

Quantum algorithms for linear and non-linear fractional reaction-diffusion equations

Dong An¹ and Konstantina Trivisa^{2,3}

¹Joint Center for Quantum Information and Computer Science, University of Maryland, MD 20742, USA

²Department of Mathematics, University of Maryland, MD 20742, USA

³Institute for Physical Science and Technology, University of Maryland, MD 20742, USA

High-dimensional fractional reaction-diffusion equations have numerous applications in the fields of biology, chemistry, and physics, and exhibit a range of rich phenomena. While classical algorithms have an exponential complexity in the spatial dimension, a quantum computer can produce a quantum state that encodes the solution with only polynomial complexity, provided that suitable input access is available. In this work, we investigate efficient quantum algorithms for linear and nonlinear fractional reaction-diffusion equations with periodic boundary conditions. For linear equations, we analyze and compare the complexity of various methods, including the second-order Trotter formula, time-marching method, and truncated Dyson series method. We also present a novel algorithm that combines the linear combination of Hamiltonian simulation technique with the interaction picture formalism, resulting in optimal scaling in the spatial dimension. For nonlinear equations, we employ the Carleman linearization method and propose a block-encoding version that is appropriate for the dense matrices that arise from the spatial discretization of fractional reaction-diffusion equations.

Contents

1	Introduction	2
2	Preliminaries	5
2.1	Setup	5
2.2	Notations	6
2.3	Quantum linear algebra	6
3	Linear equations without potential	7
4	Linear equations with potential	10
4.1	Shifting equivalence	10
4.2	Spatial discretization	10
4.3	Second-order Trotter formula	12
4.3.1	Error bound, commutator scalings, and the number of the time steps	12
4.3.2	Quantum implementation and complexity estimate	13
4.4	Time-marching method	15
4.5	Truncated Dyson series method	17
4.6	Linear combination of Hamiltonian simulation in the interaction picture	18
4.6.1	Representation	19
4.6.2	Numerical quadrature	19
4.6.3	Implementation and complexity	20
5	Nonlinear equations	23
5.1	Spatial discretization	23

5.2 Carleman linearization	23
5.3 Block-encoded implementation	24
6 Conclusion	25
A Technical lemmas	29
B Proof of Theorem 5	29
C Bounding the spatial discretization errors	31
D Bounding the Trotter error	34
D.1 Local error due to the time dependency	34
D.2 Local time-independent Trotter error	35
D.3 Global error	38
E Bounding the commutators	38
F Discretization error in the LCHS-IP method	40

1 Introduction

Reaction–diffusion equations arise in many areas in science and engineering [1, 2, 3, 4, 5]. In population dynamics models in biology, the reaction term typically accounts for growth, whereas the diffusion term accounts for migration [6, 7]. The classical diffusion term has its origin in a model in physics. Recent research investigations indicate that the classical diffusion equation is inadequate to model many real situations, where a particle plume spreads faster than that predicted by the classical model, and may exhibit significant asymmetry [8]. In a fractional diffusion equation, the classical Laplace operator in the spatial variable is replaced by a fractional Laplacian of order less than two. The fundamental solutions of these equations still exhibit useful scaling properties that make them attractive for applications.

The present article deals with

$$\begin{aligned} \partial_t u(t, x) &= -(-\Delta)^{\alpha/2} u(t, x) - c(t, x)u(t, x) + au(t, x)(1 - u(t, x)), \quad t \in [0, T], x \in [0, 1]^d, \\ u(0, x) &= u_0(x). \end{aligned} \quad (1)$$

Here $(-\Delta)^{\alpha/2}$ is the fractional Laplacian where $0 < \alpha \leq 2$, and $c(t, x)$ is the potential function. For the nonlinear term, in this work we only consider the quadratic non-linearity which yields Fisher’s equation, but our results can be generalized to equations with high-order polynomial non-linearity. For the fractional Laplacian operator, there are several different definitions of $(-\Delta)^{\alpha/2}$ on bounded domain, including spectral definition and Riesz definition [9]. This work focuses on spectral fractional Laplacian with periodic boundary condition. Classical numerical algorithms for solving Equation (1) typically require exponential computational resources when the spatial variable is in high dimension: suppose that we use N grid points or basis functions for spatial discretization in each dimension, then the dimension of semi-discretized differential equations becomes as large as N^d . Therefore we would like to explore the power of quantum algorithms for fractional reaction-diffusion equations and whether quantum algorithms can be efficient in high-dimensional case.

Quantum algorithms for differential equations aim at preparing a quantum state encoding the solutions at discrete grid points in its amplitudes. The first quantum differential equation algorithm was proposed in [10], which transforms differential equations into a linear system of equations using multi-step discretization and then applies quantum linear system algorithms such as HHL algorithm [11] or advanced ones [12, 13, 14, 15, 16]. Since then, there have been remarkable progresses on designing better quantum algorithms, for linear differential equations based on refined discretization [17, 18, 19, 20, 21], time-marching strategy [22], Schrödingerization [23], linear combination of unitaries technique [24], and for nonlinear differential equations using linearization techniques [25, 26]. With the caveat that the output is a quantum state encoding solutions

in its amplitudes, these quantum algorithms can achieve exponential speedup in the system size compared to classical algorithms.

However, when directly applied to fractional reaction-diffusion equations, existing generic quantum algorithms are not as efficient as expected due to two major difficulties. First, similar as the standard Laplacian operator, fractional Laplacian operator is an unbounded operator, so its spatially discretized version has a huge spectral norm as the spatial dimension and the number of grid points increase. Most existing quantum differential equation algorithms scale at least linearly on the spectral norm of the coefficient matrix [10, 17, 18, 19, 20, 21, 22, 23, 24], and thus can be computationally expensive in high-dimensional case for accurate simulation. Second, when the equation is genuinely fractional (i.e., $\alpha < 2$), the coefficient matrix for the linear part after spatial discretization is unavoidably dense, because fractional differential operators are global operators that depends on the function evaluated in the entire space. The dense coefficient matrix poses computational difficulties in solving nonlinear equations. This is because all the existing quantum Carleman linearization algorithms require the coefficient matrix to be sparse in order to bypass the difficulty caused by the enlarged Carleman matrix, which is a direct sum of matrices in different dimensions [25].

We remark that the (fractional- or integer-order) Laplacian operator, and more general spatial differential operators, widely appear in various types of partial differential equations. In the contexts other than fractional reaction-diffusion equations, there have been several work managing to overcome the computational difficulty brought by its large spectral norm after spatial discretization, but unfortunately those techniques do not apply to the fractional reaction-diffusion equations. For example, in real-space Schrödinger equation, a poly-logarithmic dependence on the spectral norm of the Laplacian operator can be achieved by simulating the Hamiltonian in the interaction picture [27, 28], which simulates the transformed wavefunction under the rotation associated with the Laplacian operator. The resulting interaction picture Hamiltonian becomes bounded and thus can be efficiently simulated. To avoid the large spectral norm dependence in the rotations, the algorithm takes advantage of an important feature of the Laplacian operator: it can be diagonalized by the quantum Fourier transform (QFT) circuit and thus can be fastforwarded (i.e., Hamiltonian simulation governed by the Laplacian operator can be implemented with cost independent of evolution time and its spectral norm). However, such a technique does not directly work for fractional reaction-diffusion equations. The interaction picture transformation requires both forward and backward time evolution of the Laplacian operator. This is efficient for Schrödinger equations because the dynamics is reversible and both forward and backward time evolution operators are unitary, but fractional reaction-diffusion equation is a dissipative system and implementing its backward time evolution can be prohibitively expensive. The recent work [29] proposes efficient quantum algorithms for various partial differential equations beyond Schrödinger equations. The key technique there is a generalization of the fastforwarding simulation of the Laplacian operator. However, the algorithms in [29] require the entire coefficient matrix of the linear part to be fast-forwardable (i.e., its spectral decomposition has quantumly implementable eigenstates and classically computable eigenvalues), which cannot be satisfied even by linear fractional reaction-diffusion equations with the presence of the potential $c(t, x)$.

In this work, we investigate efficient quantum algorithms for fractional reaction-diffusion equations. The majority of our work is devoted to linear fractional reaction-diffusion equations. We numerically treat the equations by the method of lines, i.e., first discretizing the spatial variable to obtain a system of ODEs

$$\frac{d}{dt}\vec{u} = -B\vec{u} - C(t)\vec{u}, \quad (2)$$

and then solving the resulting ODE system with different quantum ODE algorithms. Here $\vec{u}$ represents the solution evaluated at different spatial grid points, B is the discretized fractional Laplacian operator, and $C(t)$ is the potential matrix. For time evolution, we consider four different algorithms: second-order Trotter formula [30], time-marching method [22], truncated Dyson series method [21], and linear combination of Hamiltonian simulation in the interaction picture (LCHS-IP), which is a novel method that combines the techniques in the two references [24, 27]. We analyze the complexity of these four methods and the results are shown in Table 1. Our main results and contributions are summarized as follows:

1. Second-order Trotter formula: unlike Hamiltonian simulation, Equation (2) is not a unitary

Method	Queries to the matrices			
	d	ϵ	T	Norm
Second-order Trotter (Theorem 12)	$\tilde{\mathcal{O}}(d^{\alpha(1/2+\sigma/2)})$	$\tilde{\mathcal{O}}(\epsilon^{-1/2})$	$\tilde{\mathcal{O}}(T^{3/2})$	$\tilde{\mathcal{O}}((g(T))^{3/2})$
Time-marching (Theorem 14)	$\tilde{\mathcal{O}}(d^{\alpha(1+2\sigma)})$	$\mathcal{O}(\text{poly log}(1/\epsilon))$	$\tilde{\mathcal{O}}(T^2)$	$\tilde{\mathcal{O}}(Q)$
Dyson series (Theorem 16)	$\tilde{\mathcal{O}}(d^{\alpha(1/2+\sigma)})$	$\mathcal{O}(\text{poly log}(1/\epsilon))$	$\tilde{\mathcal{O}}(T)$	$\tilde{\mathcal{O}}(g(T))$
LCHS-IP (Theorem 18)	$\tilde{\mathcal{O}}(\text{poly log}(d))$	$\tilde{\mathcal{O}}(\epsilon^{-1})$	$\tilde{\mathcal{O}}(T)$	$\tilde{\mathcal{O}}(g(T)^2)$

Method	Queries to the state preparation			
	d	ϵ	T	Norm
Second-order Trotter (Theorem 12)	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(g(T))$
Time-marching (Theorem 14)	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(Q)$
Dyson series (Theorem 16)	$\tilde{\mathcal{O}}(d^{\alpha(1/2+\sigma)})$	$\mathcal{O}(\text{poly log}(1/\epsilon))$	$\tilde{\mathcal{O}}(T)$	$\tilde{\mathcal{O}}(g(T))$
LCHS-IP (Theorem 18)	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(g(T))$

Table 1: Query complexities of differential methods for linear fractional reaction-diffusion equations. Here d is the spatial dimension, ϵ is the tolerated error in 2-norm, T is the evolution time, α is the half order of the fractional Laplacian operator ranging in $(0, 2]$ and σ is the parameter of the Gevrey class defined in Equation (40). The function $g(T) \geq \|\vec{u}(0)\|/\|\vec{u}(T)\|$ describes the decay of the spatially discretized solution, Q is the decay corrected by the spectral norm of infinitesimal evolution operators as defined in Equation (67), and we always have $Q \leq g(T)$.

dynamics, and in the Trotter formula we need to implement non-unitary operators e^{-Bs} and $e^{-C(t)s}$. We discuss efficient construction of these operators via controlled rotations, and implementing their multiplication through a generalization of the compression gadget technique [22, 27] with only poly-logarithmic many ancilla qubits. For complexity analysis, we derive an improved Trotter error bound that avoids the exponential factor in [30] and generalizes to the time-dependent case. Compared to time-marching and truncated Dyson series methods, second-order Trotter has better dependence on the dimension d thanks to its commutator scalings, but has worse dependence on the precision.

2. Time-marching method: we directly apply the standard time-marching method in [22] and analyze its complexity for fractional reaction-diffusion equations. It has low state preparation cost and poly-logarithmic dependence on precision, but has worse scalings in the dimension d and the evolution time T .
3. Truncated Dyson series method: we directly apply the standard truncated Dyson series method in [21] and analyze its complexity for fractional reaction-diffusion equations. It still depends polynomially on the dimension due to its spectral norm dependence and has high state preparation cost due to the usage of quantum linear system algorithms, but can achieve poly-logarithmic scaling in the precision and linear scaling in time simultaneously.
4. LCHS-IP: The LCHS-IP method is a novel method that combines the linear combination of Hamiltonian simulation (LCHS) [24] technique with the interaction picture Hamiltonian simulation [27]. The LCHS method first represents the evolution operator of Equation (2) as a linear combination of several Hamiltonian simulation problems associated with the matrices B and $C(t)$. To avoid the computational overhead brought by the discretized fractional Laplacian B , we implement each Hamiltonian simulation in the interaction picture by rotating the Hamiltonian with respect to B . Therefore, the resulting algorithm only has poly-logarithm dependence on the dimension d and thus is the most preferable algorithm in the high-dimensional case. It also has low state preparation cost, but only linear scaling in precision.

For nonlinear fractional reaction-diffusion equations, we discuss a block-encoding version of the Carleman linearization technique to deal with the dense coefficient matrix. Let $\vec{u}(t)$ denote

the solution vector of the fractional reaction-diffusion equation after spatial discretization. Standard quantum Carleman linearization algorithm [25] considers the dynamics of the enlarged vector $[\vec{u}(t); \vec{u}(t)^{\otimes 2}; \dots; \vec{u}(t)^{\otimes M}]$, which approximately satisfies a linear system of ODEs governed by a so-called Carleman matrix. Since $\vec{u}(t)^{\otimes m}$'s are in different sizes for different powers, the Carleman matrix is a direct sum of matrices in different dimensions. In our work, we propose a simple generalization of the Carleman linearization, by extending the Carleman matrix to even higher dimension such that it becomes the direct sum of matrices in the same dimension and the corresponding solution vector $[\vec{w}_1; \vec{w}_2; \dots; \vec{w}_M]$ can be exactly mapped to the solution $[\vec{u}(t); \vec{u}(t)^{\otimes 2}; \dots; \vec{u}(t)^{\otimes M}]$ via discarding all the zero entries. Then the extended Carleman matrix can be easily block-encoded through the block-encoding of the original coefficient matrix, and thus the corresponding extended linearized system can be solved via quantum linear differential algorithms with block-encoding input model (e.g., [21, 22, 24]). This makes the quantum Carleman linearization algorithm applicable to the equations with dense coefficient matrices, including the fractional reaction-diffusion equations.

The rest of this paper is organized as follows. Section 2 first discusses the mathematical setup of the fractional reaction-diffusion equations, notations being used throughout the paper and some preliminary results. The main results of this paper start by Section 3 with the simplest case where the equations only involve a single fractional Laplacian operator without potential or nonlinear term, then Section 4 considers the general linear equations with potential. Quantum algorithm for nonlinear equations is presented in Section 5, followed by conclusion and open questions in Section 6.

2 Preliminaries

We start with a more rigorous setup of the fractional reaction-diffusion equation we are interested in and a summary of theoretical tools and technical lemmas being used in our analysis.

2.1 Setup

We consider the spatial fractional reaction-diffusion equations Equation (1). Let N be a positive integer and we discretize the spatial variable x using equi-distant nodes (j/N) , $j \in [N]^d$. Quantum algorithms for solving Equation (1) aim at preparing a quantum state approximately encoding $u(T, j/N)$ in its amplitude, i.e., $\frac{1}{\|u(T, j/N)\|_{j \in [N]^d}} \sum_{j \in [N]^d} u(T, j/N) |j\rangle$.

There are several different definitions of $(-\Delta)^{\alpha/2}$ on bounded domain (see the recent paper [9] for a comprehensive review). The *spectral fractional Laplacian* is defined using the eigenvalues and eigenfunctions of the original Laplacian $(-\Delta)$. Suppose that λ_j 's are the eigenvalues of $(-\Delta)$ and $e_j(x)$'s are the corresponding eigenfunctions, then the spectral fractional Laplacian is defined to be

$$(-\Delta)^{\alpha/2} v(x) = \sum_j \lambda_j^{\alpha/2} (v, e_j)_{L^2} e_j(x) \quad (3)$$

where $(\cdot, \cdot)_{L^2}$ denotes the L^2 inner product on $[0, 1]^d$. An alternative definition is the *Riesz fractional Laplacian*

$$(-\Delta)^{\alpha/2} v(x) = \frac{2^\alpha \Gamma(\frac{\alpha}{2} + \frac{d}{2})}{\pi^{d/2} |\Gamma(-\frac{\alpha}{2})|} \text{p.v.} \int_{\mathbb{R}^d} \frac{v(x) - v(y)}{|x - y|^{d+\alpha}} dy. \quad (4)$$

Here $\Gamma(s)$ denotes the gamma function and p.v. refers to the principle value integral. Notice that the definition Equation (4) is not closed yet since the integral is over the entire space $\mathbb{R}^d$ while $v(x)$ is confined in the cube $[0, 1]^d$. Therefore we must enforce boundary condition on $\mathbb{R}^d \setminus [0, 1]^d$, e.g., the homogeneous Dirichlet boundary condition $v(x) = 0$ for all $x \in \mathbb{R}^d \setminus [0, 1]^d$. For periodic boundary condition, the definition is unambiguous that we will always follow Equation (3), while either definition is commonly used with other boundary conditions. In this work, we will focus on the periodic boundary condition and use the spectral definition for the fractional Laplacian operator.

2.2 Notations

We use $\vec{a}$ (a letter with an array above) to denote a possibly unnormalized vector. When $u(t, x)$ refers to a scalar-valued function with arguments $t \in \mathbb{R}$ and $x \in [0, 1]^d$, we use $(u(t, j/N))_{j \in [N]^d}$ or simply $(u(t, j/N))$ to denote the N^d -dimensional vector with entries $u(t, j/N)$, where $j \in [N]^d$ and $[N] = \{0, 1, \dots, N-1\}$ for a positive integer N . In our analysis, we also use $\vec{u}_0$ as a shorthand notation of the initial condition vector $(u(0, j/N))$, and $\vec{u}(t)$ as $(u(t, j/N))$ for a fixed time t .

For a vector $\vec{a}$, we use $\|\vec{a}\|$ without subscript for the standard vector 2-norm of $\vec{a}$, and $\|\vec{a}\|_p$ for vector p -norm. The notation $|\vec{a}\rangle$ with ket notation denotes the corresponding quantum state, i.e., the normalized vector $\vec{a}/\|\vec{a}\|$.

Let A and B be two matrices. We use $[A, B]$ to denote the commutator between A and B , defined as $AB - BA$. For a matrix A , $\|A\|$ denotes its spectral norm or equivalently the matrix 2-norm.

2.3 Quantum linear algebra

To design and analyze quantum algorithms for the spatial fractional reaction-diffusion equations, we need to frequently implement linear algebra operations, including matrix-vector multiplication, matrix-matrix addition and multiplication. To this end, we briefly introduce the concept and properties of block-encoding, which is a widely-used quantum input model for possibly non-unitary matrix.

We start with the definition of block-encoding.

Definition 1 (Block-encoding). *Suppose that A is a 2^s -dimensional matrix, then we say that the $(s+n)$ -qubit unitary U is an (α, n, ϵ) -block-encoding of A , if*

$$\left\| A - \alpha \left((|0\rangle^{\otimes n} \otimes I) U (|0\rangle^{\otimes n} \otimes I) \right) \right\| \leq \epsilon. \quad (5)$$

Intuitively, in the block-encoding, a matrix is represented as the upper-left block of a unitary matrix as

$$U \approx \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}. \quad (6)$$

α is called the block-encoding factor such that $\alpha \geq \|A\|$, since the larger matrix U is supposed to be unitary and the norm of its sub-block should be bounded by 1.

With block-encoding structure, we may implement matrix linear algebra operations. For example, matrix addition can be directly implemented by the linear combination of unitaries (LCU) technique, which has become an important subroutine in designing various quantum algorithms including Hamiltonian simulation [31], solving linear systems [12] and differential equations [24]. Here we present the LCU lemma from [32].

Lemma 2. *Let $A = \sum_{j=0}^{m-1} y_j A_j$ and $\|y\|_1 \leq \beta$. Suppose that (P_L, P_R) is a pair of b -qubit unitaries such that $P_L |0\rangle = \sum_{j=0}^{m-1} c_j |j\rangle$ and $P_R |0\rangle = \sum_{j=0}^{m-1} d_j |j\rangle$ with $\sum_{j=0}^{m-1} |\beta(c_j^* d_j - y_j)| < \epsilon_1$, and $W = \sum_{j=0}^{m-1} |j\rangle \langle j| \otimes U_j$ where U_j is a (α, a, ϵ_2) -block-encoding of A_j . Then we can construct a $(\alpha\beta, a+b, \alpha\epsilon_1 + \alpha\beta\epsilon_2)$ -block-encoding of A with a single use of W , $P_L^\dagger$ and P_R .*

Matrix multiplication can also be implemented via block-encodings. For two arbitrary matrices A and B , a straightforward approach of constructing the block-encoding of AB is to multiply together the block-encodings of A and B , as shown in the following lemma [32].

Lemma 3. *If U_A is an $(\alpha_A, n_A, \epsilon_A)$ -block-encoding of A , and U_B is an $(\alpha_B, n_B, \epsilon_B)$ -block-encoding of B , then $(I_{n_B} \otimes U_A)(I_{n_A} \otimes U_B)$ is an $(\alpha_A \alpha_B, n_A + n_B, \alpha_A \epsilon_B + \alpha_B \epsilon_A)$ -block-encoding of AB .*

Despite its simplicity, such a straightforward approach for matrix multiplication may incur large space overhead. This is because we need to enlarge the ancilla register at each step of the multiplication, so the multiplication of J many matrices will in general require $\mathcal{O}(J)$ ancilla qubits. To overcome this issue, [27] introduces a technique called compression gadget, which is further simplified in [22]. The circuit is given in Figure 1, and the idea of compression gadget is

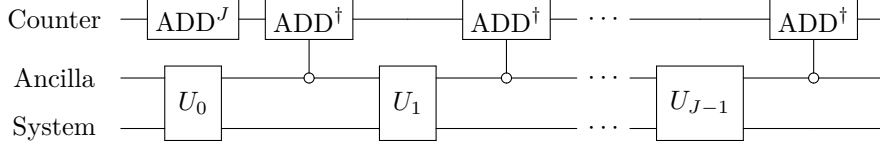


Figure 1: Quantum circuit for compression gadget to block encode $A_{J-1} \cdots A_1 A_0$. Here the Counter register contains $\lceil \log_2 J \rceil + 1$ qubits and the ancilla register contains $\max n_j$ qubits. ADD implements addition by 1 modulo $2^{\lceil \log_2 J \rceil + 1}$, and U_j is an $(\alpha_j, n_j, 0)$ -block-encoding of the matrix A_j .

to use a counter register to keep track of the multiplication in a coherent way. Specifically, let ADD implements addition by 1 modulo the 2-power of the number of the qubits in the counter register. One can first apply ADD to lift the counter register from 0 to $L - 1$. Then we sequentially apply the block-encodings using the same ancilla register, but after each block-encoding we apply a controlled $\text{ADD}^\dagger$ to reduce the counter by 1 if the corresponding block-encoding was successfully implemented. Therefore, a 0 final outcome of the counter register implies successful application of all the block-encodings. We present the result in [22], in which interested readers may find more details on its proof.

Lemma 4. For $0 \leq j \leq J - 1$, let U_j be an $(\alpha_j, n_j, 0)$ -block-encoding of A_j . Then an $(\alpha, n, 0)$ -block-encoding of $A_{J-1} \cdots A_1 A_0$ can be constructed using one application of each U_j , where $\alpha = \alpha_0 \alpha_1 \cdots \alpha_{J-1}$ and $n = \max n_j + \lceil \log_2(J) \rceil + 1$.

3 Linear equations without potential

We start with the simplest case where $c(t, x) = 0$ and $a = 0$. In this case, the right hand side of Equation (1) only involves a fractional Laplacian operator. By the spectral definition, this operator has closed-form eigenvalues and known eigenfunctions, so the corresponding time-evolution operator can be implemented fast-forwardly [29] (*i.e.*, query complexity is independent of the spectral norm and the evolution time). Here we present our algorithm in general high dimension and establish rigorous complexity estimate taking into consideration the spatial discretization errors.

We consider general d -dimensional case. The eigenvalues and eigenfunctions of $(-\Delta)$ are $4\pi^2(k_0^2 + \cdots + k_{d-1}^2)$ and $e^{2\pi i(k_0 x_0 + \cdots + k_{d-1} x_{d-1})}$, where $x = (x_0, \cdots, x_{d-1})$ denotes the spatial coordinate and $k = (k_0, \cdots, k_{d-1})$ denotes a set of integers. Let the Fourier series of $u_0(x)$ be

$$u_0(x) = \sum_{k \in \mathbb{Z}^d} \hat{u}_k e^{2\pi i(k_0 x_0 + \cdots + k_{d-1} x_{d-1})}, \quad (7)$$

where $\hat{u}_k$ denotes the Fourier coefficients. Then the solution of Equation (1) has the form

$$u(T, x) = \sum_{k \in \mathbb{Z}^d} \hat{u}_k e^{-(4\pi^2(k_0^2 + \cdots + k_{d-1}^2))^{\alpha/2} T} e^{2\pi i(k_0 x_0 + \cdots + k_{d-1} x_{d-1})}. \quad (8)$$

Numerical solutions can be obtained by truncating the Fourier series at a finite order.

Equation (8) can be quantumly implemented using quantum Fourier transform (QFT) and controlled rotations. Recall that our goal is to prepare an approximation of the quantum state encoding the normalized solution at discrete spatial grid points $(j/N)_{j \in [N]^d}$. Let N be the number of the grid points in each spatial dimension. Suppose that we are given the oracle O_{u_0} that prepares the normalized initial condition

$$|u_0\rangle = \frac{1}{\|\vec{u}_0\|} \sum_{n \in [N]^d} u_0(n_0/N, \cdots, n_{d-1}/N) |n_0\rangle \cdots |n_{d-1}\rangle, \quad (9)$$

where

$$\vec{u}_0 = \sum_{n \in [N]^d} u_0(n_0/N, \cdots, n_{d-1}/N) |n_0\rangle \cdots |n_{d-1}\rangle. \quad (10)$$

We first compute a quantum state encoding the Fourier coefficients $\hat{u}_k$ by QFT. Specifically, let $\omega_N = e^{2\pi i/N}$ and $\mathcal{F}$ denote the one-dimensional QFT, *i.e.*, for any computational basis state $|j\rangle$,

$$\mathcal{F}|j\rangle = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} \omega_N^{jl} |l\rangle. \quad (11)$$

Then

$$(\mathcal{F}^{-1})^{\otimes d} \vec{u}_0 = \frac{1}{N^{d/2}} \sum_{m \in [N]^d} u_0(n_0/N, \dots, n_{d-1}/N) \left(\sum_{m_0=0}^{N-1} \omega_N^{-n_0 m_0} |m_0\rangle \right) \cdots \left(\sum_{m_{d-1}=0}^{N-1} \omega_N^{-n_{d-1} m_{d-1}} |m_{d-1}\rangle \right) \quad (12)$$

$$= \frac{1}{N^{d/2}} \sum_{m \in [N]^d} \sum_{n \in [N]^d} u_0(n_0/N, \dots, n_{d-1}/N) \omega_N^{-n_0 m_0 - \dots - n_{d-1} m_{d-1}} |m_0\rangle \cdots |m_{d-1}\rangle \quad (13)$$

$$= \frac{1}{N^{d/2}} \sum_{m \in [N]^d} \sum_{n \in [N]^d} \sum_{k \in \mathbb{Z}^d} \hat{u}_k e^{2\pi i(k_0 n_0/N + \dots + k_{d-1} n_{d-1}/N)} \omega_N^{-n_0 m_0 - \dots - n_{d-1} m_{d-1}} |m_0\rangle \cdots |m_{d-1}\rangle \quad (14)$$

$$= \frac{1}{N^{d/2}} \sum_{m \in [N]^d} \sum_{k \in \mathbb{Z}^d} \hat{u}_k \left(\sum_{n_0=0}^{N-1} \omega_N^{n_0(k_0 - m_0)} \right) \cdots \left(\sum_{n_{d-1}=0}^{N-1} \omega_N^{n_{d-1}(k_{d-1} - m_{d-1})} \right) |m_0\rangle \cdots |m_{d-1}\rangle \quad (15)$$

$$= N^{d/2} \sum_{m \in [N]^d} \sum_{j \in \mathbb{Z}^d} \hat{u}_{m+jN} |m_0\rangle \cdots |m_{d-1}\rangle. \quad (16)$$

When the function $u_0(x)$ satisfies certain regularity assumption (*i.e.*, at least $(d+2)$ -th order continuously differentiable specified in [Theorem 5](#)), the summation $\sum_{j \in \mathbb{Z}^d} \hat{u}_{m+jN}$ is dominated by the index with smallest absolute value, because the Fourier coefficients decay rapidly with respect to its frequency. Therefore

$$(\mathcal{F}^{-1})^{\otimes d} |\vec{u}_0\rangle \approx \frac{N^{d/2}}{\|\vec{u}_0\|} \sum_{m \in [N]^d} \hat{u}_{i(m)} |m_0\rangle \cdots |m_{d-1}\rangle, \quad (17)$$

where $i(m) = (i_0(m), \dots, i_{d-1}(m))$ represents an d -dimensional vector with the j -th index

$$i_j(m) = \begin{cases} m_j, & \text{if } 0 \leq m_j \leq N/2 \\ m_j - N, & \text{if } N/2 + 1 \leq m_j \leq N-1. \end{cases} \quad (18)$$

Now we append three ancilla registers to [Equation \(17\)](#) and get the (approximate) quantum state

$$\frac{N^{d/2}}{\|\vec{u}_0\|} \sum_{m \in [N]^d} \hat{u}_{i(m)} |m_0\rangle \cdots |m_{d-1}\rangle |0\rangle |0\rangle |0\rangle. \quad (19)$$

The first two ancilla registers are used to binarily encode $e^{-(4\pi^2(i_0(m)^2 + \dots + i_{d-1}(m)^2))^{\alpha/2} T}$ and the third register is used for rotation. Specifically, suppose that we are given the oracle that encodes the eigenvalues as

$$O_1 : |m_0\rangle \cdots |m_{d-1}\rangle |0\rangle \rightarrow |m_0\rangle \cdots |m_{d-1}\rangle |(2\pi\|i(m)\|)^{\alpha}\rangle, \quad (20)$$

and the oracle for computing an exponential function as

$$O_{\text{exp},1} : |x\rangle |0\rangle \rightarrow |x\rangle |e^{-xT}\rangle. \quad (21)$$

Notice that both functions have closed-form expression so the oracles can be efficiently constructed using classical arithmetic [\[33\]](#). Applying O_1 on the index and first ancilla registers and then $O_{\text{exp},1}$

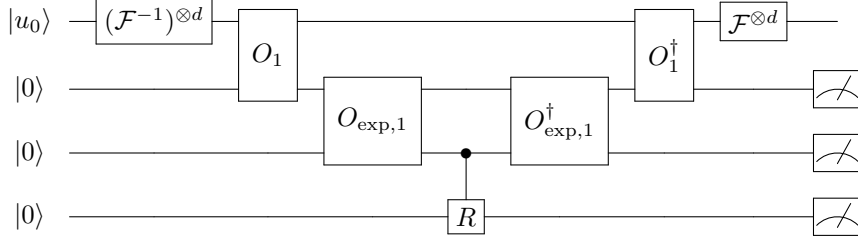


Figure 2: Quantum circuit for solving linear fractional reaction-diffusion equations without potential. Here O_1 , $O_{\text{exp},1}$ are the oracles defined in Equation (20) and Equation (21), and R is the single-qubit rotation gate defined in Equation (23).

on the first and second ancilla registers yields

$$\frac{N^{d/2}}{\|\vec{u}_0\|} \sum_{m \in [N]^d} \hat{u}_{i(m)} |m_0\rangle \cdots |m_{d-1}\rangle ||i(m)|| \rangle |e^{-(2\pi\|i(m)\|)^{\alpha}T}\rangle |0\rangle. \quad (22)$$

Let $\text{c-}R$ denote the controlled rotation

$$\text{c-}R : |\theta\rangle |0\rangle \rightarrow |\theta\rangle \left(\theta |0\rangle + \sqrt{1-|\theta|^2} |1\rangle \right). \quad (23)$$

Then, by applying $\text{c-}R$ on the last two registers in Equation (22), we obtain

$$\frac{N^{d/2}}{\|\vec{u}_0\|} \sum_{m \in [N]^d} \hat{u}_{i(m)} e^{-(2\pi\|i(m)\|)^{\alpha}T} |m_0\rangle \cdots |m_{d-1}\rangle ||i(m)|| \rangle |e^{-(2\pi\|i(m)\|)^{\alpha}T}\rangle |0\rangle + |\perp\rangle, \quad (24)$$

where $|\perp\rangle$ represents the orthogonal part with 1 in the last ancilla register. Uncomputing the first two ancilla registers yields

$$\frac{N^{d/2}}{\|\vec{u}_0\|} \sum_{m \in [N]^d} \hat{u}_{i(m)} e^{-(2\pi\|i(m)\|)^{\alpha}T} |m_0\rangle \cdots |m_{d-1}\rangle |0\rangle |0\rangle |0\rangle + |\perp\rangle, \quad (25)$$

Notice that, by replacing $u_0(x)$ with $u(T, x)$ in Equation (17) and using Equation (8), we have

$$(\mathcal{F}^{-1})^{\otimes d} \vec{u}(T, x) \approx N^{d/2} \sum_{m \in [N]^d} \hat{u}_{i(m)} e^{-(2\pi\|i(m)\|)^{\alpha}T} |m_0\rangle \cdots |m_{d-1}\rangle. \quad (26)$$

Therefore, applying the QFT $\mathcal{F}^{\otimes d}$ to Equation (25) approximately gives

$$\frac{1}{\|\vec{u}_0\|} \sum_{n \in [N]^d} u(T, n/N) |n_0\rangle \cdots |n_{d-1}\rangle |0\rangle |0\rangle |0\rangle + |\perp\rangle. \quad (27)$$

Measuring the ancilla registers to get all 0's yields an approximation of $|u(T)\rangle$, and the averaged number of repeats for success after amplitude amplification scales $\mathcal{O}(\|\vec{u}_0\|/\|\vec{u}(T)\|)$.

We summarize the quantum algorithm in Figure 2 and present its error and complexity estimates in the following result. The main result is that the overall error will decrease as the number of the grid points increases, since more grid points imply larger truncation order in the Fourier series. Detailed proof can be found in Section B.

Theorem 5. Consider solving Equation (1) with $a = c = 0$. Suppose that we are given the state preparation oracle O_{u_0} and the oracles O_1 , $O_{\text{exp},1}$ as defined in Equation (20) and Equation (21). Furthermore, suppose that there exists an integer $p \geq d + 2$ such that $u(t, x)$ is p -th order spatially continuously differentiable. Then there exists a quantum algorithm that prepares an approximation of the quantum state $|\vec{u}(T)\rangle$ with success probability $\Omega(1)$ ¹ and 2-norm error at most

$$\frac{4 \max_{j,t \in \{0,T\}} \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\|\vec{u}(T)\| (\pi/2)^p N^{p-d}}, \quad (28)$$

¹The $\Omega(1)$ here is referred to all variables, i.e., an absolute constant that is smaller than 1 (e.g., 0.9 or 0.99), so the success probability is bounded from below by it (e.g., 0.9 or 0.99).

using $\mathcal{O}(\|\vec{u}_0\|/\|\vec{u}(T)\|)$ queries to $O_{u_0}, O_1, O_{\text{exp},1}$, their inverse and controlled versions, and $\mathcal{O}((\|\vec{u}_0\|/\|\vec{u}(T)\|)d\log^2(N))$ additional gates.

4 Linear equations with potential

Consider the linear spatial fractional reaction-diffusion equations

$$\begin{aligned}\partial_t u(t, x) &= -(-\Delta)^{\alpha/2} u(t, x) - c(t, x)u(t, x), \quad t \in [0, T], x \in [0, 1]^d, \\ u(0, x) &= u_0(x).\end{aligned}\tag{29}$$

Throughout this section, we assume the potential function $c(t, x)$ to be non-negative. We will first discuss the validity of this assumption by so-called shifting equivalence. The equation [Equation \(29\)](#) can then be treated by the method of lines, *i.e.*, first discretizing the spatial variable to obtain a system of ODEs, and then solving the ODE with different quantum algorithms. We consider four different quantum algorithms: second-order Trotter formula, time-marching method, truncated Dyson series method, and linear combination of Hamiltonian simulation in the interaction picture.

4.1 Shifting equivalence

We consider the shifted PDE

$$\begin{aligned}\partial_t v(t, x) &= -(-\Delta)^{\alpha/2} v(t, x) - \tilde{c}(t, x)v(t, x), \quad t \in [0, T], x \in [0, 1]^d, \\ v(0, x) &= u_0(x).\end{aligned}\tag{30}$$

Here

$$\tilde{c}(t, x) = c(t, x) - \gamma(t)\tag{31}$$

where $\gamma(t)$ is a real scalar-valued function. An important observation is that [Equation \(29\)](#) and [Equation \(30\)](#) are *quantumly* equivalent in the sense that the normalized solutions are the same. Specifically, let u denote the solution of [Equation \(29\)](#), then

$$\begin{aligned}\partial_t (e^{\int_0^t \gamma(s) ds} u) &= e^{\int_0^t \gamma(s) ds} \partial_t u + \gamma(t) e^{\int_0^t \gamma(s) ds} u \\ &= e^{\int_0^t \gamma(s) ds} \left(-(-\Delta)^{\alpha/2} u - c(t, x)u \right) + \gamma(t) e^{\int_0^t \gamma(s) ds} u \\ &= -(-\Delta)^{\alpha/2} (e^{\int_0^t \gamma(s) ds} u) - \tilde{c}(t, x) (e^{\int_0^t \gamma(s) ds} u).\end{aligned}\tag{32}$$

Therefore $e^{\int_0^t \gamma(s) ds} u$ solves [Equation \(30\)](#) and only differs u by a multiplicative constant factor at fixed time, which implies that they are the same in a quantum state representation, *i.e.*,

$$\frac{1}{\|\vec{u}\|} \sum_{n \in [N]^d} u(t, n/N) |n\rangle = \frac{1}{\|e^{\int_0^t \gamma(s) ds} \vec{u}\|} \sum_{n \in [N]^d} e^{\int_0^t \gamma(s) ds} u(t, n/N) |n\rangle.\tag{33}$$

We will choose the shifting parameter to be

$$\gamma(t) = \min_x c(t, x).\tag{34}$$

Therefore the function $\tilde{c}(t, x)$ is non-negative. For notation simplicity, we will directly assume that the original $c(t, x)$ in [Equation \(29\)](#) to be non-negative instead of introducing new notations with tildes.

4.2 Spatial discretization

Our goal is to prepare a quantum state encoding $u(T, x)$ at equi-distant grid points $(n_0/N, \dots, n_{d-1}/N)$ where $n_j \in [N]$. Motivated by the spectral decomposition, we define

$$B = (\mathcal{F})^{\otimes d} D (\mathcal{F}^{-1})^{\otimes d}\tag{35}$$

where $\mathcal{F}$ is the one-dimensional quantum Fourier transform defined in Equation (11), and D is an N^d -dimensional diagonal matrix

$$D = \sum_{n \in [N]^d} 2^\alpha \pi^\alpha \|i(n)\|^\alpha |n_0\rangle \cdots |n_{d-1}\rangle \langle n_0| \cdots \langle n_{d-1}|. \quad (36)$$

For each t , let $C(t)$ be an N^d -dimensional diagonal matrix

$$C(t) = \sum_{n \in [N]^d} c(t, n/N) |n_0\rangle \cdots |n_{d-1}\rangle \langle n_0| \cdots \langle n_{d-1}|. \quad (37)$$

Then we consider the spatially discretized equation

$$\begin{aligned} \frac{d}{dt} \vec{u} &= -B\vec{u} - C(t)\vec{u}, \quad t \in [0, T] \\ \vec{u}(0) &= \vec{u}_0. \end{aligned} \quad (38)$$

In order to bound the spatial discretization error by ϵ , we need to choose a sufficiently large N . In the following, we derive an error bound in terms of N . The proof of this result can be found in Section C.

Lemma 6. *Let u be the exact solution of Equation (29) and $\vec{u}$ be the solution of Equation (38). Suppose $u(t, x)$ is p -th order spatially continuously differentiable where $p \geq d + \alpha + 2$, then*

$$\|(u(T, n/N))_{n \in [N]^d} - \vec{u}(T)\| \leq T \frac{2^{p+1} d^{\alpha/2} \max_{t,j} \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-d-\alpha}}. \quad (39)$$

Theorem 6 tells that, similar to the case without potential, the order of the spatial discretization error convergence depends on the smoothness of the solution. In particular, it can be exponential convergence if the solution is within the Gevrey class. The Gevrey class includes infinitely differentiable functions whose p -th order derivative grows polynomially in $p!$. Notice that Gevrey class greatly enlarges the class of real analytic functions, since the Taylor series will only have convergence radius 0 if the p -th order derivative scales super-linearly in $p!$. We give the explicit error bounds and the choice of the grid for bounded errors in the next two results. Their proof can be found in Section C as well.

Corollary 7. *Consider solving Equation (29) on discrete grid points $(T, n/N)$ where $n \in [N]^d$. Suppose that the exact solution $u(t, x)$ is in the Gevrey class G^σ in the sense that $u(t, x)$ is smooth and there exist constants $\Lambda > 0$ and $\sigma \geq 0$ such that*

$$\sup_{j \in [d], t \in [0, T], x \in [0, 1]^d} |\partial_{x_j}^p u(t, x)| \leq \Lambda^{p+1} (p!)^\sigma. \quad (40)$$

Then, for any $N \geq (2\Lambda/\pi)(d + \alpha + 2)^\sigma$, we have

$$\|(u(T, n/N))_{n \in [N]^d} - \vec{u}(T)\| \leq c_1 T (c_2 d)^{c_3 d} d^{\alpha/2} \exp(-c_4 N^{1/\sigma}), \quad (41)$$

where c_j 's are constants only depending on σ, α and Λ .

Corollary 8. *Consider solving Equation (29) on discrete grid points $(T, n/N)$ where $n \in [N]^d$. Suppose that the exact solution $u(t, x)$ is in the Gevrey class G^σ as in Theorem 7. Then, in order to bound the spatial discretization error in the quantum state (i.e., $\|(u(T, n/N))_{n \in [N]^d} - |\vec{u}(T)\rangle\|$) by ϵ , it suffices to choose*

$$N = \mathcal{O} \left(\left(d \log d + \log \left(\frac{T}{\|(u(T, n/N))_{n \in [N]^d}\|} \right) + \log \left(\frac{1}{\epsilon} \right) \right)^\sigma \right). \quad (42)$$

Before we proceed to the algorithm design and analysis, we would like to further discuss the relation between the fractional Laplacian $(-\Delta)^{\alpha/2}$ and its discretized version B . First, notice that

by the definition of B in Equation (35) and Equation (36), the operator norm of B can be bounded as

$$\|B\| = \|(\mathcal{F})^{\otimes d} D (\mathcal{F}^{-1})^{\otimes d}\| = \|D\| = \max_{n \in [N]^d} 2^\alpha \pi^\alpha \|i(n)\|^\alpha = \max_{n \in [N]^d} 2^\alpha \pi^\alpha \left(\sum_{j=0}^{d-1} |i_j(n)|^2 \right)^{\alpha/2} \quad (43)$$

$$\leq 2^\alpha \pi^\alpha \left(\sum_{j=0}^{d-1} |N/2|^2 \right)^{\alpha/2} = \mathcal{O}(d^{\alpha/2} N^\alpha), \quad (44)$$

where the inequality in the last line follows from the definition of $i(n)$ in Equation (18). Therefore, although the discretized Laplacian B has a finite operator norm, it grows as the number of grid points along each spatial dimension increases, and tends to infinity when $N \rightarrow \infty$ (in this case B also converges to the continuous fractional Laplacian $(-\Delta)^{\alpha/2}$).

Fortunately, as shown in Theorem 6, Theorem 7 and Theorem 8, we can choose a specific value of N to control the discretization error by a target ϵ . So, the way in which we resolve the unbounded operator issue is as follows: we first choose a sufficiently large N to ensure that the spatial discretization error is within the error tolerance, and then simply design quantum algorithms for the spatially discretized equation in Equation (38). The output of quantum algorithms would also be a good approximation of the solution of the original equation in Equation (30). Nevertheless, since $\|B\|$ is larger as we require more accuracy, we need to carefully analyze how the complexities of the quantum algorithms might be affected by $\|B\|$. This will be carefully discussed in the following subsections.

4.3 Second-order Trotter formula

We solve Equation (38) by Trotter formula. In particular, we divide $[0, T]$ into r equi-length segments and let $h = T/r$. Consider

$$\mathcal{T}e^{\int_0^T (-B-C(t))dt} \approx \prod_{j=0}^{r-1} S_2((j+1)h, jh) \quad (45)$$

where $S_2(jh, (j+1)h)$ is the second-order time-dependent Trotter method, aiming at approximating $\mathcal{T}e^{\int_{jh}^{(j+1)h} (-B-C(t))dt}$ and defined as

$$S_2((j+1)h, jh) = e^{-Bh/2} e^{-C((j+1/2)h)h} e^{-Bh/2}. \quad (46)$$

In this subsection, we first derive a bound of the Trotter error and show the choice of the number of the segments for bounded errors. Then we discuss how to quantumly implement the numerical scheme and estimate its complexity.

4.3.1 Error bound, commutator scalings, and the number of the time steps

We first bound the distance between the exact evolution operator and the numerical integrators.

Lemma 9. Consider solving Equation (38) using second-order Trotter formula $\prod_{j=0}^{r-1} S_2((j+1)h, jh)$ with time step size $h = T/r$, where the local integrator S_2 is defined in Equation (46). Then

$$\begin{aligned} & \left\| \mathcal{T}e^{\int_0^T (-B-C(t))dt} - \prod_{j=0}^{r-1} S_2((j+1)h, jh) \right\| \\ & \leq Th^2 \left(\frac{1}{24} \max \|C''\| + \frac{1}{4} (\|B\| + \max \|C\|) \max \|C'\| \right. \\ & \quad \left. + \frac{1}{6} \max \|[B, [B, C]]\| + \frac{1}{4} \max \|[B, C]\| \max \|C\| + \frac{1}{3} \max \|C\|^3 \right), \end{aligned} \quad (47)$$

where all the maximums are taken over $t \in [0, T]$.

The proof of [Theorem 9](#) can be found in [Section D](#), which contains two parts. First, we deal with the time-ordering operator by bounding the distance between $\mathcal{T}e^{\int_0^h (-B-C(t))dt}$ and $e^{(-B-C(h/2))h}$ using the variation of parameters formula. This part contributes to the first two terms in the error bound (the second line of [Equation \(47\)](#)) involving time derivatives of the potential matrix $C(t)$. Then we can bound the error between $e^{(-B-C(h/2))h}$ and S_2 by time-independent Trotter error bounds. Notice that the proof for the second part is different from that for Hamiltonian simulation [30], since in our case the parameters β and γ in the exponentials $e^{-\beta B}$ and $e^{-\gamma C}$ are restricted to non-negative to avoid exponential overhead, while for Hamiltonian simulation there is no such restriction. Instead, we mostly follow the procedure in [34] to establish the second part of the error bound. This contributes to the last three terms in the error bound (the third line of [Equation \(47\)](#)) involving the commutators and the spectral norm of C explicitly.

Now we compute the norm of the commutators $[B, C(t)]$ and $[B, [B, C(t)]]$. All the results are for a fixed t , so we will omit the explicit t dependence in our notation for now. Naive bounds are $\|[B, C]\| \leq \mathcal{O}(\|B\|) = \mathcal{O}(d^{\alpha/2}N^\alpha)$ and $\|[B, [B, C]]\| \leq \mathcal{O}(\|B\|^2) = \mathcal{O}(d^\alpha N^{2\alpha})$. However, these naive bounds are improvable, because the order of the commutator of the Laplacian operator can be reduced. Such a phenomenon has been observed in [34, 35]. To see this, let us take $\alpha = 2$ and consider its continuous analog. Then for any smooth function $f(x)$,

$$\begin{aligned} [-\Delta, c]f &= -\Delta(cf) - c(-\Delta f) \\ &= -(\Delta c)f - 2(\nabla c) \cdot (\nabla f) - c\Delta f + c\Delta f \\ &= -(\Delta c)f - 2(\nabla c) \cdot (\nabla f). \end{aligned} \quad (48)$$

So we expect, in the discrete setting, the first commutator is bounded by the discretized divergence operator, whose norm is only $\mathcal{O}(dN)$.

Now we state the improved bound in the discrete setting. The proof can be found in [Section E](#), which is a generalization of [34].

Lemma 10. *Suppose that $c(t, x)$ is a bounded C^{5+d} function in x . Then we have*

$$\|[B, C(t)]\| \leq \mathcal{O}(d^{\alpha/2}N^{\alpha/2}) \quad (49)$$

and

$$\|[B, [B, C(t)]]\| \leq \mathcal{O}(d^\alpha N^\alpha). \quad (50)$$

The choice of the number of time steps is a direct consequence of [Theorem 9](#) and [Theorem 10](#).

Corollary 11. *Consider solving [Equation \(38\)](#) using second-order Trotter formula $\prod_{j=0}^{r-1} S_2((j+1)h, jh)$ with time step size $h = T/r$, where the local integrator S_2 is defined in [Equation \(46\)](#). Suppose that $\|C^{(k)}(t)\|$'s are uniformly bounded in t for $k \leq d+5$. Then, in order to bound the operator splitting error by ϵ , it suffices to choose*

$$r = \mathcal{O}\left(d^{\alpha/2}N^{\alpha/2}\frac{T^{3/2}}{\epsilon^{1/2}}\right). \quad (51)$$

4.3.2 Quantum implementation and complexity estimate

Now we discuss quantum implementation of this method. The main idea is to construct the block-encodings of $e^{-Bh/2}$ and $e^{-C((j+1/2)h)}$ and multiply them together using [Theorem 4](#). The block-encodings can be constructed using controlled rotations since each evolution operator is unitarily equivalent to a diagonal matrix and the corresponding unitary transformation matrix is efficiently implementable. The nuance is that the operator $e^{-C((j+1/2)h)}$ still depends on the specific time, so we use the counter register in the compression gadget as the time clock as well.

Suppose that we are given the oracle O_1 , defined in [Equation \(20\)](#), that encodes the eigenvalues of B , and the oracle O_2 that gives the element of $C(t)$ as

$$O_2 : |t\rangle |n_0\rangle \cdots |n_{d-1}\rangle |0\rangle \rightarrow |t\rangle |n_0\rangle \cdots |n_{d-1}\rangle |c(t, n/N)\rangle. \quad (52)$$

Here, with an abuse of notations, $|t\rangle$ represents some specific way of encoding the information of t , i.e., $(2j+1)$ for the time point $(j+1/2)h$ after time discretization (the reason why it is $(2j+1)$ will be clear soon).

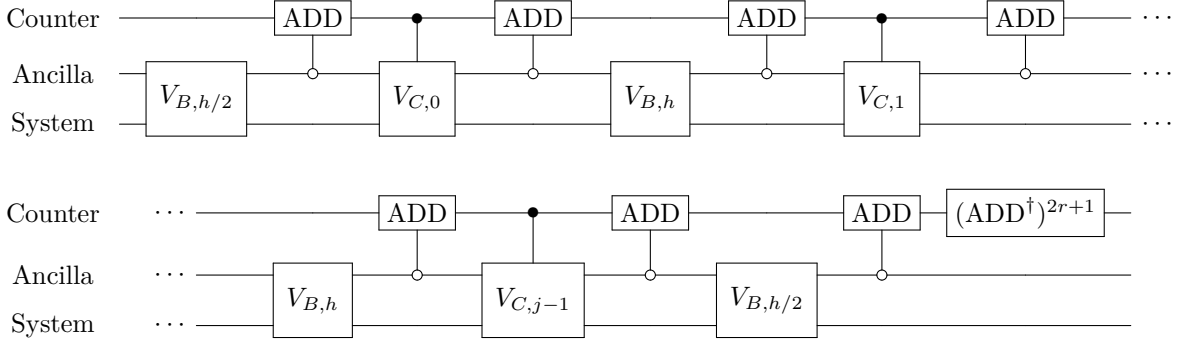


Figure 3: Quantum circuit for implementing second-order Trotter method. Here the Counter register contains $\lceil \log_2(2r+1) \rceil + 1$ qubits. ADD implements addition by 1 modulo $2^{\lceil \log_2(2r+1) \rceil + 1}$.

As discussed before, we may use the circuit in Figure 2 with a replacement of $O_{\text{exp},1}$ by $O_{\text{exp},2} : |x\rangle|0\rangle \rightarrow |x\rangle|e^{-xh/2}\rangle$ to construct a $(1, *, 0)$ -block-encoding of $e^{-Bh/2}$. Here the number of the ancilla qubits depends on those for encoding the eigenvalues of B and their exponentials, on which we do not keep track for technical simplicity. We denote this block-encoding by $V_{B,h/2}$. Similarly, a $(1, *, 0)$ -block-encoding of e^{-Bh} can be constructed with $O_{\text{exp},3} : |x\rangle|0\rangle \rightarrow |x\rangle|e^{-xh}\rangle$. We denote it by $V_{B,h}$. Furthermore, in Figure 2, if we discard the QFT steps, and replace $O_{\text{exp},1}$ by $O_{\text{exp},3}$ and the oracle O_1 by O_2 controlled by an extra counter register, we can construct a (controlled version of) $(1, *, 0)$ -block-encoding of $e^{-C((j+1/2)h)h}$, denoted by $V_{C,j}$. Notice that the constructions of block-encodings $V_{B,h/2}$, $V_{B,h}$ and $V_{C,j}$ only require $\mathcal{O}(1)$ queries to the aforementioned oracles and QFT.

Now we construct the numerical integrator. Mathematically we need to block encode the operator

$$\prod_{j=0}^{r-1} S_2((j+1)h, jh) = e^{-Bh/2} \left(\prod_{j=1}^{r-1} e^{-C((j+1/2)h)h} e^{-Bh} \right) e^{-C(h/2)h} e^{-Bh/2}. \quad (53)$$

This can be done by the circuit in Figure 3, which can be viewed as a variant of the compression gadget in Theorem 4. The idea is to use an extra counter register with $\lceil \log_2(2r+1) \rceil + 1$ qubits for both keeping track of the success/failure of the multiplication and indicating the index of the time step. If all the applications of the block-encodings of the local exponentials are successful, then the value of the counter register at each step suggests the correct time step index, and the final value is reset to be 0. Notice that the block-encoding factor of $V_{B,h}$ and $V_{C,j}$ are 1. Then the circuit in Figure 3 gives a $(1, *, 0)$ -block-encoding of $\prod_{j=0}^{r-1} S_2((j+1)h, jh)$, as desired.

We apply this block-encoding to the input state $|u_0\rangle$, and the final state is

$$\frac{1}{\|\vec{u}_0\|} \sum_{n \in [N]^d} (\vec{u}_r)_n |n_0\rangle \cdots |n_{d-1}\rangle |0\rangle + |\perp\rangle, \quad (54)$$

where

$$\vec{u}_r = \prod_{j=0}^{r-1} S_2((j+1)h, jh) \vec{u}_0 \approx \vec{u}(T), \quad (55)$$

and $|\perp\rangle$ represents the junk state with ancilla register not equal to 0. The final step is to measure the ancilla registers of Equation (54). If all the ancilla registers are 0, then we get a good approximation of $|u(T)\rangle$. The averaged number of repeats after amplitude amplification is $\mathcal{O}(\|\vec{u}_0\|/\|\vec{u}(T)\|)$.

We summarize the overall complexity as follows.

Theorem 12. Consider solving Equation (29) on discrete grid points $(T, n/N)$ where $n \in [N]^d$. Let $u(t, x)$ denote the solution of the equation Equation (29), and $\vec{u}(t)$ denote the solution of the spatially discretized Equation (38). Suppose that

1. we are given oracles O_1 and O_2 defined in [Equation \(20\)](#) and [Equation \(52\)](#), and the state preparation oracle $O_u : |0\rangle \rightarrow |\vec{u}_0\rangle$,
2. $u(t, x)$ is in the Gevrey class G^σ in the sense that $u(t, x)$ is smooth and there exist constants $\Lambda > 0$ and $\sigma \geq 0$ such that

$$\sup_{j \in [d], t \in [0, T], x \in [0, 1]^d} |\partial_{x_j}^p u(t, x)| \leq \Lambda^{p+1} (p!)^\sigma. \quad (56)$$

3. $\|(u(T, n/N))_{n \in [N]^d}\| \geq \tilde{g}(T)$ for a function $\tilde{g}$.
4. $\|\vec{u}(0)\|/\|\vec{u}(T)\| \leq g(T)$ for a function g .

Then, with second-order operator splitting method for time propagation, an ϵ -approximation of $|(u(T, n/N))_{n \in [N]^d}\rangle$ can be obtained by choosing

$$N = \mathcal{O} \left(\left(d \log d + \log \left(\frac{T}{\tilde{g}(T)} \right) + \log \left(\frac{1}{\epsilon} \right) \right)^\sigma \right). \quad (57)$$

and using

1.
$$\mathcal{O} \left((g(T))^{3/2} \frac{T^{3/2}}{\epsilon^{1/2}} d^{\alpha/2} \left(d \log d + \log \left(\frac{T}{\tilde{g}(T)} \right) + \log \left(\frac{1}{\epsilon} \right) \right)^{\alpha\sigma/2} \right) \quad (58)$$

queries to O_1, O_2 , their inverses and controlled versions,

2.
$$\mathcal{O}(g(T)) \quad (59)$$

queries to the state preparation oracle O_u and its inverse.

3.
$$\begin{aligned} & \mathcal{O} \left((g(T))^{3/2} \frac{T^{3/2}}{\epsilon^{1/2}} d^{\alpha/2+1} \left(d \log d + \log \left(\frac{T}{\tilde{g}(T)} \right) + \log \left(\frac{1}{\epsilon} \right) \right)^{\alpha\sigma/2} \right. \\ & \quad \left. \times \log^2 \left(d \log d + \log \left(\frac{T}{\tilde{g}(T)} \right) + \log \left(\frac{1}{\epsilon} \right) \right) \right) \end{aligned} \quad (60)$$

additional elementary gates.

Proof. It suffices to bound both spatial and time discretization errors by ϵ . The choice of N directly follows [Theorem 8](#). We now count the overall complexity. Each run of the algorithm requires an application of [Figure 2](#) which implements $e^{-Bh/2}$ and $e^{-C(t)h}$ for $\mathcal{O}(r)$ times. In each block-encoding of $e^{-Bh/2}$ and $e^{-C(t)h}$, we need to use $\mathcal{O}(1)$ queries to O_1 and O_2 , and $\mathcal{O}(d \log^2(N))$ additional gates mainly due to the QFT. According to [Theorem 11](#), it suffices to choose $r = \mathcal{O} \left(g(T)^{1/2} d^{\alpha/2} N^{\alpha/2} \frac{T^{3/2}}{\epsilon^{1/2}} \right)$. Here the extra factor $g(T)^{1/2}$ is because [Theorem 11](#) only bounds the operator norm, and in order to bound the error in the quantum state by ϵ , the operator norm error bound needs to be bounded by $\mathcal{O}(\epsilon \|\vec{u}(T)\|/\|\vec{u}(0)\|)$ according to [Theorem 19](#). The averaged number of repeats to succeed after amplitude amplification is $\mathcal{O}(\|\vec{u}(0)\|/\|\vec{u}(T)\|) = \mathcal{O}(g(T))$. Multiplying these together gives the overall query complexity and additional gates required. Notice that in each run we only need one query to the state preparation oracle, so the overall number of state preparation is only $\sim g(T)$. $\square$

4.4 Time-marching method

Now we consider an alternative algorithm proposed in [22] called time-marching method. The method is designed for general ODE

$$\frac{d}{dt} \phi(t) = A(t) \phi(t), \quad \phi(0) = \phi_0 \quad (61)$$

with a time-dependent matrix-valued function $A(t)$, and is a quantum analog of the classical exponential propagation methods. It first divides the time interval $[0, T]$ into small segments with mesh $0 = t_0 < t_1 < \dots < t_L = T$ and applies the short-time evolution operator sequentially. While naive applications of a sequence of non-unitary operators may incur an exponential overhead in the number of the operators, the time-marching method avoids such overhead by a technique that combines the uniform singular value transformation and the amplitude amplification. We refer to [22] for more details and only roughly summarize the main result here.

Lemma 13 (Theorem 8 of [22]). *Consider the ODE Equation (61). Suppose that we are given the prepare oracle O_ϕ of $|\phi_0\rangle$ such that $O_\phi|0\rangle = |\phi_0\rangle$ and an input model of $A(t)$, denoted by MAT_A , that simultaneously block encodes $A(t'_k)$ at some refined mesh points t'_k . Then, with the time-marching method, an ϵ -approximation of $|\phi(T)\rangle$ can be prepared using*

$$\mathcal{O}\left(\eta^2 T^2 Q \log(\eta T Q) \frac{\log(\eta T Q / \epsilon)}{\log \log(\eta T Q / \epsilon)}\right) \quad (62)$$

queries to MAT_A and $\mathcal{O}(Q)$ queries to O_ϕ , its inverse and controlled version. Here η is the block-encoding factor of MAT_A such that $\eta \geq \|A(t)\|$ for all $t \in [0, T]$, and

$$Q = \frac{\|\phi(0)\| \prod_{l=1}^L \left\| \mathcal{T} e^{\int_{t_{l-1}}^{t_l} A(t) dt} \right\|}{\|\phi(T)\|}. \quad (63)$$

Now we discuss the complexity of applying the time-marching method to spatially discretized equation Equation (38). For technical simplicity, we only estimate the query complexities.

Theorem 14. *Consider solving Equation (29) on discrete grid points $(T, n/N)$ where $n \in [N]^d$. Let $u(t, x)$ denote the solution of the equation Equation (29), and $\bar{u}(t)$ denote the solution of the spatially discretized Equation (38). Suppose that*

1. *we are given oracles encoding the eigenvalues of B and the diagonal entries of $C(t)$, i.e., O_1 and O_2 defined in Equation (20) and Equation (52), and the state preparation oracle $O_u : |0\rangle \rightarrow |\bar{u}_0\rangle$,*
2. *$v(t, x)$ is in the Gevrey class G^σ in the sense that $v(t, x)$ is smooth and there exist constants $\Lambda > 0$ and $\sigma \geq 0$ such that*

$$\sup_{j \in [d], t \in [0, T], x \in [0, 1]^d} |\partial_{x_j}^p v(t, x)| \leq \Lambda^{p+1} (p!)^\sigma. \quad (64)$$

3. *$\|(v(T, n/N))_{n \in [N]^d}\| \geq \tilde{g}(T)$ for a function $\tilde{g}$.*

Then, with the time-marching method for time propagation, an ϵ -approximation of $|(u(T, n/N))_{n \in [N]^d}\rangle$ can be obtained by choosing

$$N = \mathcal{O}\left(\left(d \log d + \log\left(\frac{T}{\tilde{g}(T)}\right) + \log\left(\frac{1}{\epsilon}\right)\right)^\sigma\right). \quad (65)$$

and using

$$1. \quad \tilde{\mathcal{O}}\left(Q T^2 d^\alpha \left(d \log d + \log\left(\frac{T}{\tilde{g}(T)}\right) + \log\left(\frac{1}{\epsilon}\right)\right)^{2\alpha\sigma} \log\left(\frac{1}{\epsilon}\right)\right) \quad (66)$$

queries to O_1 , O_2 , the d -dimensional QFT circuit, their inverses and controlled versions, where

$$Q = \frac{\|\bar{u}(0)\| \prod_{l=1}^L \left\| \mathcal{T} e^{\int_{t_{l-1}}^{t_l} (-B - C(t)) dt} \right\|}{\|\bar{u}(T)\|}, \quad (67)$$

2.

$$\mathcal{O}(Q) \tag{68}$$

queries to the state preparation oracle O_u or its inverse.

Proof. First, the oracle MAT_A mentioned in [Theorem 13](#) can be constructed through the given O_1 and O_2 and additional $\mathcal{O}(d \log^2(N))$ gates. The idea is as follows. We can first construct the block-encoding of the diagonal matrix D with O_1 , then construct the block-encoding of $B = (\mathcal{F})^{\otimes d} D (\mathcal{F}^{-1})^{\otimes d}$ using QFT. Meanwhile we may construct the simultaneous block-encoding of $C(t)$ with O_2 . Finally, we can linearly combine these two block-encodings to obtain the desired MAT_A that block encodes $A(t) = -B - C(t)$. According to [\[32, Lemma 48 & Lemma 52 & Lemma 53\]](#), such approach requires $\mathcal{O}(1)$ queries to O_1 and O_2 , so we may directly estimate the number of queries to MAT_A as that of queries to O_1 and O_2 . Note that there are additional $\mathcal{O}(d \log^2(N))$ gates required to construct MAT_A from O_1 and O_2 , but we only focus on the query complexities here for technical simplicity.

Now we may directly use [Theorem 13](#) to estimate the complexity, and it suffices to write down explicit scalings of the block-encoding factor η in the example of [Equation \(38\)](#). Under the assumption that $C(t)$ is uniformly bounded, the spectral norm $\| -B - C(t) \| = \mathcal{O}(d^{\alpha/2} N^\alpha)$, so the parameter $\eta = \mathcal{O}(d^{\alpha/2} N^\alpha)$. Using the choice of N estimated in [Theorem 6](#), we have

$$\eta = \mathcal{O} \left(d^{\alpha/2} \left(d \log d + \log \left(\frac{T}{\tilde{g}(T)} \right) + \log \left(\frac{1}{\epsilon} \right) \right)^{\alpha\sigma} \right). \tag{69}$$

Plugging this parameter back to the scalings in [Theorem 13](#) yields the desired estimates. $\square$

4.5 Truncated Dyson series method

Now we discuss the query complexity of applying the state-of-the-art generic quantum ODE solvers to our linear fractional reaction-diffusion equation. We consider the method proposed in [\[21\]](#), which is based on the truncated Dyson series. It first expands the solution via truncated Dyson series and encode it into a linear system of equations, then solve it using the optimal quantum linear system algorithm. It works for the most general linear ODE with time dependent coefficient matrix and possible inhomogeneous term. In our case, we are only interested in the homogeneous equation, so here we only summarize their main result for the homogeneous equation [Equation \(61\)](#).

Lemma 15 (Theorem 1 of [\[21\]](#)). *Consider the ODE [Equation \(61\)](#) where the coefficient matrix $A(t)$ has non-positive logarithmic norm for all t . Suppose that we are given the prepare oracle O_ϕ of $|\phi_0\rangle$ such that $O_\phi |0\rangle = |\phi_0\rangle$ and an input model of $A(t)$, denoted by MAT_A , that simultaneously block-encoding $A(t'_k)$ at some refined mesh points t'_k . Then, with the truncated Dyson series method, an ϵ -approximation of $|\phi(T)\rangle$ can be prepared using*

$$\mathcal{O} \left(\frac{\max_{t \in [0, T]} \|\phi(t)\|}{\|\phi(T)\|} \eta T \log(1/\epsilon) \log(\eta T/\epsilon) \right) \tag{70}$$

queries to MAT_A and

$$\mathcal{O} \left(\frac{\max_{t \in [0, T]} \|\phi(t)\|}{\|\phi(T)\|} \eta T \log(1/\epsilon) \right) \tag{71}$$

queries to O_ϕ . Here η is the block-encoding factor of MAT_A such that $\eta \geq \|A(t)\|$ for all $t \in [0, T]$.

Theorem 16. *Consider solving [Equation \(29\)](#) on discrete grid points $(T, n/N)$ where $n \in [N]^d$. Let $u(t, x)$ denote the solution of the equation [Equation \(29\)](#), and $\vec{u}(t)$ denote the solution of the spatially discretized [Equation \(38\)](#). Suppose that*

1. *we are given oracles encoding the eigenvalues of B and the diagonal entries of $C(t)$, i.e., O_1 and O_2 defined in [Equation \(20\)](#) and [Equation \(52\)](#), and the state preparation oracle $O_u : |0\rangle \rightarrow |\vec{u}_0\rangle$,*
2. *$u(t, x)$ is in the Gevrey class G^σ in the sense that $u(t, x)$ is smooth and there exist constants $\Lambda > 0$ and $\sigma \geq 0$ such that*

$$\sup_{j \in [d], t \in [0, T], x \in [0, 1]^d} |\partial_{x_j}^p u(t, x)| \leq \Lambda^{p+1} (p!)^\sigma. \tag{72}$$

3. $\|(u(T, n/N))_{n \in [N]^d}\| \geq \tilde{g}(T)$ for a function $\tilde{g}$,
4. $\|\vec{u}(0)\|/\|\vec{u}(T)\| \leq g(T)$ for a function g .

Then, with the truncated Dyson series method for time propagation, an ϵ -approximation of $|(u(T, n/N))_{n \in [N]^d}\rangle$ can be obtained by choosing

$$N = \mathcal{O} \left(\left(d \log d + \log \left(\frac{T}{\tilde{g}(T)} \right) + \log \left(\frac{1}{\epsilon} \right) \right)^\sigma \right). \quad (73)$$

and using

1.
$$\tilde{\mathcal{O}} \left(g(T) T d^{\alpha/2} \left(d \log d + \log \left(\frac{T}{\tilde{g}(T)} \right) + \log \left(\frac{1}{\epsilon} \right) \right)^{\alpha\sigma} \left(\log \left(\frac{1}{\epsilon} \right) \right)^2 \right) \quad (74)$$

queries to O_1 and O_2 , d -dimensional QFT, their inverses and controlled versions,

2.
$$\mathcal{O} \left(g(T) T d^{\alpha/2} \left(d \log d + \log \left(\frac{T}{\tilde{g}(T)} \right) + \log \left(\frac{1}{\epsilon} \right) \right)^{\alpha\sigma} \log \left(\frac{1}{\epsilon} \right) \right) \quad (75)$$

queries to the state preparation oracle O_u or its inverse.

Proof. As shown in the proof of [Theorem 14](#), we may directly estimate the number of queries to O_j as that of queries to MAT_A , and the block-encoding factor η can be bounded as

$$\eta = \mathcal{O} \left(d^{\alpha/2} \left(d \log d + \log \left(\frac{T}{\tilde{g}(T)} \right) + \log \left(\frac{1}{\epsilon} \right) \right)^{\alpha\sigma} \right). \quad (76)$$

Since the coefficient matrix $(-B - C(t))$ is always negative semi-definite, the norm of the solution $\|\phi(t)\|$ is non-increasing over t , so

$$\frac{\max_{t \in [0, T]} \|\vec{u}(t)\|}{\|\vec{u}(T)\|} = \frac{\|\vec{u}(0)\|}{\|\vec{u}(T)\|} \leq g(T). \quad (77)$$

Plugging these parameters back to [Theorem 15](#) completes the proof. $\square$

4.6 Linear combination of Hamiltonian simulation in the interaction picture

We have discussed and analyzed the second-order operator splitting method, the time-marching method, and the truncated Dyson series method. All the methods have extra polynomial dependence on the dimension d , which comes from the dependence on the spectral norm of the discrete fractional Laplacian operator B , although the operator splitting method may partially benefit from its commutator scalings.

In quantum dynamics, a common technique to avoid the explicit dependence on $\|B\|$ is to simulate the dynamics in the interaction picture. Specifically, if we consider the fractional Schrödinger equation

$$i \frac{d}{dt} |\psi\rangle = (-B - C(t)) |\psi\rangle, \quad (78)$$

then, by defining the interaction picture Hamiltonian $H_I(t) = -e^{-iBt}C(t)e^{iBt}$ and $|\psi_I\rangle = e^{-iBt}|\psi\rangle$, we may obtain the transformed solution $|\psi_I\rangle$ by simulating $i \frac{d}{dt} |\psi_I\rangle = H_I(t) |\psi_I\rangle$. Here $H_I(t)$ is bounded independently of $\|B\|$, and its oscillations depend on $\|B\|$. Therefore we may efficiently simulate the interaction picture Hamiltonian using truncated Dyson series method which has linearly dependence on $\|H_I\|$ but only poly-logarithmically depends on its derivatives. We refer to [27] for more details.

The success of the interaction picture Hamiltonian simulation relies on the fast-forwarded implementation of e^{iBs} , that is, e^{iBs} can be implemented for any real number s with cost independent of $\|B\|$ and $|s|$. However, in our reaction-diffusion equation, which can be viewed as the imaginary time evolution of the Schrödinger equation, we cannot directly apply similar technique. This is because e^{-Bs} is only fast-forwardable when $s \geq 0$, while the transformation into the analog of interaction picture requires both forward and backward time evolution.

4.6.1 Representation

To take advantage of the interaction picture technique, we can relate the reaction-diffusion equation with the Hamiltonian simulation problem. A recent work [24] shows that any linear ODE can be represented as a linear combination of Hamiltonian simulation. In particular, we may write the evolution operator of our fractional reaction-diffusion equation as

$$\mathcal{T}e^{-\int_0^T (B+C(s))ds} = \int_{\mathbb{R}} \frac{1}{\pi(1+\xi^2)} \mathcal{T}e^{-i\int_0^T \xi(B+C(s))ds} d\xi. \quad (79)$$

Here $\mathcal{T}$ is the time-ordering operator. The proof of Equation (79) can be found in [24]. Now we use the interaction picture simulation. Let

$$U(t) = \mathcal{T}e^{-i\int_0^t \xi(B+C(s))ds}, \quad (80)$$

then $U(t)$ satisfies the time-dependent Hamiltonian simulation problem as

$$\frac{dU}{dt} = -i(\xi B + \xi C(t))U(t), \quad U(0) = I. \quad (81)$$

Let

$$U_I(t) = e^{i\xi Bt}U(t). \quad (82)$$

We may compute that

$$\begin{aligned} \frac{dU_I}{dt} &= i\xi B e^{i\xi Bt}U(t) - i e^{i\xi Bt}(\xi B + \xi C(t))U(t) \\ &= -i e^{i\xi Bt} \xi C(t) e^{-i\xi Bt} U_I(t). \end{aligned} \quad (83)$$

Define

$$H_I(t; \xi) = e^{i\xi Bt} \xi C(t) e^{-i\xi Bt}. \quad (84)$$

Then

$$U_I(t) = \mathcal{T}e^{-i\int_0^t H_I(s; \xi)ds}, \quad (85)$$

and we may write Equation (79) as

$$\mathcal{T}e^{-\int_0^T (B+C(s))ds} = \int_{\mathbb{R}} \frac{1}{\pi(1+\xi^2)} e^{-i\xi BT} \mathcal{T}e^{-i\int_0^T H_I(s; \xi)ds} d\xi. \quad (86)$$

4.6.2 Numerical quadrature

We can truncate Equation (86) over a finite interval $[-\Xi, \Xi]$ and write it as

$$\mathcal{T}e^{-\int_0^T (B+C(s))ds} \approx \int_{-\Xi}^{\Xi} \frac{1}{\pi(1+\xi^2)} e^{-i\xi BT} \mathcal{T}e^{-i\int_0^T H_I(s; \xi)ds} d\xi. \quad (87)$$

The resulting integral can be discretized using standard numerical quadrature. Here we use the simplest Riemann sum formula with M grid points. For $0 \leq j \leq M-1$, let $\xi_j = -\Xi + 2j\Xi/M$ and $w_j = \frac{1}{\pi(1+\xi_j^2)} \frac{2\Xi}{M}$. Then

$$\mathcal{T}e^{-\int_0^T (B+C(s))ds} \approx \sum_{j=0}^{M-1} w_j e^{-i\xi_j BT} \mathcal{T}e^{-i\int_0^T H_I(s; \xi_j)ds}. \quad (88)$$

The discretization error can be bounded as follows, and its proof is given in Section F.

Lemma 17. *We have*

$$\left\| \mathcal{T}e^{-\int_0^T (B+C(s))ds} - \sum_{j=0}^{M-1} w_j e^{-i\xi_j BT} \mathcal{T}e^{-i\int_0^T H_I(s; \xi_j)ds} \right\| \leq \frac{2}{\pi\Xi} + \frac{2\Xi^2}{\pi M} (1 + T(\|B\| + \max \|C\|)). \quad (89)$$

In order to bound the discretization error by $\mathcal{O}(\epsilon)$, it suffices to choose

$$\Xi = \mathcal{O}\left(\frac{1}{\epsilon}\right), \quad M = \mathcal{O}\left(\frac{T\|B\|}{\epsilon^3}\right). \quad (90)$$

4.6.3 Implementation and complexity

Suppose that we are given the same input oracles as in previous algorithms, encoding the eigenvalues of B and the diagonal entries of $C(t)$, *i.e.*, O_1 and O_2 defined in Equation (20) and Equation (52). The sketch of the algorithm is to simulate the interaction picture Hamiltonian using truncated Dyson series method [27] and then apply the LCU technique to compute Equation (88).

We start with the HAM-T encoding of the matrix H_I . For a fixed time step size h and an integer such that $[mh, (m+1)h] \subset [0, T]$, we may first construct the HAM-T encoding of C from its sparse input oracle following the approach in [32]. The resulting HAM-T encoding satisfies

$$\langle 0|_a \text{HAM-T}_{C,m} |0\rangle_a = \sum_{l=0}^{M_H-1} |l\rangle \langle l| \otimes \frac{C(mh + lh/M_H)}{\alpha_C}. \quad (91)$$

Here α_C is the upper bound of $\|C\|$, and M_H is the number of grid points used in the truncated Dyson series method. By appending an ancilla register $|j\rangle$ encoding the index for ξ and applying the controlled rotation $|j\rangle |0\rangle \rightarrow |j\rangle \left(\frac{\xi_j}{\Xi} |0\rangle + \sqrt{1 - \frac{\xi_j^2}{\Xi^2}} |1\rangle \right)$ with the help of an additional ancilla qubit, we obtain the HAM-T encoding of ξC such that

$$\langle 0|_a \text{HAM-T}_{\xi C,m} |0\rangle_a = \sum_{j=0}^{M-1} \sum_{l=0}^{M_H-1} |j\rangle \langle j| \otimes |l\rangle \langle l| \otimes \frac{\xi_j C(mh + lh/M_H)}{\Xi \alpha_C}. \quad (92)$$

To construct the block-encoding of $e^{-i\xi B t}$, we write

$$\begin{aligned} e^{-i\xi_j B(mh + lh/M_H)} &= e^{-i(-\Xi + 2j\Xi/M)B(mh + lh/M_H)} \\ &= e^{i\Xi Bmh} \left(e^{-i(2\Xi/M)Bmh} \right)^j \left(e^{i\Xi B h/M_H} \right)^l \left(e^{-i(2\Xi/M)B h/M_H} \right)^l. \end{aligned} \quad (93)$$

Noticing that the matrix B can be diagonalized with QFT and the diagonal components are given through the oracle O_1 , we can implement e^{-iBs} for any real number s fast-forwardly with $\mathcal{O}(1)$ uses of O_1 , controlled phase gate and QFT. Then, according to the binary encoding of $0 \leq j \leq M-1$, we can use the controlled version of a total of $\log_2(M)$ operators $e^{-i(2\Xi/M)Bmh}$, $e^{-i2(2\Xi/M)Bmh}$, $e^{-i2^2(2\Xi/M)Bmh}$, $\dots$, $e^{-i2^{\log_2(M)}(2\Xi/M)Bmh}$ to implement the controlled evolution $\sum_{j=0}^{M-1} |j\rangle \langle j| \otimes (e^{-i(2\Xi/M)Bmh})^j$. Similarly, we can construct the evolution $\sum_{l=0}^{M_H-1} |l\rangle \langle l| \otimes (e^{i\Xi B h/M_H})^l$ and $\sum_{j=0}^{M-1} \sum_{l=0}^{M_H-1} |j\rangle \langle j| \otimes |l\rangle \langle l| \otimes \left((e^{-i(2\Xi/M)B h/M_H})^j \right)^l$ with logarithmic cost as well. Multiplying them together gives the select oracle

$$\text{SEL}_{B,m} = \sum_{j=0}^{M-1} \sum_{l=0}^{M_H-1} |j\rangle \langle j| \otimes |l\rangle \langle l| \otimes e^{-i\xi_j B(mh + lh/M_H)}. \quad (94)$$

Then

$$\text{HAM-T}_{H_I,m} := (I_{n_a} \otimes \text{SEL}_{B,m}^\dagger) \text{HAM-T}_{\xi C,m} (I_{n_a} \otimes \text{SEL}_{B,m}) \quad (95)$$

gives the HAM-T encoding of H_I that

$$\langle 0|_a \text{HAM-T}_{H_I,m} |0\rangle_a = \sum_{j=0}^{M-1} \sum_{l=0}^{M_H-1} |j\rangle \langle j| \otimes |l\rangle \langle l| \otimes \frac{H_I(mh + lh/M_H; \xi_j)}{\Xi \alpha_C}. \quad (96)$$

The $\text{HAM-T}_{H_I,m}$ serves as the Hamiltonian input oracle in the truncated Dyson series method. Therefore, the method in [27] gives the select oracle

$$\text{SEL}_W = \sum_{j=0}^{M-1} |j\rangle \langle j| \otimes W_j, \quad (97)$$

where W_j is an approximation of $U_I(T; \xi_j)$. We then multiply it on the left by $\sum_{j=0}^{M-1} |j\rangle \langle j| \otimes e^{-i\xi_j BT}$ (which again can be efficiently constructed according to the binary representation of j) and obtain

$$\text{SEL}_U = \sum_{j=0}^{M-1} |j\rangle \langle j| \otimes U_j. \quad (98)$$

Here U_j is an approximation of $e^{-i\xi_j BT} \mathcal{T} e^{-i \int_0^T H_I(s; \xi_j) ds}$. The SEL_U operator serves as the select oracle in the LCU subroutine. Hence the formula Equation (88) can be directly implemented by the LCU technique (Theorem 2).

The overall complexity of the algorithm is given as follows.

Theorem 18. *Consider the spatially discretized equation Equation (38). Suppose that*

1. *we are given oracles encoding the eigenvalues of B and the diagonal entries of $C(t)$, i.e., O_1 and O_2 defined in Equation (20) and Equation (52), and the state preparation oracle $O_u : |0\rangle \rightarrow |\vec{u}_0\rangle$,*
2. *$u(t, x)$ is in the Gevrey class G^σ in the sense that $u(t, x)$ is smooth and there exist constants $\Lambda > 0$ and $\sigma \geq 0$ such that*

$$\sup_{j \in [d], t \in [0, T], x \in [0, 1]^d} |\partial_{x_j}^p u(t, x)| \leq \Lambda^{p+1} (p!)^\sigma, \quad (99)$$

3. *$\|(u(T, n/N))_{n \in [N]^d}\| \geq \tilde{g}(T)$ for a function $\tilde{g}$,*
4. *$\|\vec{u}(0)\| / \|\vec{u}(T)\| \leq g(T)$ for a function g .*

Then, the linear combination of Hamiltonian simulation in the interaction picture can prepare an ϵ -approximation of $|(u(T, n/N))_{n \in [N]^d}\rangle$ using

1. *queries to O_1 and O_2 a total number of times*

$$\mathcal{O} \left(g(T)^2 \frac{T}{\epsilon} \log^3 \left(\frac{g(T) T d \log d}{\epsilon} \log \left(\frac{1}{\tilde{g}(T)} \right) \right) \right), \quad (100)$$

2. *queries to O_u for $\mathcal{O}(g(T))$ times,*
3. *additional elementary gates for*

$$\mathcal{O} \left(g(T)^2 \frac{T}{\epsilon} \log \left(\frac{g(T) T}{\epsilon} \right) \left(d \log^2 \left(d + \log \left(\frac{T}{\tilde{g}(T)} \right) + \log \left(\frac{1}{\epsilon} \right) \right) + \log \left(\frac{g(T) T}{\epsilon} \right) \right) \right). \quad (101)$$

Proof. As discussed before, the oracle $\text{HAM-T}_{\xi C, m}$ can be implemented with $\mathcal{O}(1)$ queries to O_2 and $\mathcal{O}(d \log(N))$ additional gates [32, Lemma 48], and the construction of $\text{SEL}_{B, m}$ requires $\mathcal{O}(\log(M) \log(M_H))$ O_1 and QFT (which requires $\mathcal{O}(d \log^2(N))$ gates). So each $\text{HAM-T}_{H_I, m}$ can be constructed with $\mathcal{O}(\log(M) \log(M_H))$ queries to O_1 , O_2 and $\mathcal{O}(d \log^2(N))$ additional gates. According to [27, Corollary 4], by choosing

$$\begin{aligned} M_H &= \mathcal{O} \left(\frac{T}{\Xi \alpha_C \epsilon'} (\|B\| \Xi^2 \alpha_C + \Xi \max \|C'\| + \Xi^2 \alpha_C^2) \right) \\ &= \mathcal{O} \left(\frac{T}{\epsilon'} (\|B\| \Xi + \max \|C'\|) \right), \end{aligned} \quad (102)$$

we may implement the select oracle $\text{SEL}_W = \sum_{j=0}^{M-1} |j\rangle \langle j| \otimes W_j$ such that

$$\left\| W_j - \mathcal{T} e^{-i \int_0^T H_I(s; \xi_j) ds} \right\| \leq \epsilon' \quad (103)$$

with

$$\mathcal{O} \left(\Xi \alpha_C T \frac{\log(\Xi \alpha_C T / \epsilon')}{\log \log(\Xi \alpha_C T / \epsilon')} \right) = \mathcal{O} \left(\Xi T \frac{\log(\Xi T / \epsilon')}{\log \log(\Xi T / \epsilon')} \right) \quad (104)$$

queries to $\text{HAM-T}_{H_I, m}$ and

$$\begin{aligned} & \mathcal{O} \left(\Xi \alpha_C T \frac{\log(\Xi \alpha_C T / \epsilon')}{\log \log(\Xi \alpha_C T / \epsilon')} (d \log(N) + \log(M_H)) \right) \\ &= \mathcal{O} \left(\Xi T \frac{\log(\Xi T / \epsilon')}{\log \log(\Xi T / \epsilon')} (d \log(N) + \log(M_H)) \right) \end{aligned} \quad (105)$$

additional gates. Hence the SEL_U can be implemented with asymptotically the same cost, and, taking into account the cost of constructing $\text{HAM-T}_{H_I, m}$, this step needs

$$\mathcal{O} \left(\Xi T \frac{\log(\Xi T / \epsilon')}{\log \log(\Xi T / \epsilon')} \log(M) \log(M_H) \right) \quad (106)$$

queries to O_1 and O_2 , and

$$\mathcal{O} \left(\Xi T \frac{\log(\Xi T / \epsilon')}{\log \log(\Xi T / \epsilon')} (d \log^2(N) + \log(M_H)) \right) \quad (107)$$

additional gates.

The LCU algorithm requires a single application of the select oracle and two applications of the prepare oracle for $\frac{1}{\sqrt{\|w\|_1}} \sum_{j=0}^{M-1} \sqrt{w_j} |j\rangle$. Noticing that w represents the discretized Cauchy distribution, we can implement this prepare oracle with $\mathcal{O}(\log(M))$ gates [36] and $\|w\|_1 = \mathcal{O}(1)$. The output of the LCU step can be written as $\frac{1}{\|w\| \|\vec{u}_0\|} |0\rangle \tilde{v} + |\perp\rangle$, where

$$\tilde{v} = \sum_{j=0}^{M-1} w_j U_j \vec{u}_0. \quad (108)$$

Using the inequality $\|\vec{a}/\|a\| - \vec{b}/\|b\|\| \leq 2\|\vec{a} - \vec{b}\|/\|\vec{a}\|$, we can bound the error in the quantum state as

$$\begin{aligned} \|\vec{u}(T)\rangle - |\tilde{v}\rangle\| &\leq \frac{2}{\|\vec{u}(T)\|} \|\vec{u}(T) - \tilde{v}\| \\ &\leq \frac{2\|\vec{u}_0\|}{\|\vec{u}(T)\|} \left\| \mathcal{T} e^{-\int_0^T (B+C(s))ds} - \sum_{j=0}^{M-1} w_j e^{-i\xi_j B T} \mathcal{T} e^{-i\int_0^T H_I(s; \xi_j)ds} \right\| \\ &\quad + \frac{2\|\vec{u}_0\|}{\|\vec{u}(T)\|} \left\| \sum_{j=0}^{M-1} w_j e^{-i\xi_j B T} \mathcal{T} e^{-i\int_0^T H_I(s; \xi_j)ds} - \sum_{j=0}^{M-1} w_j U_j \right\| \\ &\leq \frac{2\|\vec{u}_0\|}{\|\vec{u}(T)\|} \left\| \mathcal{T} e^{-\int_0^T (B+C(s))ds} - \sum_{j=0}^{M-1} w_j e^{-i\xi_j B T} \mathcal{T} e^{-i\int_0^T H_I(s; \xi_j)ds} \right\| + \frac{2\|\vec{u}_0\|}{\|\vec{u}(T)\|} \|w\|_1 \epsilon'. \end{aligned} \quad (109)$$

To bound the error by ϵ , according to [Theorem 17](#), it suffices to choose

$$\Xi = \mathcal{O} \left(\frac{\|\vec{u}_0\|}{\|\vec{u}(T)\|} \frac{1}{\epsilon} \right), \quad M = \mathcal{O} \left(\left(\frac{\|\vec{u}_0\|}{\|\vec{u}(T)\|} \right)^3 \frac{T\|B\|}{\epsilon^3} \right), \quad \epsilon' = \mathcal{O} \left(\frac{\|\vec{u}(T)\|}{\|\vec{u}_0\|} \epsilon \right). \quad (110)$$

With these and by $\|B\| = \mathcal{O}(d^{\alpha/2} N^\alpha)$ and N given in [Theorem 8](#), in each run of the algorithm we need queries to O_u for $\mathcal{O}(1)$ times, queries to O_1 and O_2 for

$$\begin{aligned} & \mathcal{O} \left(\Xi T \frac{\log(\Xi T / \epsilon')}{\log \log(\Xi T / \epsilon')} \log(M) \log(M_H) \right) \\ &= \mathcal{O} \left(g(T) \frac{T}{\epsilon} \log^3 \left(\frac{g(T) T d \log d}{\epsilon} \log \left(\frac{1}{\tilde{g}(T)} \right) \right) \right), \end{aligned} \quad (111)$$

and additional gates for a total number of

$$\begin{aligned} & \mathcal{O} \left(\log(M) + \Xi T \frac{\log(\Xi T/\epsilon')}{\log \log(\Xi T/\epsilon')} (d \log^2(N) + \log(M_H)) \right) \\ &= \mathcal{O} \left(g(T) \frac{T}{\epsilon} \log \left(\frac{g(T)T}{\epsilon} \right) \left(d \log^2 \left(d + \log \left(\frac{T}{\tilde{g}(T)} \right) + \log \left(\frac{1}{\epsilon} \right) \right) + \log \left(\frac{g(T)T}{\epsilon} \right) \right) \right). \end{aligned} \quad (112)$$

With amplitude amplification, the average number of repeats to get a success is $\mathcal{O}(\|w\| \|\vec{u}_0\| / \|\vec{v}\|) = \mathcal{O}(g(T))$, so the overall complexity should be multiplied by this factor. $\square$

5 Nonlinear equations

We now discuss the full non-linear fractional reaction-diffusion equation as in [Equation \(1\)](#). For simplicity we only consider time-independent potential function $c(t, x) \equiv c(t)$. After spatial discretization, [Equation \(1\)](#) becomes a system of nonlinear ODEs with quadratic nonlinear term, which can be tackled by the Carleman linearization technique [25]. Existing algorithms based on Carleman linearization assume the sparsity of the coefficient matrices in the nonlinear ODEs. This facilitates the construction of the Carleman matrix, which is a direct sum of matrices in different dimensions. However, in the fractional reaction-diffusion equation, the coefficient matrices are unavoidably dense, so we will discuss a block-encoding implementation of the Carleman linearization technique.

5.1 Spatial discretization

We use the same spatial discretization as in the linear case. Let $\vec{u}(t)$ be a N^d -dimensional vector approximating the exact solution $u(t, x)$ at equi-distant grid points $(n_1/N, \dots, n_d/N)$ where $n_j \in [N]$. The spatially discretized equation can be written as

$$\frac{d}{dt} \vec{u} = F_1 \vec{u} + F_2 \vec{u}^{\otimes 2}. \quad (113)$$

Here $\vec{u}^{\otimes 2} = \vec{u} \otimes \vec{u}$ is a N^{2d} -dimensional vector. $F_1 = -B - C + aI$. F_2 is a $N^d \times N^{2d}$ dimensional matrix that maps $\vec{u}^{\otimes 2}$ to $-a\vec{u}$, i.e., each row of F_2 only has one non-zero entry to be $-a$ at its $((j-1)N^d + j)$ -th column (for the j -th row).

5.2 Carleman linearization

The idea of Carleman linearization for [Equation \(113\)](#) is to convert the nonlinear ODE to an equivalent infinite-dimensional linear ODE. Specifically, for any positive integer m , the tensor product $\vec{u}^{\otimes m}$ satisfies the ODE

$$\frac{d}{dt} (\vec{u}^{\otimes m}) = A_m^m \vec{u}^{\otimes m} + A_{m+1}^m \vec{u}^{\otimes (m+1)}, \quad (114)$$

where (I represents the identity matrix of dimension $N^d \times N^d$)

$$A_m^m = \sum_{j=1}^m I^{\otimes (j-1)} \otimes F_1 \otimes I^{\otimes (m-j)}, \quad (115)$$

$$A_{m+1}^m = \sum_{j=1}^m I^{\otimes (j-1)} \otimes F_2 \otimes I^{\otimes (m-j)}. \quad (116)$$

So the infinite-dimensional vector $[\vec{u}; \vec{u}^{\otimes 2}; \vec{u}^{\otimes 3}; \dots]$ satisfies a system of homogeneous linear ODE with coefficient matrix

$$\begin{pmatrix} A_1^1 & A_2^1 & & & & \\ & A_2^2 & A_3^2 & & & \\ & & \ddots & \ddots & & \\ & & & A_M^M & A_{M+1}^M & \\ & & & & \ddots & \ddots \end{pmatrix}. \quad (117)$$

To implement the Carleman linearization numerically, we truncate the infinite-dimensional ODE at a specific order M and consider the ODE

$$\frac{d}{dt}\vec{w} = A\vec{w}. \quad (118)$$

Here $\vec{w} = [\vec{w}_1; \vec{w}_2; \dots; \vec{w}_M]$ and each $\vec{w}_j$ is an N^{jd} -dimensional vector expected to approximate $\vec{u}^{\otimes j}$. The matrix A can be represented as

$$A = \begin{pmatrix} A_1^1 & A_2^1 & & & \\ & A_2^2 & A_3^2 & & \\ & & \ddots & \ddots & \\ & & & A_{M-1}^{M-1} & A_M^{M-1} \\ & & & & A_M^M \end{pmatrix}. \quad (119)$$

A crucial theoretical aspect on the efficiency of the Carleman linearization method is its convergence, i.e., whether or not $\vec{w}_1$, the first block in the solution of the truncated system in Equation (118), is a good approximation of $\vec{u}$, the solution of the nonlinear ODE in Equation (113). Several existing works [25, 26, 37, 38, 39, 40, 41] have studied this topic and established rigorous convergence guarantee in different scenarios. For example, in the ODE case, [25] shows that, when all the eigenvalues of the linear part F_1 are negative and the nonlinear part F_2 is relatively small compared to the decay rate of the linear part, $\vec{w}_1$ can be a good approximation of $\vec{u}$, and the truncation order M only needs to be logarithmic in the precision.

However, as pointed out in [26, 40], such a convergence condition might be too strong or even unrealistic when N is large. Then, for standard reaction-diffusion equations (i.e., $\alpha = 2$), these two works show an improved convergence analysis of the Carleman linearization with broader condition, by leveraging a maximum norm analysis [26, 40] and a rescaling technique [40]. We expect the same convergence for the fractional order as well. This is because the only difference between standard and fractional reaction diffusion equations lies in the eigenvalues of the discretized Laplace B , stretching from λ to $\lambda^{\alpha/2}$. Therefore, we expect that one can prove the convergence of the Carleman linearization following the analysis in [26, 40] with a modified definition of the eigenvalues of B , and we leave a careful study with explicit technical statements for future work.

5.3 Block-encoded implementation

The coefficient matrix A is expressed as a partitioned matrix with blocks in different dimension. When all the blocks are sparse, as assumed in existing quantum algorithm based on Carleman linearization, the entire matrix A is still sparse and one can directly implement it from sparse input model. This is also true for (integer-order) reaction-diffusion equation when $\alpha = 2$. However, when $0 < \alpha < 2$, while the off-diagonal blocks A_{m+1}^m are still sparse, the diagonal blocks A_m^m are unavoidably dense. Though the block-encoding of each block A_m^m is still construable, it is somewhat cumbersome to assemble them together to the block-encoding of the entire A .

Inspired by the technique of [25] for state preparation, we further enlarge the dimension of the linearized ODE. The purpose is to make the resulting enlarged coefficient matrix expressed as a partitioned matrix with blocks in the same dimension, and meanwhile a subspace of the enlarged solution is still the solution of the original ODE. Specifically, let us consider the ODE

$$\frac{d}{dt}\vec{y} = \tilde{A}\vec{y}. \quad (120)$$

Here $\vec{y} = [\vec{y}_1; \dots; \vec{y}_M]$ and each $\vec{y}_j$ is an N^{Md} -dimensional vector. The initial value $\vec{y}(0)$ is chosen to be $[|0\rangle^{\otimes M-1} \otimes |\vec{u}(0)\rangle; |0\rangle^{\otimes M-2} \otimes |\vec{u}(0)\rangle^{\otimes 2}; \dots; |\vec{u}(0)\rangle^{\otimes M}]$ where $|0\rangle = (0, \dots, 0, 1)^T$ in N^d dimension. The coefficient matrix is

$$\tilde{A} = \begin{pmatrix} \tilde{A}_1^1 & \tilde{A}_2^1 & & & \\ & \tilde{A}_2^2 & \tilde{A}_3^2 & & \\ & & \ddots & \ddots & \\ & & & \tilde{A}_{M-1}^{M-1} & \tilde{A}_M^{M-1} \\ & & & & \tilde{A}_M^M \end{pmatrix}. \quad (121)$$

Here each $\tilde{A}_m^m$ and $\tilde{A}_{m+1}^m$ is an N^{Md} -dimensional square matrix. $\tilde{A}_m^m = I^{\otimes(M-m)} \otimes A_m^m$ for $m < M$ and $\tilde{A}_M^M = A_M^M$. $\tilde{A}_{m+1}^m$ contains A_{m+1}^m at its most bottom right and otherwise 0.

We now show that the non-zero entries of $\vec{y}$ exactly form $\vec{w}$. To this end, for each $\vec{y}_m$, we write it as $[\vec{y}_{m,1}; \dots; \vec{y}_{m,N^{(M-m)d}}]$, where each $\vec{y}_{m,j}$ is an N^{md} -dimensional vector. By definition of $\vec{y}(0)$, we have that for every m , all but the last component $\vec{y}_{m,N^{(M-m)d}}(0)$ of $\vec{y}_m(0)$ are zero. Furthermore, by the definition of $\tilde{A}$, the variables $\vec{y}_{m,j}$ for $j \neq N^{(M-m)d}$ does not interact with other variables outside. Therefore, $\vec{y}_{m,j}(t)$ is always 0 for all t, m and $j \neq N^{(M-m)d}$. Now, if we only focus on the ODEs that $\vec{y}_{m,N^{(M-m)d}}$'s satisfy, it is exactly the original ODE Equation (118) with the same initial condition. So we have $[\vec{y}_{1,N^{(M-1)d}}(t); \vec{y}_{2,N^{(M-2)d}}(t); \dots; \vec{y}_{M,1}(t)] = \vec{w}(t)$ for all t . This implies that, instead of solving Equation (118), we can focus on its equivalent formalism Equation (120).

Equation (120) can be solved by standard quantum algorithm for linear ODEs. For example, we can use the method based on truncated Dyson series [21] discussed in the previous section. To use this method, we need the state preparation for $\vec{y}(0)$ and the block-encoding of $\tilde{A}$. The state preparation oracle can be constructed in a similar manner as in [25]. To construct the block-encoding of $\tilde{A}$, we decompose it as $\tilde{A} = \tilde{D} + \tilde{R}$, where

$$\tilde{D} = \begin{pmatrix} \tilde{A}_1^1 & & & & \\ & \tilde{A}_2^2 & & & \\ & & \ddots & & \\ & & & \tilde{A}_{M-1}^{M-1} & \\ & & & & \tilde{A}_M^M \end{pmatrix}, \quad \tilde{R} = \begin{pmatrix} 0 & \tilde{A}_2^1 & & & \\ & 0 & \tilde{A}_3^2 & & \\ & & \ddots & \ddots & \\ & & & 0 & \tilde{A}_M^{M-1} \\ & & & & 0 \end{pmatrix}. \quad (122)$$

Notice that $\tilde{R}$ is a $\mathcal{O}(M)$ -sparse matrix. According to [32, Lemma 48], we may implement a block-encoding of $\tilde{R}$ with $\mathcal{O}(1)$ query complexity and $\mathcal{O}(M)$ block-encoding factor. For $\tilde{D}$, we start with the block-encoding of F_1 , which can be constructed from the linear combination of B, C and aI . According to Theorem 2, the block-encoding factor is $\mathcal{O}(d^\alpha N^{2\alpha})$, corresponding to the spectral norm of B , and the query complexity for block encoding F_1 is $\mathcal{O}(1)$. Denote this block-encoding by U_{F_1} . Then a block-encoding of $I^{\otimes(j-1)} \otimes F_1 \otimes I^{\otimes(m-j)}$ can be constructed by applying U_{F_1} on the correct register. Since A_m^m is the summation of $I^{\otimes(j-1)} \otimes F_1 \otimes I^{\otimes(m-j)}$, according to Theorem 2, we may further construct the block-encoding of A_m^m , denoted by $U_{A_m^m}$, with $\mathcal{O}(Md^\alpha N^{2\alpha})$ block-encoding factor. Here we choose the block-encoding factor for all A_m^m to be the same and corresponds to the worst case A_M^M with most summation terms in order to facilitate later construction for bigger matrix. Then, for each m , a block-encoding of $\tilde{A}_m^m$, denoted by $U_{\tilde{A}_m^m}$, can be constructed with a single use of $U_{A_m^m}$, and the block-encoding of $\tilde{D}$ is given by

$$\sum_{m=0}^{M-1} |m\rangle \langle m| \otimes U_{\tilde{A}_m^m}. \quad (123)$$

This can be implemented by controlled version of $U_{A_m^m}$ and requires $\mathcal{O}(M)$ query complexity, and the block-encoding factor is $\mathcal{O}(Md^\alpha N^{2\alpha})$. The final step is to use Theorem 2 again and, from the block-encoding of $\tilde{D}$ and $\tilde{R}$, we may construct the desired block-encoding of $\tilde{A}$ with $\mathcal{O}(M)$ query complexity and $\mathcal{O}(Md^\alpha N^{2\alpha})$ block-encoding factor. Notice that both query complexity and the block-encoding factor does not involve exponential dependence on d .

With the state preparation for $\vec{y}(0)$ and the block-encoding of $\tilde{A}$, we can solve Equation (120) efficiently using truncated Dyson series method [21]. Notice that the method requires $\tilde{A}$ to have non-positive logarithmic norm, which can be guaranteed if all the eigenvalues of F_1 are negative and F_2 is bounded, which corresponds to the standard assumption on the boundedness of the nonlinearity (namely the condition $R_D < 1$ in [26]).

6 Conclusion

In this paper, we study efficient quantum algorithms for linear and nonlinear fractional reaction-diffusion equations. For linear equations, we improve and analyze the complexity of four different methods: second-order Trotter formula, time-marching method, truncated Dyson series

method, and the linear combination of Hamiltonian simulation with the interaction picture formalism (LCHS-IP). Among all the methods, the LCHS-IP method achieves best scaling in the spatial dimension and thus is most suitable for high-dimensional linear fractional reaction-diffusion equations. For nonlinear equations, we generalize the quantum Carleman linearization algorithm to the case with block-encoding input oracle, making the algorithm applicable to dense coefficient matrices.

A natural extension of this work is to design better quantum algorithm for linear reaction-diffusion equations with near-optimal scalings in all parameters. A desired algorithm is expected to simultaneously scale poly-logarithmically in dimension (as LCHS-IP), poly-logarithmically in precision (as time-marching and truncated Dyson series methods), linearly in evolution time (as truncated Dyson series and LCHS-IP), and have low state preparation cost (as Trotter, time-marching and LCHS-IP). Among all the methods being considered in this paper, the LCHS-IP method is the closest one as it only misses the poly-logarithmic dependence on precision, so it is interesting to explore whether the LCHS-IP method can be further exponentially improved in terms of precision. Another possibility is to use higher-order product formula, whose asymptotic scaling tends to be $T^{1+o(1)}/\epsilon^{o(1)}$. However, as proved in [30], there might be extra exponential overhead in Trotter errors when we deal with non-unitary dynamics. Therefore a tailored design of the product formula and an improved error analysis would be necessary. Furthermore, to obtain a poly-logarithmic dependence on dimension, we may need to take advantage of the vector-norm scaling of the product formula, which states that the Trotter errors may be independent of the spectral norm of the Hamiltonians if the quantum states are within a more regular subspace with better smoothness assumption. Such a vector-norm scaling has been proved for first- and second-order Trotter applied to Hamiltonian simulation in [35], and it remains open to establish similar error bounds for higher-order product formula and the cases beyond Hamiltonian simulation. This is our ongoing work.

For non-linear equations, a natural next step is to establish a rigorous analysis with detailed computational costs. We expect the analysis presented in [26] to work with suitable modifications. However, the complexity estimate in [26] still depends polynomially on the dimension d . It is interesting to explore whether the quantum Carleman algorithm can also avoid such a polynomial overhead with the help of tighter error bounds, or new techniques are necessary to achieve this task.

Throughout this paper, we focus on the spectral fractional Laplacian with periodic boundary condition, which facilitates its spatial discretization and quantum implementation of its time evolution through the QFT circuit. Our future work will be focusing on the fractional Laplacian operator with Riesz definition and exploring the efficiency of quantum algorithms.

Acknowledgments

DA acknowledges the support by the Department of Defense through the Hartree Postdoctoral Fellowship at QuICS, and the seed grant at the NSF Quantum Leap Challenge Institute for Robust Quantum Simulation (QLCI grant OMA-2120757). KT gratefully acknowledges the support by the National Science Foundation under the grants DMS-2231533 and DMS-2008568.

References

- [1] Nicholas F. Britton. “Reaction–diffusion equations and their applications to biology”. Academic Press. (1986). url: <https://www.cabidigitallibrary.org/doi/full/10.5555/19870540782>.
- [2] Robert Stephen Cantrell and Chris Cosner. “Spatial ecology via reaction–diffusion equations”. Wiley. (2003).
- [3] Peter Grindrod. “The theory and applications of reaction-diffusion equations: Patterns and waves”. Clarendon Press. (1996).
- [4] Franz Rothe. “Global solutions of reaction-diffusion systems”. Springer-Verlag, Berlin. (1984).
- [5] Joel Smoller. “Shock waves and reaction—diffusion equations”. Springer New York, NY. (1994).

- [6] James D. Murray. “Mathematical biology”. [Springer Berlin, Heidelberg](#). (1993).
- [7] Michael G. Neubert and Hal Caswell. “Demography and dispersal: Calculation and sensitivity analysis of invasion speed for structured populations”. [Ecology](#) **81**, 1613–1628 (2000).
- [8] I. M. Sokolov and J. Klafter. “From diffusion to anomalous diffusion: A century after einstein’s brownian motion”. [Chaos](#) **15**, 026103 (2005).
- [9] Anna Lischke, Guofei Pang, Mamikon Gulian, Fangying Song, Christian Glusa, Xiaoning Zheng, Zhiping Mao, Wei Cai, Mark M. Meerschaert, Mark Ainsworth, and George Em Karniadakis. “What is the fractional Laplacian? A comparative review with new results”. [Journal of Computational Physics](#) **404**, 109009 (2020).
- [10] Dominic W. Berry. “High-order quantum algorithm for solving linear differential equations”. [Journal of Physics A: Mathematical and Theoretical](#) **47**, 105301 (2014).
- [11] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. “Quantum algorithm for linear systems of equations”. [Physical Review Letters](#) **103**, 150502 (2009).
- [12] 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).
- [13] Yigit Subaşı, Rolando D. Somma, and Davide Orsucci. “Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing”. [Physical Review Letters](#) **122**, 060504 (2019).
- [14] Dong An and Lin Lin. “Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm”. [ACM Transactions on Quantum Computing](#) **3**, 1–28 (2022).
- [15] Lin Lin and Yu Tong. “Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems”. [Quantum](#) **4**, 361 (2020).
- [16] 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).
- [17] 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).
- [18] Andrew M. Childs and Jin-Peng Liu. “Quantum spectral methods for differential equations”. [Communications in Mathematical Physics](#) **375**, 1427–1457 (2020).
- [19] Andrew M. Childs, Jin-Peng Liu, and Aaron Ostrander. “High-precision quantum algorithms for partial differential equations”. [Quantum](#) **5**, 574 (2021).
- [20] Hari Krovi. “Improved quantum algorithms for linear and nonlinear differential equations”. [Quantum](#) **7**, 913 (2023).
- [21] Dominic W. Berry and Pedro C. S. Costa. “Quantum algorithm for time-dependent differential equations using dyson series”. [Quantum](#) **8**, 1369 (2024).
- [22] Di Fang, Lin Lin, and Yu Tong. “Time-marching based quantum solvers for time-dependent linear differential equations”. [Quantum](#) **7**, 955 (2023).
- [23] Shi Jin, Nana Liu, and Yue Yu. “Quantum simulation of partial differential equations via schrödingerization”. [Physical Review Letters](#) **133**, 230602 (2024).
- [24] 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).
- [25] Jin-Peng Liu, Herman Øie Kolden, Hari K. Krovi, Nuno F. Loureiro, Konstantina Trivisa, and Andrew M. Childs. “Efficient quantum algorithm for dissipative nonlinear differential equations”. [Proceedings of the National Academy of Sciences](#) **118**, e2026805118 (2021).
- [26] Jin-Peng Liu, Dong An, Di Fang, Jiasu Wang, Guang Hao Low, and Stephen Jordan. “Efficient quantum algorithm for nonlinear reaction–diffusion equations and energy estimation”. [Communications in Mathematical Physics](#) **404**, 963–1020 (2023).
- [27] Guang Hao Low and Nathan Wiebe. “Hamiltonian simulation in the interaction picture” (2019). [arXiv:1805.00675](#).
- [28] Andrew M. Childs, Jiaqi Leng, Tongyang Li, Jin-Peng Liu, and Chenyi Zhang. “Quantum simulation of real-space dynamics”. [Quantum](#) **6**, 860 (2022).

- [29] 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).
- [30] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. “Theory of trotter error with commutator scaling”. *Physical Review X* **11**, 011020 (2021).
- [31] Andrew M. Childs and Nathan Wiebe. “Hamiltonian simulation using linear combinations of unitary operations”. *Quantum Information and Computation* **12**, 901–924 (2012).
- [32] 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*. Page 193–204. STOC ’19. ACM (2019).
- [33] Michael A. Nielsen and Isaac L. Chuang. “Quantum computation and quantum information”. *Cambridge University Press*. (2000).
- [34] Tobias Jahnke and Christian Lubich. “Error bounds for exponential operator splittings”. *BIT Numerical Mathematics* **40**, 735–744 (2000).
- [35] Dong An, Di Fang, and Lin Lin. “Time-dependent unbounded hamiltonian simulation with vector norm scaling”. *Quantum* **5**, 459 (2021).
- [36] Lov Grover and Terry Rudolph. “Creating superpositions that correspond to efficiently integrable probability distributions” (2002). [arXiv:quant-ph/0208112](https://arxiv.org/abs/quant-ph/0208112).
- [37] Marcelo Forets and Amaury Pouly. “Explicit error bounds for Carleman linearization” (2017). [arXiv:1711.02552](https://arxiv.org/abs/1711.02552).
- [38] Arash Amini, Qiyu Sun, and Nader Motee. “Error bounds for carleman linearization of general nonlinear systems”. In *2021 Proceedings of the Conference on Control and its Applications (CT)*. Pages 1–8. (2021).
- [39] Arash Amini, Cong Zheng, Qiyu Sun, and Nader Motee. “Carleman linearization of nonlinear systems and its finite-section approximations”. *Discrete and Continuous Dynamical Systems - B* **30**, 577–603 (2025).
- [40] Pedro C. S. Costa, Philipp Schleich, Mauro E. S. Morales, and Dominic W. Berry. “Further improving quantum algorithms for nonlinear differential equations via higher-order methods and rescaling”. *npj Quantum Information* **11**, 141 (2025).
- [41] Hsuan-Cheng Wu, Jingyao Wang, and Xiantao Li. “Quantum algorithms for nonlinear dynamics: Revisiting Carleman linearization with no dissipative conditions”. *SIAM Journal on Scientific Computing* **47**, A943–A970 (2025).

A Technical lemmas

Lemma 19. Let $\vec{a}$ and $\vec{b}$ be two non-zero vectors, possibly unnormalized. Then

$$\left\| \frac{\vec{a}}{\|\vec{a}\|} - \frac{\vec{b}}{\|\vec{b}\|} \right\| \leq \frac{2\|\vec{a} - \vec{b}\|}{\|\vec{a}\|}. \quad (124)$$

Lemma 20. Suppose $A(t)$ is a matrix-valued continuous function and the operator $S(t, s)$ solves the differential equation

$$\frac{d}{dt}S(t, s) = A(t)S(t, s), \quad S(s, s) = I. \quad (125)$$

Then,

1. for any matrix-valued continuous function $R(t)$, the solution of the differential equation

$$\frac{d}{dt}\tilde{S}(t, 0) = A(t)\tilde{S}(t, 0) + R(t), \quad \tilde{S}(0, 0) = I \quad (126)$$

can be represented as

$$\tilde{S}(t, 0) = S(t, 0) + \int_0^t S(t, s)R(s)ds. \quad (127)$$

2. for any vector-valued continuous function $\vec{r}(t)$, the solution of the differential equation

$$\frac{d}{dt}\vec{\psi}(t) = A(t)\vec{\psi}(t) + \vec{r}(t), \quad \vec{\psi}(0) = \vec{\psi}_0 \quad (128)$$

can be represented as

$$\vec{\psi}(t) = S(t, 0)\vec{\psi}_0 + \int_0^t S(t, s)\vec{r}(s)ds. \quad (129)$$

B Proof of Theorem 5

Proof. For any function f defined on $[0, 1]^d$, its Fourier coefficient is defined to be

$$\hat{f}_k = \int_{[0, 1]^d} f(x) e^{-2\pi i(k_0 x_0 + \dots + k_{d-1} x_{d-1})} dx. \quad (130)$$

Using integration by parts for p times, we may obtain

$$|\hat{f}_k| \leq \frac{\max_j \|\partial_{x_j}^p f(x)\|_{L^1}}{(2\pi\|k\|_\infty)^p}. \quad (131)$$

As a result, for each fixed m ,

$$\begin{aligned} \left| \sum_{j \in \mathbb{Z}^d} \hat{u}_{m+jN} - \hat{u}_{i(m)} \right| &\leq \sum_{l=1}^{\infty} \sum_{\|j\|_\infty=l} |\hat{u}_{i(m)+jN}| \\ &\leq \sum_{l=1}^{\infty} ((l+1)^d - l^d) \frac{\max_j \|\partial_{x_j}^p u_0(x)\|_{L^1}}{(\pi(2l-1)N)^p}. \end{aligned} \quad (132)$$

Suppose that $p \geq d + 2$. Using the inequality that $2l - 1 \geq l + 1$ for all $l \geq 2$, we have

$$\begin{aligned}
\left| \sum_{j \in \mathbb{Z}^d} \hat{u}_{m+jN} - \hat{u}_{i(m)} \right| &\leq \left(2^d - 1 + \sum_{l=2}^{\infty} \frac{(l+1)^d - l^d}{(2l-1)^p} \right) \frac{\max_j \|\partial_{x_j}^p u_0(x)\|_{L^1}}{\pi^p N^p} \\
&\leq \left(2^d - 1 + \sum_{l=2}^{\infty} \frac{1}{(2l-1)^{p-d}} \right) \frac{\max_j \|\partial_{x_j}^p u_0(x)\|_{L^1}}{\pi^p N^p} \\
&\leq \left(2^d - 1 + \int_1^{\infty} \frac{dx}{(2x-1)^{p-d}} \right) \frac{\max_j \|\partial_{x_j}^p u_0(x)\|_{L^1}}{\pi^p N^p} \\
&= \left(2^d - 1 + \frac{1}{2(p-d-1)} \right) \frac{\max_j \|\partial_{x_j}^p u_0(x)\|_{L^1}}{\pi^p N^p} \\
&\leq \frac{\max_j \|\partial_{x_j}^p u_0(x)\|_{L^1}}{(\pi/2)^p N^{p-d}}.
\end{aligned} \tag{133}$$

Therefore,

$$\begin{aligned}
&\left\| (\mathcal{F}^{-1})^{\otimes d} \vec{u}_0 - N^{d/2} \sum_{m \in [N]^d} \hat{u}_{i(m)} |m_0\rangle \cdots |m_{d-1}\rangle \right\| \\
&= N^{d/2} \left\| \sum_{m \in [N]^d} \left(\sum_{j \in \mathbb{Z}^d} \hat{u}_{m+jN} - \hat{u}_{i(m)} \right) |m_0\rangle \cdots |m_{d-1}\rangle \right\| \\
&\leq \frac{\max_j \|\partial_{x_j}^p u_0(x)\|_{L^1}}{(\pi/2)^p N^{p-d}}.
\end{aligned} \tag{134}$$

Similarly,

$$\left\| (\mathcal{F}^{-1})^{\otimes d} \vec{u}(T, x) - N^{d/2} \sum_{m \in [N]^d} \hat{u}_{i(m)} e^{-(2\pi\|i(m)\|)^{\alpha} T} |m_0\rangle \cdots |m_{d-1}\rangle \right\| \leq \frac{\max_j \|\partial_{x_j}^p u(T, x)\|_{L^1}}{(\pi/2)^p N^{p-d}}. \tag{135}$$

Let $\tilde{\mathcal{F}} = (\mathcal{F})^{\otimes d} \otimes I \otimes I$ and U denote the composition of O_1 , $O_{\text{exp},1}$ and c- R specified in [Figure 2](#), then

$$\begin{aligned}
&\left\| (I^{\otimes d} \otimes \langle 0|^{\otimes 3}) \tilde{\mathcal{F}} U \tilde{\mathcal{F}}^{-1} |u_0\rangle |0\rangle^{\otimes 3} - \frac{1}{\|\vec{u}_0\|} \vec{u}(T) \right\| \\
&\leq \left\| (I^{\otimes d} \otimes \langle 0|^{\otimes 3}) \tilde{\mathcal{F}} U \tilde{\mathcal{F}}^{-1} |u_0\rangle |0\rangle^{\otimes 3} - (I^{\otimes d} \otimes \langle 0|^{\otimes 3}) \tilde{\mathcal{F}} U \frac{N^{d/2}}{\|\vec{u}_0\|} \sum_{m \in [N]^d} \hat{u}_{i(m)} |m_0\rangle \cdots |m_{d-1}\rangle |0\rangle^{\otimes 3} \right\| \\
&\quad + \left\| (I^{\otimes d} \otimes \langle 0|^{\otimes 3}) \tilde{\mathcal{F}} \left(\frac{N^{d/2}}{\|\vec{u}_0\|} \sum_{m \in [N]^d} \hat{u}_{i(m)} e^{-(2\pi\|i(m)\|)^{\alpha} T} |m_0\rangle \cdots |m_{d-1}\rangle |0\rangle^{\otimes 3} + |\perp\rangle \right) \right. \\
&\quad \left. - \frac{1}{\|\vec{u}_0\|} \sum_{m \in [N]^d} u(T, m/N) |m_0\rangle \cdots |m_{d-1}\rangle \right\| \\
&\leq \left\| (\mathcal{F}^{-1})^{\otimes d} |u_0\rangle - \frac{N^{d/2}}{\|\vec{u}_0\|} \sum_{m \in [N]^d} \hat{u}_{i(m)} |m_0\rangle \cdots |m_{d-1}\rangle \right\| \\
&\quad + \left\| \mathcal{F}^{\otimes d} \frac{N^{d/2}}{\|\vec{u}_0\|} \sum_{m \in [N]^d} \hat{u}_{i(m)} e^{-(2\pi\|i(m)\|)^{\alpha} T} |m_0\rangle \cdots |m_{d-1}\rangle - \frac{1}{\|\vec{u}_0\|} \sum_{m \in [N]^d} u(T, m/N) |m_0\rangle \cdots |m_{d-1}\rangle \right\| \\
&\leq \frac{1}{\|\vec{u}_0\|} \frac{\max_j \|\partial_{x_j}^p u_0(x)\|_{L^1} + \max_j \|\partial_{x_j}^p u(T, x)\|_{L^1}}{(\pi/2)^p N^{p-d}}.
\end{aligned} \tag{136}$$

By [Theorem 19](#), the overall 2-norm error of the output quantum state after successful measurement can be bounded by

$$\frac{4}{\|\vec{u}_0\|} \frac{\max_{j,t \in \{0,T\}} \|\partial_{x_j}^p u(t,x)\|_{L^1}}{(\pi/2)^p N^{p-d}} \frac{\|\vec{u}_0\|}{\|\vec{u}(T)\|} = \frac{4 \max_{j,t \in \{0,T\}} \|\partial_{x_j}^p u(t,x)\|_{L^1}}{\|\vec{u}(T)\| (\pi/2)^p N^{p-d}}. \quad (137)$$

The claimed complexity can be shown by noticing that in each run of [Figure 2](#), we need $\mathcal{O}(1)$ queries to the aforementioned oracles, and $\mathcal{O}(d \log^2(N))$ gates due to QFT and controlled rotations, and under amplitude amplification, the averaged repeats to get a success is $\mathcal{O}(\|\vec{u}_0\|/\|\vec{u}(T)\|)$. $\square$

C Bounding the spatial discretization errors

Here we present the detailed proofs of the spatial discretization results ([Theorem 6](#), [Theorem 7](#) and [Theorem 8](#)).

Proof of [Theorem 6](#). Let $\vec{u}_e(t) = (u(t, n/N))_{n \in [N]^d}$ denote the exact solution evaluated at discrete grid points. Using [Equation \(29\)](#), for any $n \in [N]^d$,

$$\begin{aligned} \partial_t u(t, n/N) &= -(-\Delta)^{\alpha/2} u(t, n/N) - c(t, n/N) u(t, n/N) \\ &= (-B\vec{u}_e)_n - c(t, n/N) u(t, n/N) + (B\vec{u}_e)_n - (-\Delta)^{\alpha/2} u(t, n/N). \end{aligned} \quad (138)$$

Therefore the vector $\vec{u}_e(t)$ solves the differential equation

$$\frac{d}{dt} \vec{u}_e = -B\vec{u}_e - C(t)\vec{u}_e + \vec{r}(t), \quad (139)$$

where

$$(\vec{r}(t))_n = (B\vec{u}_e)_n - (-\Delta)^{\alpha/2} u(t, n/N). \quad (140)$$

Therefore, the equations $\vec{u}_e$ satisfies can be viewed as a perturbation (by $\vec{r}$) of the equations $\vec{u}$ satisfies. By [Theorem 20](#), we have

$$\vec{u}_e(t) = \vec{u}(t) + \int_0^t \mathcal{T}e^{\int_s^t (-B-C(\tau))d\tau} \vec{r}(s) ds, \quad (141)$$

and thus

$$\|\vec{u}_e(T) - \vec{u}(T)\| \leq T \max_{t \in [0,T]} \|\vec{r}(t)\|. \quad (142)$$

It remains to bound $\|\vec{r}(t)\|$, which can be done similarly to the proof of [Theorem 5](#). Suppose that the Fourier series of $u(t, x)$ is

$$u(t, x) = \sum_{k \in \mathbb{Z}^d} \hat{u}_k(t) e^{2\pi i(k_0 x_0 + \dots + k_{d-1} x_{d-1})}, \quad (143)$$

where

$$\hat{u}_k(t) = \int_{[0,1]^d} u(t, x) e^{-2\pi i(k_0 x_0 + \dots + k_{d-1} x_{d-1})} dx. \quad (144)$$

By the same reasoning of [Equation \(134\)](#),

$$\left\| (\mathcal{F}^{-1})^{\otimes d} \vec{u}_e(t) - N^{d/2} \sum_{m \in [N]^d} \hat{u}_{i(m)}(t) |m_0\rangle \dots |m_{d-1}\rangle \right\| \leq \frac{\max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{(\pi/2)^p N^{p-d}}. \quad (145)$$

Noticing that $\|D\| \leq \pi^\alpha N^\alpha d^{\alpha/2}$, we have

$$\left\| D(\mathcal{F}^{-1})^{\otimes d} \vec{u}_e(t) - N^{d/2} \sum_{m \in [N]^d} 2^\alpha \pi^\alpha \|i(m)\|^\alpha \hat{u}_{i(m)}(t) |m_0\rangle \dots |m_{d-1}\rangle \right\| \leq \frac{2^p d^{\alpha/2} \max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-d-\alpha}}. \quad (146)$$

Again using the derivation of Equation (134) and noting that $(-\Delta)^{\alpha/2}u(t, x) = \sum_{k \in \mathbb{Z}^d} 2^\alpha \pi^\alpha \|k\|^\alpha \hat{u}_k(t) e^{2\pi i(k_0 x_0 + \dots + k_{d-1} x_{d-1})}$, we have

$$\begin{aligned} & \left\| (\mathcal{F}^{-1})^{\otimes d} \left((-\Delta)^{\alpha/2} u(t, n/N) \right)_{n \in [N]^d} - N^{d/2} \sum_{m \in [N]^d} 2^\alpha \pi^\alpha \|i(m)\|^\alpha \hat{u}_{i(m)}(t) |m_0\rangle \cdots |m_{d-1}\rangle \right\| \\ &= N^{d/2} \left\| \sum_{m \in [N]^d} \left(\sum_{j \in \mathbb{Z}^d} 2^\alpha \pi^\alpha \|m + jN\|^\alpha \hat{u}_{m+jN}(t) - 2^\alpha \pi^\alpha \|i(m)\|^\alpha \hat{u}_{i(m)}(t) \right) |m_0\rangle \cdots |m_{d-1}\rangle \right\|. \end{aligned} \quad (147)$$

Using Equation (131), for each fixed m , we have

$$\begin{aligned} & \left| \sum_{j \in \mathbb{Z}^d} 2^\alpha \pi^\alpha \|m + jN\|^\alpha \hat{u}_{m+jN}(t) - 2^\alpha \pi^\alpha \|i(m)\|^\alpha \hat{u}_{i(m)}(t) \right| \\ & \leq \sum_{l=1}^{\infty} \sum_{\|j\|_\infty=l} 2^\alpha \pi^\alpha \|i(m) + jN\|^\alpha |\hat{u}_{i(m)+jN}(t)| \\ & \leq \sum_{l=1}^{\infty} \sum_{\|j\|_\infty=l} 2^\alpha \pi^\alpha \|i(m) + jN\|^\alpha \frac{\max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{(2\pi \|i(m) + jN\|_\infty)^p} \\ & \leq \sum_{l=1}^{\infty} ((l+1)^d - l^d) \pi^\alpha (2l+1)^\alpha N^\alpha d^{\alpha/2} \frac{\max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{(\pi(2l-1)N)^p} \end{aligned} \quad (148)$$

Therefore, for any $p \geq d + \alpha + 2$,

$$\begin{aligned} & \left| \sum_{j \in \mathbb{Z}^d} 2^\alpha \pi^\alpha \|m + jN\|^\alpha \hat{u}_{m+jN}(t) - 2^\alpha \pi^\alpha \|i(m)\|^\alpha \hat{u}_{i(m)}(t) \right| \\ & \leq \left(\sum_{l=1}^{\infty} \frac{((l+1)^d - l^d)(2l+1)^\alpha}{(2l-1)^p} \right) \frac{d^{\alpha/2} \max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-\alpha}} \\ & \leq \left((2^d - 1)3^\alpha + \sum_{l=2}^{\infty} \frac{((l+1)^d - l^d)(2l+1)^\alpha}{(2l-1)^p} \right) \frac{d^{\alpha/2} \max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-\alpha}} \\ & \leq \left((2^d - 1)3^\alpha + 2^\alpha \sum_{l=2}^{\infty} \frac{1}{(2l-1)^{p-d-\alpha}} \right) \frac{d^{\alpha/2} \max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-\alpha}} \\ & \leq \left((2^d - 1)3^\alpha + 2^\alpha \frac{1}{2(p-d-\alpha-1)} \right) \frac{d^{\alpha/2} \max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-\alpha}} \\ & \leq \frac{2^d 3^\alpha d^{\alpha/2} \max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-\alpha}}. \end{aligned} \quad (149)$$

As a result,

$$\begin{aligned} & \left\| (\mathcal{F}^{-1})^{\otimes d} \left((-\Delta)^{\alpha/2} u(t, n/N) \right)_{n \in [N]^d} - N^{d/2} \sum_{m \in [N]^d} 2^\alpha \pi^\alpha \|i(m)\|^\alpha \hat{u}_{i(m)}(t) |m_0\rangle \cdots |m_{d-1}\rangle \right\| \\ &= N^{d/2} \left\| \sum_{m \in [N]^d} \left(\sum_{j \in \mathbb{Z}^d} 2^\alpha \pi^\alpha \|m + jN\|^\alpha \hat{u}_{m+jN}(t) - 2^\alpha \pi^\alpha \|i(m)\|^\alpha \hat{u}_{i(m)}(t) \right) |m_0\rangle \cdots |m_{d-1}\rangle \right\| \\ &\leq \frac{2^d 3^\alpha d^{\alpha/2} \max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-d-\alpha}}. \end{aligned} \quad (150)$$

Combining Equation (146) and Equation (150), we have

$$\left\| D(\mathcal{F}^{-1})^{\otimes d} \vec{u}_e(t) - (\mathcal{F}^{-1})^{\otimes d} \left((-\Delta)^{\alpha/2} u(t, n/N) \right)_{n \in [N]^d} \right\| \leq \frac{2^{p+1} d^{\alpha/2} \max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-d-\alpha}}, \quad (151)$$

and

$$\|r(t)\| \leq \frac{2^{p+1} d^{\alpha/2} \max_j \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-d-\alpha}}. \quad (152)$$

Plugging this back into Equation (142) and we obtain

$$\|\vec{u}_e(T) - \vec{u}(T)\| \leq T \frac{2^{p+1} d^{\alpha/2} \max_{t,j} \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-d-\alpha}}. \quad (153)$$

□

Proof of Theorem 7. According to Theorem 6, the spatial discretization error can be bounded by

$$\begin{aligned} \|(u(T, n/N))_{n \in [N]^d} - \vec{u}(T)\| &\leq T \frac{2^{p+1} d^{\alpha/2} \max_{t,j} \|\partial_{x_j}^p u(t, x)\|_{L^1}}{\pi^{p-\alpha} N^{p-d-\alpha}} \\ &\leq 2\Lambda \pi^\alpha d^{\alpha/2} T \frac{(2\Lambda/\pi)^p (p!)^\sigma}{N^{p-d-\alpha}} \\ &\leq 2^{1+\sigma} \pi^{\alpha+\sigma/2} \Lambda d^{\alpha/2} T \frac{p^{\sigma/2} (2\Lambda/\pi)^p (p/e)^{p\sigma}}{N^{p-d-\alpha}} \end{aligned} \quad (154)$$

where in the second line we use $p! \leq \sqrt{2\pi p} (p/e)^p e^{1/(12p)} \leq 2\sqrt{\pi p} (p/e)^p$. We choose $p = \lfloor p_* \rfloor$ where

$$p_* = \left(\frac{\pi}{2\Lambda} \right)^{1/\sigma} N^{1/\sigma}. \quad (155)$$

We remark that here we need $p_* \geq d + \alpha + 2$ and thus $N \geq (2\Lambda/\pi)(d + \alpha + 2)^\sigma$. Then,

$$\begin{aligned} \|(u(T, n/N))_{n \in [N]^d} - \vec{u}(T)\| &\leq 2^{2+\sigma} \pi^{\alpha+\sigma/2} \Lambda d^{\alpha/2} T \frac{p_*^{\sigma/2} (2\Lambda/\pi)^{p_*} (p_*/e)^{p_*\sigma}}{N^{p_*-d-\alpha}} \\ &= 2^{3/2+\sigma} \pi^{\alpha+\sigma/2+1/2} \Lambda^{1/2} T d^{\alpha/2} N^{d+\alpha+1/2} \exp \left(-\sigma \left(\frac{\pi}{2\Lambda} \right)^{1/\sigma} N^{1/\sigma} \right). \end{aligned} \quad (156)$$

By the Taylor expansion $e^x = \sum_k \frac{1}{k!} x^k$ and thus $e^x \geq \frac{1}{k!} x^k$ for $x \geq 0$, we have

$$\begin{aligned} \exp \left(\frac{\sigma}{2} \left(\frac{\pi}{2\Lambda} \right)^{1/\sigma} N^{1/\sigma} \right) &\geq \frac{1}{\lceil \sigma(d + \alpha + 1/2) \rceil!} \left(\frac{\sigma}{2} \left(\frac{\pi}{2\Lambda} \right)^{1/\sigma} \right)^{\lceil \sigma(d + \alpha + 1/2) \rceil} N^{d+\alpha+1/2} \\ &\geq \left(\frac{\sigma}{2 \lceil \sigma(d + \alpha + 1/2) \rceil} \left(\frac{\pi}{2\Lambda} \right)^{1/\sigma} \right)^{\lceil \sigma(d + \alpha + 1/2) \rceil} N^{d+\alpha+1/2}. \end{aligned} \quad (157)$$

Therefore

$$\begin{aligned} &\|(u(T, n/N))_{n \in [N]^d} - \vec{u}(T)\| \\ &\leq 2^{3/2+\sigma} \pi^{\alpha+\sigma/2+1/2} \Lambda^{1/2} T d^{\alpha/2} \left(\frac{2 \lceil \sigma(d + \alpha + 1/2) \rceil}{\sigma} \left(\frac{2\Lambda}{\pi} \right)^{1/\sigma} \right)^{\lceil \sigma(d + \alpha + 1/2) \rceil} \exp \left(-\frac{\sigma}{2} \left(\frac{\pi}{2\Lambda} \right)^{1/\sigma} N^{1/\sigma} \right) \\ &\leq c_1 T (c_2 d)^{c_3 d} d^{\alpha/2} \exp \left(-c_4 N^{1/\sigma} \right), \end{aligned} \quad (158)$$

where we may choose

$$c_1 = 2^{3/2+\sigma} \pi^{\alpha+\sigma/2+1/2} \Lambda^{1/2}, \quad (159)$$

$$c_2 = \max \left\{ 2(2 + \alpha + 1/\sigma) \left(\frac{2\Lambda}{\pi} \right)^{1/\sigma}, 1 \right\}, \quad (160)$$

$$c_3 = \sigma(3 + \alpha), \quad (161)$$

$$c_4 = \frac{\sigma}{2} \left(\frac{\pi}{2\Lambda} \right)^{1/\sigma}. \quad (162)$$

□

Proof of Theorem 8. By Theorem 7 and Theorem 19, we have

$$\| |(u(T, n/N))_{n \in [N]^d} \rangle - |\vec{u}(T) \rangle \| \leq \frac{2c_1 T}{\|(u(T, n/N))_{n \in [N]^d}\|} (c_2 d)^{c_3 d} d^{\alpha/2} \exp(-c_4 N^{1/\sigma}). \quad (163)$$

The choice of N can be solved by letting the error bound smaller than ϵ . □

D Bounding the Trotter error

In this section we present the detailed proof of Theorem 9. For $s \in [t_0, t_0 + h]$ and $t \in [0, h]$, we consider three operators

$$S(s+t, s) = \mathcal{T} e^{\int_s^{s+t} (-B-C(\tau)) d\tau}, \quad (164)$$

$$\tilde{S}(s+t, s) = e^{(-B-C(t_0+h/2))t}, \quad (165)$$

and

$$S_2(s+t, s) = e^{-Bt/2} e^{-C(t_0+h/2)t} e^{-Bt/2}. \quad (166)$$

The proof follows the steps that we first bound $\|S - I\|$ and $\|\tilde{S} - I\|$, then separately bound $\|S - \tilde{S}\|$ and $\|\tilde{S} - S_2\|$ using variation of parameters formula.

D.1 Local error due to the time dependency

First, $S(s+t, s)$ satisfies the differential equation

$$\frac{d}{dt} S(s+t, s) = (-B - C(s+t)) S(s+t, s). \quad (167)$$

Integrate this differential equation and get

$$S(s+t, s) - I = \int_s^{s+t} (-B - C(s+\tau)) S(s+\tau, s) d\tau, \quad (168)$$

and thus we may write

$$S(s+t, s) = I + t R_1(s+t, s) \quad (169)$$

where

$$\|R_1(s+t, s)\| \leq \left(\|B\| + \max_{\tau} \|C(\tau)\| \right). \quad (170)$$

Similarly,

$$\tilde{S}(s+t, s) = I + t R_2(s+t, s) \quad (171)$$

where

$$\|R_2(s+t, s)\| \leq \left(\|B\| + \max_{\tau} \|C(\tau)\| \right). \quad (172)$$

Now we bound the distance between S and $\tilde{S}$. Notice that $\tilde{S}$ satisfies the differential equation

$$\begin{aligned} \frac{d}{dt} \tilde{S}(t_0+t, t_0) &= (-B - C(t_0+h/2)) \tilde{S}(t_0+t, t_0) \\ &= (-B - C(t_0+t)) \tilde{S}(t_0+t, t_0) + (C(t_0+t) - C(t_0+h/2)) \tilde{S}(t_0+t, t_0). \end{aligned} \quad (173)$$

By Taylor expansion, we can write

$$C(t_0+t) = C(t_0+h/2) + (t-h/2)C'(t_0+h/2) + R_3(t), \quad (174)$$

where

$$\|R_3(t)\| \leq \frac{(t-h/2)^2}{2} \max_{\tau} \|C''(\tau)\|. \quad (175)$$

Therefore the differential equation becomes

$$\frac{d}{dt}\tilde{S}(t_0+t, t_0) = (-B - C(t_0+t))\tilde{S}(t_0+t, t_0) + ((t-h/2)C'(t_0+h/2) + R_3(t))\tilde{S}(t_0+t, t_0). \quad (176)$$

Comparing this with Equation (167) and using Theorem 20, we obtain

$$\tilde{S}(t_0+h, t_0) = S(t_0+h, t_0) + \int_0^h S(t_0+h, t_0+t)((t-h/2)C'(t_0+h/2) + R_3(t))\tilde{S}(t_0+t, t_0)dt. \quad (177)$$

Plugging in Equation (169) and Equation (171) yields

$$\begin{aligned} \tilde{S}(t_0+h, t_0) &= S(t_0+h, t_0) + \int_0^h (I + (h-t)R_1(t_0+h, t_0+t))(t-h/2)C'(t_0+h/2)(I + tR_2(t_0+t, t_0))dt \\ &\quad + \int_0^h S(t_0+h, t_0+t)R_3(t)\tilde{S}(t_0+t, t_0)dt \\ &= S(t_0+h, t_0) + \int_0^h (t-h/2)C'(t_0+h/2)dt \\ &\quad + \int_0^h (h-t)(t-h/2)R_1(t_0+h, t_0+t)C'(t_0+h/2)\tilde{S}(t_0+t, t_0)dt \\ &\quad + \int_0^h t(t-h/2)C'(t_0+h/2)R_2(t_0+t, t_0)dt \\ &\quad + \int_0^h S(t_0+h, t_0+t)R_3(t)\tilde{S}(t_0+t, t_0)dt \\ &= S(t_0+h, t_0) + \int_0^h (h-t)(t-h/2)R_1(t_0+h, t_0+t)C'(t_0+h/2)\tilde{S}(t_0+t, t_0)dt \\ &\quad + \int_0^h t(t-h/2)C'(t_0+h/2)R_2(t_0+t, t_0)dt \\ &\quad + \int_0^h S(t_0+h, t_0+t)R_3(t)\tilde{S}(t_0+t, t_0)dt. \end{aligned} \quad (178)$$

Therefore

$$\begin{aligned} &\|\tilde{S}(t_0+h, t_0) - S(t_0+h, t_0)\| \\ &\leq \int_0^h t|t-h/2|dt (\max(\|R_1\|\|C'\|) + \max(\|C'\|\|R_2\|)) + \int_0^h \|R_3(t)\|dt \\ &\leq \int_0^h t|t-h/2|dt (\max(\|R_1\|\|C'\|) + \max(\|C'\|\|R_2\|)) + \int_0^h \frac{(t-h/2)^2}{2}dt \max\|C''\| \\ &= h^3 \left(\frac{1}{4}(\|B\| + \max\|C\|) \max\|C'\| + \frac{1}{24} \max\|C''\| \right). \end{aligned} \quad (179)$$

D.2 Local time-independent Trotter error

Next, we bound the distance between $\tilde{S}$ and S_2 . For notation simplicity, throughout this subsection we let $A_1 = -B$ and $A_2 = -C(t_0+h/2)$. Differentiating $\tilde{S}(t_0+t, t_0)$ and $e^{A_1(t-s)}$ yields

$$\frac{d}{dt}\tilde{S}(t_0+t, t_0) = A_1\tilde{S}(t_0+t, t_0) + A_2\tilde{S}(t_0+t, t_0), \quad (180)$$

and

$$\frac{d}{dt}(e^{A_1(t-s)}) = A_1 e^{A_1(t-s)}. \quad (181)$$

By Theorem 20, for any $t \in [0, h]$, we have

$$\tilde{S}(t_0+t, t_0) = e^{A_1 t} + \int_0^t e^{A_1(t-s)} A_2 \tilde{S}(t_0+s, t_0) ds. \quad (182)$$

We use this equation iteratively for three times and get

$$\begin{aligned}
\tilde{S}(t_0 + h, t_0) &= e^{A_1 h} + \int_0^h e^{A_1(h-t)} A_2 \tilde{S}(t_0 + t, t_0) dt \\
&= e^{A_1 h} + \int_0^h e^{A_1(h-t)} A_2 e^{A_1 t} dt + \int_0^h \int_0^t e^{A_1(h-t)} A_2 e^{A_1(t-s)} A_2 \tilde{S}(t_0 + s, t_0) ds dt \\
&= e^{A_1 h} + \int_0^h e^{A_1(h-t)} A_2 e^{A_1 t} dt + \int_0^h \int_0^t e^{A_1(h-t)} A_2 e^{A_1(t-s)} A_2 e^{A_1 s} ds dt + R_4(h),
\end{aligned} \tag{183}$$

where

$$R_4(h) = \int_0^h \int_0^t \int_0^s e^{A_1(h-t)} A_2 e^{A_1(t-s)} A_2 e^{A_1(s-\tau)} A_2 \tilde{S}(t_0 + \tau, t_0) d\tau ds dt. \tag{184}$$

Meanwhile, using the Taylor expansion of $e^{A_2 h}$, we may obtain

$$\begin{aligned}
S_2(t_0 + h, t_0) &= e^{A_1 h/2} e^{A_2 h} e^{A_1 h/2} \\
&= e^{A_1 h} + h e^{A_1 h/2} A_2 e^{A_1 h/2} + \frac{h^2}{2} e^{A_1 h/2} A_2^2 e^{A_1 h/2} + R_5(h),
\end{aligned} \tag{185}$$

where

$$R_5(h) = e^{A_1 h/2} \left(\int_0^h \frac{(h-t)^2}{2} A_2^3 e^{A_2 t} dt \right) e^{A_1 h/2}. \tag{186}$$

Therefore,

$$\begin{aligned}
&\tilde{S}(t_0 + h, t_0) - S_2(t_0 + h, t_0) \\
&= \left(\int_0^h e^{A_1(h-t)} A_2 e^{A_1 t} dt - h e^{A_1 h/2} A_2 e^{A_1 h/2} \right) \\
&\quad + \left(\int_0^h \int_0^t e^{A_1(h-t)} A_2 e^{A_1(t-s)} A_2 e^{A_1 s} ds dt - \frac{h^2}{2} e^{A_1 h/2} A_2^2 e^{A_1 h/2} \right) \\
&\quad + R_4(h) - R_5(h).
\end{aligned} \tag{187}$$

It remains to bound each term in the right hand side of [Equation \(187\)](#). Let $f(t) = e^{A_1(h-t)} A_2 e^{A_1 t}$, then

$$\begin{aligned}
&\int_0^h e^{A_1(h-t)} A_2 e^{A_1 t} dt - h e^{A_1 h/2} A_2 e^{A_1 h/2} \\
&= \int_0^h f(t) dt - h f(h/2) \\
&= \int_0^h \left(f(h/2) + (t - h/2) f'(h/2) + \int_0^t (t - s) f''(s) ds \right) dt - h f(h/2) \\
&= \int_0^h \int_0^t (t - s) f''(s) ds dt.
\end{aligned} \tag{188}$$

Notice that $f''(t) = e^{A_1(h-t)} [[A_2, A_1], A_1] e^{A_1 t}$, so we bound

$$\left\| \tilde{S}(t_0 + h, t_0) - S_2(t_0 + h, t_0) \right\| \leq \int_0^h \int_0^t |t - s| ds dt \max_{\tau \in [0, h]} \|f''(\tau)\| \leq \frac{1}{6} h^3 \|[[A_2, A_1], A_1]\|. \tag{189}$$

Similarly, let $g(t, s) = e^{A_1(h-t)} A_2 e^{A_1(t-s)} A_2 e^{A_1 s}$, then

$$\begin{aligned}
& \int_0^h \int_0^t e^{A_1(h-t)} A_2 e^{A_1(t-s)} A_2 e^{A_1 s} ds dt - \frac{h^2}{2} e^{A_1 h/2} A_2^2 e^{A_1 h/2} \\
&= \int_0^h \int_0^t g(t, s) ds dt - \frac{h^2}{2} g(h/2, h/2) \\
&= \int_0^h \int_0^t \left(g(h/2, h/2) + \int_0^1 \nabla g(h/2 + \alpha(t-h/2), h/2 + \alpha(s-h/2)) \cdot (t-h/2, s-h/2) d\alpha \right) ds dt \\
&\quad - \frac{h^2}{2} g(h/2, h/2) \\
&= \int_0^h \int_0^t \int_0^1 \nabla g(h/2 + \alpha(t-h/2), h/2 + \alpha(s-h/2)) \cdot (t-h/2, s-h/2) d\alpha ds dt.
\end{aligned} \tag{190}$$

For $0 \leq \alpha \leq 1$, $0 \leq s \leq t$ and $0 \leq t \leq h$, we always have $0 \leq h/2 + \alpha(s-h/2) \leq h/2 + \alpha(t-h/2)$ and $0 \leq h/2 + \alpha(t-h/2) \leq h$, so

$$\begin{aligned}
& \left\| \int_0^h \int_0^t e^{A_1(h-t)} A_2 e^{A_1(t-s)} A_2 e^{A_1 s} ds dt - \frac{h^2}{2} e^{A_1 h/2} A_2^2 e^{A_1 h/2} \right\| \\
&\leq \int_0^h \int_0^t \int_0^1 (|t-h/2| + |s-h/2|) d\alpha ds dt \left(\max \left\{ \max_{0 \leq s' \leq t' \leq h} \left\| \frac{\partial g(t', s')}{\partial t} \right\|, \max_{0 \leq s' \leq t' \leq h} \left\| \frac{\partial g(t', s')}{\partial s} \right\| \right\} \right) \\
&= \frac{1}{4} h^3 \left(\max \left\{ \max_{0 \leq s' \leq t' \leq h} \left\| \frac{\partial g(t', s')}{\partial t} \right\|, \max_{0 \leq s' \leq t' \leq h} \left\| \frac{\partial g(t', s')}{\partial s} \right\| \right\} \right).
\end{aligned} \tag{191}$$

We may compute that

$$\frac{\partial g}{\partial t} = e^{A_1(h-t)} [A_2, A_1] e^{A_1(t-s)} A_2 e^{A_1 s}, \tag{192}$$

and

$$\frac{\partial g}{\partial s} = e^{A_1(h-t)} A_2 e^{A_1(t-s)} [A_2, A_1] e^{A_1 s}, \tag{193}$$

so for $0 \leq s \leq t \leq h$, both $\|\partial g/\partial t\|$ and $\|\partial g/\partial s\|$ are bounded by $\|[A_2, A_1]\| \|A_2\|$. Then

$$\left\| \int_0^h \int_0^t e^{A_1(h-t)} A_2 e^{A_1(t-s)} A_2 e^{A_1 s} ds dt - \frac{h^2}{2} e^{A_1 h/2} A_2^2 e^{A_1 h/2} \right\| \leq \frac{1}{4} h^3 \|[A_2, A_1]\| \|A_2\|. \tag{194}$$

The bounds for R_4 and R_5 are straightforward from [Equation \(184\)](#) and [Equation \(186\)](#) that

$$\|R_4(h)\| \leq \frac{1}{6} h^3 \|A_2\|^3 \tag{195}$$

and

$$\|R_5(h)\| \leq \frac{1}{6} h^3 \|A_2\|^3. \tag{196}$$

Plug [Equations \(189\)](#) and [\(194\)](#) to [\(196\)](#) back to [Equation \(187\)](#), and we get

$$\begin{aligned}
\left\| \tilde{S}(t_0 + h, t_0) - S_2(t_0 + h, t_0) \right\| &\leq h^3 \left(\frac{1}{6} \|[A_1, [A_1, A_2]]\| + \frac{1}{4} \|[A_1, A_2]\| \|A_2\| + \frac{1}{3} \|A_2\|^3 \right) \\
&\leq h^3 \left(\frac{1}{6} \max \|[B, [B, C]]\| + \frac{1}{4} \max \|[B, C]\| \max \|C\| + \frac{1}{3} \max \|C\|^3 \right).
\end{aligned} \tag{197}$$

D.3 Global error

Combining Equation (179) and Equation (197), we obtain the local error bound as

$$\begin{aligned} & \|S(t_0 + h, t_0) - S_2(t_0 + h, t_0)\| \\ & \leq h^3 \left(\frac{1}{24} \max \|C''\| + \frac{1}{4} (\|B\| + \max \|C\|) \max \|C'\| \right. \\ & \quad \left. + \frac{1}{6} \max \|[B, [B, C]]\| + \frac{1}{4} \max \|[B, C]\| \max \|C\| + \frac{1}{3} \max \|C\|^3 \right). \end{aligned} \quad (198)$$

Finally we bound the global error. Notice that, for any $t \geq 0$, $\|S_2(t + s, t)\| \leq 1$ and $\|S(s + t, t)\| \leq 1$, then the global error accumulates linearly as

$$\begin{aligned} & \|S(T, 0) - \prod_{j=0}^{r-1} S_2((j+1)h, jh)\| \\ & = \left\| \prod_{j=0}^{r-1} S((j+1)h, jh) - \prod_{j=0}^{r-1} S_2((j+1)h, jh) \right\| \\ & \leq \sum_{k=0}^{r-1} \left\| \prod_{j=k+1}^{r-1} S_2((j+1)h, jh) (S((k+1)h, kh) - S_2((k+1)h, kh)) \prod_{j=0}^{k-1} S((j+1)h, jh) \right\| \\ & \leq \sum_{k=0}^{r-1} \|S((k+1)h, kh) - S_2((k+1)h, kh)\| \\ & \leq Th^2 \left(\frac{1}{24} \max \|C''\| + \frac{1}{4} (\|B\| + \max \|C\|) \max \|C'\| \right. \\ & \quad \left. + \frac{1}{6} \max \|[B, [B, C]]\| + \frac{1}{4} \max \|[B, C]\| \max \|C\| + \frac{1}{3} \max \|C\|^3 \right). \end{aligned} \quad (199)$$

E Bounding the commutators

Proof of Theorem 10. We shall consider a fixed t and omit the explicit time dependence in our notation. We first consider $[B, C]$ and write

$$\begin{aligned} \|[B, C]\| &= \|[(\mathcal{F})^{\otimes d} D (\mathcal{F}^{-1})^{\otimes d}, C]\| \\ &= \|(\mathcal{F})^{\otimes d} [D, (\mathcal{F}^{-1})^{\otimes d} C (\mathcal{F})^{\otimes d}] (\mathcal{F}^{-1})^{\otimes d}\| \\ &= \|[D, (\mathcal{F}^{-1})^{\otimes d} C (\mathcal{F})^{\otimes d}]\|. \end{aligned} \quad (200)$$

By the definition of the matrices, we have

$$C(\mathcal{F})^{\otimes d} = \frac{1}{N^{d/2}} \sum_{j, k \in [N]^d} c(j/N) \omega_N^{j \cdot k} |j_0 \cdots j_{d-1}\rangle \langle k_0 \cdots k_{d-1}| \quad (201)$$

and

$$\begin{aligned}
(\mathcal{F}^{-1})^{\otimes d} C(\mathcal{F})^{\otimes d} &= \frac{1}{N^d} \sum_{j,k \in [N]^d} \left(\sum_{l \in [N]^d} \omega_N^{-j \cdot l} c(l/N) \omega_N^{l \cdot k} \right) |j_0 \cdots j_{d-1}\rangle \langle k_0 \cdots k_{d-1}| \\
&= \frac{1}{N^d} \sum_{j,k \in [N]^d} \left(\sum_{l \in [N]^d} \left(\sum_{m \in \mathbb{Z}^d} \hat{c}_m \omega_N^{l \cdot m} \right) \omega_N^{l \cdot (k-j)} \right) |j_0 \cdots j_{d-1}\rangle \langle k_0 \cdots k_{d-1}| \\
&= \frac{1}{N^d} \sum_{j,k \in [N]^d} \left(\sum_{m \in \mathbb{Z}^d} \hat{c}_m \left(\sum_{l \in [N]^d} \omega_N^{l \cdot (m-(j-k))} \right) \right) |j_0 \cdots j_{d-1}\rangle \langle k_0 \cdots k_{d-1}| \\
&= \sum_{j,k \in [N]^d} \left(\sum_{m \in \mathbb{Z}^d} \hat{c}_{j-k+Nm} \right) |j_0 \cdots j_{d-1}\rangle \langle k_0 \cdots k_{d-1}|.
\end{aligned} \tag{202}$$

Let $\hat{b}_l = \sum_{m \in \mathbb{Z}^d} \hat{c}_{l+Nm}$, and we can compute that

$$[D, (\mathcal{F}^{-1})^{\otimes d} C(\mathcal{F})^{\otimes d}] = \sum_{j,k \in [N]^d} (\|i(j)\|^\alpha - \|i(k)\|^\alpha) \hat{b}_{j-k} |j_0 \cdots j_{d-1}\rangle \langle k_0 \cdots k_{d-1}|. \tag{203}$$

Notice the decomposition

$$\|i(j)\|^\alpha - \|i(k)\|^\alpha = (\|i(j)\|^{\alpha/2} - \|i(k)\|^{\alpha/2})^2 + 2(\|i(j)\|^{\alpha/2} - \|i(k)\|^{\alpha/2})\|i(k)\|^{\alpha/2}. \tag{204}$$

We may write

$$[D, (\mathcal{F}^{-1})^{\otimes d} C(\mathcal{F})^{\otimes d}] = A_1 + 2A_2A_3 \tag{205}$$

where

$$A_1 = \sum_{j,k \in [N]^d} \left(\|i(j)\|^{\alpha/2} - \|i(k)\|^{\alpha/2} \right)^2 \hat{b}_{j-k} |j_0 \cdots j_{d-1}\rangle \langle k_0 \cdots k_{d-1}|, \tag{206}$$

$$A_2 = \sum_{j,k \in [N]^d} \left(\|i(j)\|^{\alpha/2} - \|i(k)\|^{\alpha/2} \right) \hat{b}_{j-k} |j_0 \cdots j_{d-1}\rangle \langle k_0 \cdots k_{d-1}| \tag{207}$$

and

$$A_3 = \sum_{k \in [N]^d} \|i(k)\|^{\alpha/2} |k_0 \cdots k_{d-1}\rangle \langle k_0 \cdots k_{d-1}|. \tag{208}$$

Now we bound the norms of A_j 's. The norm of A_3 is clearly

$$\|A_3\| = \mathcal{O}(d^{\alpha/4} N^{\alpha/2}). \tag{209}$$

For A_2 , let $q = \operatorname{argmax} ||i(j_{q'})| - |i(k_{q'})||$. We use the inequality

$$\begin{aligned}
\left| \|i(j)\|^{\alpha/2} - \|i(k)\|^{\alpha/2} \right| &\leq \| |i(j)| - |i(k)| \| \|^{\alpha/2} \\
&\leq \| |i(j)| - |i(k)| \| \|^{\alpha/2} \\
&\leq d^{\alpha/4} \| |i(j_q)| - |i(k_q)| \| \|^{\alpha/2}.
\end{aligned} \tag{210}$$

Here the second line is the triangle inequality (and $|v|$ for a vector v denotes its entrywise absolute), the third line is due to the definition of q , and the first line can be proved by the following arguments: let $b \geq a \geq 0$ be two real number and $f(x) = x^{\alpha/2}$, then $b^{\alpha/2} - a^{\alpha/2} = \int_a^b f'(x) dx \leq \int_0^{b-a} f'(x) dx = (b-a)^{\alpha/2}$ because $f'(x) = \frac{\alpha}{2} x^{\alpha/2-1}$ is monotonically decreasing for $0 < \alpha \leq 2$. Notice that the Fourier coefficient $|\hat{c}_l| = \mathcal{O}(|l_{q'}|^{-(2+d)})$ for any index q' , and when $|i(j_q) - i(k_q)| \leq N/2$ (c.f., $i(j_q) - i(k_q) > N/2$ or $i(j_q) - i(k_q) < -N/2$), $\hat{c}_{i(j)-i(k)}$ (c.f., $\hat{c}_{i(j)-i(k)-N}$ or $\hat{c}_{i(j)-i(k)+N}$) is the

term with lowest frequency in $\hat{b}_{j-k}$, We have

$$\begin{aligned} \sum_{k \in [N]^d} \left| \|i(j)\|^{\alpha/2} - \|i(k)\|^{\alpha/2} \right| |\hat{b}_{j-k}| &\leq d^{\alpha/4} \sum_{k \in [N]^d} \|i(j_q) - i(k_q)\|^{\alpha/2} |\hat{b}_{j-k}| \\ &\leq d^{\alpha/4} \sum_{k \in [N]^d, m \in [N]^d} \|j - k + Nm\|_{\infty}^{\alpha/2} |\hat{c}_{j-k+Nm}| \\ &= \mathcal{O}(d^{\alpha/4}). \end{aligned} \quad (211)$$

Similarly, the summation over j is also bounded by $\mathcal{O}(d^{\alpha/4})$. These imply $\|A_2\|_1 = \mathcal{O}(d^{\alpha/4})$ and $\|A_2\|_{\infty} = \mathcal{O}(d^{\alpha/4})$, which further implies that

$$\|A_2\| = \mathcal{O}(d^{\alpha/4}) \quad (212)$$

since $\|A\|^2 \leq \|A\|_1 \|A\|_{\infty}$ for any matrix A . For A_1 , we use [Equation \(210\)](#) again that

$$\left(\|i(j)\|^{\alpha/2} - \|i(k)\|^{\alpha/2} \right)^2 \leq d^{\alpha/2} \|i(j_q) - i(k_q)\|^{\alpha}, \quad (213)$$

and

$$\|A_1\| = \mathcal{O}(d^{\alpha/2}) \quad (214)$$

by the same argument for A_2 (with the only difference that we need the Fourier coefficient to decay as $|\hat{c}_l| = \mathcal{O}(|l_{q'}|^{-(3+d)})$). Plugging [Equations \(209\), \(212\) and \(214\)](#) back to [Equation \(205\)](#) yields

$$\|[B, C]\| = \mathcal{O}(d^{\alpha/2} N^{\alpha/2}). \quad (215)$$

The second estimate $\|[B, [B, C]]\| = \mathcal{O}(d^{\alpha} N^{\alpha})$ can be proved in the same way by using $|\hat{c}_l| = \mathcal{O}(|l_{q'}|^{-(5+d)})$ and noticing that

$$\|[B, [B, C]]\| = \|[D, [D, (\mathcal{F}^{-1})^{\otimes d} C (\mathcal{F})^{\otimes d}]]\|, \quad (216)$$

$$[D, [D, (\mathcal{F}^{-1})^{\otimes d} C (\mathcal{F})^{\otimes d}]] = \sum_{j, k \in [N]^d} (\|i(j)\|^{\alpha} - \|i(k)\|^{\alpha})^2 \hat{b}_{j-k} |j_0 \cdots j_{d-1}\rangle \langle k_0 \cdots k_{d-1}|, \quad (217)$$

and

$$\begin{aligned} &(\|i(j)\|^{\alpha} - \|i(k)\|^{\alpha})^2 \\ &= (\|i(j)\|^{\alpha/2} - \|i(k)\|^{\alpha/2})^4 + 4(\|i(j)\|^{\alpha/2} - \|i(k)\|^{\alpha/2})^3 \|i(k)\|^{\alpha/2} + 4(\|i(j)\|^{\alpha/2} - \|i(k)\|^{\alpha/2})^2 \|i(k)\|^{\alpha}. \end{aligned} \quad (218)$$

□

F Discretization error in the LCHS-IP method

Proof of [Theorem 17](#). Let $U(t; \xi) = \mathcal{T} e^{-i \int_0^t \xi(B+C(s))ds} = e^{-i \xi B t} \mathcal{T} e^{-i \int_0^t H_I(s; \xi)ds}$ and $V(t; \xi) = \frac{1}{\pi(1+\xi^2)} U(t; \xi)$. The truncation error can be bounded by

$$\left\| \mathcal{T} e^{-\int_0^T (B+C(s))ds} - \int_{-\Xi}^{\Xi} \frac{1}{\pi(1+\xi^2)} U(T; \xi) d\xi \right\| \leq 2 \int_{\Xi}^{\infty} \frac{d\xi}{\pi(1+\xi^2)} = \frac{2}{\pi} \left(\frac{\pi}{2} - \arctan(\Xi) \right) \leq \frac{2}{\pi \Xi}. \quad (219)$$

According to the standard quadrature error bound, we have

$$\left\| \mathcal{T} e^{-\int_0^T (B+C(s))ds} - \sum_{j=0}^{M-1} w_j U(T; \xi_j) \right\| \leq \frac{2\Xi^2}{M} \sup_{\xi \in [-\Xi, \Xi]} \left\| \frac{\partial V}{\partial \xi} \right\|. \quad (220)$$

To estimate $\partial V / \partial \xi$, we first compute $\partial U / \partial \xi$. Differentiating [Equation \(81\)](#) yields

$$\frac{\partial}{\partial t} \frac{\partial U}{\partial \xi} = -i(B + C(t))U - i(\xi B + \xi C(t)) \frac{\partial U}{\partial \xi}, \quad \frac{\partial U}{\partial \xi}(0; \xi) = 0. \quad (221)$$

By the variation of constants formula, we have

$$\frac{\partial U}{\partial \xi}(t; \xi) = \int_0^t \left(\mathcal{T} e^{-i \int_s^t (\xi B + \xi C(\tau)) d\tau} \right) (-i)(B + C(s))U(s; \xi) ds, \quad (222)$$

and thus

$$\left\| \frac{\partial U}{\partial \xi} \right\| \leq T(\|B\| + \max \|C\|). \quad (223)$$

Therefore, by the product rule, we have

$$\begin{aligned} \left\| \frac{\partial V}{\partial \xi} \right\| &\leq \frac{2\pi|\xi|}{\pi^2(1 + \xi^2)} + \frac{1}{\pi(1 + \xi^2)} \left\| \frac{\partial U}{\partial \xi} \right\| \\ &\leq \frac{1}{\pi} (1 + T(\|B\| + \max \|C\|)). \end{aligned} \quad (224)$$

Plugging this back into [Equation \(220\)](#) and [Equation \(219\)](#) gives the desired error bound, and the choices of Ξ and M directly follow from the error bound. $\square$