \|b(t)\|^2}{(1 - e^{-\eta h/2})^4} + \sum_{k=1}^M \frac{\epsilon_2^2}{(1 - e^{-\eta h/2})^2}} \quad (75)$$

$$= \sqrt{3} \sqrt{\sum_{k=1}^M k^2 \epsilon_1^2 e^{-\eta(k-1)h} \|u_0\|^2 + \frac{\epsilon_1^2 T h \max \|b(t)\|^2}{(1 - e^{-\eta h/2})^4} + \frac{\epsilon_2^2 (T/h)}{(1 - e^{-\eta h/2})^2}}. \quad (76)$$

Notice that

$$\sum_{k=1}^M k^2 e^{-\eta(k-1)h} = \frac{(2M^2 + 2M - 1)e^{-\eta(M+1)h} - M^2 e^{-\eta(M+2)h} - (M+1)^2 e^{-\eta Mh} + e^{-\eta h} + 1}{(1 - e^{-\eta h})^3} \quad (77)$$

$$= \frac{1 + e^{-\eta h}}{(1 - e^{-\eta h})^3} - \frac{M^2 e^{-\eta Mh} (1 - e^{-\eta h})^2}{(1 - e^{-\eta h})^3} - \frac{2M e^{-\eta Mh} (1 - e^{-\eta h})}{(1 - e^{-\eta h})^3} - \frac{e^{-\eta Mh} (1 + e^{-\eta h})}{(1 - e^{-\eta h})^3} \quad (78)$$

$$\leq \frac{2}{(1 - e^{-\eta h})^3}, \quad (79)$$

so, together with the inequality $1 - e^{-x} \geq x/2$ for $x \in [0, 1]$, we have

$$\|\mathbf{u} - \mathbf{u}_{\text{exact}}\| \leq \sqrt{3} \sqrt{\frac{2\epsilon_1^2 \|u_0\|^2}{(1 - e^{-\eta h})^3} + \frac{\epsilon_1^2 T h \max \|b(t)\|^2}{(1 - e^{-\eta h/2})^4} + \frac{\epsilon_2^2 (T/h)}{(1 - e^{-\eta h/2})^2}} \quad (80)$$

$$\leq \sqrt{\frac{48\epsilon_1^2 \|u_0\|^2}{\eta^3 h^3} + \frac{48\epsilon_1^2 T \max \|b(t)\|^2}{\eta^4 h^3} + \frac{12\epsilon_2^2 T}{\eta^2 h^3}}. \quad (81)$$

The error in the (normalized) quantum states is bounded as

$$\| |\mathbf{u}\rangle - |\mathbf{u}_{\text{exact}}\rangle \| \leq \frac{2}{\|\mathbf{u}_{\text{exact}}\|} \|\mathbf{u} - \mathbf{u}_{\text{exact}}\|, \quad (82)$$

so we need to find a lower bound of $\|\mathbf{u}_{\text{exact}}\|$. To this end, we will find a lower bound of the number of $u(kh)$ such that its norm is bounded from below by a constant. Specifically, by Duhamel's principle and [Lemma 5](#), we have

$$\|u(kh)\| = \left\| \mathcal{T} e^{\int_0^{kh} A(s) ds} u_0 + \int_0^{kh} \mathcal{T} e^{\int_s^{kh} A(\tau) d\tau} b(s) ds \right\| \quad (83)$$

$$\geq \left\| \mathcal{T} e^{\int_0^{kh} A(s) ds} u_0 \right\| - \left\| \int_0^{kh} \mathcal{T} e^{\int_s^{kh} A(\tau) d\tau} b(s) ds \right\| \quad (84)$$

$$\geq e^{-kh \max \|A(t)\|} \|u_0\| - kh \max \|b(t)\|. \quad (85)$$

Notice that for $k \leq \frac{1}{3h \max\{\max \|A(t)\|, \max \|b(t)\|/\|u_0\|\}}$, we always have $kh \leq \frac{\log(3/2)}{\max \|A(t)\|}$ and $kh \leq \frac{\|u_0\|}{3 \max \|b(t)\|}$, which imply $e^{-kh \max \|A(t)\|} \geq 2/3$ and $kh \max \|b(t)\| \leq \|u_0\|/3$, and thus $\|u(kh)\| \geq \|u_0\|/3$. Then

$$\|\mathbf{u}_{\text{exact}}\| = \sqrt{\sum_{k=0}^M \|u(kh)\|^2} \geq \sqrt{\frac{\|u_0\|^2}{9} \frac{1}{3h \max\{\max \|A(t)\|, \max \|b(t)\|/\|u_0\|\}}} \quad (86)$$

$$= \frac{\|u_0\|}{3\sqrt{3}} \frac{1}{\sqrt{h \max\{\max \|A(t)\|, \max \|b(t)\|/\|u_0\|\}}}. \quad (87)$$

Plugging Eqs. (81) and (87) back to Eq. (82), we have

$$\| |\mathbf{u}\rangle - |\mathbf{u}_{\text{exact}}\rangle \| \leq 36 \sqrt{\max \left\{ \max \|A(t)\|, \max \frac{\|b(t)\|}{\|u_0\|} \right\}} \sqrt{\frac{4\epsilon_1^2}{\eta^3 h^2} + \frac{4\epsilon_1^2 T \max \|b(t)\|^2}{\eta^4 h^2 \|u_0\|^2} + \frac{\epsilon_2^2 T}{\eta^2 h^2 \|u_0\|^2}}, \quad (88)$$

which is further upper bounded by $\epsilon/2$ by the choices of ϵ_1 and ϵ_2 in Eq. (52).

Then, solving the linear system of equations $\mathbf{A}_{M,0} \mathbf{u} = \mathbf{b}_{M,0}$ up to error $\epsilon/2$ gives the ϵ -approximation of $|\mathbf{u}_{\text{exact}}\rangle$. According to [13], it requires $\mathcal{O}(\kappa \log(1/\epsilon))$ queries to the block-encoding of $\mathbf{A}_{M,0}$ and the state preparation oracle of $\mathbf{b}_{M,0}$, where $\kappa = \|\mathbf{A}_{M,0}\| \|\mathbf{A}_{M,0}^{-1}\|$ is the condition number of $\mathbf{A}_{M,0}$. According to Lemma 7, we have

$$\kappa = \mathcal{O} \left(\left(1 + \max_j \|L_j\| + \max_j \|R_j\| \right) \left(1 + \max_j \|L_j^{-1}\| \right) \frac{M}{\eta T} \right) \quad (89)$$

and complete the proof. $\square$

4.2.2 Homogeneous case

Now we consider the special homogeneous case where $b(t) \equiv 0$. Notice that Theorem 8 directly applies to the homogeneous case by letting $b(t) = 0$ in the complexity. However, we can obtain a better homogeneous complexity estimate by leveraging tighter discretization error bound. We show it in the next result and give its proof in Appendix C.

Theorem 9. Consider the ODE in Eq. (1) with $b(t) \equiv 0$ and $A(t) + A(t)^\dagger \leq -2\eta < 0$. Let $\epsilon > 0$ be the target error, $T \geq (\max \|A(t)\|)^{-1}$ be the evolution time, and M be the number of time steps. Suppose we use the single step method in Eq. (15) for its time discretization, the time step size $h = T/M$ is chosen such that $\eta h \leq 1$ and

$$\left\| L_j^{-1} R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \min \left\{ \frac{1}{2} \eta h e^{-\eta h}, \frac{\eta^{3/2} h \epsilon}{32 \sqrt{\max \|A(t)\|}} \right\}, \quad 0 \leq j \leq T/h - 1. \quad (90)$$

Then, we can obtain an ϵ -approximation of the history state $\mathbf{u}_{\text{exact}}/\|\mathbf{u}_{\text{exact}}\|$, using

$$\mathcal{O} \left(\left(1 + \max_j \|L_j\| + \max_j \|R_j\| \right) \left(1 + \max_j \|L_j^{-1}\| \right) \frac{M}{\eta T} \log \left(\frac{1}{\epsilon} \right) \right) \quad (91)$$

queries to the block-encoding of $\mathbf{A}_{M,0}$ and the state preparation oracle of $|u_0\rangle$.

4.3 Complexity of final state preparation

For the final state preparation, there are two components of the complexity: the cost of solving linear system of equations $\mathbf{A}_{M,M_p-1} \mathbf{u} = \mathbf{b}_{M,M_p-1}$, and the probability of measuring the ancilla register onto the correct index. As we discussed earlier, the choice of M_p heavily affects the overall complexity. Here we first present a complexity estimate for any choice of M_p .

Theorem 10. Consider the ODE in Eq. (1) such that $A(t) + A(t)^\dagger \leq -2\eta < 0$. Let $\epsilon > 0$ be the target error, $T \geq (\max \|A(t)\|)^{-1}$ be the evolution time, M be the number of time steps, and M_p be the number of copies of the final state in the linear system of equations. Suppose we use the single step method in Eq. (15) for its time discretization, the time step size $h = T/M$ is chosen such that $\eta h \leq 1$ and for all $0 \leq j \leq T/h - 1$,

$$\left\| L_j^{-1} R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \min \left\{ \frac{1}{2} \eta h e^{-\eta h}, \frac{\|u(T)\|}{\|u_0\| + \max \|b(t)\|/\eta} \frac{\eta h \epsilon}{128} \right\}, \quad (92)$$

and

$$\left\| L_j^{-1} v_j - \int_{jh}^{(j+1)h} \mathcal{T} e^{\int_s^{(j+1)h} A(\tau) d\tau} b(s) ds \right\| \leq \frac{\|u(T)\| \eta h \epsilon}{32}. \quad (93)$$

Then, we can obtain an ϵ -approximation of the final state $u(T)/\|u(T)\|$, using

$$\tilde{\mathcal{O}}\left(\left(1 + \max_j \|L_j\| + \max_j \|R_j\|\right)\left(1 + \max_j \|L_j^{-1}\|\right)\frac{\max_t \|u(t)\|}{\|u(T)\|}\sqrt{\frac{M + M_p}{M_p}}\left(\frac{M}{\eta T} + M_p\right)\log\left(\frac{1}{\epsilon}\right)\right) \quad (94)$$

queries to the block-encoding of $\mathbf{A}_{M,M_p-1}$ and the state preparation oracle of $|\mathbf{b}_{M,M_p-1}\rangle$.

Proof. Let $M = T/h$ and

$$\epsilon_1 = \frac{\|u(T)\|}{\|u_0\| + \max \|b(t)\|/\eta} \frac{\eta h \epsilon}{128}, \quad \epsilon_2 = \frac{\|u(T)\| \eta h \epsilon}{32}. \quad (95)$$

Then the assumptions on the local operators become

$$\left\|L_j^{-1}R_j - \mathcal{T}e^{\int_{jh}^{(j+1)h} A(s)ds}\right\| \leq \epsilon_1, \quad (96)$$

and

$$\left\|L_j^{-1}v_j - \int_{jh}^{(j+1)h} \mathcal{T}e^{\int_s^{(j+1)h} A(\tau)d\tau} b(s)ds\right\| \leq \epsilon_2. \quad (97)$$

In the proof of [Theorem 8](#), we show that the time discretization error at step k can be bounded as

$$\|u_k - u(kh)\| \leq k\epsilon_1 e^{-\eta(k-1)h/2} \|u_0\| + \frac{\epsilon_1 h \max \|b(t)\|}{(1 - e^{-\eta h/2})^2} + \frac{\epsilon_2}{1 - e^{-\eta h/2}}. \quad (98)$$

Using the facts that $1 - e^{-x} \geq x/2$ for $x \in [0, 1]$ and the function $xe^{-\eta(x-1)h/2}$ attains its maximum at $x = 2/(\eta h)$, we can further bound the time discretization error as

$$\|u_k - u(kh)\| \leq \frac{2\|u_0\|}{\sqrt{e}} \frac{\epsilon_1}{\eta h} + \frac{16\epsilon_1 \max \|b(t)\|}{\eta^2 h} + \frac{4\epsilon_2}{\eta h} \quad (99)$$

$$\leq \frac{16\epsilon_1}{\eta h} \left(\|u_0\| + \frac{\max \|b(t)\|}{\eta}\right) + \frac{4\epsilon_2}{\eta h}. \quad (100)$$

We can verify that the choices of ϵ_1 and ϵ_2 in [Eq. \(95\)](#) ensures

$$\|u_k - u(kh)\| \leq \frac{\|u(T)\| \epsilon}{4}. \quad (101)$$

[Eq. \(101\)](#) also holds for $k = M$. Then the error in the normalized solutions becomes

$$\left\|\frac{u_M}{\|u_M\|} - \frac{u(T)}{\|u(T)\|}\right\| \leq \frac{2}{\|u(T)\|} \|u_M - u(T)\| \leq \frac{\epsilon}{2}. \quad (102)$$

Now let us determine the required accuracy in solving $\mathbf{A}_{M,M_p-1}\mathbf{u} = \mathbf{b}_{M,M_p-1}$. Let $\mathbf{u} = \mathbf{A}_{M,M_p-1}^{-1}\mathbf{b}_{M,M_p-1}$ and $|\mathbf{u}'\rangle$ be the state produced by a quantum linear system algorithm up to error ϵ' , in the sense that

$$\| |\mathbf{u}'\rangle - \mathbf{u} / \|\mathbf{u}\| \| \leq \epsilon'. \quad (103)$$

Let $|\mathbf{u}'\rangle = [u'_0; u'_1; \dots; u'_{M+M_p-1}]$ be partitioned in the same way as $\mathbf{u}$. Then for any $k \geq M$, we have

$$\|u'_k - u_M / \|\mathbf{u}\| \| \leq \epsilon'. \quad (104)$$

Upon a successful measurement, we would get the state $|u'_k\rangle$, which satisfies

$$\| |u'_k\rangle - |u_M\rangle \| = \left\| \frac{u'_k}{\|u'_k\|} - \frac{u_M / \|\mathbf{u}\|}{\|u_M / \|\mathbf{u}\|\|} \right\| \leq \frac{2\|\mathbf{u}\|}{\|u_M\|} \|u'_k - u_M / \|\mathbf{u}\|\| \leq \frac{2\|\mathbf{u}\|}{\|u_M\|} \epsilon'. \quad (105)$$

Using [Eq. \(101\)](#) and triangle inequality, we have, for any $0 \leq k \leq M$,

$$\|u(kh)\| - \frac{\|u(T)\| \epsilon}{4} \leq \|u_k\| \leq \max_t \|u(t)\| + \frac{\|u(T)\| \epsilon}{4}, \quad (106)$$

so

$$\|u_M\| \geq \|u(T)\|(1 - \epsilon/4), \quad (107)$$

and

$$\begin{aligned} \|\mathbf{u}\| &\leq \sqrt{M + M_p} \max_k \|u_k\| \leq \sqrt{M + M_p} \left(\max_t \|u(t)\| + \frac{\|u(T)\|\epsilon}{4} \right) \\ &\leq \sqrt{M + M_p} \max_t \|u(t)\| \left(1 + \frac{\epsilon}{4} \right). \end{aligned} \quad (108)$$

Then Eq. (105) can be further bounded as

$$\| |u'_k\rangle - |u_M\rangle \| \leq \frac{2\sqrt{M + M_p} \max_t \|u(t)\| (1 + \epsilon/4)}{\|u(T)\|(1 - \epsilon/4)} \epsilon' \leq \frac{4\sqrt{M + M_p} \max_t \|u(t)\|}{\|u(T)\|} \epsilon'. \quad (109)$$

In order to bound this by $\epsilon/2$, it suffices to choose

$$\epsilon' = \frac{\|u(T)\|}{8\sqrt{M + M_p} \max_t \|u(t)\|} \epsilon. \quad (110)$$

Then

$$\| |u'_k\rangle - |u_M\rangle \| \leq \frac{\epsilon}{2}. \quad (111)$$

We have shown in Eqs. (102) and (111) that, by solving the linear system up to ϵ' defined in Eq. (110) and successfully measuring the index register, the produced quantum state is an ϵ -approximation of the exact final state $|u(T)\rangle$. According to [13], solving the linear system requires $\mathcal{O}(\kappa \log(1/\epsilon'))$ queries to the block-encoding of $\mathbf{A}_{M, M_p-1}$ and the state preparation oracle of $\mathbf{b}_{M, M_p-1}$, where $\kappa = \|\mathbf{A}_{M, M_p-1}\| \|\mathbf{A}_{M, M_p-1}^{-1}\|$ is the condition number of $\mathbf{A}_{M, M_p-1}$. According to Lemma 7, we have

$$\kappa = \mathcal{O} \left(\left(1 + \max_j \|L_j\| + \max_j \|R_j\| \right) \left(1 + \max_j \|L_j^{-1}\| \right) \left(\frac{M}{\eta T} + M_p \right) \right). \quad (112)$$

So the query complexity in each run of the algorithm before measurement is

$$\mathcal{O} \left(\left(1 + \max_j \|L_j\| + \max_j \|R_j\| \right) \left(1 + \max_j \|L_j^{-1}\| \right) \left(\frac{M}{\eta T} + M_p \right) \log \left(\frac{(M + M_p) \max_t \|u(t)\|}{\|u(T)\|\epsilon} \right) \right). \quad (113)$$

Now we estimate the number of repeats of solving the linear system. The successful case is when the measurement outcome $k \geq M$. To boost the successful probability to $\Omega(1)$ with amplitude amplification, we need $\mathcal{O}(1/\sqrt{\sum_{k=M}^{M+M_p-1} \|u'_k\|^2})$ rounds. For each $k \geq M$, according to Eqs. (104), (107) and (108) and the definition of ϵ' in Eq. (110), we can bound

$$\begin{aligned} \|u'_k\| &\geq \frac{\|u_M\|}{\|\mathbf{u}\|} - \epsilon' \geq \frac{\|u(T)\|(1 - \epsilon/4)}{\sqrt{M + M_p} \max_t \|u(t)\|(1 + \epsilon/4)} - \frac{\|u(T)\|}{8\sqrt{M + M_p} \max_t \|u(t)\|} \epsilon \\ &\geq \frac{\|u(T)\|}{4\sqrt{M + M_p} \max_t \|u(t)\|}. \end{aligned} \quad (114)$$

So the number of amplitude amplification rounds becomes

$$\mathcal{O} \left(\frac{1}{\sqrt{\sum_{k=M}^{M+M_p-1} \|u'_k\|^2}} \right) = \mathcal{O} \left(\frac{\max_t \|u(t)\|}{\|u(T)\|} \sqrt{\frac{M + M_p}{M_p}} \right), \quad (115)$$

which contributes to another multiplicative factor in the overall complexity and completes the proof. $\square$

In [Theorem 10](#), we can see two opposite effects of the parameter M_p . The factor $\sqrt{\frac{M+M_p}{M_p}}$ is due to the amplitude amplification, and will decrease as we choose larger M_p to have more copies of the final solution. However, larger M_p will increase the condition number of the linear system and thus the complexity of solving it, as shown in the factor $\frac{M}{\eta T} + M_p$. The following result shows the optimal choice of M_p to best balance these two effects for large T .

Corollary 11. *Under the same conditions as [Theorem 10](#) and by choosing*

$$M_p = \lceil \frac{M}{\eta T} \rceil, \quad (116)$$

we can obtain an ϵ -approximation of the final state $u(T)/\|u(T)\|$, using

$$\tilde{O}\left(\left(1 + \max_j \|L_j\| + \max_j \|R_j\|\right)\left(1 + \max_j \|L_j^{-1}\|\right) \frac{\max_t \|u(t)\|}{\|u(T)\|} \frac{M\sqrt{1+\eta T}}{\eta T} \log\left(\frac{1}{\epsilon}\right)\right) \quad (117)$$

queries to the block-encoding of $\mathbf{A}_{M,M_p-1}$ and the state preparation oracle of $|\mathbf{b}_{M,M_p-1}\rangle$.

Proof. It is a straightforward consequence of [Theorem 10](#) by noticing that $f(x) = \sqrt{\frac{M+x}{x}}\left(\frac{M}{\eta T} + x\right)$ achieves its minimum at $x = \frac{2M}{\eta T(1+\sqrt{1+8/(\eta T)})}$. $\square$

5 Specific numerical methods

We have established a complexity estimate for preparing a history state and a final state of a dissipative ODE in [Section 4](#). Notice that the results in [Section 4](#) work for any single-step numerical method, yet the overall complexity in terms of T and ϵ still remains unclear as the number of time steps M still has implicit time and precision dependence. Here we discuss several examples of numerical methods and present their complexity estimates in a more explicit way. Examples include the state-of-the-art truncated Dyson series method, and two lower-order methods.

Throughout this section, we assume the state preparation oracle O_{u_0} for $|u_0\rangle$ such that

$$O_{u_0} |0\rangle = |u_0\rangle, \quad (118)$$

the input model O_A for $A(t)$ such that

$$O_A |0\rangle |t\rangle |\psi\rangle = \frac{1}{\alpha_A} |0\rangle |t\rangle A(t) |\psi\rangle + |\perp\rangle, \quad (119)$$

and the input model O_b for $b(t)$ such that

$$O_b |0\rangle |t\rangle |0\rangle = \frac{1}{\alpha_b} |0\rangle |t\rangle |b(t)\rangle + |\perp\rangle. \quad (120)$$

Here O_A is a time-dependent block-encoding of $A(t)$, and O_b is a time-dependent state preparation oracle of $b(t)$. The normalization factors α_A and α_b satisfy $\alpha_A \geq \max_t \|A(t)\|$ and $\alpha_b \geq \max_t \|b(t)\|$. $|t\rangle$ represents an encoding of the time which usually refers to the index in specific methods. Such time-dependent version of the block-encoding model of $A(t)$ and state preparation of $b(t)$ have also been assumed in existing quantum algorithms for Hamiltonian simulation [\[46\]](#) and general differential equations [\[6, 7, 10\]](#).

5.1 Truncated Dyson series method

The truncated Dyson series method is introduced in [\[6\]](#) and achieves high-order convergence. In this method, we have

$$L_j = I, \quad (121)$$

$$R_j = \sum_{k=0}^K \int_{jh}^{(j+1)h} dt_1 \int_{jh}^{t_1} dt_2 \cdots \int_{jh}^{t_{k-1}} dt_k A(t_1) A(t_2) \cdots A(t_k), \quad (122)$$

and

$$v_j = \sum_{k=1}^K \int_{jh}^{(j+1)h} dt_1 \int_{jh}^{t_1} dt_2 \cdots \int_{jh}^{t_{k-1}} dt_k A(t_1) A(t_2) \cdots A(t_{k-1}) b(t_k). \quad (123)$$

Here K is the truncation order in the Dyson series and only has a logarithmic dependence on the local truncation error, according to [6, Section 1].

Notice that in the actual implementation, integrals in Eqs. (122) and (123) need to be further discretized with sufficiently many grid points, but this only introduces logarithmically many extra ancilla qubits and gates, and does not affect the overall number of queries to the input model of $A(t)$. For technical simplicity, here we will omit this further step of integral discretization as well as its discretization error, and refer to [6] for more details.

We first discuss the complexity in the general inhomogeneous case, which is given in the following result.

Corollary 12. *Consider using the truncated Dyson series method for solving the ODE Eq. (1) with $A(t)$ being a matrix such that $A(t) + A(t)^\dagger \leq -2\eta < 0$. Let $\epsilon > 0$ be the target error, $T \geq \eta^{-1}$ be the evolution time, $\alpha_A \geq \max \|A(t)\|$ be the block-encoding factor of $A(t)$, and $\alpha_b \geq \max \|b(t)\|$ be the normalization factor of $b(t)$. Then, we can obtain an ϵ -approximation of*

1. the history state $\mathbf{u}_{\text{exact}}/\|\mathbf{u}_{\text{exact}}\|$ using

$$\mathcal{O}\left(\frac{\alpha_A}{\eta} \log\left(\frac{(\alpha_A + \alpha_b)T}{\|u_0\|\eta\epsilon}\right) \log\left(\frac{1}{\epsilon}\right)\right) \quad (124)$$

queries to the time-dependent block-encoding O_A of $A(t)$ and

$$\mathcal{O}\left(\frac{\alpha_A}{\eta} \log\left(\frac{1}{\epsilon}\right)\right) \quad (125)$$

queries to the state preparation O_{u_0} of $|u_0\rangle$ and O_b of $|b(t)\rangle$.

2. the final state $u(T)/\|u(T)\|$ using

$$\tilde{\mathcal{O}}\left(\frac{\max_t \|u(t)\|}{\|u(T)\|} \frac{\alpha_A \sqrt{T}}{\sqrt{\eta}} \log\left(\frac{\|u_0\| + \alpha_b \alpha_A}{\|u(T)\| \eta \epsilon}\right) \log\left(\frac{1}{\epsilon}\right)\right) \quad (126)$$

queries to the time-dependent block-encoding O_A of $A(t)$ and

$$\tilde{\mathcal{O}}\left(\frac{\max_t \|u(t)\|}{\|u(T)\|} \frac{\alpha_A \sqrt{T}}{\sqrt{\eta}} \log\left(\frac{1}{\epsilon}\right)\right) \quad (127)$$

queries to the state preparation O_{u_0} of $|u_0\rangle$ and O_b of $|b(t)\rangle$.

Proof. We first consider the history state preparation. According to [6, Section 3], constructing a block-encoding of $\mathbf{A}_{M, M_p-1}$ needs K calls to O_A , and constructing a state preparation oracle of $|\mathbf{b}_{M, M_p-1}\rangle$ needs $K-1$ calls to O_A , one call to O_b , and one call to O_{u_0} . Therefore, Theorem 8 tells that the overall query complexity is

$$\mathcal{O}\left(\left(1 + \max_j \|R_j\|\right) \frac{KM}{\eta T} \log\left(\frac{1}{\epsilon}\right)\right) \quad (128)$$

queries to O_A , and

$$\mathcal{O}\left(\left(1 + \max_j \|R_j\|\right) \frac{M}{\eta T} \log\left(\frac{1}{\epsilon}\right)\right) \quad (129)$$

queries to O_{u_0} and O_b .

We need to choose K and M such that the assumptions in Theorem 8 hold. Notice that, under the assumption $\eta h \leq 1$ which will be satisfied by the choice of h specified later, the first term on

the right hand side of Eq. (49) is always larger than the second term. So we need to choose K and M such that

$$\left\| R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \mathcal{O} \left(\frac{\eta^{3/2} h \epsilon}{\sqrt{1 + \frac{T \max \|b\|^2}{\eta \|u_0\|^2}} \sqrt{\max \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}} \right), \quad (130)$$

and

$$\left\| v_j - \int_{jh}^{(j+1)h} \mathcal{T} e^{\int_s^{(j+1)h} A(\tau) d\tau} b(s) ds \right\| \leq \mathcal{O} \left(\frac{\|u_0\| \eta h \epsilon}{\sqrt{T} \sqrt{\max \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}} \right). \quad (131)$$

We first choose

$$M = 2\alpha_A T, \quad (132)$$

then $\alpha_A h = 1/2$ and thus the block-encoding of the Dyson series is well-behaved [6]. As a result, by the definition of R_j in Eq. (122), we have

$$\|R_j\| \leq \sum_{k=0}^K \frac{h^k}{k!} \alpha_A^k \leq e^{\alpha_A h} = \mathcal{O}(1). \quad (133)$$

Furthermore, according to [6, Eq.(6)], the local truncation error can be bounded as

$$\left\| R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \mathcal{O} \left(\frac{(\alpha_A h)^{K+1}}{(K+1)!} \right) \leq \mathcal{O} \left(\frac{1}{(K+1)!} \right) \leq \mathcal{O}(e^{-K}). \quad (134)$$

In order to bound this error according to Eq. (130), it suffices to choose

$$K = \mathcal{O} \left(\log \left(\frac{\sqrt{1 + \frac{T \max \|b\|^2}{\eta \|u_0\|^2}} \sqrt{\max \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}}{\eta^{3/2} h \epsilon} \right) \right) = \mathcal{O} \left(\log \left(\frac{(\alpha_A + \alpha_b) T}{\|u_0\| \eta \epsilon} \right) \right). \quad (135)$$

For the inhomogeneous error, [6, Eq.(14)] suggests that

$$\left\| v_j - \int_{jh}^{(j+1)h} \mathcal{T} e^{\int_s^{(j+1)h} A(\tau) d\tau} b(s) ds \right\| \leq \mathcal{O} \left(\frac{\alpha_A^K h^{K+1} \alpha_b}{(K+1)!} \right) \leq \mathcal{O} \left(\frac{\alpha_b h}{(K+1)!} \right) \leq \mathcal{O}(\alpha_b h e^{-K}). \quad (136)$$

In order to bound this error according to Eq. (131), it suffices to choose

$$K = \mathcal{O} \left(\log \left(\frac{\alpha_b \sqrt{T} \sqrt{\max \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}}{\|u_0\| \eta \epsilon} \right) \right) = \mathcal{O} \left(\log \left(\frac{(\alpha_A + \alpha_b) T}{\|u_0\| \eta \epsilon} \right) \right), \quad (137)$$

which is asymptotically the same as the choice of K for bounding the homogeneous error. Plugging the choices of M and K back to Eqs. (128) and (129) yields the claimed complexity estimates for history state preparation.

We now move on to the final state preparation, where the proof is quite similar. Corollary 11 tells that the overall query complexity is

$$\begin{aligned} & \tilde{\mathcal{O}} \left(\left(1 + \max_j \|R_j\| \right) \frac{\max_t \|u(t)\|}{\|u(T)\|} \frac{KM \sqrt{1 + \eta T}}{\eta T} \log \left(\frac{1}{\epsilon} \right) \right) \\ &= \tilde{\mathcal{O}} \left(\left(1 + \max_j \|R_j\| \right) \frac{\max_t \|u(t)\|}{\|u(T)\|} \frac{KM}{\sqrt{\eta T}} \log \left(\frac{1}{\epsilon} \right) \right) \end{aligned} \quad (138)$$

queries to O_A , and

$$\begin{aligned} & \tilde{\mathcal{O}} \left(\left(1 + \max_j \|R_j\| \right) \frac{\max_t \|u(t)\|}{\|u(T)\|} \frac{M \sqrt{1 + \eta T}}{\eta T} \log \left(\frac{1}{\epsilon} \right) \right) \\ &= \tilde{\mathcal{O}} \left(\left(1 + \max_j \|R_j\| \right) \frac{\max_t \|u(t)\|}{\|u(T)\|} \frac{M}{\sqrt{\eta T}} \log \left(\frac{1}{\epsilon} \right) \right) \end{aligned} \quad (139)$$

queries to O_{u_0} and O_b . So we need to determine the choice of K and M in this case. We can use the same choice $M = 2\alpha_A T$. Then $\|R_j\|$'s are still $\mathcal{O}(1)$, and we still have the local truncation error bound in Eqs. (134) and (136). To satisfy the assumptions in Theorem 10 and Corollary 11, it suffices to require

$$\mathcal{O}(e^{-K}) \leq \mathcal{O}\left(\frac{\|u(T)\|\eta h\epsilon}{\|u_0\| + \max\|b(t)\|/\eta}\right), \quad (140)$$

and

$$\mathcal{O}(\alpha_b h e^{-K}) \leq \mathcal{O}(\|u(T)\|\eta h\epsilon). \quad (141)$$

By solving these two inequalities, we can derive a sufficient condition for K to be

$$K = \mathcal{O}\left(\log\left(\frac{\|u_0\| + \alpha_b \alpha_A}{\|u(T)\| \eta \epsilon}\right)\right). \quad (142)$$

Plugging these choices back to Eqs. (138) and (139) completes the second part of the proof. $\square$

In the special homogeneous case, the complexity for preparing a history state can be slightly improved by applying the tighter estimate in Theorem 9. We state its complexity in the following result and give its proof in Appendix D.

Corollary 13. *Consider using the truncated Dyson series method for solving the ODE Eq. (1) with $b(t) \equiv 0$ and $A(t)$ being a matrix such that $A(t) + A(t)^\dagger \leq -2\eta < 0$. Let $\epsilon > 0$ be the target error, $T \geq (\max\|A(t)\|)^{-1}$ be the evolution time, and $\alpha_A \geq \max\|A(t)\|$ be the block-encoding factor of $A(t)$. Then, we can obtain an ϵ -approximation of the history state $\mathbf{u}_{\text{exact}}/\|\mathbf{u}_{\text{exact}}\|$, using*

$$\mathcal{O}\left(\frac{\alpha_A}{\eta} \log\left(\frac{\alpha_A}{\eta \epsilon}\right) \log\left(\frac{1}{\epsilon}\right)\right) \quad (143)$$

queries to the time-dependent block-encoding O_A of $A(t)$ and

$$\mathcal{O}\left(\frac{\alpha_A}{\eta} \log\left(\frac{1}{\epsilon}\right)\right) \quad (144)$$

queries to the state preparation O_{u_0} of $|u_0\rangle$.

5.2 Lower-order methods

We have shown that the truncated Dyson series method can scale $\mathcal{O}(\text{poly log}(T/\epsilon))$ for history state preparation and $\tilde{\mathcal{O}}(\sqrt{T} \text{poly log}(1/\epsilon))$ for final state preparation. The dependence on time T achieves fast-forwarding compared to the previous linear scaling. In this subsection, we will show that such a fast-forwarded time dependence can also be achieved by much simpler lower-order methods such as forward Euler method and trapezoidal rule.

To highlight our main point and simplify the technical analysis, throughout this subsection, we will only focus on the explicit complexity in terms of time T and error ϵ , and treat all the other parameters as constants and absorb them into the big-O notation. We will require the time-discretization $h = T/N$ to satisfy $h\alpha_A = \mathcal{O}(1)$. Thus we are able to construct the block-encoding of $\mathbf{A}_{M,M_p-1}$ defined in Eq. (26) corresponding to different numerical schemes with a $\mathcal{O}(1)$ normalization factor. The constructing process is postponed to Appendix E.

5.2.1 Forward Euler method

The forward Euler method is to approximate $\mathcal{T}e^{\int_{jh}^{(j+1)h} A(s)ds}$ by $I + hA(jh)$. According to the analysis in Appendix F, we know

$$\left\| \mathcal{T}e^{\int_{jh}^{(j+1)h} A(s)ds} - (I + hA(jh)) \right\| = \mathcal{O}(h^2) \quad (145)$$

and

$$\left\| hb((j+1)h) - \int_0^h \mathcal{T}e^{\int_s^{(j+1)h} A(t)dt} b(s)dt \right\| = \mathcal{O}(h^2). \quad (146)$$

We first present its complexity estimate for the general inhomogeneous ODE.

Corollary 14. To solve the ODE Eq. (1) with $A(t)$ satisfying the stability condition Eq. (11) by the forward Euler method, let ϵ be the desired precision, $T > (\max \|A(t)\|)^{-1}$ be the evolution time, then we can obtain an ϵ -approximation of

1. the history state $\mathbf{u}_{\text{exact}}/\|\mathbf{u}_{\text{exact}}\|$ using

$$\mathcal{O}\left(\frac{\sqrt{T}}{\epsilon} \log\left(\frac{1}{\epsilon}\right)\right) \quad (147)$$

queries to the time-dependent block-encoding O_A of $A(t)$ and the state preparation oracle.

2. the final state $u(T)/\|u(T)\|$ using

$$\tilde{\mathcal{O}}\left(\frac{\sqrt{T}}{\epsilon} \log\left(\frac{1}{\epsilon}\right)\right) \quad (148)$$

queries to the time-dependent block-encoding O_A of $A(t)$ and the state preparation oracle.

Proof. According to the discussion in Appendix E, we need one query of O_A to construct the block-encoding of $\mathbf{A}_{M,M_p-1}$. Theorem 8 tells us that we need Eq. (49) and Eq. (50) hold for the numerical scheme.

In the inhomogeneous case, the first requirement is

$$\begin{aligned} & \left\| L(j, h)^{-1} R(j, h) - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \\ & \leq \min \left\{ \frac{1}{2} \eta h e^{-\eta h}, \frac{\eta^{3/2} h \epsilon}{144 \sqrt{2} \sqrt{1 + \frac{T \max \|b(t)\|^2}{\eta \|u_0\|^2}} \sqrt{\max \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}} \right\}, \end{aligned} \quad (149)$$

and what we have is

$$\left\| L(j, h)^{-1} R(j, h) - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| = \mathcal{O}(h^2), \quad (150)$$

so we can just require

$$h \lesssim \min \left\{ \eta, \frac{\eta^{3/2} \epsilon}{\sqrt{1 + \frac{T \max \|b(t)\|^2}{\eta \|u_0\|^2}} \sqrt{\alpha_A + \frac{\max \|b(t)\|}{\|u_0\|}}} \right\} = \mathcal{O}\left(\frac{\epsilon}{\sqrt{T}}\right). \quad (151)$$

For the second requirement

$$\left\| L(j, h)^{-1} v(j, h) - \int_{jh}^{(j+1)h} \mathcal{T} e^{\int_s^{(j+1)h} A(\tau) d\tau} b(s) ds \right\| \leq \frac{\|u_0\| \eta h \epsilon}{72 \sqrt{2} \sqrt{T} \sqrt{\max_t \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}}, \quad (152)$$

we would require

$$h \lesssim \frac{\|u_0\| \eta \epsilon}{\sqrt{T} \sqrt{\alpha_A + \frac{\max \|b(t)\|}{\|u_0\|}}} = \mathcal{O}\left(\frac{\epsilon}{\sqrt{T}}\right). \quad (153)$$

For the complexity of T and ϵ , equation Eq. (151) and equation Eq. (153) indicate that

$$M = \mathcal{O}\left(T^{3/2} \frac{1}{\epsilon}\right). \quad (154)$$

So according to Theorem 8, we will need

$$\mathcal{O}\left(T^{1/2} \frac{1}{\epsilon} \log\left(\frac{1}{\epsilon}\right)\right) \quad (155)$$

queries to O_A and the state preparation oracle that produces $|\mathbf{b}\rangle$.

For the final state preparation, the first requirement in [Theorem 10](#) is

$$\left\| L_j^{-1} R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \min \left\{ \frac{1}{2} \eta h e^{-\eta h}, \frac{\|u(T)\|}{\|u_0\| + \max \|b(t)\|/\eta} \frac{\eta h \epsilon}{128} \right\}, \quad (156)$$

and what we have is

$$\left\| L(j, h)^{-1} R(j, h) - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| = \mathcal{O}(h^2). \quad (157)$$

So we can just require

$$h \lesssim \min \left\{ \eta, \frac{\|u(T)\| \eta \epsilon}{\|u_0\| + \max \|b(t)\|/\eta} \right\} = \mathcal{O}(\epsilon). \quad (158)$$

For the second requirement

$$\left\| L_j^{-1} v_j - \int_{jh}^{(j+1)h} \mathcal{T} e^{\int_s^{(j+1)h} A(\tau) d\tau} b(s) ds \right\| \leq \frac{\|u(T)\| \eta h \epsilon}{32}, \quad (159)$$

we would require

$$h \lesssim \|u(T)\| \eta \epsilon = \mathcal{O}(\epsilon). \quad (160)$$

It is clear that

$$M = \mathcal{O}\left(\frac{T}{\epsilon}\right). \quad (161)$$

So according to [Corollary 11](#), we will need

$$\tilde{\mathcal{O}}\left(T^{1/2} \frac{1}{\epsilon} \log\left(\frac{1}{\epsilon}\right)\right) \quad (162)$$

queries to O_A and the state preparation oracle. $\square$

[Corollary 14](#) shows that complexities of the forward Euler method for history state and final state are the same $\mathcal{O}(\sqrt{T})$. For homogeneous ODE, the complexity for history state preparation can be further improved, as shown in the next result.

Corollary 15. *To solve the ODE [Eq. \(1\)](#) with $A(t)$ satisfying the stability condition [Eq. \(11\)](#) and $b(t) \equiv 0$ by the forward Euler method, let ϵ be the desired precision, $T > (\max \|A(t)\|)^{-1}$ be the evolution time, then we can obtain an ϵ -approximation of the history state $\mathbf{u}_{\text{exact}}/\|\mathbf{u}_{\text{exact}}\|$ using*

$$\mathcal{O}\left(\frac{1}{\epsilon} \log\left(\frac{1}{\epsilon}\right)\right) \quad (163)$$

queries to O_A and the state preparation oracle.

Proof. According to [Theorem 9](#), for the homogeneous case, we only require that

$$\left\| L_j^{-1} R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \min \left\{ \frac{1}{2} \eta h e^{-\eta h}, \frac{\eta^{3/2} h \epsilon}{32 \sqrt{\max \|A(t)\|}} \right\}. \quad (164)$$

This leads to

$$h \lesssim \min \left\{ \eta, \frac{\eta^{3/2} \epsilon}{\alpha_A^{1/2}} \right\} = \mathcal{O}(\epsilon), \quad (165)$$

meaning

$$M = \mathcal{O}\left(\frac{T}{\epsilon}\right). \quad (166)$$

So according to [Theorem 9](#), there is no dependence on T , and the dependence on ϵ is $\mathcal{O}(\frac{1}{\epsilon} \log(\frac{1}{\epsilon}))$. $\square$

5.2.2 Trapezoidal rule

The trapezoidal rule is to approximate $\mathcal{T}e^{\int_{jh}^{(j+1)h} A(s)ds}$ by $(I - \frac{h}{2}A((j+1)h))^{-1}(I + \frac{h}{2}A(jh))$. According to the analysis in [Appendix F](#), we know

$$\left\| \mathcal{T}e^{\int_{jh}^{(j+1)h} A(s)ds} - \left(I - \frac{h}{2}A((j+1)h)\right)^{-1} \left(I + \frac{h}{2}A(jh)\right) \right\| = \mathcal{O}(h^3) \quad (167)$$

and

$$\left\| \left(I - \frac{h}{2}A((j+1)h)\right)^{-1} \frac{h}{2}(b(h) + b(0)) - \int_0^h \mathcal{T}e^{\int_s^{(j+1)h} A(t)dt} b(s)ds \right\| = \mathcal{O}(h^3). \quad (168)$$

Corollary 16. *To solve the ODE [Eq. \(1\)](#) with $A(t)$ satisfying the stability condition [Eq. \(11\)](#) by the trapezoidal rule, let ϵ be the desired precision, $T > (\max \|A(t)\|)^{-1}$ be the evolution time, then we can obtain an ϵ -approximation of*

1. the history state $\mathbf{u}_{\text{exact}}/\|\mathbf{u}_{\text{exact}}\|$ using

$$\mathcal{O}\left(T^{1/4} \sqrt{\frac{1}{\epsilon}} \log\left(\frac{1}{\epsilon}\right)\right) \quad (169)$$

queries to the time-dependent block-encoding O_A of $A(t)$ and the state preparation oracle.

2. the final state $u(T)/\|u(T)\|$ using

$$\tilde{\mathcal{O}}\left(T^{1/2} \sqrt{\frac{1}{\epsilon}} \log\left(\frac{1}{\epsilon}\right)\right) \quad (170)$$

queries to the time-dependent block-encoding O_A of $A(t)$ and the state preparation oracle.

Proof. According to the discussion in [Appendix E](#), we need two queries of O_A to construct the block-encoding of $\mathbf{A}_{M,M_p-1}$. [Theorem 8](#) tells us that we need [Eq. \(49\)](#) and [Eq. \(50\)](#) hold for the numerical scheme. In the inhomogeneous case, the first requirement is

$$\begin{aligned} & \left\| L(j, h)^{-1} R(j, h) - \mathcal{T}e^{\int_{jh}^{(j+1)h} A(s)ds} \right\| \\ & \leq \min \left\{ \frac{1}{2} \eta h e^{-\eta h}, \frac{\eta^{3/2} h \epsilon}{144 \sqrt{2} \sqrt{1 + \frac{T \max \|b(t)\|^2}{\eta \|u_0\|^2}} \sqrt{\max \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}} \right\}, \end{aligned} \quad (171)$$

and what we have is

$$\left\| L(j, h)^{-1} R(j, h) - \mathcal{T}e^{\int_{jh}^{(j+1)h} A(s)ds} \right\| = \mathcal{O}(h^3), \quad (172)$$

so we can just require

$$h \lesssim \min \left\{ \sqrt{\eta}, \left(\frac{\eta^{3/2} \epsilon}{\sqrt{1 + \frac{T \max \|b(t)\|^2}{\eta \|u_0\|^2}} \sqrt{\alpha_A + \frac{\max \|b(t)\|}{\|u_0\|}}} \right)^{1/2} \right\} = \mathcal{O}\left(\frac{\epsilon^{1/2}}{T^{1/4}}\right). \quad (173)$$

For the second requirement

$$\left\| L(j, h)^{-1} v(j, h) - \int_{jh}^{(j+1)h} \mathcal{T}e^{\int_s^{(j+1)h} A(\tau)d\tau} b(s)ds \right\| \leq \frac{\|u_0\| \eta h \epsilon}{72 \sqrt{2} \sqrt{T} \sqrt{\max_t \|A(t)\| + \frac{\max \|b(t)\|}{\|u_0\|}}}, \quad (174)$$

we would require

$$h \lesssim \left(\frac{\|u_0\| \eta \epsilon}{\sqrt{T} \sqrt{\alpha_A + \frac{\max\|b(t)\|}{\|u_0\|}}} \right)^{1/2} = \mathcal{O}\left(\frac{\epsilon^{1/2}}{T^{1/4}}\right). \quad (175)$$

For the complexity of T and ϵ , equation [Eq. \(173\)](#) and equation [Eq. \(175\)](#) indicate that

$$M = \mathcal{O}\left(T^{5/4} \frac{1}{\sqrt{\epsilon}}\right). \quad (176)$$

So according to theorem [Theorem 8](#), we will need

$$\mathcal{O}\left(T^{1/4} \frac{1}{\sqrt{\epsilon}} \log\left(\frac{1}{\epsilon}\right)\right) \quad (177)$$

queries to O_A and the state preparation oracle.

For the final state preparation, the first requirement in [Theorem 10](#) is

$$\left\| L_j^{-1} R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \min \left\{ \frac{1}{2} \eta h e^{-\eta h}, \frac{\|u(T)\|}{\|u_0\| + \max\|b(t)\|/\eta} \frac{\eta h \epsilon}{128} \right\}, \quad (178)$$

and what we have is

$$\left\| L(j, h)^{-1} R(j, h) - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| = \mathcal{O}(h^3), \quad (179)$$

so we can just require

$$h \lesssim \min \left\{ \sqrt{\eta}, \left(\frac{\|u(T)\| \eta \epsilon}{(\|u_0\| + \max\|b(t)\|/\eta)} \right)^{1/2} \right\} = \mathcal{O}(\sqrt{\epsilon}). \quad (180)$$

For the second requirement

$$\left\| L_j^{-1} v_j - \int_{jh}^{(j+1)h} \mathcal{T} e^{\int_s^{(j+1)h} A(\tau) d\tau} b(s) ds \right\| \leq \frac{\|u(T)\| \eta h \epsilon}{32}, \quad (181)$$

we would require

$$h \lesssim (\|u(T)\| \eta \epsilon)^{1/2} = \mathcal{O}(\sqrt{\epsilon}). \quad (182)$$

It is clear then

$$M = \mathcal{O}\left(T \frac{1}{\sqrt{\epsilon}}\right). \quad (183)$$

So according to [Corollary 11](#), we will need

$$\tilde{\mathcal{O}}\left(T^{1/2} \frac{1}{\sqrt{\epsilon}} \log\left(\frac{1}{\epsilon}\right)\right) \quad (184)$$

queries to O_A and the state preparation oracle. $\square$

In the homogeneous case, we have the following further improved complexity estimate for history state preparation.

Corollary 17. *To solve the ODE [Eq. \(1\)](#) with $A(t)$ satisfying the stability condition [Eq. \(11\)](#) and $b(t) \equiv 0$ by the trapezoidal rule, let ϵ be the desired precision, $T > (\max\|A(t)\|)^{-1}$ be the evolution time, then we can obtain an ϵ -approximation of the history state $\mathbf{u}_{\text{exact}}/\|\mathbf{u}_{\text{exact}}\|$ using*

$$\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}} \log\left(\frac{1}{\epsilon}\right)\right) \quad (185)$$

queries to O_A and the state preparation oracle.

Proof. According to [Theorem 9](#), for the homogeneous case, we only require that

$$\left\| L_j^{-1} R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \min \left\{ \frac{1}{2} \eta h e^{-\eta h}, \frac{\eta^{3/2} h \epsilon}{32 \sqrt{\max \|A(t)\|}} \right\}, \quad \forall 0 \leq j \leq T/h - 1. \quad (186)$$

This leads to

$$h \lesssim \min \left\{ \sqrt{\eta}, \sqrt{\frac{\eta^{3/2} \epsilon}{\alpha_A^{1/2}}} \right\} = \mathcal{O}(\sqrt{\epsilon}), \quad (187)$$

meaning

$$M = \mathcal{O} \left(T \frac{1}{\sqrt{\epsilon}} \right). \quad (188)$$

So according to [Theorem 9](#), there is no dependence on T , and the dependence on ϵ is $\mathcal{O} \left(\frac{1}{\sqrt{\epsilon}} \log \left(\frac{1}{\epsilon} \right) \right)$. $\square$

6 Applications

In this section, we discuss the complexity of quantum algorithm for some applications of dissipative ODEs, including quantum dynamics with non-Hermitian Hamiltonian and generalized heat equation. For the best query complexity, in this section we only focus on the complexity of using the truncated Dyson series method.

6.1 Quantum dynamics with non-Hermitian Hamiltonian

General form of quantum dynamics with non-Hermitian Hamiltonian is given as

$$i \frac{du(t)}{dt} = (H(t) + iL(t))u(t), \quad u(0) = |u_0\rangle. \quad (189)$$

Here the entire matrix $H(t) + iL(t)$ is called the non-Hermitian Hamiltonian, where both $H(t)$ and $L(t)$ are Hermitian. [Eq. \(189\)](#) is a generalization of the time-dependent Hamiltonian simulation problem. The $H(t)$ matrix is the standard (Hermitian) Hamiltonian in a closed quantum system, and $iL(t)$ represents the non-Hermitian correction term to the system.

To apply our fast-forwarded algorithm, we assume [Eq. \(189\)](#) to be dissipative, namely that there exists a constant $\eta > 0$ such that $L(t) \leq -\eta < 0$. We suppose that we are given access to the time-dependent block-encodings O_L and O_H of $L(t)$ and $H(t)$, respectively. The block-encoding of $H(t) + iL(t)$ can be easily constructed by linearly adding them together using the linear combination of unitaries (LCU) technique [\[47, 48\]](#), and this only requires one query to each of the block-encodings.

Notice that [Eq. \(189\)](#) is a homogeneous ODE, so the dynamics decays exponentially and we only focus on the task of history state preparation. Quantum complexity is given in the following result, which is just a reformulation of [Corollary 13](#) and has no explicit time dependence.

Corollary 18. *Consider using the truncated Dyson series method for simulating dissipative quantum dynamics with non-Hermitian Hamiltonian in [Eq. \(189\)](#) with $L(t) \leq -\eta < 0$. Let $\epsilon > 0$ be the target error, T be the evolution time, and $\alpha_L \geq \max \|L(t)\|$ and $\alpha_H \geq \max \|H(t)\|$ be the block-encoding factors of $L(t)$ and $H(t)$, respectively. Then, we can obtain an ϵ -approximation of the history state with complexity*

$$\mathcal{O} \left(\frac{\alpha_L + \alpha_H}{\eta} \log \left(\frac{\alpha_L + \alpha_H}{\eta \epsilon} \right) \log \left(\frac{1}{\epsilon} \right) \right). \quad (190)$$

6.2 Generalized heat equation

Consider the following evolutionary partial differential equation (PDE) on a d -dimensional unit cube

$$\partial_t u(t, x) = a\Delta u(t, x) + b\nabla \cdot u(t, x) + c(t, x)u(t, x) + f(t, x), \quad t \in [0, T], \quad x \in [0, 1]^d, \quad (191)$$

$$u(0, x) = u_0(x), \quad x \in [0, 1]^d, \quad (192)$$

$$u(t, x) = 0, \quad x \in \partial([0, 1]^d). \quad (193)$$

Here $a > 0$ is a positive parameter called the thermal diffusivity, b is a real parameter called the flow velocity, $c(t, x) \leq 0$ is the potential function, and $f(t, x)$ is the source term. We impose homogeneous Dirichlet boundary condition on the cube, so the heat process is dissipative.

A standard way of solving Eq. (191) numerically is by the method of lines, which first semi-discretizes the PDE into an ODE by only performing spatial discretization then solves the resulting ODE by any time propagator. For spatial discretization, along each direction we use $(n_x + 1)$ many equi-distant grid points j/n_x , $0 \leq j \leq n_x$. We discretize the spatial derivatives by central difference, and the semi-discretized heat equation is given as

$$\frac{d\vec{u}(t)}{dt} = aL\vec{u}(t) + bD\vec{u}(t) + C(t)\vec{u}(t) + \vec{f}(t), \quad (194)$$

$$\vec{u}(0) = \vec{u}_0. \quad (195)$$

Here $\vec{u}(t)$ is an $(n_x + 1)^d$ -dimensional vector approximating the solution $u(t, x)$ at the point $x = (j_1/n_x, j_2/n_x, \dots, j_d/n_x)$, $0 \leq j_k \leq n_x$. $\vec{f}(t)$ is a vector encoding $f(t, x)$ at those points as well. L , D and $C(t)$ are the discretized version of the Laplace operator Δ , the divergence operator $\nabla \cdot$, and the potential function $c(t, x)$, respectively. They are given as

$$L = \sum_{j=1}^d I^{\otimes(j-1)} \otimes L_1 \otimes I^{\otimes(d-j)}, \quad L_1 = n_x^2 \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix}, \quad (196)$$

$$D = \sum_{j=1}^d I^{\otimes(j-1)} \otimes D_1 \otimes I^{\otimes(d-j)}, \quad D_1 = \frac{1}{2}n_x \begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & -1 & 0 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 0 & 1 \\ & & & & -1 & 0 \end{pmatrix}, \quad (197)$$

and

$$C(t) = \text{diag}(c(t, x_j)), \quad x_j = (j_1/n_x, j_2/n_x, \dots, j_d/n_x), \quad 0 \leq j_k \leq n_x. \quad (198)$$

We assume that n_x is sufficiently large such that the spatial discretization error is already reasonably bounded, and we only consider applying quantum algorithm to approximate the semi-discretized solution $\vec{u}(t)$ in Eq. (194).

The coefficient matrix of Eq. (194) is $A(t) = aL + bD + C(t)$ and has the following properties. First, both L and $C(t)$ are Hermitian, and D is anti-Hermitian. The matrix $C(t)$ is negative semi-definite since $c(t, x) \leq 0$, and eigenvalues of L_1 are given as $\lambda = -4n_x^2 \left(\sin\left(\frac{j\pi}{2(n_x+2)}\right) \right)^2$, $1 \leq j \leq n_x + 1$. So, we have

$$A(t) + A(t)^\dagger = 2aL + 2C(t) \leq -8adn_x^2 \left(\sin\left(\frac{\pi}{2(n_x+2)}\right) \right)^2 \leq -\frac{8adn_x^2}{(n_x+2)^2}, \quad (199)$$

where we have used the fact that $\sin(\frac{\pi x}{2}) \geq x$ for $x \in [0, 1]$, and thus Eq. (194) satisfies our assumption for fast-forwarding by choosing $\eta \sim d$. We also have $\|A(t)\| = \mathcal{O}(dn_x^2)$, supposing $c(t, x)$ is on $\mathcal{O}(1)$. So, we have the following complexity estimate for solving Eq. (194) from Corollary 12.

Corollary 19. Consider using the truncated Dyson series method for solving the semi-discretized heat equation in [Eq. \(194\)](#). Let $\epsilon > 0$ be the target error, T be the evolution time, d be the spatial dimension and n_x be the number of spatial grid points along each dimension. Then, we can obtain an ϵ -approximation of

1. the history state with complexity

$$\tilde{\mathcal{O}}\left(n_x^2 \log(T) \left(\log\left(\frac{1}{\epsilon}\right)\right)^2\right), \quad (200)$$

2. the final state with complexity

$$\tilde{\mathcal{O}}\left(\frac{\max_t \|\vec{u}(t)\|}{\|\vec{u}(T)\|} n_x^2 \sqrt{dT} \left(\log\left(\frac{1}{\epsilon}\right)\right)^2\right). \quad (201)$$

When $b = 0$, $c \equiv 0$ and $f \equiv 0$, [Eq. \(191\)](#) represents the standard heat equation. A detailed study on the quantum and classical algorithms for heat equations has been carried out in [\[49\]](#). Notice that the problem setups of [\[49\]](#) and our work are different: our work aims to prepare a quantum state encoding the solution, while [\[49\]](#) studies an estimation of the total heat over a subdomain. It would be interesting to further investigate the complexity of measuring the total heat using our fast-forwarded algorithm as a subroutine, and our work is likely to yield a better time dependence compared to [\[49\]](#) for the linear-equations based algorithms.

Acknowledgements

We thank Andrew Childs, David Jennings, and Matteo Lostaglio for helpful discussions.

References

- [1] Dominic W. Berry. “High-order quantum algorithm for solving linear differential equations”. [Journal of Physics A: Mathematical and Theoretical](#) **47**, 105301 (2014).
- [2] Aram W. Harrow, Avinandan Hassidim, and Seth Lloyd. “Quantum algorithm for linear systems of equations”. [Physical Review Letters](#) **103**, 150502 (2009).
- [3] 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).
- [4] Andrew M. Childs and Jin-Peng Liu. “Quantum spectral methods for differential equations”. [Communications in Mathematical Physics](#) **375**, 1427–1457 (2020).
- [5] Hari Krovi. “Improved quantum algorithms for linear and nonlinear differential equations”. [Quantum](#) **7**, 913 (2023).
- [6] Dominic W. Berry and Pedro C. S. Costa. “Quantum algorithm for time-dependent differential equations using dyson series”. [Quantum](#) **8**, 1369 (2024).
- [7] Di Fang, Lin Lin, and Yu Tong. “Time-marching based quantum solvers for time-dependent linear differential equations”. [Quantum](#) **7**, 955 (2023).
- [8] Guang Hao Low and Yuan Su. “Quantum eigenvalue processing”. In 2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS). [Page 1051–1062](#). IEEE (2024).
- [9] Dong An, Jin-Peng Liu, and Lin Lin. “Linear combination of Hamiltonian simulation for nonunitary dynamics with optimal state preparation cost”. [Physical Review Letters](#) **131**, 150603 (2023).
- [10] Dong An, Andrew M. Childs, and Lin Lin. “Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters”. [Communications in Mathematical Physics](#) **407**, 19 (2026).
- [11] Shi Jin, Nana Liu, and Yue Yu. “Quantum simulation of partial differential equations via schrödingerization”. [Physical Review Letters](#) **133**, 230602 (2024).

- [12] Junpeng Hu, Shi Jin, Nana Liu, and Lei Zhang. “Dilation theorem via schrödingerisation, with applications to the quantum simulation of differential equations” (2023). [arXiv:2309.16262](#).
- [13] 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 (2022).
- [14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. “Exponential improvement in precision for simulating sparse Hamiltonians”. In Proceedings of the forty-sixth annual ACM symposium on Theory of computing. Pages 283–292. (2014).
- [15] Yosi Atia and Dorit Aharonov. “Fast-forwarding of hamiltonians and exponentially precise measurements”. [Nature Communications](#) **8**, 1572 (2017).
- [16] Shouzhen Gu, Rolando D. Somma, and Burak Şahinoğlu. “Fast-forwarding quantum evolution”. [Quantum](#) **5**, 577 (2021).
- [17] Lin Lin and Yu Tong. “Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems”. [Quantum](#) **4**, 361 (2020).
- [18] Alexander M. Dalzell. “A shortcut to an optimal quantum linear system solver” (2024). [arXiv:2406.12086](#).
- [19] Dong An, Jin-Peng Liu, Daochen Wang, and Qi Zhao. “Quantum differential equation solvers: Limitations and fast-forwarding”. [Communications in Mathematical Physics](#) **406**, 189 (2025).
- [20] David Jennings, Matteo Lostaglio, Robert B. Lowrie, Sam Pallister, and Andrew T. Sornborger. “The cost of solving linear differential equations on a quantum computer: fast-forwarding to explicit resource counts”. [Quantum](#) **8**, 1553 (2024).
- [21] Carl M. Bender. “Making sense of non-Hermitian Hamiltonians”. [Reports on Progress in Physics](#) **70**, 947–1018 (2007).
- [22] M. A. Rego-Monteiro and F. D. Nobre. “Classical field theory for a non-Hermitian Schrödinger equation with position-dependent masses”. [Physical Review A](#) **88**, 032105 (2013).
- [23] Giulio G. Giusteri, Francesco Mattiotti, and G. Luca Celardo. “Non-Hermitian Hamiltonian approach to quantum transport in disordered networks with sinks: Validity and effectiveness”. [Physical Review B](#) **91**, 094301 (2015).
- [24] Ramy El-Ganainy, Konstantinos G. Makris, Mercedeh Khajavikhan, Ziad H. Musslimani, Stefan Rotter, and Demetrios N. Christodoulides. “Non-Hermitian physics and PT symmetry”. [Nature Physics](#) **14**, 11–19 (2018).
- [25] Zongping Gong, Yuto Ashida, Kohei Kawabata, Kazuaki Takasan, Sho Higashikawa, and Masahito Ueda. “Topological phases of non-Hermitian systems”. [Physical Review X](#) **8**, 031079 (2018).
- [26] Nobuyuki Okuma, Kohei Kawabata, Ken Shiozaki, and Masatoshi Sato. “Topological origin of non-Hermitian skin effects”. [Physical Review Letters](#) **124**, 086801 (2020).
- [27] Yuto Ashida, Zongping Gong, and Masahito Ueda. “Non-Hermitian physics”. [Advances in Physics](#) **69**, 249–435 (2020).
- [28] Norifumi Matsumoto, Kohei Kawabata, Yuto Ashida, Shunsuke Furukawa, and Masahito Ueda. “Continuous phase transition without gap closing in non-Hermitian quantum many-body systems”. [Physical Review Letters](#) **125**, 260601 (2020).
- [29] Kun Ding, Chen Fang, and Guancong Ma. “Non-Hermitian topology and exceptional-point geometries”. [Nature Reviews Physics](#) **4**, 745–760 (2022).
- [30] Guangze Chen, Fei Song, and Jose L. Lado. “Topological spin excitations in non-Hermitian spin chains with a generalized kernel polynomial algorithm”. [Physical Review Letters](#) **130**, 100401 (2023).
- [31] Mingchen Zheng, Yi Qiao, Yupeng Wang, Junpeng Cao, and Shu Chen. “Exact solution of the Bose-Hubbard model with unidirectional hopping”. [Physical Review Letters](#) **132**, 086502 (2024).
- [32] Pei-Xin Shen, Zhide Lu, Jose L. Lado, and Mircea Trif. “Non-Hermitian Fermi-Dirac distribution in persistent current transport”. [Physical Review Letters](#) **133**, 086301 (2024).
- [33] Lawrence C. Evans. “Partial differential equations”. [American Mathematical Society](#). (2010).
- [34] David Vernon Widder. “The heat equation”. Academic Press. (1976).
- [35] John Rozier Cannon. “The one-dimensional heat equation”. [Cambridge University Press](#). (1984).

- [36] R. K. Michael Thambynayagam. “The diffusion handbook: applied solutions for engineers”. McGraw Hill Professional. (2011).
- [37] Gengzhi Yang, Akwum Onwunta, and Dong An. “Quantum differential equation solvers with low state preparation cost: Eliminating the time dependence in dissipative equations” (2025). [arXiv:2508.15170](#).
- [38] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. “Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics”. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing. Pages 193–204. (2019).
- [39] B. David Clader, Alexander M. Dalzell, Nikitas Stamatopoulos, Grant Salton, Mario Berta, and William J. Zeng. “Quantum resources required to block-encode a matrix of classical data”. *IEEE Transactions on Quantum Engineering* **3**, 1–23 (2022).
- [40] Quynh T. Nguyen, Bobak T. Kiani, and Seth Lloyd. “Block-encoding dense and full-rank kernels using hierarchical matrices: applications in quantum numerical linear algebra”. *Quantum* **6**, 876 (2022).
- [41] Daan Camps, Lin Lin, Roel Van Beeumen, and Chao Yang. “Explicit quantum circuits for block encodings of certain sparse matrices”. *SIAM Journal on Matrix Analysis and Applications* **45**, 801–827 (2024).
- [42] Christoph Sünderhauf, Earl Campbell, and Joan Camps. “Block-encoding structured matrices for data input in quantum computing”. *Quantum* **8**, 1226 (2024).
- [43] Lov Grover and Terry Rudolph. “Creating superpositions that correspond to efficiently integrable probability distributions” (2002). [arXiv:quant-ph/0208112](#).
- [44] Xiao-Ming Zhang, Tongyang Li, and Xiao Yuan. “Quantum state preparation with optimal circuit depth: Implementations and applications”. *Physical Review Letters* **129**, 230504 (2022).
- [45] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. “Quantum amplitude amplification and estimation”. *Contemporary Mathematics* **305**, 53–74 (2002).
- [46] Guang Hao Low and Nathan Wiebe. “Hamiltonian simulation in the interaction picture” (2019). [arXiv:1805.00675](#).
- [47] Andrew M. Childs and Nathan Wiebe. “Hamiltonian simulation using linear combinations of unitary operations”. *Quantum Information and Computation* **12**, 901–924 (2012).
- [48] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. “Quantum algorithm for systems of linear equations with exponentially improved dependence on precision”. *SIAM Journal on Computing* **46**, 1920–1950 (2017).
- [49] Noah Linden, Ashley Montanaro, and Changpeng Shao. “Quantum vs. classical algorithms for solving the heat equation”. *Communications in Mathematical Physics* **395**, 601–641 (2022).

A Proof of Lemma 5

Proof of Lemma 5. Let $\Delta t = t_1 - t_0$. We first prove the upper bound. We use the convergence of the first-order time-dependent Trotter formula and obtain

$$\mathcal{T}e^{\int_{t_0}^{t_1} A(s)ds} = \lim_{r \rightarrow \infty} \prod_{j=0}^{r-1} e^{\frac{\Delta t}{r} A(t_0 + j\Delta t/r)}. \quad (202)$$

For each fixed integer r and $0 \leq j \leq r-1$, by the definition of η , we have

$$\left\| e^{\frac{\Delta t}{r} A(t_0 + j\Delta t/r)} \right\| \leq e^{-\eta\Delta t/r}, \quad (203)$$

so

$$\left\| \mathcal{T}e^{\int_{t_0}^{t_1} A(s)ds} \right\| \leq \lim_{r \rightarrow \infty} \prod_{j=0}^{r-1} \left\| e^{\frac{\Delta t}{r} A(t_0 + j\Delta t/r)} \right\| \leq \lim_{r \rightarrow \infty} \prod_{j=0}^{r-1} e^{-\eta\Delta t/r} = e^{-\eta\Delta t}. \quad (204)$$

For the lower bound, notice that $\left(\mathcal{T}e^{\int_{t_0}^{t_1} A(s)ds} \right)^{-1} = \mathcal{T}e^{-\int_{t_0}^{t_1} A(t_0 + t_1 - s)ds}$. We have

$$1 = \| |v\rangle \| = \left\| \left(\mathcal{T}e^{-\int_{t_0}^{t_1} A(t_0 + t_1 - s)ds} \right) \left(\mathcal{T}e^{\int_{t_0}^{t_1} A(s)ds} \right) |v\rangle \right\| \leq \left\| \mathcal{T}e^{-\int_{t_0}^{t_1} A(t_0 + t_1 - s)ds} \right\| \left\| \mathcal{T}e^{\int_{t_0}^{t_1} A(s)ds} |v\rangle \right\|, \quad (205)$$

and thus

$$\left\| \mathcal{T}e^{\int_{t_0}^{t_1} A(s)ds} |v\rangle \right\| \geq \left\| \mathcal{T}e^{-\int_{t_0}^{t_1} A(t_0 + t_1 - s)ds} \right\|^{-1}. \quad (206)$$

It suffices to derive an upper bound of $\mathcal{T}e^{-\int_{t_0}^{t_1} A(t_0 + t_1 - s)ds}$, which can be achieved by the similar technique as in the first part of the proof. Specifically, by the Trotter formula again, we have

$$\mathcal{T}e^{-\int_{t_0}^{t_1} A(t_0 + t_1 - s)ds} = \lim_{r \rightarrow \infty} \prod_{j=0}^{r-1} e^{-\frac{\Delta t}{r} A(t_1 - j\Delta t/r)}. \quad (207)$$

Noticing that $\left\| e^{-\frac{\Delta t}{r} A(t_1 - j\Delta t/r)} \right\| \leq e^{(\Delta t/r) \max_t \|A(t)\|}$ for every r and $0 \leq j \leq r-1$, we have

$$\left\| \mathcal{T}e^{-\int_{t_0}^{t_1} A(t_0 + t_1 - s)ds} \right\| \leq \lim_{r \rightarrow \infty} \prod_{j=0}^{r-1} \left\| e^{-\frac{\Delta t}{r} A(t_1 - j\Delta t/r)} \right\| \leq \lim_{r \rightarrow \infty} \prod_{j=0}^{r-1} e^{(\Delta t/r) \max_t \|A(t)\|} = e^{\Delta t \max_t \|A(t)\|}. \quad (208)$$

Therefore

$$\left\| \mathcal{T}e^{\int_{t_0}^{t_1} A(s)ds} |v\rangle \right\| \geq \left\| \mathcal{T}e^{-\int_{t_0}^{t_1} A(t_0 + t_1 - s)ds} \right\|^{-1} \geq e^{-\Delta t \max_t \|A(t)\|}. \quad (209)$$

□

B Proof of Lemma 6

Proof of Lemma 6. The idea is that, for any matrix $\mathbf{M}$, its spectral norm is the square root of the spectral norm of the Hermitian matrix $\mathbf{M}^\dagger \mathbf{M}$, and the spectral norm of a Hermitian matrix is bounded by its “1-norm” in terms of the blocks.

Specifically, we write $\mathbf{M}^\dagger \mathbf{M} = (B_{i,j})$, where

$$B_{i,j} = \sum_{k=0}^{n-1} M_{k,i}^\dagger M_{k,j}. \quad (210)$$

Let λ be the largest eigenvalue of $\mathbf{M}^\dagger \mathbf{M}$, and $\mathbf{x} = [x_0; x_1; \dots; x_{n-1}]$ be its eigenvector. Suppose that $q = \operatorname{argmax}_j \|x_j\|$, and by the definition of an eigenvector we have

$$\sum_{j=0}^{n-1} B_{q,j} x_j = \lambda x_q. \quad (211)$$

By triangle inequality,

$$\lambda \|x_q\| \leq \sum_{j=0}^{n-1} \|B_{q,j}\| \|x_j\|, \quad (212)$$

and

$$\lambda \leq \sum_{j=0}^{n-1} \|B_{q,j}\| \frac{\|x_j\|}{\|x_q\|} \leq \sum_{j=0}^{n-1} \|B_{q,j}\| \leq \max_i \sum_{j=0}^{n-1} \|B_{i,j}\|. \quad (213)$$

Therefore

$$\|\mathbf{M}\| = \sqrt{\lambda} \leq \max_i \sqrt{\sum_{j=0}^{n-1} \|B_{i,j}\|} \leq \max_i \sqrt{\sum_{j=0}^{n-1} \sum_{k=0}^{n-1} \|M_{k,i}\| \|M_{k,j}\|} = \max_i \sqrt{\sum_{k=0}^{n-1} \sum_{j=0}^{n-1} \|M_{k,i}\| \|M_{k,j}\|}. \quad (214)$$

For each fixed i and k , we have

$$\sum_{j=0}^{n-1} \|M_{k,i}\| \|M_{k,j}\| = \|M_{k,i}\| \left(\sum_{j=0}^{n-1} \|M_{k,j}\| \right) \leq \|M_{k,i}\| \left(\max_l \sum_{j=0}^{n-1} \|M_{l,j}\| \right). \quad (215)$$

So

$$\|\mathbf{M}\| \leq \max_i \sqrt{\sum_{k=0}^{n-1} \left(\|M_{k,i}\| \left(\max_l \sum_{j=0}^{n-1} \|M_{l,j}\| \right) \right)} = \sqrt{\left(\max_i \sum_{k=0}^{n-1} \|M_{k,i}\| \right) \left(\max_l \sum_{j=0}^{n-1} \|M_{l,j}\| \right)}, \quad (216)$$

which is the claimed inequality despite different notation of indices. $\square$

C Proof of Theorem 9

Proof of Theorem 9. The idea is the same as proving Theorem 8 but with tighter upper bound for the numerical discretization error. For the ease of reading, here we still present a complete proof, which has some overlap with the proof of Theorem 8 but can be read independently.

Let $M = T/h$, $\mathbf{u} = \mathbf{A}_{M,0}^{-1} \mathbf{b}_{M,0}$ and $|\mathbf{u}\rangle = \mathbf{u}/\|\mathbf{u}\|$, and we denote $L_{-1} = I$. We first show that $|\mathbf{u}\rangle$ is indeed an $\epsilon/2$ -approximation of $|\mathbf{u}_{\text{exact}}\rangle$. Let

$$\epsilon' = \frac{\eta^{3/2} h \epsilon}{32 \sqrt{\max \|A(t)\|}}. \quad (217)$$

Now the local truncation error is bounded as

$$\left\| L_j^{-1} R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \epsilon'. \quad (218)$$

Then, for each $1 \leq k \leq T/h$, we have

$$\|u_k - u(kh)\| = \left\| \prod_{j=0}^{k-1} L_j^{-1} R_j u_0 - \mathcal{T} e^{\int_0^{kh} A(s) ds} u_0 \right\| \quad (219)$$

$$\leq \left\| \prod_{j=0}^{k-1} L_j^{-1} R_j - \mathcal{T} e^{\int_0^{kh} A(s) ds} \right\| \|u_0\| \quad (220)$$

$$= \left\| \prod_{j=0}^{k-1} P_j - \prod_{j=0}^{k-1} Q_j \right\| \|u_0\|, \quad (221)$$

where we denote $P_j = L_j^{-1} R_j$ and $Q_j = \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds}$. We use the triangle inequality to write

$$\|u_k - u(kh)\| \leq \left\| \sum_{l=0}^{k-1} \left[\left(\prod_{j=l+1}^{k-1} P_j \right) (P_l - Q_l) \left(\prod_{j=0}^{l-1} Q_j \right) \right] \right\| \|u_0\| \quad (222)$$

$$\leq \sum_{l=0}^{k-1} \left[\left(\prod_{j=l+1}^{k-1} \|P_j\| \right) \|P_l - Q_l\| \left(\prod_{j=0}^{l-1} \|Q_j\| \right) \right] \|u_0\| \quad (223)$$

$$\leq \epsilon' \sum_{l=0}^{k-1} \left(\prod_{j=l+1}^{k-1} \|P_j\| \right) \left(\prod_{j=0}^{l-1} \|Q_j\| \right) \|u_0\|. \quad (224)$$

By [Lemma 5](#) and the proof of [Lemma 7](#), we have $\|Q_j\| \leq e^{-\eta h} \leq e^{-\eta h/2}$ and $\|P_j\| \leq e^{-\eta h/2}$, and thus

$$\|u_k - u(kh)\| \leq k \epsilon' e^{-\eta(k-1)h/2} \|u_0\|. \quad (225)$$

Therefore, the error in the unnormalized history vector can be bounded as

$$\|\mathbf{u} - \mathbf{u}_{\text{exact}}\| \quad (226)$$

$$= \sqrt{\sum_{k=1}^M \|u_k - u(kh)\|^2} \quad (227)$$

$$\leq \|u_0\| \epsilon' \sqrt{\sum_{k=1}^M k^2 e^{-\eta(k-1)h}} \quad (228)$$

$$= \|u_0\| \epsilon' \sqrt{\frac{(2M^2 + 2M - 1)e^{-\eta(M+1)h} - M^2 e^{-\eta(M+2)h} - (M+1)^2 e^{-\eta Mh} + e^{-\eta h} + 1}{(1 - e^{-\eta h})^3}} \quad (229)$$

$$= \|u_0\| \epsilon' \sqrt{\frac{1 + e^{-\eta h}}{(1 - e^{-\eta h})^3} - \frac{M^2 e^{-\eta Mh} (1 - e^{-\eta h})^2}{(1 - e^{-\eta h})^3} - \frac{2M e^{-\eta Mh} (1 - e^{-\eta h})}{(1 - e^{-\eta h})^3} - \frac{e^{-\eta Mh} (1 + e^{-\eta h})}{(1 - e^{-\eta h})^3}} \quad (230)$$

$$\leq \|u_0\| \epsilon' \sqrt{\frac{2}{(1 - e^{-\eta h})^3}} \quad (231)$$

$$\leq 4 \frac{\|u_0\| \epsilon'}{(\eta h)^{3/2}}. \quad (232)$$

The error in the quantum states becomes

$$\|\|\mathbf{u}\rangle - |\mathbf{u}_{\text{exact}}\rangle\| \leq \frac{2}{\|\mathbf{u}_{\text{exact}}\|} \|\mathbf{u} - \mathbf{u}_{\text{exact}}\| \leq 8 \frac{\|u_0\| \epsilon'}{\|\mathbf{u}_{\text{exact}}\| (\eta h)^{3/2}}. \quad (233)$$

According to the lower bound in [Lemma 5](#), the norm of the exact unnormalized history vector satisfies

$$\|\mathbf{u}_{\text{exact}}\| = \sqrt{\sum_{j=0}^M \|u(jh)\|^2} \quad (234)$$

$$= \sqrt{\sum_{j=0}^M \left\| \mathcal{T} e^{\int_0^{jh} A(s) ds} u_0 \right\|^2} \quad (235)$$

$$\geq \|u_0\| \sqrt{\sum_{j=0}^M e^{-2jh \max\|A(t)\|}} \quad (236)$$

$$= \|u_0\| \sqrt{\frac{1 - e^{-2(T+h) \max\|A(t)\|}}{1 - e^{-2h \max\|A(t)\|}}} \quad (237)$$

$$\geq \frac{\|u_0\|}{2\sqrt{h \max\|A(t)\|}}. \quad (238)$$

Then [Eq. \(233\)](#) becomes

$$\|\mathbf{u} - \mathbf{u}_{\text{exact}}\| \leq 16 \frac{\sqrt{\max\|A(t)\|} \epsilon'}{h\eta^{3/2}} \leq \frac{\epsilon}{2}, \quad (239)$$

where the last inequality is due to the choice of ϵ' in [Eq. \(217\)](#).

Then, solving the linear system of equations $\mathbf{A}_{M,0}\mathbf{u} = \mathbf{b}_{M,0}$ up to error $\epsilon/2$ gives the ϵ -approximation of $|\mathbf{u}_{\text{exact}}\rangle$. According to [\[13\]](#), it requires $\mathcal{O}(\kappa \log(1/\epsilon))$ queries to the block-encoding of $\mathbf{A}_{M,0}$ and the state preparation oracle of $\mathbf{b}_{M,0}$, where $\kappa = \|\mathbf{A}_{M,0}\| \|\mathbf{A}_{M,0}^{-1}\|$ is the condition number of $\mathbf{A}_{M,0}$. According to [Lemma 7](#), we have

$$\kappa = \mathcal{O}\left(\left(1 + \max_j \|L_j\| + \max_j \|R_j\|\right) \left(1 + \max_j \|L_j^{-1}\|\right) \frac{M}{\eta T}\right) \quad (240)$$

and complete the proof. $\square$

D Proof of [Corollary 13](#)

Proof of [Corollary 13](#). According to [\[6, Section 3\]](#), constructing a block-encoding of $\mathbf{A}_{M,0}$ needs K calls to O_A . Also notice that $L_j = I$ in the truncated Dyson series method. Then [Theorem 9](#) tells that the overall query complexity is

$$\mathcal{O}\left(\left(1 + \max_j \|R_j\|\right) \frac{KM}{\eta T} \log\left(\frac{1}{\epsilon}\right)\right) \quad (241)$$

queries to O_A and

$$\mathcal{O}\left(\left(1 + \max_j \|R_j\|\right) \frac{M}{\eta T} \log\left(\frac{1}{\epsilon}\right)\right) \quad (242)$$

queries to O_{u_0} .

We need to choose K and M such that [Eq. \(90\)](#) holds. Notice that, under the assumption $\eta h \leq 1$ which will be satisfied by the choice of h specified later, we have

$$\eta h e^{-\eta h} = \Omega(\eta h) = \Omega\left(\frac{\eta^{3/2} h}{\sqrt{\max\|A(t)\|}}\right) = \Omega\left(\frac{\eta^{3/2} h \epsilon}{\sqrt{\max\|A(t)\|}}\right). \quad (243)$$

Then it suffices to choose K and M such that

$$\left\| R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \mathcal{O}(\epsilon') \quad (244)$$

with

$$\epsilon' = \frac{\eta^{3/2} h \epsilon}{\sqrt{\max \|A(t)\|}}. \quad (245)$$

We first choose

$$M = 2\alpha_A T, \quad (246)$$

then $\alpha_A h = 1/2$ and thus the block-encoding of the Dyson series is well-behaved [6]. As a result, by the definition of R_j in Eq. (122), we have

$$\|R_j\| \leq \sum_{k=0}^K \frac{h^k}{k!} \alpha_A^k \leq e^{\alpha_A h} = \mathcal{O}(1). \quad (247)$$

Furthermore, according to [6, Eq.(6)], the local truncation error can be bounded as

$$\left\| R_j - \mathcal{T} e^{\int_{jh}^{(j+1)h} A(s) ds} \right\| \leq \mathcal{O}\left(\frac{(\alpha_A h)^{K+1}}{(K+1)!}\right) \leq \mathcal{O}\left(\frac{1}{(K+1)!}\right) \leq \mathcal{O}(e^{-K}). \quad (248)$$

Therefore, to bound the local truncation error by ϵ' , it suffices to choose $K = \mathcal{O}(\log(1/\epsilon'))$, which in turn, according to the choice of ϵ' in Eq. (245), becomes

$$K = \mathcal{O}\left(\log\left(\frac{\max \|A(t)\|}{\eta h \epsilon}\right)\right) = \mathcal{O}\left(\log\left(\frac{\alpha_A}{\eta \epsilon}\right)\right). \quad (249)$$

Plugging Eqs. (246), (247) and (249) back to Eqs. (241) and (242) yields the claimed complexity estimates. $\square$

E Constructing block-encodings in lower-order methods

In order to design a quantum algorithm, it is crucial to determine the input model. Recall Eq. (119), here we mainly follow the paradigm in [6], by assuming we can query $A(t)$ in the following way:

$$O_A |0^a\rangle |t\rangle |\psi\rangle = \frac{1}{\alpha_A} |0^a\rangle |t\rangle A(t) |\psi\rangle + |\perp\rangle. \quad (250)$$

Here a is the number of ancilla qubits, and the first register is used to indicate a successful or failed measurement. $|t\rangle$ ranges from $|0\rangle$ to $|M-1\rangle$, representing a uniform grid on the time interval $[0, T]$. $|\perp\rangle$ represents a possibly unnormalized state such that $(|0^a\rangle \langle 0^a| \otimes I) |\perp\rangle = 0$, and $\alpha_A \geq \max \|A(t)\|$ is the block-encoding factor.

Because of the definition of Eq. (26), we could encounter the scenario that the clock register may contain some state that is beyond time T . However we could pay the price of at most M_p controlled- X gate acting on the first register controlled on the clock register is greater than $|M-1\rangle$, so the definition in Eq. (119) will not be influenced.

Another useful operator is the ADD operator, namely

$$O_{\text{ADD}} := \sum_{t=0}^{M+M_p-1} |(t+1) \bmod (M+M_p-1)\rangle \langle t|. \quad (251)$$

E.1 Forward Euler method

Lemma 20. *When using the Forward Euler method, we are able to construct a block-encoding of the matrix $\mathbf{A}_{M, M_p-1}$ defined in Eq. (26) with block-encoding factor $2 + h\alpha_A$, using 1 query to O_A , 2 queries to the ADD operator, 2 queries to the preparation oracles defined in Eq. (257), and at most $M_p + 1$ multi-qubit controlled- X gate.*

Proof. The Forward Euler method is defined as follows:

$$u_{j+1} = (I + hA(jh))u_j. \quad (252)$$

So in this scenario we have

$$L_j = I, \quad R_j = I + hA(jh). \quad (253)$$

Thus the matrix $\mathbf{A}_{M, M_p-1}$ becomes

$$\begin{aligned} \mathbf{A}_{M, M_p-1} &= \begin{bmatrix} I & & & & & & & & \\ -I - hA(0) & I & & & & & & & \\ & -I - hA(h) & I & & & & & & \\ & & \ddots & \ddots & & & & & \\ & & & -I - hA((M-1)h) & I & & & & \\ & & & & -I & I & & & \\ & & & & & -I & I & & \\ & & & & & & \ddots & \ddots & \\ & & & & & & & -I & I \end{bmatrix} \\ &= \mathbb{I} + \begin{bmatrix} 0 & & & & & & & & \\ -I - hA(0) & 0 & & & & & & & \\ & -I - hA(h) & 0 & & & & & & \\ & & \ddots & \ddots & & & & & \\ & & & -I - hA((M-1)h) & 0 & & & & \\ & & & & -I & 0 & & & \\ & & & & & -I & 0 & & \\ & & & & & & \ddots & \ddots & \\ & & & & & & & -I & 0 \end{bmatrix} \\ &= \mathbb{I} - \sum_{t=0}^{M+M_p-2} |t+1\rangle \langle t| \otimes I - \begin{bmatrix} 0 & & & & & & & & \\ -hA(0) & 0 & & & & & & & \\ & -hA(h) & 0 & & & & & & \\ & & \ddots & \ddots & & & & & \\ & & & -hA((M-1)h) & 0 & & & & \\ & & & & 0 & 0 & & & \\ & & & & 0 & 0 & 0 & & \\ & & & & & 0 & 0 & \ddots & \ddots \\ & & & & & & & 0 & 0 \end{bmatrix} \\ &= \mathbb{I} - \sum_{t=0}^{M+M_p-2} |t+1\rangle \langle t| \otimes I - h \sum_{t=0}^{M-1} |t+1\rangle \langle t| \otimes A(t). \end{aligned} \quad (254)$$

Let us now consider the queries needed for constructing a block-encoding of the $\mathbf{A}_{M+M_p-1}$ above. Notice that

$$(I \otimes O_{\text{ADD}} \otimes I)(|0^a\rangle |t\rangle |\psi\rangle) = |0^a\rangle |(t+1) \bmod (M+M_p-1)\rangle |\psi\rangle, \quad (255)$$

then controlled on the second register being $|0\rangle$, flip the first register to $|1\rangle |0^{a-1}\rangle$, we then have a block-encoding of $\sum_{j=0}^{M-1} |j+1\rangle \langle j| \otimes I$. Denote this block-encoding as U_2 .

Similarly,

$$(I \otimes O_{\text{ADD}} \otimes I)O_A(|0^a\rangle |t\rangle |\psi\rangle) = \frac{1}{\alpha_A} |0^a\rangle |t+1\rangle A(t) |\psi\rangle + |\perp\rangle, \quad (256)$$

thus we have U_3 , which is a block-encoding of $h \sum_{j=0}^{M-1} |j+1\rangle \langle j| \otimes A(jh)$ with block-encoding factor $\alpha_A h$. By leveraging the linear combination of unitaries technique [47], we need two additional ancilla qubits. We construct the preparation oracle as

$$O_{\text{prep}} = \frac{1}{\sqrt{2 + h\alpha_A}} \begin{bmatrix} 1 & * & * & * \\ i & * & * & * \\ i\sqrt{\alpha_A h} & * & * & * \\ 0 & * & * & * \end{bmatrix} \quad (257)$$

and the select oracle to be

$$O_{\text{sel}} = |01\rangle\langle 01| \otimes U_2 + |10\rangle\langle 10| \otimes U_3. \quad (258)$$

Simple computation yields

$$(\overline{O_{\text{prep}}} \otimes I_a \otimes \mathbb{I})^\dagger O_{\text{sel}} (O_{\text{prep}} \otimes I_a \otimes \mathbb{I}) = |00\rangle\langle 00| \frac{1}{2 + h\alpha_A} (I_a \otimes \mathbb{I} - U_2 - (\alpha_A h) U_3) + P^\perp. \quad (259)$$

Here $\overline{O_{\text{prep}}}$ is the conjugate of O_{prep} , and $P^\perp$ satisfies $(\langle 00| \otimes I_a \otimes \mathbb{I}) P^\perp (|00\rangle \otimes I_a \otimes \mathbb{I}) = 0$. Thus we are able to have a block-encoding for $\mathbf{A}_{M, M_p-1}$ with block-encoding factor $2 + h\alpha_A$. $\square$

E.2 Trapezoidal rule

Lemma 21. *When using the trapezoidal rule, we are able to construct a block-encoding of the matrix $\mathbf{A}_{M, M_p-1}$ defined in Eq. (26) with block-encoding factor $2 + h\alpha_A$, using 2 queries to O_A , 2 queries to the ADD operator, 2 queries to the preparation oracles defined in Eq. (257), and at most $2M_p + 1$ multi-qubit controlled- X gate.*

Proof. For the trapezoidal rule, we have

$$u_{j+1} = u_j + \frac{1}{2} h (A(jh)u_j + b(jh) + A((j+1)h)u_{j+1} + b((j+1)h)). \quad (260)$$

$$\left(I - \frac{h}{2} A((j+1)h) \right) u_{j+1} = \left(I + \frac{h}{2} A(jh) \right) u_j + \frac{h}{2} (b(jh) + b((j+1)h)). \quad (261)$$

So we would have

$$L_j = I - \frac{h}{2} A((j+1)h), \quad R_j = I + \frac{h}{2} A(jh). \quad (262)$$

Thus the matrix $\mathbf{A}_{M,M_p-1}$ becomes

$$\begin{aligned}
& \mathbf{A}_{M,M_p-1} \\
&= \begin{bmatrix} I & & & & & & & & & \\ -I - \frac{h}{2}A(0) & I - \frac{h}{2}A(h) & & & & & & & & \\ & -I - \frac{h}{2}A(h) & I - \frac{h}{2}A(2h) & & & & & & & \\ & & & \ddots & & & & & & \\ & & & & -I - \frac{h}{2}A((M-1)h) & I - \frac{h}{2}A(Mh) & & & & \\ & & & & & -I & I & & & \\ & & & & & & -I & I & & \\ & & & & & & & \ddots & \ddots & \\ & & & & & & & & -I & I \end{bmatrix} \\
&= \mathbb{I} + \begin{bmatrix} 0 & & & & & & & & & \\ -I - \frac{h}{2}A(0) & -\frac{h}{2}A(h) & & & & & & & & \\ & -I - \frac{h}{2}A(h) & -\frac{h}{2}A(2h) & & & & & & & \\ & & & \ddots & & & & & & \\ & & & & -I - \frac{h}{2}A((M-1)h) & -\frac{h}{2}A(Mh) & & & & \\ & & & & & -I & 0 & & & \\ & & & & & & -I & 0 & & \\ & & & & & & & \ddots & \ddots & \\ & & & & & & & & -I & 0 \end{bmatrix} \\
&= \mathbb{I} - \sum_{t=0}^{M+M_p-2} |t+1\rangle \langle t| \otimes I - \begin{bmatrix} 0 & & & & & & & & & \\ -\frac{h}{2}A(0) & -\frac{h}{2}A(h) & & & & & & & & \\ & -\frac{h}{2}A(h) & -\frac{h}{2}A(2h) & & & & & & & \\ & & & \ddots & & & & & & \\ & & & & -\frac{h}{2}A((M-1)h) & -\frac{h}{2}A(Mh) & & & & \\ & & & & & 0 & 0 & & & \\ & & & & & 0 & 0 & & & \\ & & & & & & 0 & 0 & & \\ & & & & & & & \ddots & \ddots & \\ & & & & & & & & 0 & 0 \end{bmatrix} \\
&= \mathbb{I} - \sum_{t=0}^{M+M_p-2} |t+1\rangle \langle t| \otimes I - \frac{h}{2} \sum_{t=0}^{M-1} |t+1\rangle \langle t| \otimes A(t) - \frac{h}{2} \sum_{t=1}^M |t\rangle \langle t| \otimes A(t).
\end{aligned} \tag{263}$$

The rest is pretty similar to the process in the forward Euler part. Recall [Eq. \(255\)](#), we have a block-encoding V_2 of $\sum_{t=0}^{M+M_p-2} |t+1\rangle \langle t| \otimes I$ using one query to O_{ADD} and one multi-qubit controlled-X gate.

Similarly, we have a block-encoding V_3 of $\frac{h}{2} \sum_{t=0}^{M-1} |t+1\rangle \langle t| \otimes A(t)$ with block-encoding factor $\frac{1}{2}h\alpha_A$ and a block-encoding V_4 of $\frac{h}{2} \sum_{t=1}^M |t\rangle \langle t| \otimes A(t)$ with block-encoding factor $\frac{1}{2}h\alpha_A$. Constructing them will cost us at most $2M_p$ multi-qubit controlled-X gate additionally.

By leveraging the linear combination of unitaries technique [\[47\]](#), we will need two additional ancilla qubits. We construct the preparation oracle as

$$O_{\text{prep}} = \frac{1}{\sqrt{2+h\alpha_A}} \begin{bmatrix} 1 & * & * & * \\ i & * & * & * \\ i\sqrt{\alpha_A h/2} & * & * & * \\ i\sqrt{\alpha_A h/2} & * & * & * \end{bmatrix} \tag{264}$$

and the select oracle to be

$$O_{\text{sel}} = |01\rangle \langle 01| \otimes V_2 + |10\rangle \langle 10| \otimes V_3 + |11\rangle \langle 11| \otimes V_4. \tag{265}$$

Simple computation yields

$$(\overline{O_{\text{prep}}} \otimes I_a \otimes \mathbb{I})^\dagger O_{\text{sel}}(O_{\text{prep}} \otimes I_a \otimes \mathbb{I}) = |00\rangle\langle 00| \frac{1}{2 + h\alpha_A} \left(I_a \otimes \mathbb{I} - V_2 - \frac{\alpha_A h}{2} V_3 - \frac{\alpha_A h}{2} V_4 \right) + P^\perp. \quad (266)$$

Here $\overline{O_{\text{prep}}}$ is the conjugate of O_{prep} , and $P^\perp$ satisfies $(\langle 00| \otimes I_a \otimes \mathbb{I}) P^\perp (|00\rangle \otimes I_a \otimes \mathbb{I}) = 0$. Thus we are able to have a block-encoding for $\mathbf{A}_{M, M_p-1}$ with block-encoding factor $2 + h\alpha_A$. $\square$

F Error analysis of lower-order numerical schemes

In the following, we demonstrate that Forward Euler and Trapezoidal rule have local error scale as $\mathcal{O}(h^2)$ and $\mathcal{O}(h^3)$ respectively, when approximating the time-ordering matrix exponential. For simplicity, we only consider the time interval $[0, h]$, where h is the step size. Furthermore, we would require that $A(t)$ is in C^3 .

F.1 Forward Euler

F.1.1 Evolution operator

Since

$$\left\| \mathcal{T}e^{\int_0^h A(s)ds} - (I + hA(0)) \right\| = \sup_{\|v\|=1} \left\| \left(\mathcal{T}e^{\int_0^h A(s)ds} - (I + hA(0)) \right) v \right\|. \quad (267)$$

Let us now consider

$$\begin{aligned} \frac{du}{dt} &= A(t)u, \\ u(0) &= v, \end{aligned} \quad (268)$$

we can then denote $\mathcal{T}e^{\int_0^h A(s)ds} v$ as $u(h)$.

$$\begin{aligned} \sup_{\|v\|=1} \|u(h) - (I + hA(0))v\| &= \sup_{\|v\|=1} \left\| v + A(0)h + \int_0^h \frac{d^2u}{dt^2} t dt - (I + hA(0))v \right\| \\ &= \sup_{\|v\|=1} \left\| \int_0^h \frac{d^2u}{dt^2} t dt \right\| \leq \sup_{\|v\|=1} \sup_{0 \leq t \leq h} \frac{h^2}{2} \left\| \frac{d^2u}{dt^2} \right\|. \end{aligned} \quad (269)$$

Notice that

$$\frac{d^2u}{dt^2} = \frac{d}{dt}(A(t)u(t)) = A'(t)u(t) + A^2(t)u(t), \quad (270)$$

so we have the estimation

$$\sup_{\|v\|=1} \sup_{0 \leq t \leq h} \left\| \frac{d^2u}{dt^2} \right\| = \sup_{\|v\|=1} \sup_{0 \leq t \leq h} \|A'(t)u(t) + A^2(t)u(t)\| \leq \sup_{0 \leq t \leq h} (\|A(t)\|^2 + \|A'(t)\|). \quad (271)$$

Plugging [Eq. \(271\)](#) back into [Eq. \(269\)](#), we have

$$\left\| \mathcal{T}e^{\int_0^h A(s)ds} - (I + hA(0)) \right\| \leq \frac{1}{2} \sup_{0 \leq t \leq h} (\|A(t)\|^2 + \|A'(t)\|) h^2 = \mathcal{O}(h^2). \quad (272)$$

F.1.2 Inhomogeneous term

Here, we want to bound

$$\left\| hb(h) - \int_0^h \mathcal{T}e^{\int_s^h A(t)dt} b(s)ds \right\| = \mathcal{O}(h^2). \quad (273)$$

Since

$$f(s)|_{s=h} := \mathcal{T}e^{\int_s^h A(t)dt} b(s) a \Big|_{s=h} = b(h), \quad (274)$$

then we have

$$\int_0^h f(s)ds - hf(h) = \int_0^h f(h) + (s-h)f'(h) + \mathcal{O}((s-h)^2) ds - hf(h) = \mathcal{O}(h^2). \quad (275)$$

F.2 Trapezoidal rule

F.2.1 Evolution operator

For the trapezoidal rule, we are estimating

$$\left\| \mathcal{T}e^{\int_0^h A(s)ds} - \left(I - \frac{h}{2}A(h) \right)^{-1} \left(I + \frac{h}{2}A(0) \right) \right\| = \sup_{\|v\|=1} \left\| \left(\mathcal{T}e^{\int_0^h A(s)ds} - \left(I - \frac{h}{2}A(h) \right)^{-1} \left(I + \frac{h}{2}A(0) \right) \right) v \right\|. \quad (276)$$

According to the Neumann series, we know that

$$\begin{aligned} \left(I - \frac{h}{2}A(h) \right)^{-1} \left(I + \frac{h}{2}A(0) \right) &= \sum_{k=0}^{\infty} \left(\frac{h}{2}A(h) \right)^k \left(I + \frac{h}{2}A(0) \right) \\ &= I + \frac{h}{2}A(0) + \frac{h}{2}A(h) + \frac{h^2}{4}A(h)A(0) + \frac{h^2}{4}A^2(h) + \mathcal{O}(h^3). \end{aligned} \quad (277)$$

Now, since

$$A(h) = A(0) + hA'(0) + \frac{h^2}{2}A''(0) + \mathcal{O}(h^3), \quad (278)$$

we can further write [Eq. \(277\)](#) as

$$\left(I - \frac{h}{2}A(h) \right)^{-1} \left(I + \frac{h}{2}A(0) \right) = I + hA(0) + \frac{h^2}{2}A'(0) + \frac{h^2}{2}A^2(0) + \mathcal{O}(h^3). \quad (279)$$

Next, using a similar notation in the above section, we know that

$$\begin{aligned} &\sup_{\|v\|=1} \left\| u(h) - \left(I - \frac{h}{2}A(h) \right)^{-1} \left(I + \frac{h}{2}A(0) \right) v \right\| \\ &= \sup_{\|v\|=1} \left\| v + hu'(0) + \frac{h^2}{2}u''(0) + \frac{1}{2} \int_0^h \frac{d^3u}{dt^3} t^2 dt - \left(I - \frac{h}{2}A(h) \right)^{-1} \left(I + \frac{h}{2}A(0) \right) v \right\|. \end{aligned} \quad (280)$$

Notice that

$$u'(t) = A(t)u(t), \quad (281)$$

$$u''(t) = \frac{d}{dt}u'(t) = \frac{d}{dt}A(t)u(t) = A'(t)u(t) + A^2(t)u(t), \quad (282)$$

$$u'''(t) = \frac{d}{dt}u''(t) = \frac{d}{dt}(A'(t)u(t) + A^2(t)u(t)) = A''(t)u(t) + A'(t)A(t)u(t) + 2A(t)A'(t)u(t) + A^3(t)u(t), \quad (283)$$

we know

$$v + hu'(0) + \frac{h^2}{2}u''(0) = \left(I + hA(0) + \frac{h^2}{2}(A'(0) + A^2(0)) \right) v, \quad (284)$$

Compare [Eq. \(284\)](#) and [Eq. \(279\)](#) and plug everything in then equation [Eq. \(276\)](#) could be $\mathcal{O}(h^3)$.

F.2.2 Inhomogeneous term

Here we need to bound

$$\left\| \left(I - \frac{h}{2} A(h) \right)^{-1} \frac{h}{2} (b(h) + b(0)) - \int_0^h \mathcal{T} e^{\int_s^h A(t) dt} b(s) ds \right\| = \mathcal{O}(h^3), \quad (285)$$

which can be achieved by the following computation

$$\begin{aligned} & \left\| \left(I - \frac{h}{2} A(h) \right)^{-1} \frac{h}{2} (b(h) + b(0)) - \int_0^h \mathcal{T} e^{\int_s^h A(t) dt} b(s) ds \right\| \\ &= \left\| \left(I + \frac{h}{2} A(h) + \mathcal{O}(h^2) \right) \frac{h}{2} (b(h) + b(0)) - \int_0^h \mathcal{T} e^{\int_s^h A(t) dt} b(s) ds \right\| \\ &\leq \left\| \left(I + \frac{h}{2} A(h) + \mathcal{O}(h^2) \right) \frac{h}{2} (b(h) + b(0)) - \frac{h}{2} \left(b(h) + e^{\int_0^h A(t) dt} b(0) \right) \right\| \\ &\quad + \left\| \frac{h}{2} \left(b(h) + e^{\int_0^h A(t) dt} b(0) \right) - \int_0^h \mathcal{T} e^{\int_s^h A(t) dt} b(s) ds \right\| \\ &\leq \left\| \left(I + \frac{h}{2} A(h) + \mathcal{O}(h^2) \right) \frac{h}{2} (b(h) + b(0)) - \frac{h}{2} \left(b(h) + e^{\int_0^h A(t) dt} b(0) \right) \right\| + \mathcal{O}(h^3) \\ &= \left\| \frac{h^2}{4} A(h) b(h) + \left(I + \frac{h}{2} A(h) \right) \frac{h}{2} b(0) - \frac{h}{2} e^{\int_0^h A(t) dt} b(0) \right\| + \mathcal{O}(h^3) \\ &= \left\| \frac{h^2}{4} A(h) b(h) + \left(I + \frac{h}{2} A(h) \right) \frac{h}{2} b(0) - \frac{h}{2} (b(0) + h A(0) b(0)) \right\| + \mathcal{O}(h^3) \\ &= \left\| \frac{h^2}{4} A(0) b(0) + \left(I + \frac{h}{2} A(0) \right) \frac{h}{2} b(0) - \frac{h}{2} (b(0) + h A(0) b(0)) \right\| + \mathcal{O}(h^3) = \mathcal{O}(h^3). \end{aligned} \quad (286)$$