

Divergence-free algorithms for solving nonlinear differential equations on quantum computers

Katsuhiro Endo^{*} and Kazuaki Z. Takahashi[†]*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*

(Received 20 July 2025; accepted 15 December 2025; published 20 January 2026)

From weather to neural networks, modeling is not only useful for understanding various phenomena but also has a wide range of potential applications. Although nonlinear differential equations are extremely useful tools in modeling, their solutions are difficult to obtain. Based on the expectation of quantum transcendence, quantum algorithms for efficiently solving nonlinear differential equations continue to be developed. However, even the latest promising algorithm has been pointed out to have an evolution time limit. This limit is the theoretically predestined divergence of solutions. We propose algorithms of divergence-free simulation for nonlinear differential equations in quantum computers. For Hamiltonian simulations, a pivot state s is introduced in the neighborhood of state x to be solved. Divergence of the solutions is prevented by moving s to a neighborhood of x whenever x leaves the neighborhood of s . Since updating s is directly related to computational cost, to minimize the number of updates, the nonlinear differential equations are approximated by nonlinear polynomials around s , which are then Carleman linearized. Hamiltonian simulations of nonlinear differential equations based on several representative models are performed to show that the proposed methods break through the theoretical evolution time limit. Furthermore, we demonstrate that the proposed methods work well on an actual quantum computer. The solution of nonlinear differential equations free from evolution time constraints opens the door to practical applications of quantum computers.

DOI: [10.1103/2hfp-qyz1](https://doi.org/10.1103/2hfp-qyz1)

I. INTRODUCTION

Modeling, the attempt to create a well-characterized model to better capture a subset of the world, is an essential and inseparable part of many scientific disciplines, including physics, biology, medicine, engineering, and economics [1]. Good modeling based on deep insights can predict the future or evolve into fundamental technologies: Climate models [2,3] predict current global warming, and neural network models [4,5] support machine learning technologies. Differential equations play a pivotal role in modeling. Nonlinear differential equations can represent in mathematical form even complex phenomena such as collective motion of molecules [6], atmospheric turbulence [7], and the spread of infectious diseases [8], where interactions affect themselves. However, obtaining solutions to nonlinear differential equations is much more difficult than formulating the equations. Generally, it is not easy to obtain an explicit formula for the general solution, and in most cases, numerical solutions are obtained by relying on computers. Moreover, if the behavior of the

solutions changes significantly with slight differences in the initial conditions, long-term prediction is next to impossible [9], no matter how powerful classical computers are used.

Quantum computers are expected to have quantum transcendence, i.e., to perform certain calculations much more efficiently than their classical counterparts [10]. At present, various quantum approaches related to dynamical problems have been explored, including method for extracting spectral information from unitary dynamics [11] and variational circuit-based implementation of nonunitary dynamics [12]. Quantum transcendence has already been theoretically confirmed for unitary linear differential equation systems. Babbush, *et al.* have exponentially accelerated the prediction of oscillator dynamics by Hamiltonian simulations of vast numbers of interconnected harmonic oscillators [13]. Several methods have also been devised to unitarize and compute nonunitary linear differential equation systems. Gonzalez-Conde, *et al.* have solved the Black-Scholes equation by constructing an extended system of nonunitary systems to be unitary [14]. Jin, *et al.* have solved the diffusion equation by adding a dimension to the spatial dimension of the system and transforming it into a unitary time-evolution equation [15,16]. Childs and coworkers have proposed an algorithm for nonunitary time evolution by stochastically applying unitary linear sums to quantum states [17,18], and a modified form of the method has been shown to achieve a certain kind of theoretical optimum [19]. These studies demonstrate that all linear differential equation systems can be Hamiltonian simulated on quantum computers. There have also been many reports

^{*}Contact author: katsuhiro.endo@aist.go.jp[†]Contact author: kazu.takahashi@aist.go.jp

on how to efficiently implement Hamiltonian simulations [20–26] and how to simulate scientific computations [27–31] in quantum computers.

The remarkable developments in linear differential equation systems are since the physics underlying quantum computers is itself fundamentally linear [32]. Approaches to nonlinear differential equation systems are still in the primitive stage of research, because these require methods to mathematically transform nonlinear systems into linear systems. Various algorithm ideas have been proposed, including those based on unrealized technologies and unproven prospects [33]. Here, we focus on algorithms that can be realized with currently available quantum computers. Joseph has proposed an algorithm that uses the Koopman operator to represent a finite-dimensional nonlinear system by an infinite linear system and then truncates part of the operator [34]. This method is arbitrary in the truncation of the operator and in the choice of basis functions that are well matched to the characteristics of the nonlinear system. Even though the choice of basis functions is important not only for the Koopman operator but also for the efficiency of Hamiltonian simulations [35], the protocol is not clearly stated.

Using Carleman linearization, Liu, *et al.* have proposed an algorithm that transforms a nonlinear differential equation into an infinite array of linear differential equations and then truncates part of the array [36]. Although this method is arbitrary in the truncation of the array, the mathematical background of the basis functions is clear. Furthermore, they have quantified the nonlinearity by introducing a parameter that represents the ratio between the tendency of the system toward chaos and the “friction” that keeps the system on orbit, and have proven that the equations can be solved within a certain range of the parameter. However, the algorithm based on Carleman linearization has been pointed out to have evolution time limitations. Sanavio, *et al.* have employed the Carleman linearization for each of the lattice Boltzmann method, Navier-Stokes equations, and Grad formulations, and have performed classical fluid simulations [37]. In all simulations, the solutions diverge when a certain evolution time is reached. Other similar reports also indicate the limit to the evolution time [38,39]. Theoretical analysis of this limit suggests that divergence is an unavoidable numerical phenomenon destined for Hamiltonian simulations applying Carleman linearization [39], casting doubt on the practicality of the algorithm.

We propose divergence-free simulation methods for nonlinear differential equations in quantum computers. First, for a state $\mathbf{x}$ to be solved, we assume a pivot state $\mathbf{s}$ in its neighborhood. Since $\mathbf{x}$ is expected to behave linearly in the neighborhood of $\mathbf{s}$, Hamiltonian simulations are possible. Next, each time $\mathbf{x}$ leaves the neighborhood of $\mathbf{s}$, $\mathbf{s}$ is moved to the neighborhood of $\mathbf{x}$. By repeating the above, the divergence-free algorithm is achieved. Although quantum state tomography measurements of $\mathbf{x}$ are necessary to update $\mathbf{s}$, the cost of measurements can be suppressed to a certain level because $\mathbf{s}$ only needs to satisfy the condition that it is in the neighborhood of $\mathbf{x}$. Furthermore, Carleman linearization after approximating the behavior of $\mathbf{x}$ in the neighborhood of $\mathbf{s}$ with nonlinear polynomials can effectively reduce the number of updates of $\mathbf{s}$, which is directly related to the cost. As shown

below, the proposed methods are able to track the behavior of the solutions of nonlinear equations on quantum computers over a long period of time.

II. PROBLEM STATEMENT

We focus on the evolution time limit that comes to the quantum algorithm based on Carleman linearization (see Appendix A for details of the algorithm). We clarify the problem by applying the algorithm to a simple nonlinear differential equation. The logistic equation is

$$\frac{dx}{dt} = x(1 - x), \quad (1)$$

where t is time. Carleman linearization of Eq. (1) yields the following infinite-dimensional linear differential equation:

$$\frac{dy_k}{dt} = k(y_k - y_{k+1}) \quad (k > 0), \quad (2)$$

where y_k is a real number satisfying $y_k(t=0) = x(t=0)^k$. The infinite-dimensional linear differential equation is approximated in $K+1$ dimensions by setting k to a finite value less than or equal to K and $y_k(t) = 0$ for $K > k$. For $K=3$, the approximate equation is

$$\frac{d}{dt} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}. \quad (3)$$

Figure 1(a) shows the time evolution of x obtained by solving the approximate equation numerically. The analytical solution is plotted as a dotted line. The results from the algorithm based on the Carleman linearization diverge around $t = 2.5$, regardless of K . This divergence characterizes the evolution time limit. In fact, since the logistic equation is simple, the $K+1$ -dimensional approximate equation by Carleman linearization has an analytical solution [37]. The analytical solution shows that x diverges in time evolution above the evolution time limit $t_{\text{lim}} \sim 2.39$, which is consistent with Fig. 1(a). As discussed below, similar divergence also occurs in Hamiltonian simulations of the KPP-Fisher equation and the phase-field method.

Here, we state that the evolution time limit is an unavoidable numerical phenomenon destined to the conventional algorithm based on Carleman linearization. Our goal is to overcome the evolution time limit by developing divergence-free quantum algorithms.

III. ALGORITHMS

We present the pivot switching (PS) method as a fundamental divergence-free algorithm for solving nonlinear differential equations on quantum computers. The nonlinear time differential equation for the state $\mathbf{x}$ represented by an n -dimensional real vector is

$$\frac{d\mathbf{x}}{dt} = \sum_{m=0}^M \mathbf{F}_m \mathbf{x}^{\otimes m} = \mathbf{F}_0 + \mathbf{F}_1 \mathbf{x} + \mathbf{F}_2 \mathbf{x}^{\otimes 2} + \cdots + \mathbf{F}_M \mathbf{x}^{\otimes M}, \quad (4)$$

$$\mathbf{x}^{\otimes m} = \underbrace{\mathbf{x} \otimes \cdots \otimes \mathbf{x}}_{m \text{ times}}, \quad (5)$$

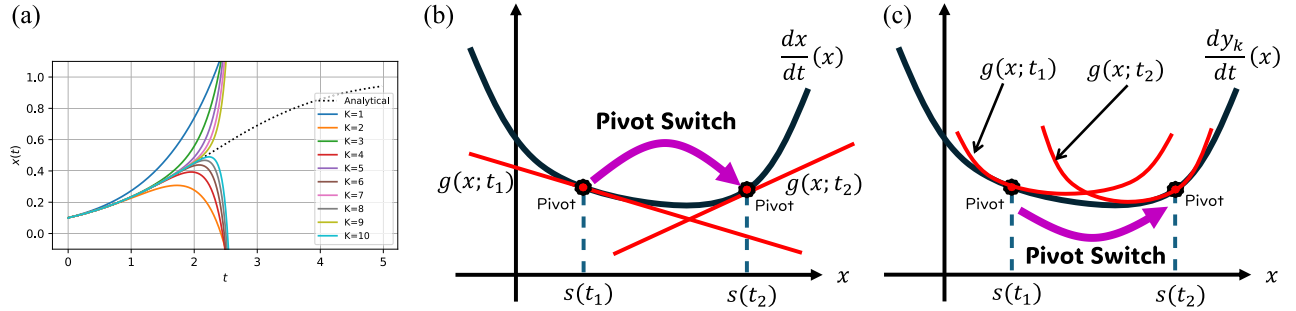


FIG. 1. (a) Time evolution of solutions of the conventional Carleman linearization-based algorithm with various K for the logistic equation. The dashed line is analytical solution. All simulations using the algorithm diverge regardless of the value of K . (b), (c) Conceptual illustrations of the PS and PSC methods proposed in this paper, respectively. $g(x, t)$ is a linear or polynomial approximation of the time derivative in the neighborhood of the pivot state s .

where t is time, M is a natural number, $x^{\otimes m}$ is an n^m -dimensional vector, and $F_m \in \mathbb{R}^n \times \mathbb{R}^{n^m}$ is a constant matrix (coefficient matrix of $x^{\otimes m}$). For convenience, $x^{\otimes 0} = 1$. From Eq. (4), the time derivative of state x is the M th order polynomial of x unless F_M is a zero matrix. Equation (4) is a nonlinear differential equation, but it can be approximated as a linear differential equation in the neighborhood of a certain state $x = s$. Here, the state s is called the pivot state. Based on the approximation, we introduce a tangent plane of the pivot state, i.e.,

$$\sum_{m=0}^M F_m x^{\otimes m} = G_0 + G_1(x - s) + O((x - s)^{\otimes 2}), \quad (6)$$

where G_0 and G_1 are constant matrices, and $O((x - s)^{\otimes a})$ is an asymptotic notation indicating that the error order is about a power of $x - s$ (a is an integer). The linearly approximated nonlinear differential equation is

$$\frac{dx}{dt} = G_0 + G_1(x - s) = (G_0 - G_1 s) + G_1 x. \quad (7)$$

Therefore, when state $x(t)$ is in the neighborhood of the pivot state, the nonlinear time differential equation can be approximated by performing the Hamiltonian simulation of Eq. (7). When x leaves the neighborhood of the pivot state, Eq. (7) is no longer valid. Therefore, each time x leaves the neighborhood of s , s is moved to the neighborhood of x , as shown in Fig. 1(b). In other words, the pivot state is updated every time interval T so that the approximate solution obtained by Eq. (7) does not exceed an allowed threshold of the error

$$E = \|x_{\text{True}}(t) - x(t)\|, \quad (8)$$

where we denote the true solution of Eq. (4) as $x_{\text{True}}(t)$. Updating the pivot state determines the cost of the PS method because it requires the measurements of x by quantum state tomography. Importantly, s needs only be in the neighborhood of x . This means that the determination of s does not require an exact measurement of x , and the cost is kept to a certain level. This switching of pivot states updates the valid range of Eq. (7) and allows the Hamiltonian simulation to continue. Through repeated switching, a divergence-free algorithm is achieved. From Eq. (7), the linear differential equation to be

solved by the PS method is

$$\frac{d}{dt} \begin{pmatrix} 1 \\ x \end{pmatrix} = \begin{pmatrix} 0 & \mathbf{0} \\ G_0 - G_1 s & G_1 \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}, \quad (9)$$

for the state $y_{\text{PS}} = (1, x)$. The dimension of the state y_{PS} is $n + 1$. The effectiveness of the PS method is discussed below.

The number of times the pivot state s is switched in the PS method is directly related to the number of measurements of state x and should be reduced. We present the pivot-switching Carleman (PSC) method as an advanced algorithm to increase the efficiency of the PS method. While the PS method approximates the $x = s$ neighborhood linearly with the tangent plane, the PSC method approximates the $x = s$ neighborhood with a polynomial surface, i.e.,

$$\sum_{m=0}^M F_m x^{\otimes m} = \sum_{p=0}^P G_p^1 (x - s)^{\otimes p} + O((x - s)^{\otimes P+1}), \quad (10)$$

where $G_p^1 \in \mathbb{R}^n \times \mathbb{R}^{n^p}$ and P is the degree of the polynomial approximation. The approximated nonlinear differential equation is

$$\frac{dx}{dt} = \sum_{p=0}^P G_p^1 (x - s)^{\otimes p}. \quad (11)$$

Although Eq. (11) is expected to have higher approximation accuracy than the linear approximation of the PS method, Hamiltonian simulations cannot be performed because it is a nonlinear differential equation. Therefore, we consider the time evolution of the state $y_k = x^{\otimes k}$ for all nonnegative integers k as in the Carleman linearization. The time derivative of state y_k is also approximated by the polynomial surface, i.e.,

$$\begin{aligned} \frac{dy_k}{dt} &= \frac{dx^{\otimes k}}{dx} \sum_{m=0}^M F_m x^{\otimes m} \\ &= \sum_{p=0}^P G_p^k (x - s)^{\otimes p} + O((x - s)^{\otimes P+1}), \end{aligned} \quad (12)$$

where $G_p^k \in \mathbb{R}^{n^k} \times \mathbb{R}^{n^p}$. The approximated equation is

$$\frac{dy_k}{dt} = \sum_{p=0}^P G_p^k (x - s)^{\otimes p}. \quad (13)$$

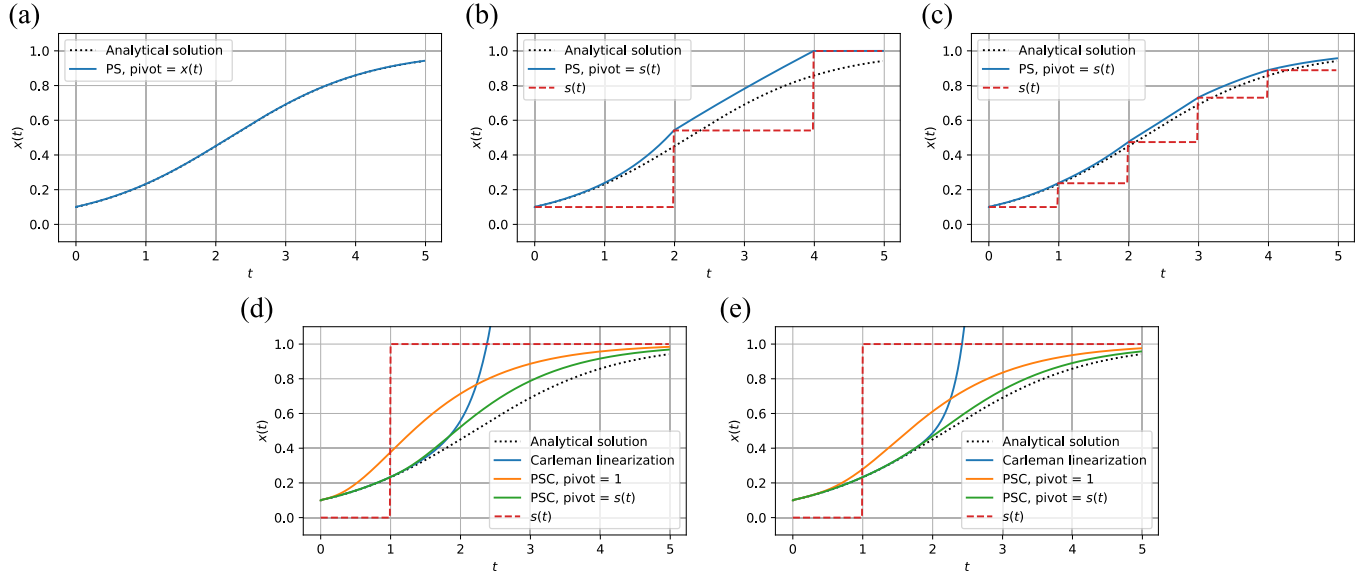


FIG. 2. Time evolution of the solution of the logistic equation. (a) Solution of the PS method with s always synchronized with x . The dashed line is the analytical solution. (b), (c) Solutions of the PS method with s switched every 1 and 2 time intervals, respectively. The red dashed lines are the time variations of s . (d), (e) Solution of the PSC method with P set to 3 and 5, respectively. The orange lines are s fixed at 1. The green lines are the initial value of s set to 0 and switched to 1 at $t = 1$. The blue lines are the solution of the conventional Carleman linearization-based algorithm.

The right-hand side is obviously a linear mapping of the vector $(1, y_1, \dots, y_P)$, so Eq. (13) is a linear differential equation for the state $\mathbf{y}_{\text{PSC}} = (1, y_1, \dots, y_P)$. The dimension of the state $\mathbf{y}_{\text{PSC}}$ is $(n^{P+1} - 1)/(n - 1)$, which is same as that of the state for Carleman-linearized systems. By performing the Hamiltonian simulation of Eq. (13), the nonlinear time differential equation in Eq. (11) can be approximately solved. Under several assumptions, the error between the approximated and true dynamics can be analytically shown to decrease exponentially with respect to the approximation order P , while increasing proportionally with the evolution time t . Moreover, both the theoretical and numerical analyses demonstrate that although the choice of the pivot state s governs the rate of error reduction with increasing P , small variations in the pivot state do not significantly affect the error. The detailed derivations supporting these error analyses are provided in Appendixes B and C.

It is not necessary to calculate the time evolution of $\mathbf{y}_k$ with $k > P$, since $\mathbf{y}_k$ with $k > P$ does not contribute to the time evolution of $\mathbf{y}_1$. Even with polynomial surfaces, Eqs. (11) and (13) are no longer valid when $\mathbf{x}$ moves away from the neighborhood of the pivot state. Therefore, as with the PS method, each time $\mathbf{x}$ leaves the neighborhood of s , s is moved to the neighborhood of $\mathbf{x}$ [see Fig. 1(c)]. Since polynomial surfaces yield higher approximation accuracy than planar surfaces, the PSC method is expected to effectively reduce the number of times s is determined. By a small number of repetitions of switching, a divergence-free and efficient algorithm is achieved. The effectiveness of the PSC method is discussed below.

IV. APPLICATIONS

Our algorithms can break through the evolution time limit imposed on the conventional algorithm based on Carleman

linearization. To demonstrate this, our algorithms were applied not only to the logistic equation illustrated in Sec. II, but also to the reaction-diffusion equation found in ecology, physiology, combustion, and plasma physics, and to the phase-field model applied to general phase-transition problems. Furthermore, the logistic equation was solved on a real quantum computer machine. The linear equations obtained by our algorithms were integrated by the first-order forward Euler method with time step size $\Delta t = 0.01$ in all the applications below.

A. Logistic equation

As mentioned in Sec. II, the conventional Carleman linearization-based algorithm cannot solve the logistic equation over long evolution times in Hamiltonian simulations, so the PS and PSC methods were applied. With the PS method, the logistic equation was approximated as follows:

$$\frac{dx}{dt} = s^2 - (1 - 2s)x, \quad (14)$$

where s is the pivot state. The ideal computational accuracy of the PS method is achieved by always having $s(t) = x(t)$. Figure 2(a) shows the solution from the PS method when $s(t) = x(t)$ always holds. The ideal PS method is in perfect agreement with the analytical solution [cf. Fig. 1(a)]. However, from a practical point of view, obtaining the state $x(t)$ from the quantum state is not easy. This is because it requires estimation of $x(t)$ by quantum state tomography or other methods, and multiple independent Hamiltonian simulations must be repeated many times. Moreover, while time-independent Hamiltonian simulations can be implemented efficiently [26], efficient implementations of time-dependent Hamiltonian simulations are not known. For these reasons, the number of times to switch pivot states

should be reduced as much as possible. Figures 2(b) and 2(c) show the results of the PS method with the pivot state switched every 1 and 2 time intervals, respectively. The time-dependent pivot state $s(t)$ is also plotted. Switching at every 1 time interval produced results close to the analytical solution [Fig. 2(b)], while switching at every 2 time interval showed a slight deviation from the analytical solution [Fig. 2(c)]. Note that this deviation is acceptable depending on the degree of tolerance for the error E . The maximum values of E in Figs. 2(b) and 2(c) were 0.142 and 0.042, respectively. It is worth noting that in both cases, the evolution time limit of the conventional Carleman linearization-based algorithm was clearly surpassed.

By the PSC method, the logistic equation was approximated as follows:

$$\frac{dy_k}{dt} = \sum_{p=0}^P g_p^k (x-s)^p, \quad (15)$$

$$g_p^k = \frac{1}{p!} \left. \frac{d^p(x(1-x)kx^{k-1})}{dx^p} \right|_{x=s}. \quad (16)$$

From Eqs. (15) and (16), the linear differential equation to be solved was obtained. For example, when $P = 3$, the logistic equation is approximated as follows:

$$\frac{d}{dt} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 2 & -2 \\ 3s^4 & -12s^3 & 18s^2 & 3-12s \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}. \quad (17)$$

Figures 2(d) and 2(e) show the results of the PSC method when $P = 3$ and 5, respectively. The analytical solution, the conventional Carleman linearization, and the time-dependent pivot state $s(t)$ are also plotted. The pivot state of the PSC method was either fixed at 1 or switched from 0 to 1 once at $t = 1$. The timing of the pivot switching can be estimated approximately from the analysis provided in Appendix B. In this study, while referring to this estimate, we chose to perform at most a single pivot switch during the simulation so that the effect of pivot switching could be clearly highlighted in comparison with Carleman linearization. In Figs. 2(d) and 2(e), the maximum values of the error E in the fixed-pivot case were 0.263 for $P = 3$ and 0.169 for $P = 5$, whereas in the pivot-switching case they were 0.100 for $P = 3$ and 0.045 for $P = 5$.

Surprisingly, the PSC method broke the evolution time limit without switching the pivot state, and the obtained solution was acceptable depending on the tolerance E . Furthermore, switching the pivot state only once yielded results closer to the analytical solution. Increasing the degree of the polynomial approximation P also improved the accuracy of the results. In particular, when $P = 5$ and the pivot state was switched once, the accuracy was equivalent to that of the PS method, which switched the pivot state every 1 time interval. This means that the PSC method reduced the number of times the pivot state was switched to 1/5 without any loss of accuracy. Based on this finding, the following sections focus on the evaluation of the PSC method.

B. Demonstration on actual quantum computers

To demonstrate that the PSC method works well in practice, we solved the logistic equation using actual quantum computers. Equation (17) (i.e., $P = 3$) was employed and the pivot state s was set to 1. However, it is almost impossible to implement complex quantum circuits on currently available quantum computers. Therefore, in implementing Eq. (17), we devised the following methods. First, let $dz/dt = A(t)z$ be the linear time evolution equation to be solved, where the dimension of z is a power of two. Since A is not skew Hermitian in general and therefore the time evolution by A is not necessarily unitary, we adopted a method based on a unitary dilation [14], which allows nonunitary time evolution. Specifically, the time-evolution operator

$$V_T = \mathcal{T} \exp \left[\int_0^T A(t) dt \right] \quad (18)$$

is used to construct a unitary time evolution

$$U_T = \begin{bmatrix} sV_T & \sqrt{I - s^2 V_T V_T^\dagger} \\ \sqrt{I - s^2 V_T V_T^\dagger} & -sV_T \end{bmatrix} \quad (19)$$

for the expanded system, where $\mathcal{T}$ is the time ordering operator and s is the scaling parameter (a real number) for the existence of the square root of the matrix $I - s^2 V_T V_T^\dagger$. In this experiment, s was set to the reciprocal of the largest eigenvalue of V_T . Applying U_T to the initial quantum state $|\psi_0\rangle = [z/\|z\|, \mathbf{0}]^T$, we obtain a vector of $V_T z(0)$ multiplied by a constant as the first-half component of the quantum state as follows:

$$U_T |\psi_0\rangle = \left[\frac{s}{\|z\|} V_T z(0), \frac{s}{\|z\|} \sqrt{I - s^2 V_T V_T^\dagger} z(0) \right]^T. \quad (20)$$

Since $V_T z(0) = z(T)$, by measuring this quantum state multiple times in the standard basis, we can obtain the simulation results of $z(T)$, except for the positive and negative signs. That is, if the total number of measurements is n_{ALL} and the number of times the i th vector of the standard basis $|i\rangle$ has been measured is n_i , the absolute value of the estimate of the i th component of $z(T)$ can be estimated by

$$|z(T)| \simeq \frac{s}{\|z(0)\|} \sqrt{\frac{n_i}{n_{\text{ALL}}}}. \quad (21)$$

Let the matrix in Eq. (17) be $B(s_t)$, the above method is theoretically usable to simulate Eq. (17) on real quantum computers by replacing $z = y = [y_0, y_1, y_2, y_3]^T$ and $A(t) = B(s_t)$. In practice, however, this is difficult with the current machine because the estimation error for y_1 originating from statistical errors and noise in the quantum gates is too large. To reduce estimation errors of the simulation results, we scaled each component of the state as

$$z = [r_0 y_0, r_1 y_1, r_2 y_2, r_3 y_3]^T = R y, \quad (22)$$

where r_i is a constant and $R = \text{diag}(r_0, r_1, r_2, r_3)$. With this scaling, the equation to be solved has the following form:

$$\frac{dz}{dt} = R B(t) R^{-1} z. \quad (23)$$

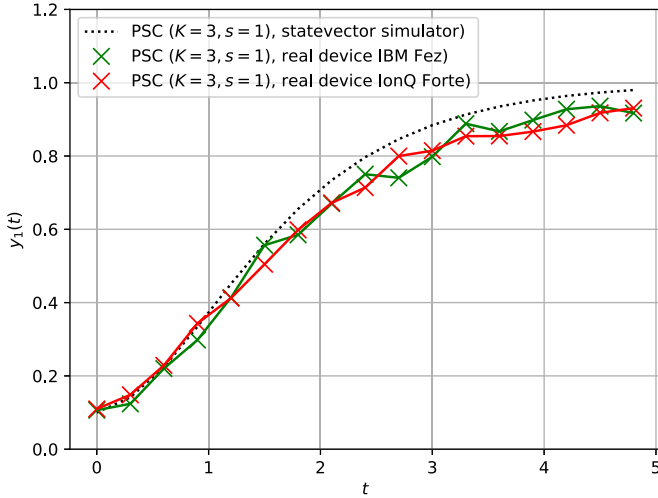


FIG. 3. The obtained simulation results of logistic equation using PSC method by actual quantum computers IBM_fez (green) and IonQ_Forte (red). The black dotted line is the result of the simulator calculation.

Note that we used $R = \text{diag}(1, 3, 3, 1)$ based on our preliminary test.

Based on the above methods, a state preparation circuit to prepare $|\psi_0\rangle$ and the unitary gate U_T for time evolution were applied in turn, and then all qubits were measured in the standard basis to get the simulation result of the logistic equation on an actual quantum computer. IBM's IBM_fez [40] and IonQ's IonQ_Forte [41] were used as the real machines, and the quantum circuits were converted to the native gates of the real machines by `qiskit.transpiler`. For error mitigation, Q-ctrl Fire Opal [42] was used for IBM_fez and IonQ's default error mitigation function was used for IonQ_Forte. n_{ALL} was set to 4096. Note that since z is four-dimensional, the expansion system $|\psi_0\rangle$ is eight-dimensional, requiring three qubits.

Figure 3 shows the simulation results of the logistic equation by the actual quantum computers. The IBM_fez and IonQ_Forte results were plotted as green and red dots, respectively, and the result from the noiseless simulation with error-free estimation (statevector simulator) was shown as a black dashed line. Both of the actual machines are in good agreement with the results from the statevector simulator. The results highlight that our proposed PSC method is a practical method that works well on real quantum computers.

C. Reaction-diffusion equation

The KPP-Fisher equation is a class of reaction-diffusion equations that can exhibit traveling wave solutions that switch between equilibrium states and has been applied to problems in ecology and physiology, combustion and plasma physics, and crystallization. The KPP-Fisher equation is expressed as follows:

$$\frac{\partial u(z, t)}{\partial t} = \frac{\partial^2 u}{\partial z^2} + u(1 - u), \quad (24)$$

where $u(z, t) \in \mathbb{R}$ is the traveling wave solutions. The first term on the right-hand side is a linear diffusion term and

the second term on the right-hand side is a second-order nonlinear source term. Equation (24) is a differential equation with spatial coordinate z as a continuous variable, which was difficult to compute directly, so it was discretized with respect to z . Assuming that the computational point $i \in \mathbb{Z}$, equally spaced with width Δz , was at coordinate $z = i\Delta z$, and applying second-order precision central differencing to the diffusion term, Eq. (24) was approximated by the following difference equation:

$$\frac{du_i}{dt} = \frac{1}{\Delta z^2} (u_{i-1} - 2u_i + u_{i+1}) + u_i(1 - u_i), \quad (25)$$

where $u_i \in \mathbb{R}$. By setting $\Delta z^2 = 1$ and imposing the periodic boundary condition $u_i = u_{i+8}$, Eq. (25) became the finite-dimensional nonlinear differential equation for state $\mathbf{u} = (u_1, u_2, \dots, u_7) \in \mathbb{R}^8$. This was solved using the conventional Carleman linearization and the PSC method and compared with the classical numerical solution. The initial condition for the simulation was

$$u_i(t=0) = \begin{cases} 0.9, & i = 3, 4 \\ 0.1, & \text{otherwise} \end{cases}. \quad (26)$$

Figure 4(a) is the classical numerical solution, which is considered the most accurate. Figure 4(b) is the solution from the conventional Carleman linearization for $K = 3$. Similar to the logistic equation results [Fig. 1(a)], the solution from the conventional Carleman linearization diverged when the evolution time limit was reached. Figures 4(c) and 4(d) show the results of the PSC method with $P = 3$ and 5, respectively, and with the pivot state fixed at 1. For the KPP-Fisher equation, the PSC method also broke the evolution time limit without switching the pivot state, and the obtained solution was acceptable depending on the tolerance E . Figures 4(e) and 4(f) show the results of the PSC method with $P = 3$ and 5, respectively, with the pivot state set to the same value as the initial condition and switched to 1 only once at $t = 1$. By switching the pivot state only once, the solution approached that of the classical numerical method. In particular, when $P = 5$ and the pivot state was switched only once, $u(t)$ was in good agreement with those of the classical numerical simulation. This means that the KPP-Fisher equation can be solved on quantum computers by the PSC method without divergence or loss of accuracy.

D. Phase-field model

The phase-field model is one of the practical models applied to general phase-transition problems. We considered the model expressed as follows:

$$\frac{\partial \phi(z, t)}{\partial t} = \frac{\partial^2 \phi}{\partial z^2} + (\phi - 1)(\phi + \beta)(\phi + 1), \quad (27)$$

where $\phi(z, t)$ is the order parameter. The first term on the right-hand side is a linear term and the second term on the right-hand side is a third-order nonlinear reaction term. The reaction term drives the order parameter to 1 or -1 . β determines the stability of the order parameter: if $\beta > 0$, $\phi = 1$ is stable; if $\beta < 0$, $\phi = -1$ is stable. Equation (27) is a differential equation with spatial coordinate z as a continuous variable, so we discretized it with respect to z . Assuming that

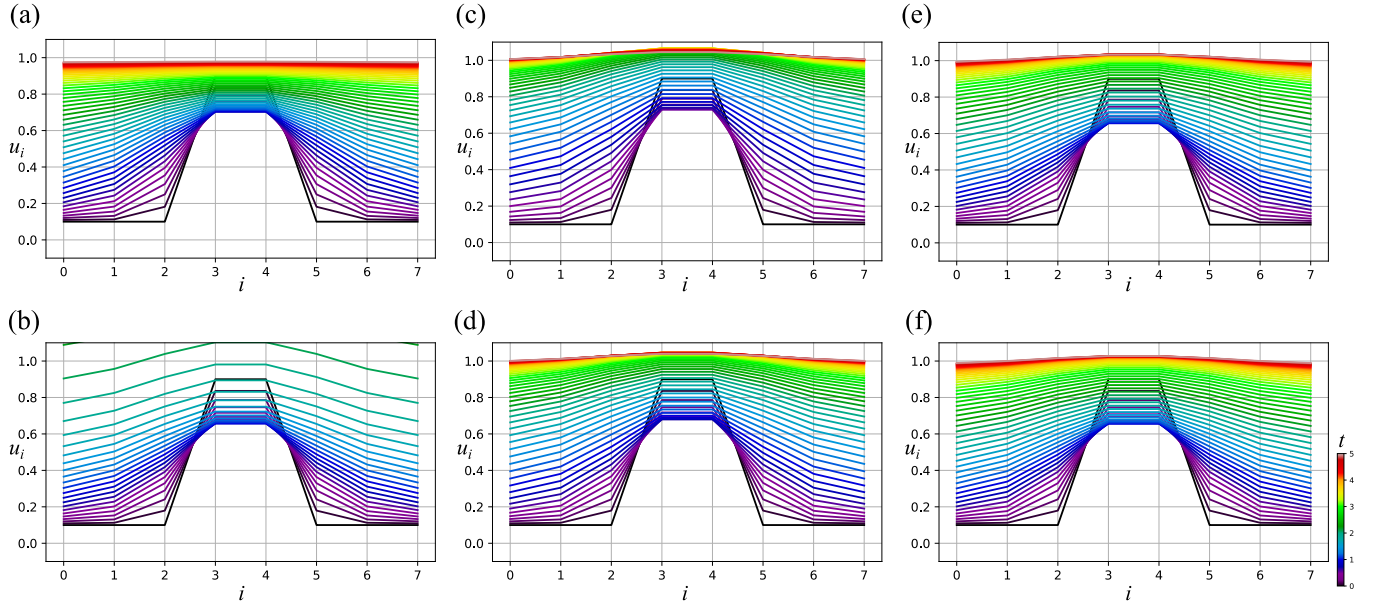


FIG. 4. Time evolution of the solution of the KPP-Fisher equation. (a) Classical computational solution. (b) Solution of the conventional Carleman linearization-based algorithm with $K = 3$. (c), (d) Solution of the PSC method with P set to 3 and 5, respectively, and s fixed to $\mathbf{1}$. (e), (f) Solution of the PSC method with P set to 3 and 5, respectively, and S initially set to $\phi(t = 0)$ and switched to $\mathbf{1}$ at $t = 1$.

the computational point i , equally spaced with width Δz , was at coordinate $z = i\Delta z$, and applying second-order precision central differencing to the diffusion term, Eq. (27) was approximated by the following difference equation:

$$\frac{d\phi_i}{dt} = \frac{1}{\Delta z^2}(\phi_{i-1} - 2\phi_i + \phi_{i+1}) + (\phi_i - 1)(\phi_i + \beta)(\phi_i + 1). \quad (28)$$

By setting $\Delta z^2 = 1$ and imposing the periodic boundary condition $\phi_i = \phi_{i+8}$, Eq. (28) became the finite-dimensional nonlinear differential equation for the state $\phi = (\phi_0, \phi_1, \dots, \phi_7) \in \mathbb{R}^8$. This equation was solved using the conventional Carleman linearization and PSC methods and compared with the classical numerical solution. β was set to -0.2 and the initial condition for the simulation was

$$\begin{aligned} \phi(t = 0) \\ = (-0.90, -0.56, 0.56, 0.90, 0.90, 0.56, -0.56, -0.90). \end{aligned} \quad (29)$$

This initial condition was set so that $\phi(t) = \phi(0)$ when $\beta = 0$, i.e., an equilibrium state.

Figure 5(a) is the classical numerical solution, which is considered the most accurate. Since $\beta < 0$, the solution evolved to $\phi = -\mathbf{1}$. Figure 5(b) is the solution from the conventional Carleman linearization for $K = 3$. Similar to the results for the logistic equation [Fig. 1(a)] and the KPP-Fisher equation [Fig. 4(b)], the solution from the conventional Carleman linearization diverged when the evolution time limit was reached. Figures 5(c) and 5(d) show the results of the PSC method with $P = 3$ and 5, respectively, and with the pivot state fixed at $-\mathbf{1}$. In the phase-field model, the PSC method broke the evolution time limit without switching the pivot

state. However, the solution deviations in the early simulation time were significant; recall that the theoretical range of values for ϕ is from -1 to 1 . Figures 5(e) and 5(f) show the results of the PSC method with $P = 3$ and 5, respectively, with the pivot state set to the same value as the initial condition and switched to $-\mathbf{1}$ only once at $t = 2.9$. By switching the pivot state only once, the solution approached that of the classical numerical solution. In particular, when $P = 5$ and the pivot state was switched only once, the phase fields $\phi(t)$ were in good agreement with those of the classical numerical simulation. This means that the phase-field model can be solved on a quantum computer by the PSC method with sufficient accuracy and without divergence.

V. DISCUSSION

In this study, we have presented the PS method as the fundamental divergence-free algorithm for solving nonlinear differential equations on quantum computers. Furthermore, we have presented the PSC method as the advanced algorithm to improve the efficiency of the PS method. These algorithms break through the evolution time limit destined for the conventional Carleman linearization-based quantum algorithm [36], as confirmed by the logistic equation, the KPP-Fisher equation, and the phase-field model. Applications of the PS method to logistic equations have shown that the basic idea of switching pivot states makes a substantial contribution to achieving divergence-free algorithms. The more frequent the switching, the closer the solution by the PS method is to the analytical solution, but the further away the method is from practicality. This trade-off has been greatly mitigated by the PSC method. The idea of approximating the $x = s$ neighborhood nonlinearly with polynomial surface rather than linearly with tangent plane and then solving by Carleman linearization has been shown to significantly reduce the number of pivot

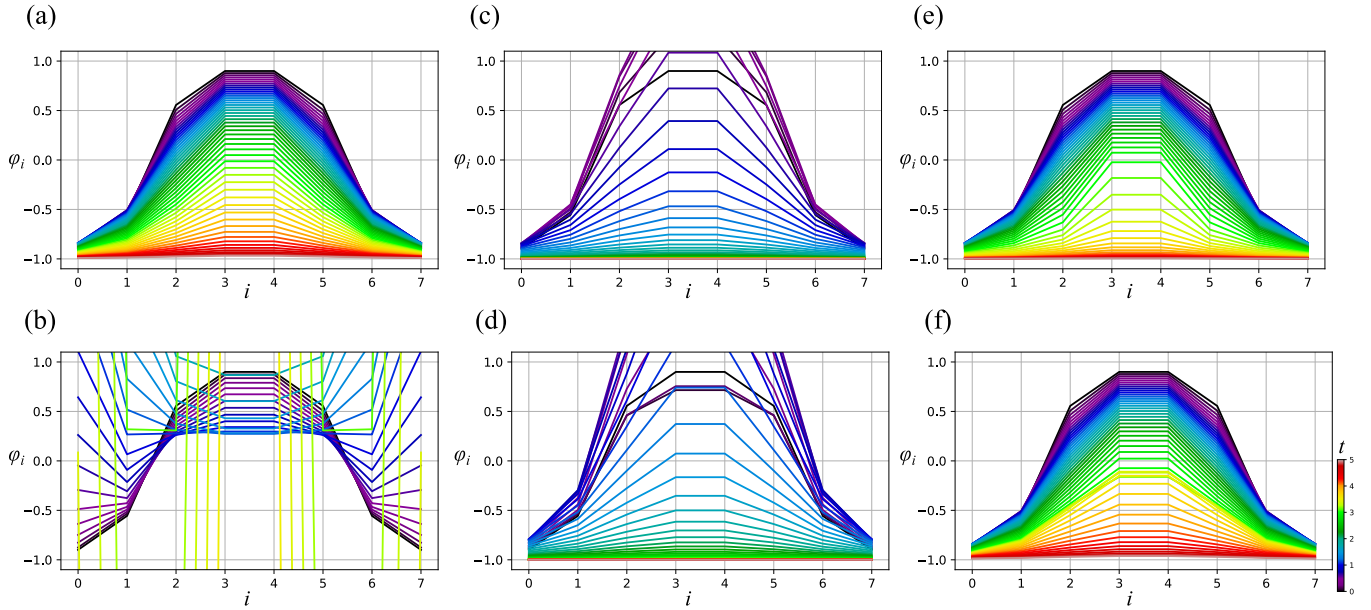


FIG. 5. Time evolution of the solution of the phase-field model. (a) Classical computational solution. (b) Solution of the conventional Carleman linearization-based algorithm with $K = 3$. (c), (d) Solution of the PSC method with P set to 3 and 5, respectively, and s fixed to $\mathbf{1}$. (e), (f) Solution of the PSC method with P set to 3 and 5, respectively, and s initially set to $\phi(t = 0)$ and switched to $\mathbf{1}$ at $t = 2.9$.

state updates while maintaining solution accuracy. Applied to the KPP-Fisher equation and phase-field model, the PSC method not only has achieved divergence-free Hamiltonian simulations but also has yielded solutions that close to those of classical numerical simulations with only one coarse pivot-state switching operation. This means that practical problems described by nonlinear differential equation systems can be solved on quantum computers with sufficient accuracy using the PSC method.

Here, let us emphasize that the PSC method was successfully used on the actual quantum computers. The PSC method was run on two different machines and the logistic equation was solved, although secondary methods were introduced to compensate for the inability of actual machines to run complex quantum circuits. The results were in good agreement with those obtained when error-free estimates were obtained in noiseless simulations. This means that the PSC method is a practical method that can be used with currently available quantum computers.

The mathematical differences between the PSC method and the conventional Carleman linearization-based algorithm are interesting. For example, comparing Eqs. (3) and (17), the only difference is in the highest order y_3 differential equation. The difference in the results produced by the two methods is due to the behavior of the solution of the highest-order equation. This is also true for general nonlinear systems of equations. Before Taylor truncation, the time derivative dy_k/dt is the product of $(k - 1)$ -degree polynomial $dx^{\otimes k}/dx$ and M -degree polynomial dx/dt [see Eq. (12)], which together give a polynomial of degree at most $M + k - 1$. Since the truncating terms of degree higher than P does not change any polynomial of degree at most P , no truncation occurs when $k < (P - M)$. This indicates that the time derivative of y_k [$k < (P - M)$] is invariant regardless of the pivot state. Thus, stabilizing the behavior of higher-order terms in a

finite Carleman linearization is a prescription to prevent catastrophic results due to solution divergence. The PSC method is one method for stabilizing the behavior and presents the possibility of other methods based on similar concepts.

The PS and PSC methods rely on “loose estimations” to obtain an estimate of state x with error from the small number of measurement operations, which is then used as the pivot state s . This paper has demonstrated that loose estimations are sufficient, since s only needs to satisfy the condition that it is in the neighborhood of x . Although loose estimations can reduce the cost of measurement to the certain level, the PSC method has been proposed for further cost reduction. Thus, loose estimations are a necessary cost to prevent solution divergence. This cost could be further reduced by combining loose estimations with weak measurements [43].

A possible future strategy is to determine the pivot state directly by quantum computation rather than obtaining it from loose estimations. Under quantum superposition, multiple candidate pivot states are employed, and Hamiltonian simulations are performed in parallel. By using an amplitude amplification algorithm [44] to remove only those candidate pivot states that move toward divergence, the pivot state can be determined without loose estimations. More broadly, intrinsically quantum subroutines such as amplitude estimation would provide more efficient and scalable approaches that avoid intermediate measurements. Another promising direction is to develop adaptive pivot-updating strategies using machine-learning and reinforcement-learning methods to reduce the update frequency. We expect that our methods, in combination with other promising methods, lead to efficient and robust quantum algorithms for nonlinear systems of equations.

A further direction is to establish formal and concrete complexity analysis for our divergence-free algorithms. There currently exists no general and systematic procedure for

designing quantum circuits derived not only from our PSC method but also from Carleman-linearized systems. Developing such a method for quantum circuit construction is an essential next step toward realizing practical implementation of Carleman- and PSC-based nonlinear simulations on actual quantum hardware. Such progress would enable definitive complexity evaluations and allow quantitative comparison with other linearization-based approaches.

Finally, let us emphasize that the algorithms presented in this paper are significant in that these can track the behavior of the solution to systems of nonlinear equations over the long period of time. For example, in general phase-transition problems, the behavior of the solution over time is important for understanding and predicting the phenomenon. This importance will become more pronounced when quantum computers can be applied to larger and high-dimensional nonlinear systems, i.e., large systems that represent phenomena in two- or three-dimensional spatial coordinate systems. We expect that divergence-free quantum algorithms for nonlinear equations represent a milestone in practical applications of quantum computers and will motivate research into more general-purpose algorithms.

ACKNOWLEDGMENT

This paper is partially based on results obtained from a project, Council for Science, Technology and Innovation (CSTI), Cross-ministerial Strategic Innovation Promotion Program (SIP), “Promoting the application of advanced quantum technology platforms to social issues” (Funding agency: QST).

DATA AVAILABILITY

The data that support the findings of this article are openly available [45]; embargo periods may apply.

APPENDIX A: CONVENTIONAL QUANTUM ALGORITHM BASED ON CARLEMAN LINEARIZATION

The method based on Carleman linearization has been employed to solve Eq. (4) by Hamiltonian simulation. In this method, the time evolution of extended variables $\mathbf{y}_k = \mathbf{x}^{\otimes k}$ for all non-negative integer k is taken into account. From Eq. (4), the time-evolution equation for $\mathbf{y}_k$ is

$$\frac{d\mathbf{y}_k}{dt} = \frac{d\mathbf{x}^{\otimes k}}{dt} = \frac{d\mathbf{x}^{\otimes k}}{d\mathbf{x}} \frac{d\mathbf{x}}{dt} = \frac{d\mathbf{x}^{\otimes k}}{d\mathbf{x}} \sum_{m=0}^M \mathbf{F}_m \mathbf{x}^{\otimes m}. \quad (\text{A1})$$

Equation (A1) is further expanded as

$$\frac{d\mathbf{y}_k}{dt} = \sum_{l=k}^{k+M} A_{k,l} \mathbf{y}_l \quad (k \geq 0), \quad (\text{A2})$$

where $A_{k,l} \in \mathbb{R}^{n^k \times n^l}$ is a constant coefficient matrix defined as

$$A_{k,l} = \left(\sum_{v=0}^{k-1} \underbrace{I \otimes \cdots \otimes I}_{v \text{ times}} \otimes \mathbf{F}_{l-k-1} \otimes \underbrace{I \otimes \cdots \otimes I}_{(k-1-v) \text{ times}} \right). \quad (\text{A3})$$

The following relationship is used to derive Eq. (A2):

$$\frac{d\mathbf{x}^{\otimes k}}{d\mathbf{x}} = \sum_{v=0}^{k-1} \underbrace{\mathbf{x} \otimes \cdots \otimes \mathbf{x}}_{v \text{ times}} \otimes I \otimes \underbrace{\mathbf{x} \otimes \cdots \otimes \mathbf{x}}_{(k-1-v) \text{ times}}. \quad (\text{A4})$$

Equation (A2) is the system of linear equations for infinite-dimensional vector $\mathbf{y}_{\text{inf}} = (y_0, y_1, y_2, \dots)$. Equation (A2) implies that the solution to the nonlinear differential equation [Eq. (4)] can be obtained by solving the system of infinite-dimensional linear differential equations. However, since $\mathbf{y}_{\text{inf}}$ is infinite dimensional, it cannot be solved realistically even if a quantum computer is used. Therefore, the higher order terms in Eq. (A2) are truncated as follows:

$$\frac{d\mathbf{y}_k}{dt} = \sum_{l=k}^K A_{k,l} \mathbf{y}_l \quad (K \geq k \geq 0). \quad (\text{A5})$$

Equation (A5) is the system of equations to be solved by the quantum algorithm based on Carleman linearization. Note that even if the system of equations is not unitary, a solution can be obtained by applying nonunitary Hamiltonian simulation methods.

APPENDIX B: ANALYTICAL ERROR BOUNDS FOR THE PROPOSED METHODS

By considering the time derivative of the higher-order tensor product $\mathbf{y}_k = \mathbf{x}^{\otimes k}$ with respect to the state of system $\mathbf{x} \in \mathbb{R}^N$, we obtain the time-evolving equation of $\mathbf{y}_k$

$$\frac{d\mathbf{y}_k}{dt} = \mathbf{g}_k(\mathbf{x}) := \frac{d\mathbf{x}^{\otimes k}}{d\mathbf{x}} \sum_{m=0}^M \mathbf{F}_m \mathbf{x}^{\otimes m}, \quad (\text{B1})$$

as in Eq. (12). In the PSC method, $\mathbf{g}_k(\mathbf{x})$ is expanded in a Taylor series around $\mathbf{x} = \mathbf{s}$ and truncated at order P , and we obtain

$$\frac{d\mathbf{y}_k}{dt} = \sum_{p=0}^P \mathbf{G}_p^k (\mathbf{x} - \mathbf{s})^{\otimes p}, \quad (\text{B2})$$

which is reproduced from Eq. (13). Rearranging the right-hand side of Eq. (B2) in terms of $\mathbf{y} = (y_0, y_1, \dots, y_P)$, we obtain the linear differential equation

$$\frac{d\mathbf{y}}{dt} = \mathbf{C}\mathbf{y}, \quad (\text{B3})$$

which is the linearized matrix obtained by PS method ($P = 1$) or PSC method ($P > 1$).

Here, the truncation error introduced by the Taylor expansion in Eq. (B2) can be evaluated by the Lagrange remainder, that is, there exists a vector ξ dividing the vector $\mathbf{x}$ and $\mathbf{s}$ internally such that, by defining the remainder term

$$\mathbf{r}_k(\mathbf{x}) = \frac{[\nabla_{\mathbf{x}}^{\otimes(P+1)} \mathbf{g}_k](\xi^{\otimes(P+1)})}{(P+1)!} (\mathbf{x} - \mathbf{s})^{\otimes(P+1)}, \quad (\text{B4})$$

the following relation holds:

$$\mathbf{g}_k(\mathbf{x}) = \sum_{p=0}^P \mathbf{G}_p^k (\mathbf{x} - \mathbf{s})^{\otimes p} + \mathbf{r}_k(\mathbf{x}). \quad (\text{B5})$$

Therefore, for the true solution of the original system $\mathbf{x}_{\text{True}}(t)$, if we define $\mathbf{y}_k^{\text{True}} = \mathbf{x}_{\text{True}}^{\otimes k}$, then the time evolution of $\mathbf{y}_{\text{True}} = (\mathbf{y}_0^{\text{True}}, \mathbf{y}_1^{\text{True}}, \dots, \mathbf{y}_P^{\text{True}})$ follows from Eq. (B2) as

$$\frac{d\mathbf{y}_{\text{True}}}{dt} = C\mathbf{y}_{\text{True}} + \mathbf{r}(\mathbf{x}_{\text{True}}), \quad (\text{B6})$$

where we defined $\mathbf{r} = (r_0, r_1, \dots, r_P)$.

The differential equation in Eq. (B6) can be formally solved by Duhamel's formula as

$$\mathbf{y}_{\text{True}} = e^{Ct}\mathbf{y}_{\text{True}}(0) + \int_0^t e^{C(t-\tau)}\mathbf{r}(\mathbf{x}_{\text{True}}(\tau))d\tau. \quad (\text{B7})$$

In contrast, the analytical solution of the linearized equation obtained by the PSC method [Eq. (B3)] is

$$\mathbf{y}(t) = e^{Ct}\mathbf{y}(0). \quad (\text{B8})$$

Assuming that the initial condition of $\mathbf{y}$ is identical to the true initial state $\mathbf{y}_{\text{True}}$, their difference is therefore given by

$$\Delta\mathbf{y} := \mathbf{y} - \mathbf{y}_{\text{True}} = \int_0^t e^{C(t-\tau)}\mathbf{r}(\mathbf{x}_{\text{True}}(\tau))d\tau. \quad (\text{B9})$$

It follows that the error between the true solution $\mathbf{x}_{\text{True}}$ and the solution obtained by the PSC method $\mathbf{x}(t)$ can be evaluated as

$$\begin{aligned} E(t) &:= \|\mathbf{x}(t) - \mathbf{x}_{\text{True}}(t)\| = \|\Pi_1 \Delta\mathbf{y}\| \\ &= \left\| \Pi_1 \int_0^t e^{C(t-\tau)}\mathbf{r}(\mathbf{x}_{\text{True}}(\tau))d\tau \right\|. \end{aligned} \quad (\text{B10})$$

Here, Π_1 denotes the projection matrix that extract $\mathbf{x}$ from $\mathbf{y}$.

To estimate the error of the PSC method, we derive an upper bound for the above error term. First, by the submultiplicativity of the norm and the triangle inequality, we obtain

$$\begin{aligned} &\left\| \Pi_1 \int_0^t e^{C(t-\tau)}\mathbf{r}(\mathbf{x}_{\text{True}}(\tau))d\tau \right\| \\ &\leq \|\Pi_1\| \int_0^t \|e^{C(t-\tau)}\|\|\mathbf{r}(\mathbf{x}_{\text{True}}(\tau))\|d\tau. \end{aligned} \quad (\text{B11})$$

Since Π_1 is a projection matrix that extracts a subset of the vector components, the norm of the projected vector never exceed of the original vector, and hence $\|\Pi_1\| \leq 1$.

From Eq. (B4), we also have

$$\|\mathbf{r}_k\| \leq \frac{\|[\nabla_x^{\otimes(P+1)}\mathbf{g}_k](\xi^{\otimes(P+1)})\|}{(P+1)!} \|\mathbf{x} - \mathbf{s}\|^{\otimes(P+1)}. \quad (\text{B12})$$

If the true state $\mathbf{x}_{\text{True}}$ stays within the region $\|\mathbf{x}_{\text{True}} - \mathbf{s}\| \leq \rho$ during the time interval between $(0, t)$, then

$$\|\mathbf{r}_k\| \leq \frac{M_k}{(P+1)!} \rho^{P+1}, \quad (\text{B13})$$

where

$$M_k := \sup_{\|\mathbf{u}-\mathbf{s}\| \leq \rho} \|[\nabla_x^{\otimes(P+1)}\mathbf{g}_k](\mathbf{u}^{\otimes(P+1)})\|. \quad (\text{B14})$$

The constant M_k depends only on the property of the function $\mathbf{g}_k$. As Eq. (B1) shows that the function $\mathbf{g}_k$ is a polynomial determined from the coefficient set $\{\mathbf{F}_m\}$. Using this fact, we

can derive a detailed bound for M_k . From Eq. (B1), we have

$$\begin{aligned} &\nabla_x^{\otimes(P+1)}\mathbf{g}_k \\ &= \nabla_x^{\otimes(P+1)}\left(\frac{d\mathbf{x}^{\otimes k}}{dx} \sum_{m=0}^M \mathbf{F}_m \mathbf{x}^{\otimes m}\right) \\ &= \nabla_x^{\otimes(P+1)}\left(\sum_{v=0}^{k-1} \sum_{m=0}^M \mathbf{x}^{\otimes v} \otimes \mathbf{F}_m \mathbf{x}^{\otimes m} \otimes \mathbf{x}^{\otimes k-1-v}\right). \end{aligned} \quad (\text{B15})$$

Here, using the relation $\|\nabla_x^{\otimes c}(\mathbf{A}\mathbf{x}^{\otimes b})\| = 0$, for $b - c < 0$ and

$$\|\nabla_x^{\otimes c}(\mathbf{A}\mathbf{x}^{\otimes b})\| \leq \frac{b!}{(b-c)!} \|\mathbf{A}\| \|\mathbf{x}\|^{b-c}, \quad (b-c \geq 0), \quad (\text{B16})$$

we obtain

$$[\nabla_x^{\otimes(P+1)}\mathbf{g}_k](\mathbf{u}^{\otimes(P+1)}) = 0, \quad (\text{B17})$$

for $k < (P+2-M)$ and

$$\begin{aligned} &[\nabla_x^{\otimes(P+1)}\mathbf{g}_k](\mathbf{u}^{\otimes(P+1)}) \\ &\leq \sum_{m=P+2-k}^M {}_{m+k-1}C_{P+1} k (P+1)! \|\mathbf{F}_m\| \|\mathbf{u}\|^{m+k-1-(P+1)}, \end{aligned} \quad (\text{B18})$$

for $(P+2-M) \leq k \leq P$, where we use combination notation ${}_aC_b = a!/b!(a-b)!$. In the region $\|\mathbf{u} - \mathbf{s}\| \leq \rho$, $\|\mathbf{u}\| \leq R := \|\mathbf{s}\| + \rho$ is also satisfied, and therefore, we have the bound as

$$M_k \leq \sum_{m=P+2-k}^M {}_{m+k-1}C_{P+1} k (P+1)! \|\mathbf{F}_m\| R^{m+k-1-(P+1)}, \quad (\text{B19})$$

for $(P+2-M) \leq k \leq P$. For subsequent calculation, the above bound can be simplified and looser upper bound

$$M_k \leq e^M P^M R_1^M (P+1)! \sum_{m=P+2-k}^M \|\mathbf{F}_m\|, \quad (\text{B20})$$

where we used the relation ${}_{m+k-1}C_{P+1} k \leq e^M P^M$ and defined $R_1 := \max(1, R)$. Therefore, Eq. (B13) can be bounded as

$$\|\mathbf{r}_k\| \leq e^M P^M R_1^M \rho^{P+1} \sum_{m=P+2-k}^M \|\mathbf{F}_m\|. \quad (\text{B21})$$

Since $\|\mathbf{r}(\mathbf{x}_{\text{True}})\|$ can be expressed as

$$\|\mathbf{r}(\mathbf{x}_{\text{True}})\| = \sqrt{\sum_{k=0}^P \|\mathbf{r}_k(\mathbf{x}_{\text{True}})\|^2}, \quad (\text{B22})$$

we obtain

$$\|\mathbf{r}(\mathbf{x}_{\text{True}})\| \leq e^M P^M R_1^M \rho^{P+1} \sqrt{\sum_{k=P+2-M}^P \left(\sum_{m=P+2-k}^M \|\mathbf{F}_m\| \right)^2}. \quad (\text{B23})$$

Reindexing the sum as $j = m + k - (P + 1)$ yields

$$\|r(x_{\text{True}})\| \leq e^M P^M R_1^M \rho^{P+1} \sqrt{\sum_{j=2}^M \left(\sum_{m=j}^M \|F_m\| \right)^2}. \quad (\text{B24})$$

If we denote by μ a constant independent of P , we finally obtain

$$\|r(x_{\text{True}})\| \leq \mu P^M \rho^{P+1}. \quad (\text{B25})$$

This indicates the value $\|r(x_{\text{True}})\|$ decreases exponentially fast as P increases, provided that $\rho < 1$.

We are now in position to derive an upper bound for the error $E(t)$ in Eq. (B11). Combining Eq. (B25) with Eq. (B11), we obtain

$$E(t) \leq \mu P^M \rho^{P+1} \int_0^t \|e^{C\tau}\| d\tau. \quad (\text{B26})$$

Among this obtained upper bound, the integral part represents the cumulative growth of the spectral norm of the time evolution operator e^{Ct} generated by the matrix C obtained through PSC method. Since the behavior of this integral strongly depends on the properties of the dynamical system under consideration and selection of the pivot state s , it is generally difficult to evaluate it. However, we can still discuss the growth conditions on $\|e^{Ct}\|$ under which both the convergence of the error $E(t)$ to zero and the characterization of its rate of convergence are ensured as P increases. If the integral part satisfies the following growth bound:

$$\int_0^t \|e^{C\tau}\| d\tau = \mathcal{O}(\xi_t^P) \quad (\text{B27})$$

for some time-dependent constant ξ_t , then Eq. (B26) implies

$$E(t) = \mathcal{O}((\rho \xi_t)^P P^M). \quad (\text{B28})$$

This indicates that, if $\rho \xi_t < 1$, the error $E(t)$ can be arbitrary small by increasing P , and its rate of decrease is exponential.

In addition, the time dependence of the constant ξ_t can be estimated under a mild assumption. Suppose that the matrix C is diagonalizable and that all of its eigenvalues have nonpositive real parts. (Note that when the real part of an eigenvalue is positive, the time-evolution operator e^{Ct} typically diverges, so it is not necessary to consider such cases.) Under this assumption, it can be shown that

$$\int_0^t \|e^{C\tau}\| d\tau < \kappa t, \quad (\text{B29})$$

where κ is a condition number of C . Therefore, there exists time-independent constant ξ that satisfies

$$E(t) \leq \mathcal{O}((\rho \xi)^P P^M t). \quad (\text{B30})$$

This means that the error increases linearly with time t , and that its coefficient can be made exponentially small in P as long as $\rho \xi < 1$.

For completeness, we provide a brief proof of Eq. (B29) below. Since C is diagonalizable, it can be written as $C = V \Lambda V^{-1}$, where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ and n is the size of the matrix C . Hence,

$$e^{Ct} = V e^{\Lambda t} V^{-1}. \quad (\text{B31})$$

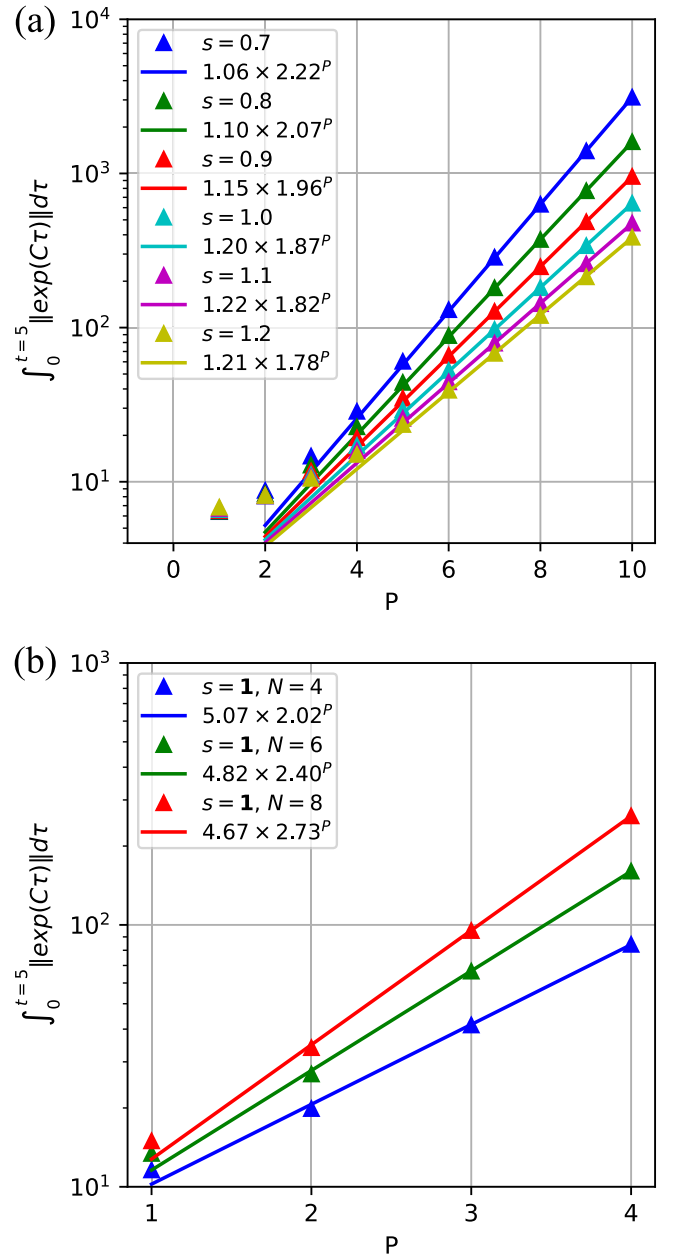


FIG. 6. (a) Numerically computed values of $\int_0^5 \|e^{C\tau}\| d\tau$ for the logistic function with various choices of the approximation order P and the pivot state s , together with the fitted curves for estimating ξ_t . (b) Numerically computed values of $\int_0^5 \|e^{C\tau}\| d\tau$ for the KPP-Fisher equation with the different approximation orders P and state dimensions n , along with the fitted curves.

Because all eigenvalues satisfy $\text{Re}[\lambda_i] \leq 0$, we have, for any $\tau \geq 0$,

$$\|e^{\Lambda \tau}\| \leq e^{\max \text{Re}[\lambda_i] \tau} \leq 1. \quad (\text{B32})$$

Therefore,

$$\|e^{Ct}\| \leq \|V\| \|e^{\Lambda t}\| \|V^{-1}\| \leq \|V\| \|V^{-1}\| = \kappa. \quad (\text{B33})$$

Consequently,

$$\int_0^t \|e^{C\tau}\| d\tau \leq \kappa t. \quad (\text{B34})$$

APPENDIX C: NUMERICALLY SUPPORT OF ANALYTICAL ERROR BOUNDS

To apply the above error bound to a specific system, it is necessary to estimate the quantities ρ and ξ_t appearing in Eq. (B28). Here, we present numerical estimates of ρ and ξ_t for logistic equation, a one-dimensional system used in the main text.

First, consider $\rho = \max |x - s|$. If the initial state is x_0 , then the state x takes values between x_0 and 1. Therefore, the value of ρ can be upper bounded as

$$\rho = \max(|1 - s|, |x_0 - s|). \quad (\text{C1})$$

Hence, ρ becomes small when the initial state x_0 is close to pivot state s and the pivot state s is close to 1.

Next, we estimate ξ_t . Even when we consider only the logistic equations, it is difficult to analytically bound the value of ξ_t for general values of s and P . Therefore, we evaluate ξ_t numerically. Figure 6(a) shows the numerically computed values of $\int_0^t \|e^{C\tau}\| d\tau$ at $t = 5$, for $P = 2, 3, \dots, 10$ and $s = 0.7, 0.8, \dots, 1.2$. From this figure, we first observe that this integral grows exponentially with increasing P . By fitting data to a function of the form $\alpha \xi_t^P$, we obtain a numerical estimate of α and ξ_t . This fitted curve is also plotted in the figure.

The fitting results indicate that although ξ_t tends to decrease as s increases, its variation remains relatively small and

its values lie between 1.7 and 2.2. Thus, in this case, when ρ is less than approximately 0.5, the condition $\rho \xi_t < 1$ is satisfied, and Eq. (B28) guarantees that the error $E(t)$ converges to zero exponentially as P increases.

As a point to note, in the numerical experiments presented in the main text, we used $x_0 = 0.1$ and $s = 1$, for which we observed the PSC method still exhibited convergence with respect to P . From Fig. 6(a), the corresponding values are $\xi_t = 1.87$ and $\rho = 0.9$, yielding $\rho \xi_t > 1$. This is because Eq. (B28) provides only a loose upper bound on the error, and thus actual convergence occurs over a broader range of ξ_t, ρ .

As an example of a system have a multidimensional state, Fig. 6(b) shows the numerically obtained value of the integral value $\int_0^t \|e^{C\tau}\| d\tau$ at $t = 5$ for the KPP-Fisher equation, which was also used in the numerical experiments in the main text. While the main text considered the case with state dimension $n = 8$, we additionally examined the cases $n = 4, 6, 8$. For constructing C matrix using the PSC method, all conditions other than the choice of n were kept the same as in the original experiments. We evaluated the cases $P = 1, 2, 3, 4$ for each value of n .

As shown in Fig. 6(b), the integral value exhibits exponential growth with respect to P , which is behavior similar to that observed for the logistic equation. The estimated values of ξ_t obtained by curve fitting, together with the fitted curves, are also shown in the figure. From these results, the growth of ξ_t remains sublinear with respect to n .

-
- [1] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation* (Academic Press, New York, 2000).
 - [2] S. Manabe and R. T. Wetherald, Thermal equilibrium of the atmosphere with a given distribution of relative humidity, *J. Atmos. Sci.* **24**, 241 (1967).
 - [3] P. M. Cox, R. A. Betts, C. D. Jones, S. A. Spall, and I. J. Totterdell, Acceleration of global warming due to carbon-cycle feedbacks in a coupled climate model, *Nature (London)* **408**, 184 (2000).
 - [4] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychol. Rev.* **65**, 386 (1958).
 - [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Adv. Neural Inf. Process. Syst.* **25**, 1097 (2012).
 - [6] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Elsevier, San Diego, 2023).
 - [7] J. H. Ferziger, M. Perić, and R. L. Street, *Computational Methods for Fluid Dynamics* (Springer, Cham, 2019).
 - [8] H. W. Hethcote, The mathematics of infectious diseases, *SIAM Rev.* **42**, 599 (2000).
 - [9] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering* (CRC Press, Boca Raton, 2018).
 - [10] A. Steane, Quantum computing, *Rep. Prog. Phys.* **61**, 117 (1998).
 - [11] T. E. O'Brien, B. Tarasinski, and B. M. Terhal, Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments, *New J. Phys.* **21**, 023022 (2019).
 - [12] H. Alghassi, A. Deshmukh, N. Ibrahim, N. Robles, S. Woerner, and C. Zoufal, A variational quantum algorithm for the Feynman-Kac formula, *Quantum* **6**, 730 (2022).
 - [13] R. Babbush, D. W. Berry, R. Kothari, R. D. Somma, and N. Wiebe, Exponential quantum speedup in simulating coupled classical oscillators, *Phys. Rev. X* **13**, 041041 (2023).
 - [14] J. Gonzalez-Conde, Á. Rodríguez-Rozas, E. Solano, and M. Sanz, Efficient Hamiltonian simulation for solving option price dynamics, *Phys. Rev. Res.* **5**, 043220 (2023).
 - [15] S. Jin, N. Liu, and Y. Yu, Quantum simulation of partial differential equations via Schrödingerization, *Phys. Rev. Lett.* **133**, 230602 (2024).
 - [16] S. Jin, N. Liu, and Y. Yu, Quantum simulation of partial differential equations: Applications and detailed analysis, *Phys. Rev. A* **108**, 032603 (2023).
 - [17] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, *Quantum Inf. Comput.* **12**, 901 (2012).
 - [18] D. An, A. M. Childs, and L. Lin, Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters, *Commun. Math. Phys.* **407**, 19 (2026).
 - [19] D. An, J.-P. Liu, and L. Lin, Linear combination of Hamiltonian simulation for nonunitary dynamics with optimal state preparation cost, *Phys. Rev. Lett.* **131**, 150603 (2023).
 - [20] M. Kieferová, A. Scherer, and D. W. Berry, Simulating the dynamics of time-dependent Hamiltonians with a truncated Dyson series, *Phys. Rev. A* **99**, 042314 (2019).

- [21] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian dynamics with a truncated Taylor series, *Phys. Rev. Lett.* **114**, 090502 (2015).
- [22] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, *Phys. Rev. Lett.* **118**, 010501 (2017).
- [23] V. Balasubramanian, M. DeCross, A. Kar, and O. Parrikar, Quantum complexity of time evolution with chaotic Hamiltonians, *J. High Energy Phys.* **01** (2020) 134.
- [24] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, Variational ansatz-based quantum simulation of imaginary time evolution, *npj Quantum Inf.* **5**, 75 (2019).
- [25] Z.-J. Zhang, J. Sun, X. Yuan, and M.-H. Yung, Low-depth Hamiltonian simulation by an adaptive product formula, *Phys. Rev. Lett.* **130**, 040601 (2023).
- [26] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, *Quantum* **3**, 163 (2019).
- [27] T. Kharazi, A. M. Alkadri, J.-P. Liu, K. K. Mandadapu, and K. B. Whaley, Explicit block encodings of boundary value problems for many-body elliptic operators, *Quantum* **9**, 1764 (2025).
- [28] S. Jin and N. Liu, Analog quantum simulation of parabolic partial differential equations using Jaynes-Cummings-like models, [arXiv:2407.01913](https://arxiv.org/abs/2407.01913).
- [29] D. Lee, S. Shon, S. Lee, and J. Ha, Size and topology optimization of truss structures using quantum-based HS algorithm, *Buildings* **13**, 1436 (2023).
- [30] B. Liu, M. Ortiz, and F. Cirak, Towards quantum computational mechanics, *Comput. Methods Appl. Mech. Eng.* **432**, 117403 (2024).
- [31] D. Wawrzyniak, J. Winter, S. Schmidt, T. Indinger, C. F. Janßen, U. Schramm, and N. A. Adams, A quantum algorithm for the lattice-Boltzmann method advection-diffusion equation, *Comput. Phys. Commun.* **306**, 109373 (2025).
- [32] R. P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 467 (1982).
- [33] S. Lloyd, G. De Palma, C. Gokler, B. Kiani, Z.-W. Liu, M. Marvian, F. Tennie, and T. Palmer, Quantum algorithm for nonlinear differential equations, [arXiv:2011.06571](https://arxiv.org/abs/2011.06571).
- [34] I. Joseph, Koopman–von Neumann approach to quantum simulation of nonlinear classical dynamics, *Phys. Rev. Res.* **2**, 043102 (2020).
- [35] D. Shi and X. Yang, Koopman spectral linearization vs. Carleman linearization: A computational comparison study, *Mathematics* **12**, 2156 (2024).
- [36] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs, Efficient quantum algorithm for dissipative nonlinear differential equations, *Proc. Natl. Acad. Sci. USA* **118**, e2026805118 (2021).
- [37] C. Sanavio, R. Scatamacchia, C. de Falco, and S. Succi, Three Carleman routes to the quantum simulation of classical fluids, *Phys. Fluids* **36**, 057143 (2024).
- [38] W. Itani, K. R. Sreenivasan, and S. Succi, Quantum algorithm for lattice Boltzmann (QALB) simulation of incompressible fluids with a nonlinear collision term, *Phys. Fluids* **36**, 017112 (2024).
- [39] W. Itani and S. Succi, Analysis of Carleman linearization of lattice Boltzmann, *Fluids* **7**, 24 (2022).
- [40] IBM quantum computers compute resources, <https://quantum.cloud.ibm.com/computers>, accessed on July 21, 2025.
- [41] IonQ quantum systems forte, <https://ionq.com/quantum-systems/forte>, accessed on July 21, 2025.
- [42] Fire opal, <https://docs.q-ctrl.com/fire-opal>, accessed on July 21, 2025.
- [43] J. Dressel, M. Malik, F. M. Miatto, A. N. Jordan, and R. W. Boyd, Colloquium: Understanding quantum weak values: Basics and applications, *Rev. Mod. Phys.* **86**, 307 (2014).
- [44] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, *Contemp. Math.* **305**, 53 (2002).
- [45] <https://github.com/Katsuhiko-Endo/PivotSwitchingCarleman>