

Hamiltonian simulation for nonlinear partial differential equation by Schrödingerization

Shoya Sasaki,¹ Katsuhiro Endo,² and Mayu Muramatsu^{3,*}

¹*Department of Science for Open and Environmental Systems, Keio University,
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan*

²*National Institute of Advanced Industrial Science and Technology (AIST),
Research Center for Computational Design of Advanced Functional Materials,
Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan*

³*Department of Mechanical Engineering, Keio University,
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan*

Hamiltonian simulation is a fundamental algorithm in quantum computing that has attracted considerable interest owing to its potential to efficiently solve the governing equations of large-scale classical systems. Exponential speedup through Hamiltonian simulation has been rigorously demonstrated in the case of coupled harmonic oscillators. The question arises as to whether Hamiltonian simulations in other physical systems also accelerate exponentially. Schrödingerization is a technique that transforms the governing equations of classical systems into the Schrödinger equation. However, since the Schrödinger equation is a linear equation, Hamiltonian simulation is often limited to linear equations. The research on Hamiltonian simulation methods for nonlinear governing equations remains relatively limited. In this study, we propose a Hamiltonian simulation method for nonlinear partial differential equations (PDEs). The proposed method is named Carleman linearization + Schrödingerization (CLS), which combines Carleman linearization (CL) and warped phase transformation (WPT). CL is first applied to transform a nonlinear PDE into a linear differential equation. This linearized equation is then mapped to the Schrödinger equation via WPT. The original nonlinear PDE can be solved efficiently by the Hamiltonian simulation of the resulting Schrödinger equation. By applying this method, we transform the original governing equation into the Schrödinger equation. Solving the transformed Schrödinger equation then enables the analysis of the original nonlinear equation. As a specific application, we apply this method to the nonlinear reaction-diffusion equation to demonstrate that Hamiltonian simulations are applicable to nonlinear PDEs.

Keywords: Quantum computing, Hamiltonian simulation, Schrödingerization, Warped phase transformation, Carleman linearization, Partial differential equations

I. INTRODUCTION

Partial differential equations (PDEs) describe a wide range of physical phenomena, such as heat conduction, microstructure evolution in materials, and fluid dynamics. PDE-based analysis plays an important role in analyzing physical phenomena in the real world. One major challenge in PDE-based analysis is the difficulty in solving PDEs for extremely large-scale systems within practical time frames [1, 2]. Quantum computing is a promising approach to overcoming this challenge. It has been increasingly attracting attention owing to its potential to accelerate the large-scale analysis of PDEs [3]. Quantum computing utilizes fundamental principles of quantum mechanics, such as superposition and entanglement, to perform calculations [4]. Compared with classical computing, quantum computing provides significant advantages in the analysis of large-scale PDEs [5, 6]. Various methodologies have been investigated for simulating the time evolution of PDEs by quantum computing. These approaches can be broadly classified into two main categories.

The first category includes methods of solving difference equations obtained by discretizing time-dependent PDEs in the time and space directions using matrix operations. The first category involves the application of quantum algorithms originally developed for linear algebra problems [7, 8], such as the quantum linear systems algorithm (QLSA) [9–12] including the Harrow–Hassidim–Lloyd algorithm [13–17].

The second category includes Hamiltonian simulation methods [18–20]. The task of solving the time evolution of the solution to a Schrödinger equation for a time-independent Hamiltonian is called the Hamiltonian simulation problem [21]. The Hamiltonian simulation problem is rewritten in short as follows: give an initial state and Hamiltonian, and then find the time evolution of the solution. Since the Hamiltonian determines the time evolution of a system, the time evolution of various physical systems can be obtained by Hamiltonian simulation through the design of the Hamiltonian. Babbush et al. [22] rigorously demonstrated an exponential quantum speedup by Hamiltonian simulation to a system of classical harmonic oscillators. Hamiltonian simulation is attracting attention as a method that has the potential to accelerate PDE-based analysis.

An analysis method based on nonlinear PDEs is also important because many real-world phenomena are non-

* muramatsu@mech.keio.ac.jp

linear such as large deformation in materials, turbulence in fluid flow, chaotic systems and reaction–diffusion phenomena. The Hamiltonian simulation of nonlinear PDEs is difficult, and there has not been sufficient discussion or verification yet. There are two key challenges in performing Hamiltonian simulations of nonlinear PDEs.

Firstly, the target equation is a nonlinear PDE, whereas a Schrödinger equation is a linear PDE. This means that the target nonlinear PDEs must be transformed into a linear equation. Some algorithms for linearization that can be realized in quantum computing have been proposed. Joseph [23] considered Koopman von Neumann (KvN) linearization based on the Koopman operator [24] in quantum computing. KvN linearization is a general linearization method with a high degree of freedom in basis functions. Liu and coworkers [25, 26] applied Carleman linearization (CL) [25–32] to linearize the nonlinear reaction–diffusion equation and employed QLSA to compute physical quantities such as energy. CL is a linearization method used when selecting polynomials as basis functions in KvN. Endo and Takahashi [33] proposed an algorithm that mitigates the divergence of solutions caused by CL when implemented on a quantum computer. In this study, we focused on CL as a linearization method because it is a fundamental linearization method and has been extensively studied for its applications in quantum computing.

Secondly, a linearized equation is generally a dissipative system, whereas a Schrödinger equation is a conservative system. In this study, a conservative system is defined as the system with the time evolution operator represented by a unitarity operator. In contrast, a dissipative system is defined as the system with the time evolution operator represented by a non-unitarity operator. In conservative systems, linear PDEs can be easily transformed into a Schrödinger equation. Costa et al. [34] proposed a quantum algorithm for simulating the wave equation under Dirichlet and Neumann boundary conditions, using Hamiltonian simulation as a subroutine. Sato et al. [1] proposed a method of explicitly implementing quantum circuits for Hamiltonian simulation, applied it to linear advection and wave equations, and highlighted its potential for exponential speedup. However, these studies have been limited to conservative systems. Several methods have been proposed for handling dissipative systems on quantum computers. Gonzalez-Conde et al. [35] converted the Black–Scholes equation in a dissipative system into the Schrödinger equation in a conservative system by unitary dilation. Unitary dilation introduces an additional ancilla qubit to the system, allowing the time evolution operator to be unitary. An et al. [36] proposed the linear combination of Hamiltonian simulation (LCHS) as an approach to handling dissipative systems. LCHS can be viewed as a special case of linear combination of unitaries (LCU) [37–39]. Jin and coworkers [40–42] proposed Schrödingerization, which is a method for mapping a general linear ordinary differential equation (ODE) including a dissipative system to a

Schrödinger equation. The core of Schrödingerization is warped phase transformation (WPT), which converts a dissipative system into a conservative system by adding new auxiliary variables to the spatial dimensions of the system. In this study, we focused on WPT because unitary dilation and LCHS can only succeed in Hamiltonian simulation probabilistically, but WPT does not require probabilistic Hamiltonian simulation. However, it is unclear what advantages there are when postselection is included. In Fig. 1 two key challenges to performing Hamiltonian simulations of nonlinear PDEs are summarized and previous studies are classified.

In this study, we propose a method for the Hamiltonian simulation of nonlinear PDEs, named Carleman linearization + Schrödingerization (CLS). The proposed CLS framework integrates the following two components: first, CL is applied to transform the target nonlinear PDE into a system of linear ODEs. Then, the resulting linear system is converted into a Schrödinger equation using WPT. Finally, Hamiltonian simulation is applied to the transformed Schrödinger equation, allowing the time evolution of the original nonlinear PDE to be recovered.

In this study, we apply CLS to a nonlinear reaction–diffusion equation as a representative example of a nonlinear PDE. We evaluate the time evolution of the solution obtained by CLS, assess the associated errors, and verify the computational accuracy, thereby demonstrating the effectiveness of the CLS method.

II. THEORY

A. Nonlinear reaction–diffusion equation

The nonlinear reaction–diffusion equation describes the time evolution of a system in which two processes, reaction and diffusion, proceed simultaneously. These processes include ecology, combustion, phase separation, and tissue formation phenomena. Letting t denote the time, \mathbf{x} represent the spatial coordinates, and $\phi(t, \mathbf{x})$ the field variable, the reaction–diffusion equation is generally given by [43]

$$\frac{\partial \phi(t, \mathbf{x})}{\partial t} = D\nabla^2\phi(t, \mathbf{x}) + f(\phi(t, \mathbf{x})), \quad (1)$$

where $D \in \mathbb{R}_+$ is the diffusion coefficient and $f(\phi(t, \mathbf{x}))$ is at least the C^1 class smooth function. In Eq. (1), the term $D\nabla^2\phi$ is the linear component, and $f(\phi(t, \mathbf{x}))$ corresponds to the nonlinear component. Therefore, the reaction–diffusion equation can be classified as a type of nonlinear PDE. In this study, we assume that $f(\phi(t, \mathbf{x}))$ is expressed as a quadratic function for ϕ , given by

$$f(\phi(t, \mathbf{x})) = Q\phi(t, \mathbf{x}) + R\phi(t, \mathbf{x})^2, \quad (2)$$

where $Q \in \mathbb{R}$ is the coefficient of the first term and $R \in \mathbb{R}$ is the coefficient of the second term. Substituting Eq. (2)

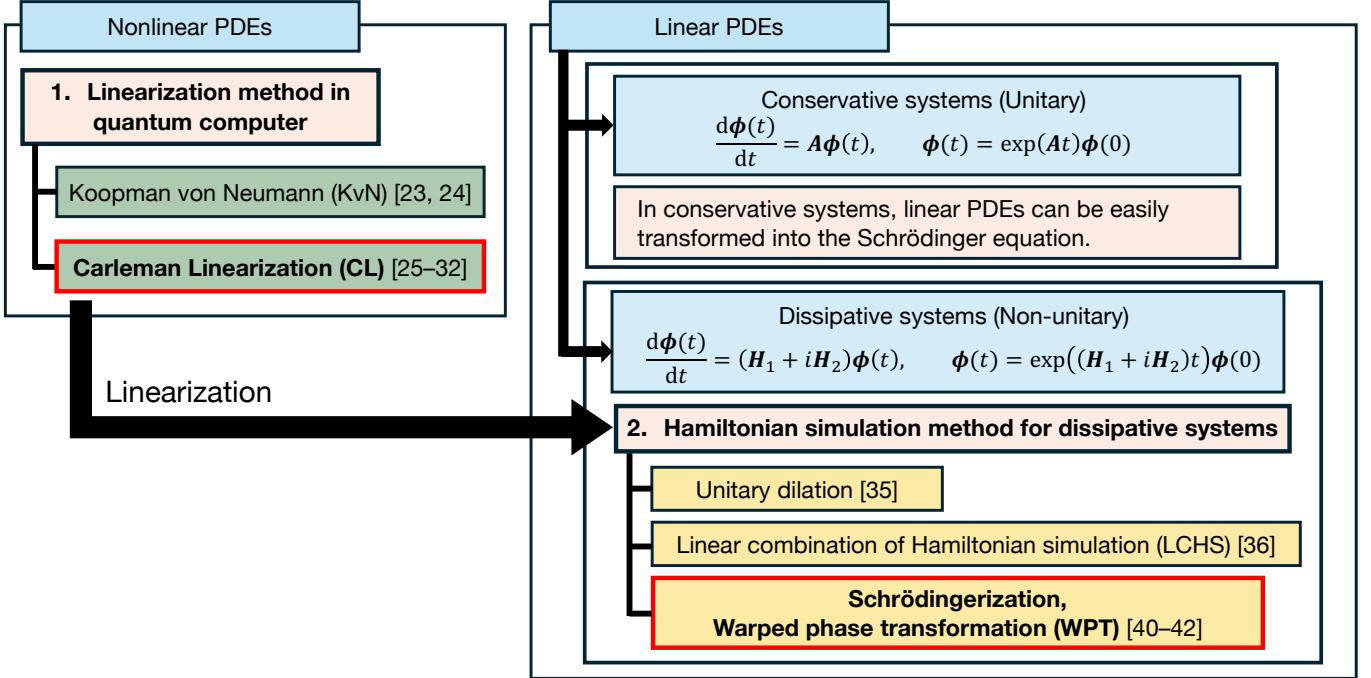


FIG. 1. Two key challenges to performing Hamiltonian simulations of nonlinear PDEs and classification of previous methods for Hamiltonian simulation.

into Eq. (1), we obtain the following equation:

$$\frac{\partial \phi(t, \mathbf{x})}{\partial t} = D\nabla^2\phi(t, \mathbf{x}) + Q\phi(t, \mathbf{x}) + R\phi(t, \mathbf{x})^2. \quad (3)$$

B. CL

CL is a linearization technique that transforms a finite-dimensional nonlinear system into an infinite-dimensional linear system by extending the state variables into an infinite-dimensional space. In this section, we consider the application of CL to a general nonlinear differential equation. Nonlinear differential equations are generally expressed as

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}). \quad (4)$$

Here, $\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ is the state vector of the system and $\mathbf{f}(t, \mathbf{x}) \in \mathbb{R}^n$ is an analytic function of \mathbf{x} , defined as $\mathbf{f} : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$. Approximating $\mathbf{f}(t, \mathbf{x})$ with a polynomial is given by

$$\mathbf{f}(t, \mathbf{x}(t)) = \sum_{m=0}^{\infty} \mathbf{F}_m \mathbf{x}^{\otimes m} = \mathbf{F}_0 + \mathbf{F}_1 \mathbf{x} + \mathbf{F}_2 \mathbf{x}^{\otimes 2} + \dots, \quad (5)$$

where \otimes represents Kronecker's product, $\mathbf{x}^{\otimes m} = \underbrace{\mathbf{x} \otimes \dots \otimes \mathbf{x}}_{m \text{ times}} \in \mathbb{R}^{n^m}$ for any given non-negative integer m , and $\mathbf{F}_m \in \mathbb{R}^{n \times n^m}$ is a coefficient matrix of $\mathbf{x}^{\otimes m}$. For notational convenience, $\mathbf{x}^{\otimes 0} := 1$. The Kronecker product is an operation on two matrices of arbitrary size. The

result of the operation is given as a matrix expanding the set of bases (i.e., $\otimes : (\mathbb{R}^a \times \mathbb{R}^b), (\mathbb{R}^c \times \mathbb{R}^d) \rightarrow (\mathbb{R}^{ac} \times \mathbb{R}^{bd})$). In CL, we consider the time evolution of extended variables $\mathbf{y}_k := \mathbf{x}^{\otimes k}$ for any non-negative integer k . According to Eqs. (4) and (5), and applying the chain rule, we obtain the time derivative of \mathbf{y}_k as

$$\frac{d\mathbf{y}_k}{dt} = \frac{d\mathbf{x}^{\otimes k}}{dt} = \frac{d\mathbf{x}^{\otimes k}}{d\mathbf{x}} \frac{d\mathbf{x}}{dt} = \frac{d\mathbf{x}^{\otimes k}}{d\mathbf{x}} \sum_{m=0}^{\infty} \mathbf{F}_m \mathbf{x}^{\otimes m}. \quad (6)$$

Applying the product rule of differentiation, we calculate $d\mathbf{x}^{\otimes k}/d\mathbf{x}$ as:

$$\frac{d\mathbf{x}^{\otimes k}}{d\mathbf{x}} = \sum_{v=0}^{k-1} \mathbf{x}^{\otimes v} \otimes \mathbf{I} \otimes \mathbf{x}^{\otimes k-1-v}. \quad (7)$$

Substituting Eq. (7) into Eq. (6), we obtain

$$\frac{d\mathbf{y}_k}{dt} = \sum_{m=0}^{\infty} \left(\sum_{v=0}^{k-1} \mathbf{I}^{\otimes v} \otimes \mathbf{F}_m \otimes \mathbf{I}^{\otimes k-1-v} \right) \mathbf{x}^{\otimes m+k-1}. \quad (8)$$

Here, we use the relationship $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$, which holds true when the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and \mathbf{D} are of a size such that the matrix products \mathbf{AC} and \mathbf{BD} can be defined. Here, by letting l be $m + k - 1$, we obtain the time evolution of \mathbf{y}_k by calculating the following infinite-dimensional linear differential equation:

$$\frac{d\mathbf{y}_k}{dt} = \sum_{l=0}^{\infty} \mathbf{A}_{k,l} \mathbf{y}_l, \quad (9)$$

where

$$\mathbf{A}_{k,l} = \sum_{v=0}^{k-1} \mathbf{I}^{\otimes v} \otimes \mathbf{F}_{l-k+1} \otimes \mathbf{I}^{\otimes k-1-v}. \quad (10)$$

Eq. (9) is an infinite-dimensional linear differential equation. Since it is not possible to solve it in an infinite-dimensional space, we truncate Eq. (9) at the order of K and compute it as the following approximate finite-dimensional linear ODE:

$$\frac{d\mathbf{y}_k}{dt} = \sum_{l=0}^K \mathbf{A}_{k,l} \mathbf{y}_l, \quad 0 \leq k \leq K. \quad (11)$$

Note that Eq. (11) is a dissipative system. To perform the Hamiltonian simulation of Eq. (11), a Hamiltonian simulation framework for dissipative systems is required.

C. Schrödingerization using WPT

In this section, we introduce Schrödingerization framework which is a method for mapping a general linear ODE including a dissipative system to a Schrödinger equation. The core of Schrödingerization is WPT. In WPT, a dissipative system can be transformed into a conservative one by introducing an auxiliary variable independent of the spatial dimensions. We consider the application of WPT to a general linear ODE. A linear ODE is generally expressed as

$$\frac{d\mathbf{u}(t)}{dt} = \mathbf{A}\mathbf{u}(t), \quad (12)$$

where $\mathbf{u}(t) \in \mathbb{C}^n$ is the state vector in the system and $\mathbf{A} \in \mathbb{C}^{n \times n}$ is a coefficient matrix. The WPT of dissipative systems into conservative systems is achieved by introducing an auxiliary variable independent of the spatial dimensions. First, since the coefficient matrix \mathbf{A} is a square matrix, it is decomposed into its Hermitian part \mathbf{H}_1 and skew-Hermitian part $i\mathbf{H}_2$ as

$$\mathbf{A} = \mathbf{H}_1 + i\mathbf{H}_2. \quad (13)$$

The Hermitian part \mathbf{H}_1 and the skew-Hermitian part $i\mathbf{H}_2$ are defined as

$$\mathbf{H}_1 = \frac{\mathbf{A} + \mathbf{A}^\dagger}{2}, \quad i\mathbf{H}_2 = \frac{\mathbf{A} - \mathbf{A}^\dagger}{2}. \quad (14)$$

In WPT, we introduce an auxiliary variable, $p \geq 0$. WPT is formulated as

$$\mathbf{v}(t, p) = e^{-p}\mathbf{u}(t). \quad (15)$$

Multiplying both sides of Eq. (12) by e^{-p} yields

$$\frac{d}{dt}(e^{-p}\mathbf{u}(t)) = e^{-p}\mathbf{A}\mathbf{u}(t). \quad (16)$$

Substituting Eq. (13) into Eq. (16) yields

$$\frac{d}{dt}(e^{-p}\mathbf{u}(t)) = e^{-p}(\mathbf{H}_1 + i\mathbf{H}_2)\mathbf{u}(t). \quad (17)$$

Eq. (17) can be transformed as

$$\frac{d}{dt}(e^{-p}\mathbf{u}(t)) = \left(-\mathbf{H}_1 \frac{\partial}{\partial p} + i\mathbf{H}_2\right)e^{-p}\mathbf{u}(t). \quad (18)$$

Substituting Eq. (15) into Eq. (18) yields

$$\frac{d\mathbf{v}(t, p)}{dt} = -\mathbf{H}_1 \frac{\partial\mathbf{v}(t, p)}{\partial p} + i\mathbf{H}_2\mathbf{v}(t, p). \quad (19)$$

The first term on the right-hand side of Eq. (19) captures the advection of $\mathbf{v}(t, p)$. Therefore, Eq. (19) should be discretized in the p -direction by the upwind difference method. The upwind difference method is a discretization technique in which the spatial differential term is approximated using the difference between a reference point and an upstream point. Furthermore, even if the initial value is extended to the region of $p < 0$, the solution $\mathbf{v}(t, p)$ does not impact the region $p \geq 0$, because it flows from right to left in the p -direction. Therefore, we extend Eq. (19) to $p < 0$ with the following initial condition:

$$\mathbf{v}(0, p) = e^{-|p|}\mathbf{u}(0). \quad (20)$$

As a result, the ODE represented by Eq. (12) is transformed into the following system:

$$\begin{cases} \frac{d\mathbf{v}(t, p)}{dt} = \mathbf{A}'\mathbf{v}(t, p), \\ \mathbf{v}(0, p) = e^{-|p|}\mathbf{u}(0), \end{cases} \quad (21)$$

where

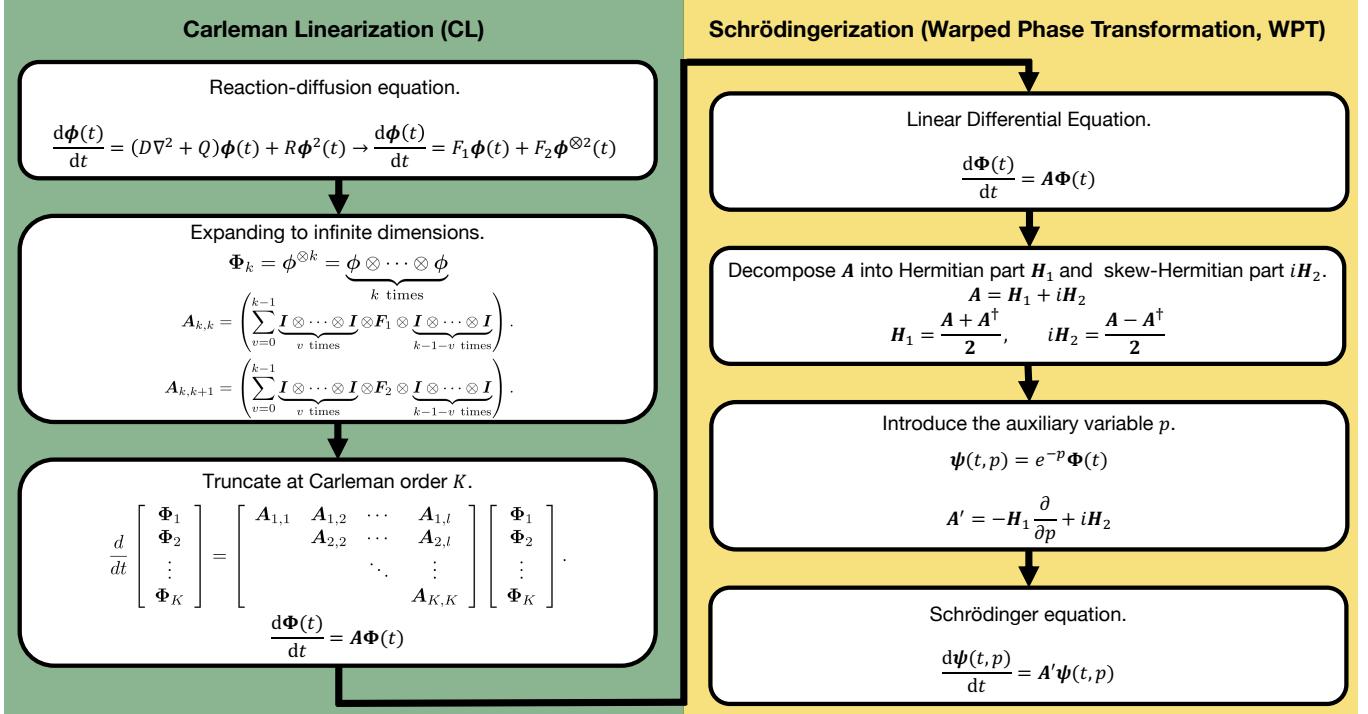
$$\mathbf{A}' := -\mathbf{H}_1 \frac{\partial}{\partial p} + i\mathbf{H}_2. \quad (22)$$

Note that the matrix \mathbf{A}' is a skew-Hermitian matrix and the proof is shown in A. Therefore, the time evolution of Eq. (21) is unitary and Eq. (21) is a Schrödinger equation. The unitarity of the time evolution of Eq. (21) implies that Eq. (12) is suitable for Hamiltonian simulation via WPT.

III. METHOD

A. CLS

In this study, we propose CLS as a Hamiltonian simulation method for nonlinear PDEs. Specifically, CL is used to transform a nonlinear PDE into a linear ODE, and WPT is used to transform a linearized ODE in a dissipative system into a Schrödinger equation. In this study, we apply CLS to nonlinear reaction-diffusion equations



ϕ : Field variables, D : diffusion coefficient, Q : first order coefficients, R : second order coefficients, Φ : Finite dimension state vector, A : Finite dimension coefficient matrix, H_1 : Hermitian part matrix, iH_2 : skew-Hermitian part matrix, p : auxiliary variable, ψ : Warped Phase Transformed Φ , A' : Warped Phase Transformed A .

FIG. 2. Flow of CLS.

and examine its usefulness. First, we consider the discretization of the nonlinear reaction–diffusion equation shown in Eq. (3) in the x -direction. Let $\Omega_x := (0, x_R)$ denote a one-dimensional spatial domain, and $x_R \in \mathbb{R}_+$ is the length of the spatial domain in the x -direction. We discretize the spatial domain Ω_x using n_x grid points uniformly distributed with spacing $\Delta x = x_R/n_x$, where $n_x \in \mathbb{R}_+$ is the number of computational points in the x -direction. Then, the nonlinear diffusion-reaction equation is discretized as

$$\frac{d\phi(t)}{dt} = (D\Delta + Q)\phi(t) + R\phi(t)^2, \quad (23)$$

where $\phi(t) = \phi = [\phi(t, x_0), \phi(t, x_1), \dots, \phi(t, x_{n_x-1})]^T$ is the discretized field variable, x_j for $j = 0, 1, \dots, n_x - 1$ indicates the spatial coordinates of the j -th node of the x -direction, and Δ is the Laplace operator discretized by the second-order central difference method. For notational convenience, we use ϕ_j given by $\phi_j := \phi(t, x_j)$. Given the Dirichlet boundary conditions $\phi_{-1} = \phi_{n_x} = 0$, the Laplace operator Δ discretized by the second-order central difference method is as follows:

$$\Delta = \frac{1}{(\Delta x)^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & \end{bmatrix}. \quad (24)$$

Eq. (23) is also written as

$$\frac{d\phi}{dt} = F_1\phi + F_2\phi^{\otimes 2}, \quad (25)$$

where $F_1 = D\Delta + Q$ and F_2 is a linear mapping of $\phi^{\otimes 2}$ to $R\phi_j^2$ for $j = 0, 1, \dots, n_x - 1$.

Subsequently, consider converting Eq. (25) into a linear ODE using CL. Let $\Phi_k(t) = \phi^{\otimes k}(t)$ and transform it into the following infinite-dimensional linear differential equation:

$$\frac{d\Phi_k(t)}{dt} = A_{k,k}\Phi_k(t) + A_{k,k+1}\Phi_{k+1}(t). \quad (26)$$

where $A_{k,k}$ and $A_{k,k+1}$ are respectively expressed as

$$A_{k,k} = \sum_{v=0}^{k-1} \mathbf{I}^{\otimes v} \otimes F_1 \otimes \mathbf{I}^{\otimes k-1-v}, \quad (27)$$

$$A_{k,k+1} = \sum_{v=0}^{k-1} \mathbf{I}^{\otimes v} \otimes F_2 \otimes \mathbf{I}^{\otimes k-1-v}. \quad (28)$$

Since Eq. (26) is not solved in infinite dimensions, we truncate Eq. (26) at order K to obtain a finite-dimensional approximation. The approximate finite-dimensional linear differential equation truncated Eq. (26) at order K is

$$\frac{d\Phi(t)}{dt} = A\Phi(t), \quad (29)$$

where Φ and \mathbf{A} are respectively defined as

$$\Phi(t) = [\Phi_1(t), \Phi_2(t), \dots, \Phi_K(t)]^T, \quad (30)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & & \\ & \mathbf{A}_{2,2} & \mathbf{A}_{2,3} & \\ & \ddots & \ddots & \\ & & \mathbf{A}_{K-1,K-1} & \mathbf{A}_{K-1,K} \\ & & & \mathbf{A}_{K,K} \end{bmatrix}. \quad (31)$$

Matrix \mathbf{A} is called the Carleman matrix [44] and vector Φ is called the Carleman state vector. The Carleman matrix is always a square matrix. Next, Eq. (29) is transformed into the Schrödinger equation using WPT. The Carleman matrix in Eq. (29) is decomposed into its Hermitian part \mathbf{H}_1 and skew-Hermitian part $i\mathbf{H}_2$. Then, the Carleman matrix that is not a skew-Hermitian matrix is

$$\mathbf{A} = \mathbf{H}_1 + i\mathbf{H}_2. \quad (32)$$

Introduce a new auxiliary variable $p \geq 0$ into the space variable x of the system. As described in Sec. II C, WPT is expressed as

$$\psi(t, p) = e^{-p}\Phi(t). \quad (33)$$

According to Eq. (19), this variable $\psi(t, p)$ satisfies the following equation:

$$\frac{d\psi(t, p)}{dt} = \left(-\mathbf{H}_1 \frac{\partial}{\partial p} + i\mathbf{H}_2\right)\psi(t, p). \quad (34)$$

Let $\Omega_p := (p_L, p_R)$, $p_L < p_R$ denote a one-dimensional domain, and p_L and p_R be the endpoints of the domain in the p -direction. We discretize the spatial domain Ω_p using n_p grid points uniformly distributed with spacing $\Delta p = (p_R - p_L)/n_p$, where $n_p \in \mathbb{R}_+$ is the number of computational points in the p -direction. p_j for $j = 0, 1, \dots, n_p - 1$ indicates the spatial coordinates of the j -th node of the p -direction. We define the following vector:

$$\mathbf{p} := [e^{-p_0}, e^{-p_1}, \dots, e^{-p_{n_p-1}}]^T \in \mathbb{R}^{n_p}. \quad (35)$$

Subsequently, we define $\psi(t)$ as

$$\psi(t) := \mathbf{p} \otimes \Phi(t), \quad (36)$$

where $\psi(t) = [\psi(t, p_0), \psi(t, p_1), \dots, \psi(t, p_{n_p-1})]^T$. For notational convenience, we use $\psi_j(t)$ given by $\psi_j(t) := \psi(t, p_j)$. The variable ψ_j can be expressed as $\psi_j(t) = e^{-p_j}\Phi(t)$ for $j = 0, 1, \dots, n_p - 1$. The first-order upwind difference method is used for the p -direction as the difference scheme in Eq. (34). The upwind difference method is a discretization technique in which the spatial differential term is approximated by the difference between a reference point and an upstream point. Since Eq. (34) advects in the p -negative direction, applying the upwind difference method yields

$$\frac{d\psi_j(t)}{dt} = -\mathbf{H}_1 \frac{\psi_{j+1}(t) - \psi_j(t)}{\Delta p} + i\mathbf{H}_2\psi_j(t). \quad (37)$$

In this study, the time evolution of the solution of the nonlinear reaction-diffusion equation by CLS is obtained by time evolving Eq. (37).

We consider another representation of Eq. (37). Applying a left tensor product with \mathbf{p} to both sides of Eq. (29) yields

$$\frac{d}{dt}(\mathbf{p} \otimes \Phi(t)) = \mathbf{p} \otimes \mathbf{H}_1\Phi(t) + \mathbf{p} \otimes i\mathbf{H}_2\Phi(t). \quad (38)$$

Eq. (38) can be transformed into

$$\frac{d}{dt}(\mathbf{p} \otimes \Phi(t)) = -\nabla_p \mathbf{p} \otimes \mathbf{H}_1\Phi(t) + \mathbf{p} \otimes i\mathbf{H}_2\Phi(t), \quad (39)$$

where ∇_p is the gradient operator in the p -direction discretized by the first-order upwind difference method. Given the periodic boundary condition, considering the advection from right to left in the p domain, the upwind difference method is defined as

$$\nabla_p = \frac{1}{\Delta p} \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ 1 & & & & -1 \end{bmatrix}. \quad (40)$$

Eq. (39) can be transformed into

$$\frac{d}{dt}(\mathbf{p} \otimes \Phi(t)) = (-\nabla_p \otimes \mathbf{H}_1 + \mathbf{I} \otimes i\mathbf{H}_2)(\mathbf{p} \otimes \Phi(t)). \quad (41)$$

Substituting Eq. (36) into Eq. (41) yields

$$\frac{d\psi}{dt} = \tilde{\mathbf{H}}\psi, \quad (42)$$

where

$$\tilde{\mathbf{H}} := -\nabla_p \otimes \mathbf{H}_1 + \mathbf{I} \otimes i\mathbf{H}_2. \quad (43)$$

Eqs. (42) and (43) are other representations of Eq. (37). When we consider the implementation of CLS on quantum circuits, the representations of Eqs. (42) and (43) are more suitable than that of Eq. (37). We extend Eqs. (37) and (42) to $p < 0$ ($p_L < 0, p_R > 0$) with the following initial data:

$$\psi(0) = \mathbf{P} \otimes \Phi(0), \quad (44)$$

where

$$\mathbf{P} = [e^{-|p_0|}, e^{-|p_1|}, \dots, e^{-|p_{n_p-1}|}]^T \in \mathbb{R}^{n_p}. \quad (45)$$

B. Classical numerical methods for CLS

The discretization in the x - and p -directions discussed in Section III A, which was originally in the context of classical computation, can also be applied when implementing the method on a quantum computer. Since

Hamiltonian simulation operates analogously rather than digitally, it does not update the state step-by-step in time during simulations; instead, it can directly generate the dynamics corresponding to the desired time evolution in an analog manner. In other words, when considering the implementation of CLS on a quantum computer, the time step size in classical simulation can effectively be set to zero. Note that, while this eliminates the time step error inherent in classical simulation, it does not render the quantum simulation error-free. Other types of error may arise depending on the quantum algorithm employed, such as time discretization errors in Suzuki–Trotter decompositions or approximation errors in the construction of time-evolution operators via quantum singular value transformation [45].

However, in this study, the primary objective is to investigate the utility and characteristics of CLS through classical simulations. Therefore, for classical simulations, we consider discretization in the time direction. Specifically, we examine the time evolution from the initial time $t = 0$ up to the integration time $t = T$. Letting n_t denote the number of time steps, we can express the time step size Δt as $\Delta t = T/n_t$. By applying the first-order forward difference scheme to the time derivative term in Eq. (37), we obtain the following expression:

$$\frac{\psi_j(n+1) - \psi_j(n)}{\Delta t} = -\mathbf{H}_1 \frac{\psi_{j+1}(n) - \psi_j(n)}{\Delta p} + i\mathbf{H}_2 \psi_j(n). \quad (46)$$

Here, $n = 0, 1, \dots, n_t - 1$ is the number of time steps and $\psi_j(n)$ represents $\psi_j(t)$ at $t = n\Delta t$. Rearranging with respect to $\psi_j(n)$, we obtain the following equation:

$$\begin{aligned} \psi_j(n+1) &= -\mathbf{H}_1 \frac{\Delta t}{\Delta p} \psi_{j+1}(n) \\ &\quad + \left(1 + \mathbf{H}_1 \frac{\Delta t}{\Delta p} + i\mathbf{H}_2 \Delta t\right) \psi_j(n). \end{aligned} \quad (47)$$

By introducing \mathbf{B}_1 and \mathbf{B}_2 , we can rewrite the equation as

$$\psi_j(n+1) = \mathbf{B}_1 \psi_{j+1}(n) + \mathbf{B}_2 \psi_j(n), \quad (48)$$

where

$$\mathbf{B}_1 := -\mathbf{H}_1 \frac{\Delta t}{\Delta p}, \quad \mathbf{B}_2 := 1 + \mathbf{H}_1 \frac{\Delta t}{\Delta p} + i\mathbf{H}_2 \Delta t. \quad (49)$$

Given the boundary condition $\psi_0 = \psi_{n_p}$, the final iterative system is as follows:

$$\psi(n+1) = \mathbf{B}\psi(n), \quad n = 0, 1, \dots, n_t - 1, \quad (50)$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_2 & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{B}_1 \\ & \ddots & \ddots \\ & & \mathbf{B}_2 & \mathbf{B}_1 \\ \mathbf{B}_1 & & & \mathbf{B}_2 \end{bmatrix}. \quad (51)$$

By iteratively computing Eq. (50), we can obtain the solution vector of ψ at the desired time.

IV. RESULTS AND DISCUSSION

The nonlinear reaction–diffusion equation shown in Eq. (1) is analyzed by the proposed method CLS shown in Fig. 2. In this study, to investigate the usefulness of CLS, a discretized and time-evolved version of the finite differential method (FDM) using the central difference method and a time-evolved version of the linearized equation using CL were prepared and compared. The validity of the proposed method was evaluated by determining the accuracy of calculation by CLS. The computational conditions in this study are shown in Table I.

A. Time evolution of the solution by CLS

The time evolution of the solutions of Eq. (3) by FDM, CL and CLS are shown in Figs. 3(a)–(b) and 3(c), respectively. It can be seen that the time evolution of the solution by the proposed method CLS qualitatively coincides with those of the solutions by FDM and CL in Figs. 3(a) and 3(b).

B. Accuracy of CLS calculations

In this section, we discuss the accuracy of CLS calculations. The error of CL with FDM as the true value is shown in Fig. 4, and that of CLS with CL as the true value is shown in Fig. 5. From Figs. 4 and 5, we see that the error due to CLS is larger than that due to CL. Therefore, it can be assumed that the error due to CLS is dominated by that introduced by the WPT process.

Next, we investigated the accuracy of the calculation for the truncated order K of CL. The relative error of CL when FDM is taken as the true value for the truncated order K of CL is shown in Fig. 6. From Fig. 6, we see that the relative error is parallel to the line with slope 1/1 on both logarithmic plots. Therefore, we can consider that CL is first-order-accurate for the truncated order K .

Next, the accuracy of the calculation is investigated for the x -direction. The relative error of CLS when FDM is taken as the true value for the spatial step size in the x -direction, Δx , is shown in Fig. 7. From Fig. 7, we see that the relative error is parallel to the straight line with slope of 2/1 on both logarithmic plots. CLS shows second-order accuracy for the x spatial step size Δx . This result implies that it is based on the discretization using a second-order central difference method for the x -direction.

Next, we examine the accuracy of the calculation for the p -direction. The relative error of CLS when CL is the true value for CLS with respect to the spatial step size in the p -direction Δp is shown in Fig. 8. From Fig. 8, we see that the relative error is parallel to the line with slope 1 on both logarithmic plots. Therefore, we can consider that CLS is first-order-accurate for the p spatial step size

TABLE I. Computational conditions for FDM, CL, and CLS.

Computation domain of x	$x \in (0 \text{ m}, 1 \text{ m})$
Computation domain of p	$p \in [p_L, p_R] = [-20 \text{ m}, 20 \text{ m}]$
Number of calculation points for x	$n_x = 36$
Number of calculation points for p	$n_p = 256$
Discretization method for x	Second-order central difference method
Discretization method for p	First-order upwind difference method
Time step size Δt	$\Delta t = 1.0 \times 10^{-6} \text{ s}$
Number of time steps n_t	$n_t = 0.4 \times 10^6$
Initial distribution $\phi(0, x)$	$\phi(0, x) = 0.5 - 0.5 \cos(2\pi x)$
Boundary condition for x	Dirichlet condition ($\phi_{-1} = \phi_{n_x} = 0$)
Boundary condition for p	Periodic condition ($\psi_0 = \psi_{n_p}$)
CL truncation order K	3
Variables in equation P, Q, R	$P = 1, Q = 1, R = -1$

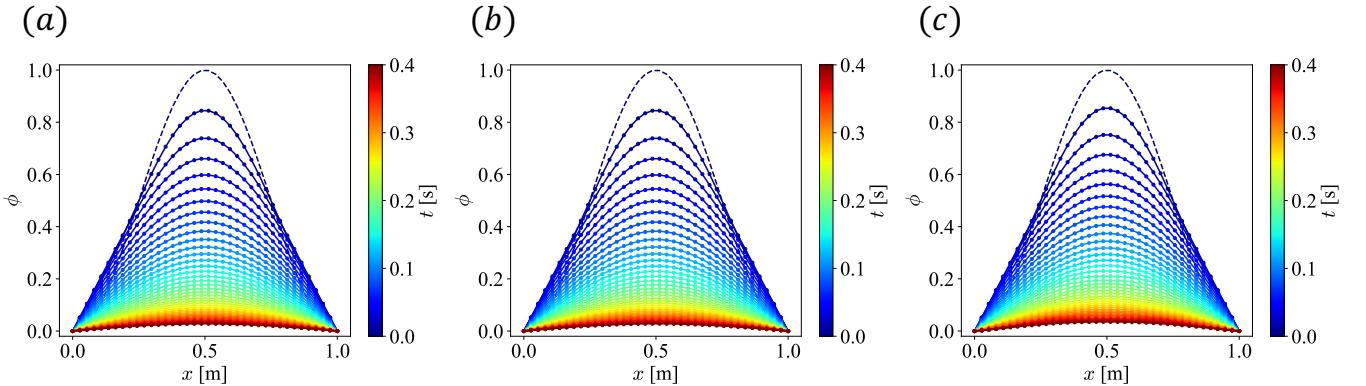


FIG. 3. Time evolution of the solution of the nonlinear reaction–diffusion equation: (a) solution of FDM, (b) solution of CL, and (c) solution of CLS. Dashed lines represent the initial condition.

Δp . This result implies that it is based on the discretization using the first-order upwind difference method for the p -direction.

We consider the error caused by advection in WPT. In WPT, the solution advects in the direction of $p < 0$. Fig. 9 shows a 3D plot of the initial distribution in WPT. Fig. 10 shows the time evolution of the solution obtained by WPT on the xp plane. The red circle surrounds the wave that first affects the calculation accuracy. The brown circle surrounds the wave that next affects the calculation accuracy. Note that periodic boundary conditions are imposed in the p -direction. The red and brown circles have different propagation speeds. Although the computational domain is extended to the $p < 0$ region under the assumption that it does not affect the solution in the $p \geq 0$ region, it is considered that this affects the computational accuracy because it affects the $p \geq 0$ region. For long-term simulations, the errors caused by this advection in WPT can be a problem.

V. CONCLUSION

In this study, the proposed CLS method was applied to the nonlinear reaction–diffusion equation, and the time

evolution of the solution by CLS and its computational accuracy were investigated. The time evolution of the solution by CLS was almost the same as that by the conventional method. The computational accuracy of CLS was found to be first-order accuracy for the truncated order of CL, second-order accuracy for the spatial variable x -direction, and first-order accuracy for the auxiliary variable p -direction. The computational accuracy in the x - and p -directions was considered to be the result of discretization by the second-order accuracy central difference and first-order accuracy upwind difference methods, respectively. This indicated that the computations performed using CLS are consistent with the theoretical predictions. The proposed CLS method extended the framework of time evolution simulation in quantum computing and newly shows that Hamiltonian simulation can be applied to nonlinear PDEs.

ACKNOWLEDGMENTS

This work was supported by JST FOREST Program, Japan (Grant Number JPMJFR212K).

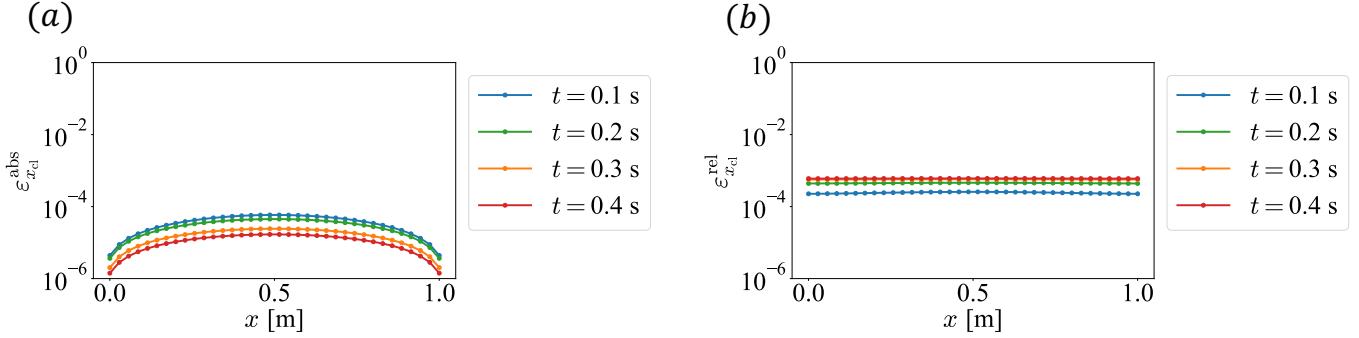


FIG. 4. Error between CL and FDM based on the dynamics of the nonlinear reaction–diffusion equation: (a) absolute error $\varepsilon_{x_{\text{cl}}}^{\text{abs}}$ and (b) relative error $\varepsilon_{x_{\text{cl}}}^{\text{rel}}$.

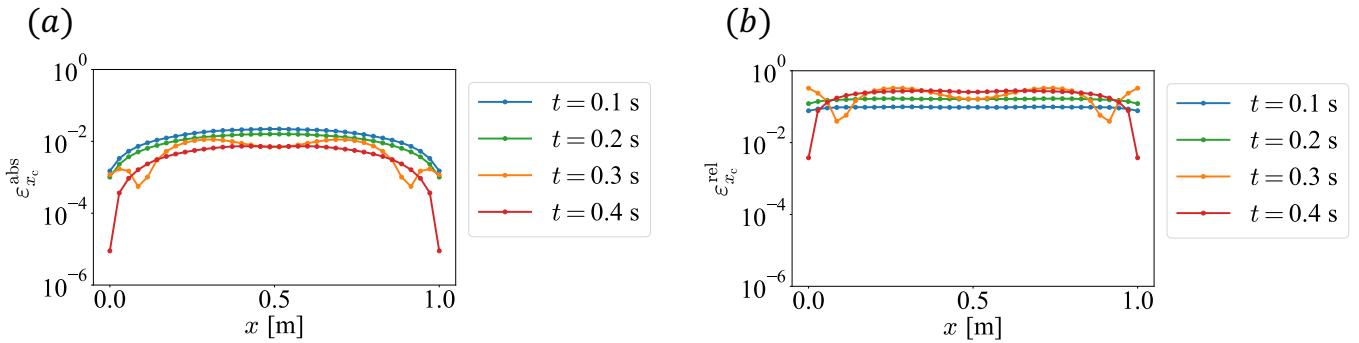


FIG. 5. Error between CL and CLS: (a) absolute error $\varepsilon_{x_c}^{\text{abs}}$ and (b) relative error $\varepsilon_{x_c}^{\text{rel}}$.

-
- [1] Yuki Sato, Ruhō Kondo, Ikko Hamamura, Tamiya Ono-dera, and Naoki Yamamoto. Hamiltonian simulation for hyperbolic partial differential equations by scalable quantum circuits. *Physical Review Research*, 6:033246, 9 2024.
- [2] Yuki Sato, Hiroyuki Tezuka, Ruhō Kondo, and Naoki Yamamoto. Quantum algorithm for partial differential equations of non-conservative systems with spatially varying parameters. *arXiv*, 7 2024.
- [3] Elizabeth Gibney. Hello quantum world! google publishes landmark quantum supremacy claim. *Nature*, 574:461–462, 10 2019.
- [4] Narueethip Sukulthanasan, Junsen Xiao, Koya Wagatsuma, Reika Nomura, Shuji Moriguchi, and Kenjiro Terada. A novel design update framework for topology optimization with quantum annealing: Application to truss and continuum structures. *Computer Methods in Applied Mechanics and Engineering*, 437:117746, 3 2025.
- [5] Burigede Liu, Michael Ortiz, and Fehmi Cirak. Towards quantum computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 432:117403, 12 2023.
- [6] Abhijit Sarma, Thomas W. Watts, Mudassir Moosa, Yilian Liu, and Peter L. McMahon. Quantum variational solving of nonlinear and multidimensional partial differential equations. *Physical Review A*, 109:062616, 6 2024.
- [7] 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, 10 2022.
- [8] Danial Motlagh and Nathan Wiebe. Generalized quantum signal processing. *PRX Quantum*, 5, 4 2024.
- [9] Andrew M. Childs and Jin-Peng Liu. Quantum spectral methods for differential equations. *Communications in Mathematical Physics*, 375:1427–1457, 4 2020.
- [10] 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, 1 2017.
- [11] Dominic W Berry. High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical*, 47:105301, 3 2014.
- [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, 1 2017.
- [13] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103:150502, 10 2009.

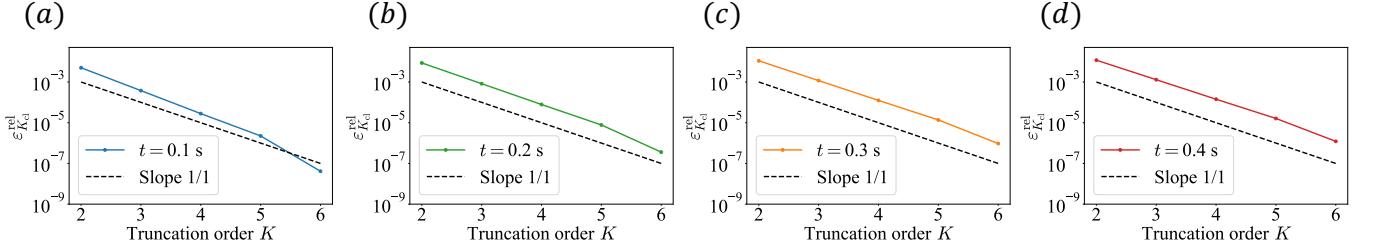


FIG. 6. Relationship between the truncation order of CL and the relative error for the nonlinear reaction–diffusion equation solved using CL: (a) relative error at $t = 0.1 \text{ s}$, (b) relative error at $t = 0.2 \text{ s}$, (c) relative error at $t = 0.3 \text{ s}$, and (d) relative error at $t = 0.4 \text{ s}$.

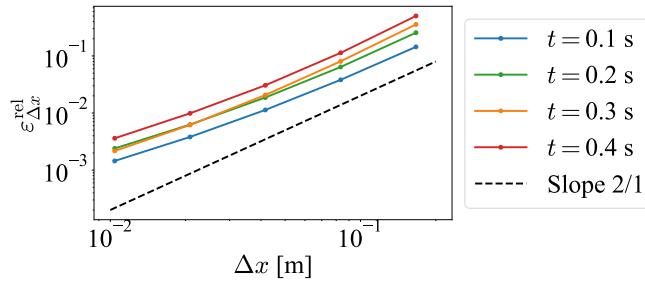


FIG. 7. Computational accuracy of CLS with respect to Δx .

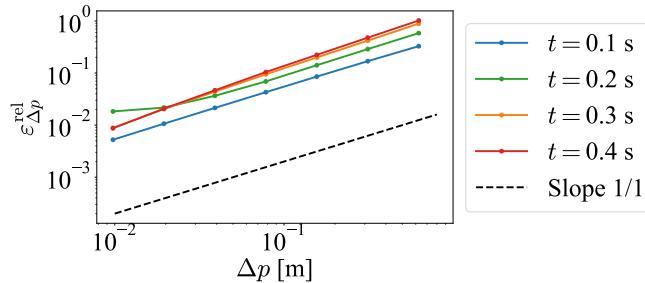


FIG. 8. Computational accuracy of CLS with respect to Δp .

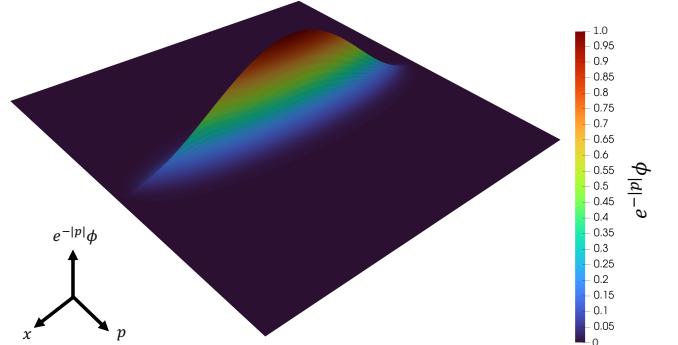


FIG. 9. 3D visualization of initial distribution $\psi(0)$.

- [14] Danial Dervovic, Mark Herbster, Peter Mountney, Simone Severini, Naiři Usher, and Leonard Wossnig. Quantum linear systems algorithms: a primer. *arXiv preprint*, 2 2018.
- [15] 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.
- [16] Linlin Ye, Zhaoqi Wu, and Shao Ming Fei. Coherence dynamics in quantum algorithm for linear systems of equations. *Physica Scripta*, 98, 12 2023.
- [17] Cezary Pilaszwicz and Marian Margraf. Black-box security of stream ciphers under the quantum algorithm for linear systems of equations. *Discover Computing*, 28:31, 4 2025.
- [18] Guang Hao Low and Isaac L. Chuang. Optimal hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118:010501, 1 2017.
- [19] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating

hamiltonian dynamics with a truncated taylor series. *Physical Review Letters*, 114:090502, 3 2015.

- [20] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115:9456–9461, 11 2018.
- [21] Lin Lin. Lecture notes on quantum algorithms for scientific computation. 1 2022.
- [22] Ryan Babbush, Dominic W. Berry, Robin Kothari, Rolando D. Somma, and Nathan Wiebe. Exponential quantum speedup in simulating coupled classical oscillators. *Physical Review X*, 13:041041, 12 2023.
- [23] Ilon Joseph. Koopman-von Neumann approach to quantum simulation of nonlinear classical dynamics. *Physical Review Research*, 2(4), 2020.
- [24] B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17:315–318, 5 1931.
- [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, 8 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, 12 2023.

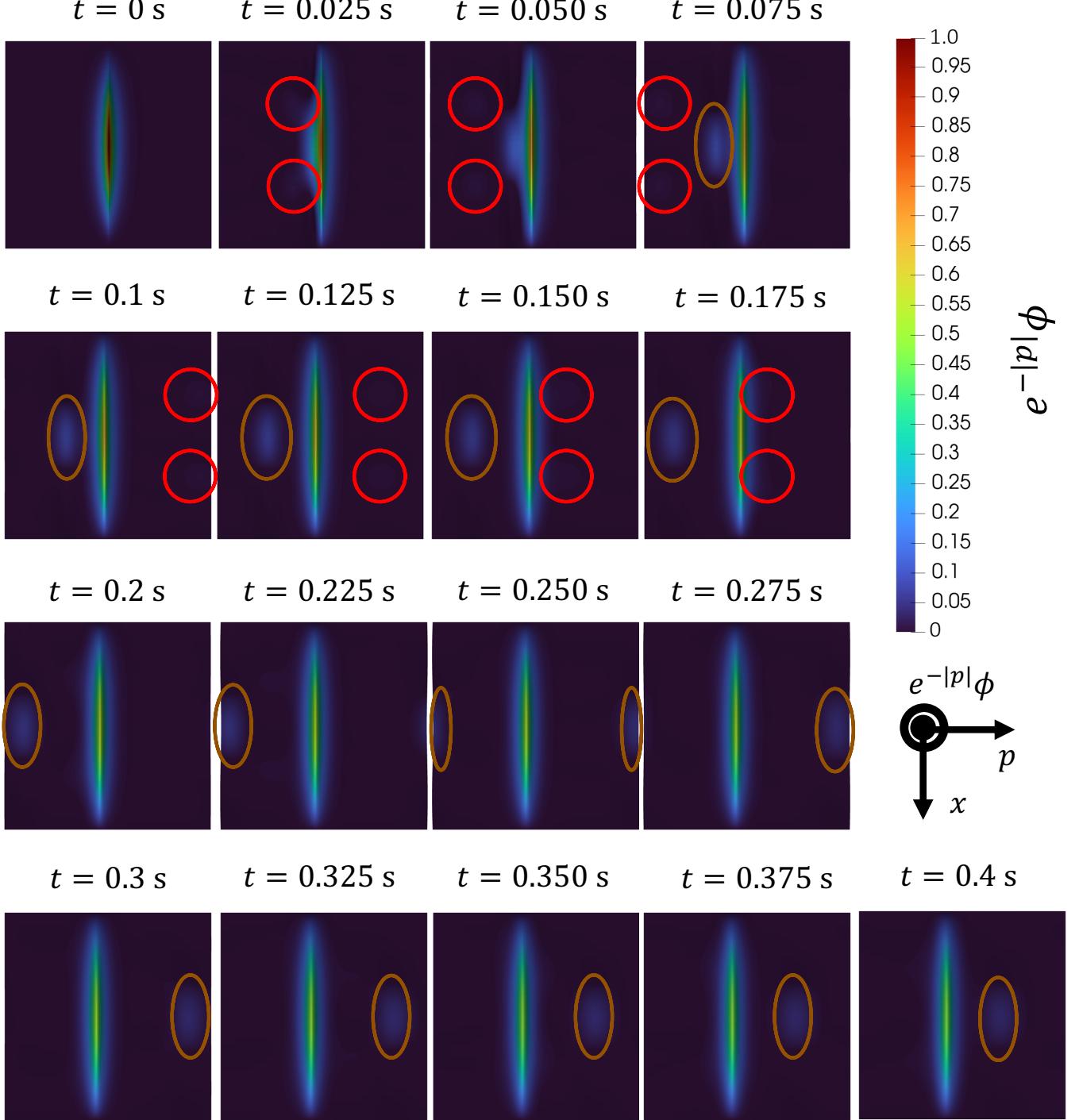


FIG. 10. Time evolutions of the solutions by CLS and conventional method.

- [27] 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.
- [28] Takaki Akiba, Youhi Morii, and Kaoru Maruta. Carleman linearization approach for chemical kinetics integration toward quantum computation. *Scientific Reports*, 13:3935, 3 2023.
- [29] Marcelo Forets and Christian Schilling. *Reachability of Weakly Nonlinear Systems Using Carleman Linearization*, pages 85–99. Springer International Publishing, 8 2021.
- [30] Marcelo Forets and Amaury Pouly. Explicit error bounds for carleman linearization. *arXiv preprint*, 11 2017.

- [31] Claudio Sanavio, Enea Mauri, and Sauro Succi. Carleman-grad approach to the quantum simulation of fluids. *arXiv preprint*, 6 2024.
- [32] Dongwei Shi and Xiu Yang. Koopman spectral linearization vs. carleman linearization: A computational comparison study. *Mathematics*, 12, 7 2024.
- [33] Katsuhiro Endo and Kazuaki Z. Takahashi. Divergence-free algorithms for solving nonlinear differential equations on quantum computers. *arXiv preprint*, 11 2024.
- [34] Pedro C. S. Costa, Stephen Jordan, and Aaron Ostrander. Quantum algorithm for simulating the wave equation. *Physical Review A*, 99:012323, 1 2019.
- [35] Javier Gonzalez-Conde, Ángel Rodríguez-Rozas, Enrique Solano, and Mikel Sanz. Efficient Hamiltonian simulation for solving option price dynamics. *Physical Review Research*, 5(4):043220, 12 2023.
- [36] 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(15):150603, 2023.
- [37] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *arXiv preprint*, 2 2012.
- [38] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Physical Review X*, 11, 2 2021.
- [39] Richard Meister, Simon C. Benjamin, and Earl T. Campbell. Tailoring term truncations for electronic structure calculations using a linear combination of unitaries. *Quantum*, 6:637, 2 2022.
- [40] Shi Jin, Nana Liu, and Yue Yu. Quantum simulation of partial differential equations via schrödingerization. *Physical Review Letters*, 133:230602, 12 2024.
- [41] Shi Jin, Nana Liu, and Yue Yu. Quantum simulation of partial differential equations: Applications and detailed analysis. *Physical Review A*, 108:032603, 9 2023.
- [42] Shi Jin, Xiantao Li, Nana Liu, and Yue Yu. Quantum simulation for partial differential equations with physical boundary or interface conditions. *Journal of Computational Physics*, 498:112707, 2 2024.
- [43] A. De Masi, P. A. Ferrari, and J. L. Lebowitz. Reaction-diffusion equations for interacting particle systems. *Journal of Statistical Physics*, 44:589–644, 8 1986.
- [44] Claudio Sanavio, Enea Mauri, and Sauro Succi. Explicit quantum circuit for simulating the advection-diffusion-reaction dynamics. *IEEE Transactions on Quantum Engineering*, 6:1–12, 2025.
- [45] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2:040203, 12 2021.

Appendix A: Skew-Hermitianity of \mathbf{A}'

The operator \mathbf{A}' obtained from WPT is

$$\mathbf{A}' = -\mathbf{H}_1 \frac{\partial}{\partial p} + i\mathbf{H}_2, \quad (\text{A1})$$

where \mathbf{H}_1 is the Hermitian operator and \mathbf{H}_2 is the skew-Hermitian operator. We define $\mathbb{R}_p = \{p \in \mathbb{R}\}$ and consider the following Hilbert space \mathcal{H} :

$$\begin{aligned} \mathcal{H} &= L^2(\mathbb{R}_p) \\ &= \left\{ f : \mathbb{R} \rightarrow \mathbb{C} \mid \int_{-\infty}^{\infty} |f(p)|^2 dp < \infty \right\}. \end{aligned} \quad (\text{A2})$$

At this time, the inner product is defined as

$$\langle \varphi, \psi \rangle = \int_{-\infty}^{\infty} \varphi^*(p)\psi(p)dp, \quad \varphi, \psi \in \mathcal{H}, \quad (\text{A3})$$

where $\varphi^*(p)$ is the complex conjugate of $\varphi(p)$. When an operator \mathbf{G} satisfies the following equation, \mathbf{G} is called a skew-Hermitian operator:

$$\langle \varphi, \mathbf{G}\psi \rangle = -\langle \mathbf{G}\varphi, \psi \rangle, \quad \varphi, \psi \in \mathcal{H}. \quad (\text{A4})$$

Consider the following inner product to confirm the skew-Hermitian property of the operator $\partial/\partial p$:

$$\begin{aligned} \left\langle \varphi, \frac{\partial \psi}{\partial p} \right\rangle &= \int_{-\infty}^{\infty} \varphi^*(p) \frac{\partial \psi(p)}{\partial p} dp, \\ &= [\varphi^*(p)\psi(p)]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \frac{\partial \varphi^*(p)}{\partial p} \psi(p) dp, \\ &= - \int_{-\infty}^{\infty} \left(\frac{\partial \varphi(p)}{\partial p} \right)^* \psi(p) dp, \\ &= - \left\langle \frac{\partial \varphi}{\partial p}, \psi \right\rangle. \end{aligned} \quad (\text{A5})$$

Therefore, the operator $\partial/\partial p$ is a skew-Hermitian operator. The complex transpose of operator \mathbf{A}' is as follows:

$$\begin{aligned} \mathbf{A}'^\dagger &= \left(-\mathbf{H}_1 \frac{\partial}{\partial p} + i\mathbf{H}_2 \right)^\dagger, \\ &= \left(-\mathbf{H}_1 \frac{\partial}{\partial p} \right)^\dagger + (i\mathbf{H}_2)^\dagger, \\ &= -\left(\frac{\partial}{\partial p} \right)^\dagger \mathbf{H}_1^\dagger - i\mathbf{H}_2, \\ &= \frac{\partial}{\partial p} \mathbf{H}_1 - i\mathbf{H}_2, \\ &= -\left(\mathbf{H}_1 \frac{\partial}{\partial p} + i\mathbf{H}_2 \right), \\ &= -\mathbf{A}'. \end{aligned} \quad (\text{A6})$$

Therefore, \mathbf{A}' is a skew-Hermitian operator.