Quantum simulation of partial differential equations: Applications and detailed analysis

Shi Jin,[1,2,*] Nana Liu,[1,2,3,†] and Yue Yu[1,‡]

[1]*School of Mathematical Sciences, Institute of Natural Sciences, MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, China*

[2]*Shanghai Artificial Intelligence Laboratory, Shanghai 200240, China*

[3]*University of Michigan–Shanghai Jiao Tong University Joint Institute, Shanghai 200240, China*

(Received 9 March 2023; accepted 20 July 2023; published 12 September 2023)

We study a recently introduced simple method [S. Jin, N. Liu, and Y. Yu, Quantum simulation of partial differential equations via Schrödingerisation, arXiv:2212.13969] for solving general linear partial differential equations with quantum simulation. This method converts linear partial differential equations into a Hamiltonian system, using a simple transformation called the warped phase transformation. Here we provide a more-in-depth technical discussion and expand on this approach in a more detailed and pedagogical way. We apply this to examples of partial differential equations, including heat, convection, Fokker-Planck, linear Boltzmann, and Black-Scholes equations. This approach can also be extended to general linear partial differential equations, including the Vlasov-Fokker-Planck equation and the Liouville representation equation for nonlinear ordinary differential equations. Extension to higher-order time derivatives is also possible.

DOI: 10.1103/PhysRevA.108.032603

I. INTRODUCTION

Quantum algorithms for solving partial differential equations (PDEs) have received extensive attention in recent years [1–14]. This is due to the fact that many classical methods for solving PDEs suffer from the curse of dimensionality, whereas quantum methods are observed to be less costly due, for instance, to the development of quantum algorithms with up to exponential advantage in linear algebraic problems [4,7,12,15–19]. For time-dependent PDEs, one usually discretizes the spatial variables to get a system of ordinary differential equations (ODEs), which in turn is solved by quantum ODE solvers [4,7,15]. In particular, when the resulting ODE is also a Hamiltonian system, quantum simulation methods can be performed. In general, quantum simulations have less time complexity than quantum ODE solvers or other quantum linear algebra solvers (e.g., the quantum difference methods [4,12]) and thus the design for quantum simulation algorithms for solving linear PDEs is important for a wide range of applications. A very recent proposal is based on block encoding [20].

This paper presents more-in-depth technical details for a protocol that transforms a general linear PDE into a quantum Hamiltonian system. We call this method the Schrödingerization method, introduced in our paper in [21]. From the simplest example of the heat equation, for instance, a corresponding set of Schrödinger equations can be derived. In Sec. II we present the heat equation example in detail, as an example to familiarize the reader with the technique. We also present the example of the convection equation.

This method is inspired by a recently developed transformation given in [10], though originally for a completely different motivation. Here we introduce an auxiliary variable and construct a transformation, referred to as the warped phase transformation, that converts the original equation into an equation that has the structure of the Schrödinger operator. Then it can subsequently be simulated by quantum Hamiltonian simulation. Since the method introduces only a one-dimensional auxiliary variable, the additional computational cost is small.

When discretizing a general linear PDE by spectral methods or other numerical methods, a Hamiltonian system is not necessarily obtained. There are two reasons for this: One is that the coefficient matrix arising from each term of the equation is not always preceded by the imaginary number $i = \sqrt{-1}$ and the other is that despite the imaginary number, the coefficient matrix is usually not symmetric, especially for problems with variable coefficients or for the discretization schemes that are not centered (with a symmetric stencil).

We observe that applying the warped phase transformation works for equations with constant coefficients and it also works for some variable-coefficient problems. However, for general variable-coefficient problems, the direct use of the method does not necessarily achieve the goal of getting a Hamiltonian system. For example, see the Vlasov-Fokker-Planck equation discussed in Sec. IV D.

For this reason, we further design a universal algorithm for the system of linear ODEs based on warped transformation, where the system of ODEs is obtained after spatial discretizations of any linear PDEs. In Sec. III we introduce an algorithm for this general linear system of ODEs. This idea works for ODEs resulting from spatial discretizations for all constant-coefficient and even some variable-coefficient PDEs. We also discuss some alternative methods to our approach.

*shijin-m@sjtu.edu.cn

†nana.liu@quantumlah.org

‡terenceyuyue@sjtu.edu.cn

We remark that this approach will also find a variety of applications in solving problems with a (time-dependent or -independent) source term and boundary-value problems (with time-dependent or -independent boundary conditions). In fact, when constructing Hamiltonian systems for general boundary-value problems (for example, with the Dirichlet boundary condition), the inhomogeneous right-hand side in the resulting ODE system may arise after spatial discretization of the boundary conditions for the PDEs. We propose a simple augmentation technique (see Remark 6 and [22,23]) to resolve this issue, which together with the warped phase transformation allows our method to be applied to time-dependent boundary-value problems. This approach has also been recently applied to linear algebra problems like the linear systems of equations and finding maximum eigenvalues and eigenvectors [24].

In Sec. IV we show that our method is applicable to a variety of important linear partial differential equations, including the linear heat, convection, Vlasov-Fokker-Planck, linear Boltzmann, and Black-Scholes equations. For nonlinear problems, we give an application via the linear representation (Liouville representation) approach for nonlinear dynamical systems. It is worth pointing out that the Liouville representation can be symmetrized using the Koopman–von Neumann representation [11,25–27], but it involves the square root of the Dirac delta function $\delta(x)$, which is not well defined mathematically, even in the weak sense. Thus one needs to be more careful in interpreting its solution and the consequent numerical convergence in a suitable solution space [26]. Our approach allows a direct treatment of the Liouville equation and hence does not have difficulties in this regard.

II. QUANTUM SIMULATION OF THE HEAT AND CONVECTION EQUATIONS

A. The heat equation rewritten as a system of Schrödinger equations

This section demonstrates how to transform a linear heat equation into Schrödinger-type PDEs.

1. Heat equation

Consider the initial-value problem of the linear heat equation

$$\partial_t u - \Delta u = 0,$$
$$u(0,x) = u_0(x),$$

where $u = u(t,x)$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ is the position, and $t \geqslant 0$. Introduce an auxiliary variable $p > 0$ and define

$$w(t,x,p) = e^{-p} u(t,x), \quad p > 0.$$

A simple calculation shows that w solves

$$\partial_t w + \partial_p \Delta_x w = 0, \quad p > 0. \tag{1}$$

From w one can recover u via

$$u(t,x) = \int_0^\infty w(t,x,p)dp = \int_{-\infty}^\infty \chi(p) w(t,x,p)dp, \tag{2}$$

where $\chi(p) = 1$ for $p > 0$ and $\chi(p) = 0$ for $p < 0$ or, since $u(t,x) = e^p w(t,x,p)$ for all $p > 0$, one can simply choose any $p_* > 0$ and let

$$u(t,x) = e^{p_*} w(t,x,p_*). \tag{3}$$

Applying the Fourier transform on x (here we assume x is defined in a periodic domain) and letting $\hat{w}(t,\xi,p)$, with $\xi = [\xi_1, \ldots, \xi_d]^T$ the Fourier modes, be the corresponding Fourier transform of w, one gets a convection equation

$$\partial_t \hat{w} - |\xi|^2 \partial_p \hat{w} = 0,$$

where $|\xi|^2 = \xi_1^2 + \cdots + \xi_d^2$. Clearly, the solution $\hat{w}$ moves from the right to the left, so no boundary condition is needed at $p = 0$, while the initial data of w are given by

$$w(0,x,p) = e^{-p} u_0(x), \quad p > 0.$$

Moreover, if we extend w to $p < 0$, then the solution does not impact the region $p > 0$ for w. For this reason, we symmetrically extend the initial data of w to $p < 0$ but keep Eq. (1):

$$\partial_t w + \Delta_x \partial_p w = 0, \quad p \in (-\infty, \infty),$$
$$w(0,x,p) = e^{-|p|} u_0(x). \tag{4}$$

This equation for w will be called the phase-space heat equation. The solution obviously coincides with the solution of (1) when $p > 0$. Due to the exponential decay of $e^{-|p|}$, one can (computationally) impose the periodic boundary condition $w(t,x,p = -L) = w(t,x,p = L)$ (=0) in the p direction for some $L > 0$ sufficiently large. Then the Fourier transform on p gives

$$\partial_t \tilde{w} - i\eta \Delta \tilde{w} = 0 \text{ or } i\partial_t \tilde{w} = -\eta \Delta \tilde{w}, \tag{5}$$

where $\tilde{w}(t,x,\eta)$, $\eta \in \mathbb{R}$, is the Fourier transform of w in p. Equation (5) is clearly the Schrödinger equation, for every η.

This means that $\tilde{w}$ evolves according to $\tilde{w}(t,x,\eta) = \exp(-it\eta\Delta)\tilde{w}(0,x,\eta)$. If we embed the values of $\tilde{w}(0,x,\eta)$ into the amplitudes of a quantum state $|\tilde{w}(0)\rangle$, then we can recover $|\tilde{w}(t)\rangle = \exp(-i\hat{p}^2 \otimes \hat{\eta})|\tilde{w}(0)\rangle$ with a unitary quantum circuit $\exp(-i\hat{p}^2 \otimes \hat{\eta})$, where $\hat{p}$ is the momentum operator and $\hat{\eta}$ is another quadrature operator with eigenvalue η. In a more general scenario, instead of using Δ, we can have some other Hamiltonian H, so we have a more general unitary operator $\exp(-iH \otimes \hat{\eta}t)$. The preparation of $|\tilde{w}(0)\rangle$ is straightforward. When given $|u(0)\rangle$, we form a product state with the auxiliary mode proportional to $\int dp \exp(-|p|)|p\rangle$ and then take a quantum Fourier transform with respect to the auxiliary mode. After application of $\exp(-iH \otimes \hat{\eta}t)$ we have $|\tilde{w}(t)\rangle$. To recover $|u(t)\rangle$, we apply an inverse quantum Fourier transform to the auxiliary mode and obtain $|w(t)\rangle$. Finally, from Eq. (2) we see we need only the $p > 0$ parts of the auxiliary mode to recover $|u(t)\rangle$, so in the simplest setting we can apply $\hat{P}$ to the auxiliary mode to project out only the $p > 0$ parts (see Fig. 1).

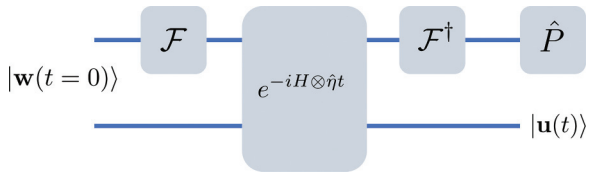

FIG. 1. Quantum circuit for the preparation of $|\boldsymbol{u}(t)\rangle$.

For a fully continuous-variable formulation of Schrödingerization, with H also a continuous-variable operator, see [28].

Remark 1. One can justify the validity of the setup (4) (which is more convenient than the half-space problem for w if one uses, for example, spectral methods in p space) in another way. Applying the Fourier transform on x to (4), one gets

$$\partial_t \hat{w}_t(t,\xi,p) - |\xi|^2 \partial_p \hat{w}(t,\xi,p) = 0, \quad p \in (-\infty,\infty),$$
$$\hat{w}(0,\xi,p) = e^{-|p|}\hat{u}_0(\xi).$$

Using the method of characteristics, the analytic solution is given by

$$\hat{w}(t,\xi,p) = \hat{w}(0,p+|\xi|^2 t) = e^{-|p+|\xi|^2 t|}\hat{u}_0(\xi). \tag{6}$$

If $p > 0$, then

$$\hat{w}(t,\xi,p) = e^{-|p+|\xi|^2 t|}\hat{u}_0 = e^{-p}[e^{-|\xi|^2 t}\hat{u}_0(\xi)].$$

The inverse transform implies

$$w(t,p) = e^{-p}\mathcal{F}^{-1}(e^{-|\xi|^2 t}\hat{u}_0(\xi)) = e^{-p}u(t,x),$$

where $\mathcal{F}$ represents the Fourier transform. This is exactly the solution of (1).

From (6) one sees that $|\hat{w}(t,\xi,p)| \leqslant |\hat{u}_0(\xi)|$. Note that if $u_0(x) \in C^k$, then $\hat{u}_0(\xi) \sim O(1/|\xi|^k)$. Therefore, if $u_0(x)$ is sufficiently smooth, $\hat{w}$ decays very fast in ξ, which means only those ξ such that $|\xi| = O(1)$ are important; hence $\hat{w}$ will move to the left with $O(1)$ speed. Thus, for $T = O(1)$, $|L| = O(1)$ is sufficient for the computational domain of $p \in [L,R]$, where $L < 0$ and $R > 0$.

From Remark 1 and Eq. (4) one sees easily that

$$\int_{-\infty}^{\infty}\int_{0}^{\infty} w(t,x,p)^2 dp\,dx = \frac{1}{2}\int_{-\infty}^{\infty} u(t,x)^2 dp\,dx,$$

$$\int_{-\infty}^{\infty}\int_{\infty}^{\infty} w(t,x,p)^2 dp\,dx = \int_{-\infty}^{\infty} u_0(x)^2 dx.$$

Standard PDE analysis using Poincaré's inequality gives

$$\int_{-\infty}^{\infty}\int_{0}^{\infty} w(t,x,p)^2 dp\,dx \leqslant e^{-\alpha t}\int_{-\infty}^{\infty}\int_{\infty}^{\infty} w(t,x,p)^2 dp\,dx \tag{7}$$

for some positive constant α that depends on the (finite) domain size in x.

Example 1. We conduct a numerical test in one dimension to justify the above idea:

$$u_t - u_{xx} = 0, \quad x \in (-1,1),$$
$$u(0,x) = u_0(x),$$
$$u(t,-1) = u(t,1), \quad u_x(t,-1) = u_x(t,1).$$

The exact solution is given by $u(t,x) = e^{-\pi^2 t}\sin(\pi x)$.

For numerical implementation, it is natural and convenient to introduce $\alpha = \alpha(p)$ in the initial data of (4) for $p < 0$:

$$\partial_t w + \partial_{xxp} w = 0, \quad p \in (-\infty,\infty),$$
$$w(0,x,p) := w_0(x,p) = e^{-\alpha|p|}u_0(x). \tag{8}$$

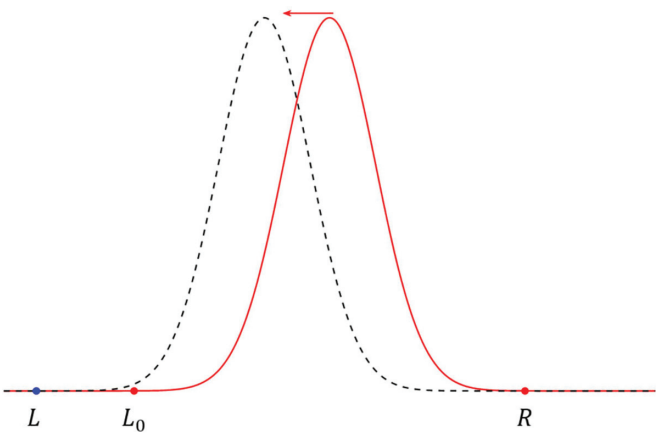

FIG. 2. Schematic diagram for the computational domain of p.

To match the exact solution, $\alpha(p) = 1$ is necessary for the region $p > 0$. In the $p > 0$ domain, we will truncation the domain at $p = R$, where R is sufficiently large such that $e^{-R} \approx 0$. We will choose a large α for $p < 0$ so the solution (see Fig. 2) will have a support within a relatively small domain. Since the wave $\hat{w}$ moves to the left, one needs to choose the artificial boundary at $p = L < 0$ for $|L|$ large enough such that $\hat{w}$, initially almost compact at $[L_0,R]$, will not reach the point $p = L$ throughout the duration of the computation. This will allow one to use periodic boundary condition in p for the spectral approximation.

The Fourier spectral approach will be used to discretize the spatial and the auxiliary variables. Let M and N be two even numbers. We choose uniform mesh sizes $\Delta x = 2/M$ and $\Delta p = (R-L)/N$ for the spatial and the auxiliary variables, with the grid points denoted by $x_0 < x_1 < \cdots < x_M$ and $p_0 < p_1 < \cdots < p_N$, respectively. Let $\boldsymbol{w}(t,p) = [w(t,x_0,p), w(t,x_1,p), \ldots, w(t,x_{M-1},p)]^T$. The discrete Fourier transform (DFT) on x gives

$$\partial_t \boldsymbol{w}(t,p) - P_\mu^2 \partial_p \boldsymbol{w}(t,p) = 0, \quad p \in (L,R),$$
$$\boldsymbol{w}(0,p) = e^{-\alpha|p|}\boldsymbol{u}_0, \tag{9}$$

where $\boldsymbol{u}_0 = [u(0,x_1),\ldots,u(0,x_{M-1})]^T$ and P_μ is the matrix representation of the momentum operator $-i\partial_x$ in the original variables. For details on the derivation of (9), refer to the next section for notation. The matrix P_μ can be diagonalized via $D_\mu = \Phi^{-1}P_\mu\Phi$, where $D_\mu = \mathrm{diag}(\mu_{-M/2},\ldots,\mu_{M/2-1})$ is a diagonal matrix with $\mu_l = \pi l$ for $l = -M/2,\ldots,M/2-1$. Let $\hat{\boldsymbol{w}}(t,p) = \Phi^{-1}\boldsymbol{w}(t,p)$. Then one has

$$\partial_t \hat{\boldsymbol{w}}(t,p) - D_\mu^2 \partial_p \hat{\boldsymbol{w}}(t,p) = 0, \quad p \in (L,R),$$
$$\hat{\boldsymbol{w}}(0,p) = e^{-\alpha|p|}\hat{\boldsymbol{u}}_0, \tag{10}$$

where $\hat{\boldsymbol{u}}_0 = \Phi^{-1}\boldsymbol{u}_0$ and the lth component of $\hat{\boldsymbol{w}}$, denoted by $\hat{\boldsymbol{w}}_l$, corresponds to a linear hyperbolic system and the wave moves from the right to the left with speed $s_l = \mu_l^2$. The analytic solution to (10) is obviously given by

$$\hat{\boldsymbol{w}}_l(t,p) = e^{-\alpha(p+s_l t)|p+s_l t|}\hat{\boldsymbol{u}}_{0,l}, \tag{11}$$

where $l = -M/2,\ldots,M/2-1$. According to Remark 1, when $u_0(x)$ is sufficiently smooth, we know that the fastest left-moving wave will have a speed $s_* = O(1)$. Given the

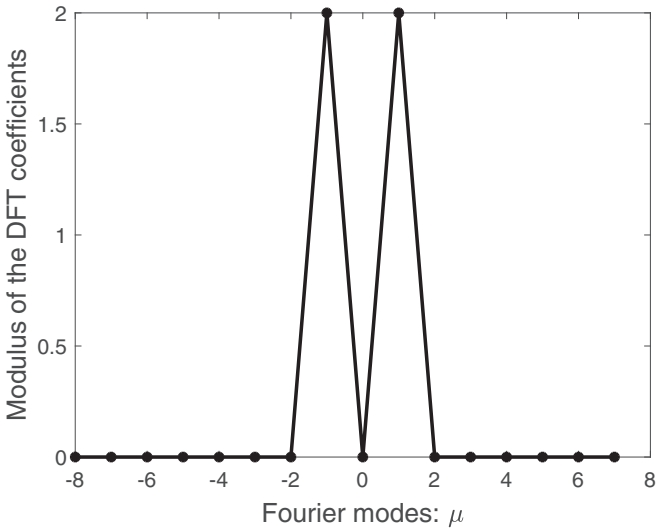

FIG. 3. Discrete Fourier transform of the initial data $u_0(x) = \sin(\pi x)$.

evolution time T, we can estimate a large enough $|L|$ such that $|s_* T| < |L|$.

Now we consider the full discretization. Let the grid value matrix $W(t) := (w(t, x_i, p_j))_{M\times N}$, which can be straightened as

$$\boldsymbol{w}(t) = [\boldsymbol{w}_0; \boldsymbol{w}_1; \ldots; \boldsymbol{w}_{M-1}] = \sum_{ij} w(t, x_i, p_j)|i, j\rangle,$$

where the semicolon indicates the straightening of $\{\boldsymbol{w}_i\}_{i\geqslant 1}$ into a column vector and

$$\boldsymbol{w}_i = [w(t, x_i, p_0), w(t, x_i, p_1), \ldots, w(t, x_i, p_{N-1})]^T.$$

Performing the DFT on both x and p yields

$$\partial_t \boldsymbol{w}(t) - i\big(P_\mu^2 \otimes P_\mu\big)\boldsymbol{w}(t) = \boldsymbol{0}, \tag{12}$$

where we use the same denotation P_μ for both variables since no confusion will arise. Let F_x and F_p be the discrete Fourier transform matrices for x and p, respectively. One can translate Eq. (12) into the frequency space

$$\partial_t \boldsymbol{c}(t) - i\big(D_\mu^2 \otimes D_\mu\big)\boldsymbol{c}(t) = \boldsymbol{0},$$

where $\boldsymbol{c}(t) = (F_x^{-1} \otimes F_p^{-1})\boldsymbol{w}(t)$. If $\boldsymbol{c}(t)$ is arranged as a matrix $C(t)$ in the form of $W(t)$, then the relation $W = QCP^T = F_p C F_x^T$ or $C = Q^{-1}WP^{-T} = F_p^{-1}WF_x^{-T}$ is easily found, which avoids the use of memory-consuming tensor products. Therefore, the numerical realization can be effectively implemented via the discrete Fourier transform (see Remark 2). The solution $\hat{\boldsymbol{w}}_l(t, p)$ of (10) can be extracted from the lth row of $F_x^{-1}W$.

In the numerical test, we choose $M = 2^4 = 16$ and $N = 2^9 = 512$. In Fig. 3 we plot the modulus of the DFT coefficients $\hat{\boldsymbol{u}}_0$. Clearly, the amplitudes decay very fast in the Fourier mode μ_l, and $\mu_{\pm 1}$ contribute most to the propagation. Therefore, for this example one can choose time t such that

$$s_1 t \leqslant L_0 - L \text{ or } t \leqslant T_* := (L_0 - L)/s_1, \tag{13}$$

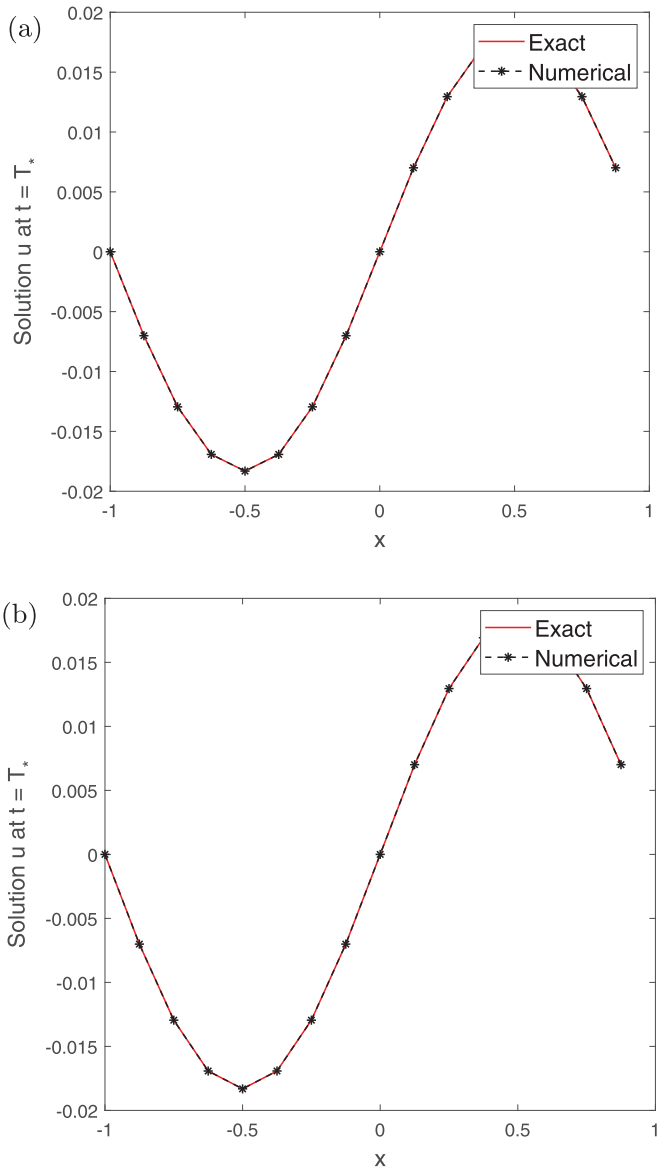

FIG. 4. Numerical and exact solutions of $\boldsymbol{u}(t = T_*)$ for the spectral method using (a) Eq. (3) and (b) Eq. (2).

where $s_1 = \pi^2 \in O(1)$. For other parameters, we choose $L = -5$, $R = 5$, $t = T_*$, $\alpha = 10$ for $p < 0$, and $L_0 = -1$ (the estimated $T_* = 0.4053$). The numerical solutions for $\boldsymbol{u}(t) = [u(t, x_0), \ldots, u(t, x_M)]^T$ are displayed in Fig. 4, with (3) and (2) used to restore the solutions, respectively. Note that it is better to pick the point $p_* > 0$ near $p = 0$, for example, $p_* = p_{N/2+3}$, to avoid the loss of significant digits of e^{-p*} for large p^*.

To validate the choice of computational domain shown in Fig. 2, we now take snapshots of the movement of the wave corresponding to Eq. (10), with the result shown in Fig. 5(a). Since the wave amplitude in the frequency space is a complex number and its real part is small, we display in this figure the modulus of these complex numbers, i.e., $|\hat{\boldsymbol{w}}_{l_*}(t, p_j)|$ for $j = 0, 1, \ldots, N-1$, where l_* corresponds to the speed s_*. The blue dashed line represents the initial wave $|\hat{\boldsymbol{w}}(0, p)|$ and the red circles and black solid lines are the analytic

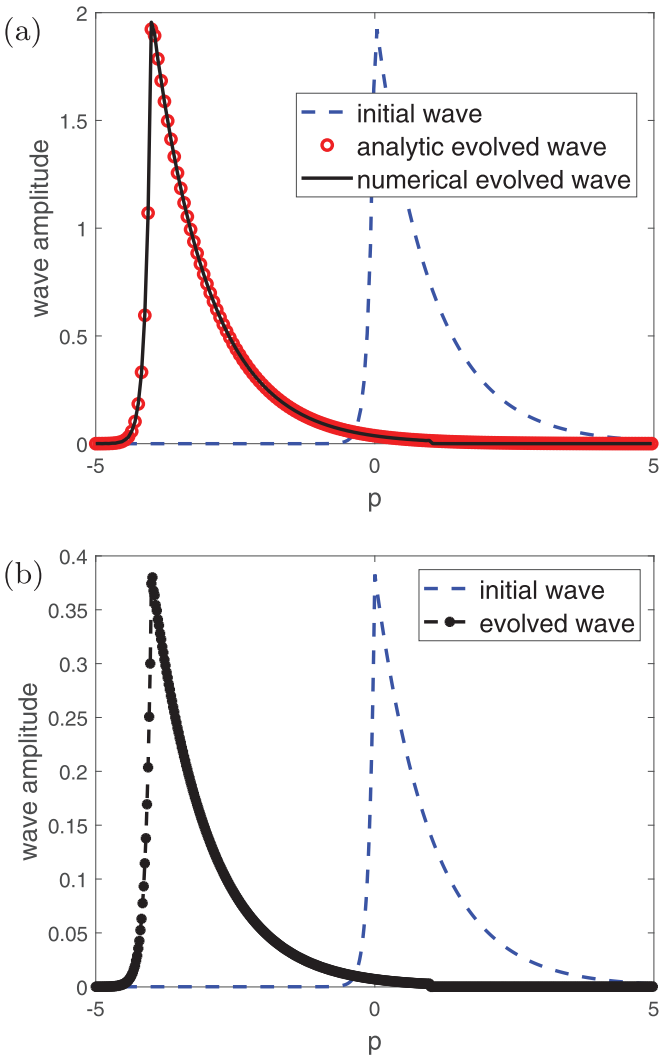

FIG. 5. Initial and evolved waves in the frequency and original spaces ($T_* = 0.4053$): (a) modulus of the wave amplitude $|\hat{\boldsymbol{w}}_{l_*}(t, p_j)|$ corresponding to (10) in the frequency space and (b) wave amplitude $\boldsymbol{w}_{i_*}(t, p_j)$ corresponding to (9) in the original space.

solution given by (11) and the numerical solution at time $t = T_*$, respectively. We also plot the wave amplitudes $\boldsymbol{w}_{i_*}(t, p_j)$ in the original space in Fig. 5(b), where i_* corresponds to local index of s_* in $\{s_i\}$. As observed, the waves in both spaces have almost moved to the left end, which validates the previous arguments.

Given $t = T = 1$, we can choose a large enough L in absolute value to get satisfactory numerical results. In view of the relation (13), we take $L = L_0 - Ts_1 \approx 11$. Considering the periodic condition, one can choose $\alpha = 40$ and $R = 10$, for example, with the results shown in Fig. 6.

One can also use the finite-difference discretization to further validate the above arguments. For the spatial discretization, we use the central difference to get

$$\partial_t w_i(t, p) + \partial_p \frac{w_{i-1}(t, p) - 2w_i(t, p) + w_{i+1}(t, p)}{\Delta x^2} = 0,$$
$$w_0(t, p) = w_M(t, p),$$

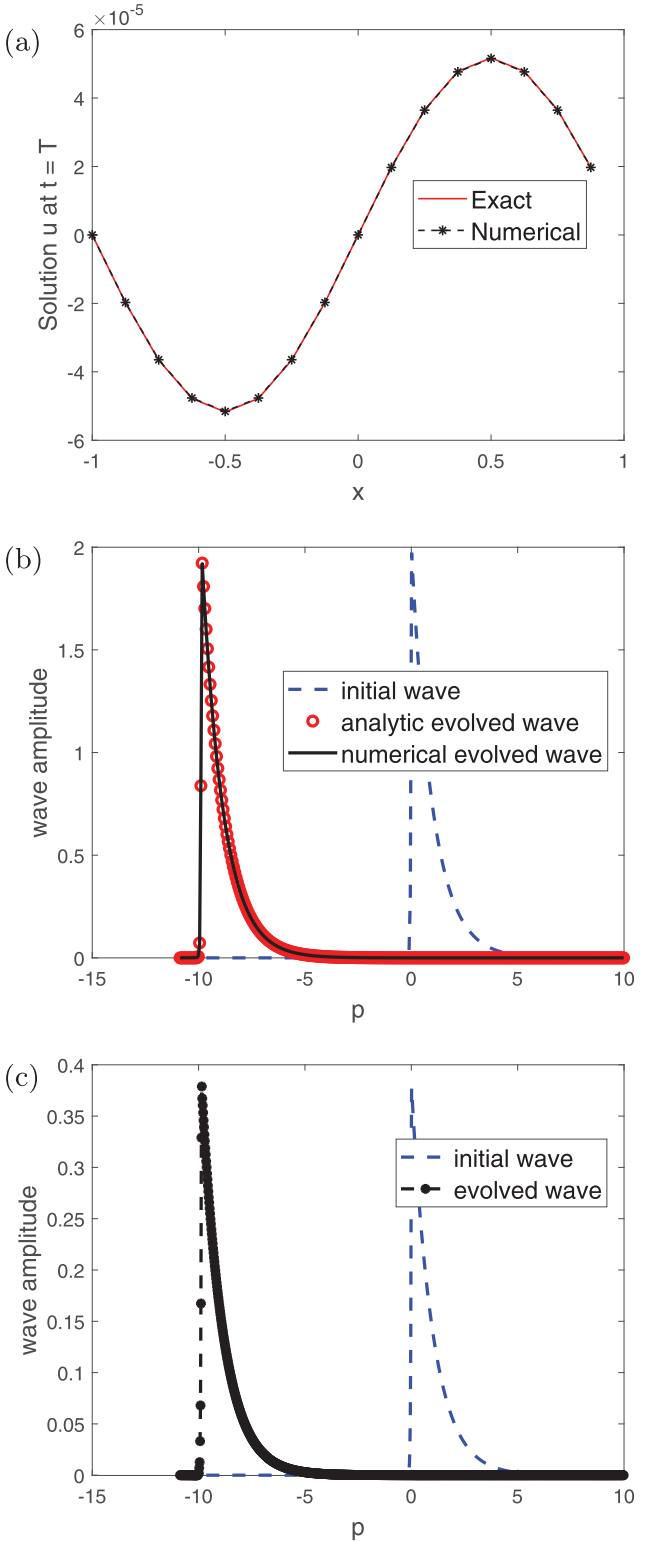

FIG. 6. Numerical results for given evolution time $t = T = 1$: (a) numerical solution for the heat equation recovered by (2), (b) modulus of the wave amplitude $|\hat{\boldsymbol{w}}_{l_*}(t, p_j)|$, and (c) wave amplitude $\boldsymbol{w}_{i_*}(t, p_j)$.

$$\frac{w_1(t, p) - w_{-1}(t, p)}{2\Delta x} = \frac{w_{M+1}(t, p) - w_{M-1}(t, p)}{2\Delta x}$$

for $i = 1, 2, \ldots, M-1$, where we have introduced the ghost points x_{-1} and x_{M+1}. To get a closed system, we assume the discretization is valid on $x = x_0$ and $x = x_M$:

$$\partial_t w_0(t,p) + \partial_p \frac{w_{-1}(t,p) - 2w_0(t,p) + w_1(t,p)}{\Delta x^2} = 0$$

at $i = 0$ and

$$\partial_t w_M(t,p) + \partial_p \frac{w_{M-1}(t,p) - 2w_M(t,p) + w_{M+1}(t,p)}{\Delta x^2} = 0$$

at $i = M$. Summing the above two equations and eliminating the ghost values

$$\partial_t w_0(t,p) + \partial_p \frac{-2w_0(t,p) + w_1(t,p) + w_{M-1}(t,p)}{\Delta x^2} = 0, \quad i = 0,$$

we then obtain the system

$$\partial_t \boldsymbol{w}(t,p) + A\partial_p \boldsymbol{w}(t,p) = \boldsymbol{0},$$

where

$$\boldsymbol{w}(t,p) = \begin{bmatrix} w_0(t,p) \\ w_1(t,p) \\ \vdots \\ w_{M-2}(t,p) \\ w_{M-1}(t,p) \end{bmatrix},$$

$$A = \frac{1}{\Delta x^2}\begin{bmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{bmatrix}.$$

One can check that the eigenvalues of A are $\lambda_k(A) = -\frac{4}{\Delta x^2}\sin^2\frac{k\pi}{M}$ for $k = 0, 1, \ldots, M-1$. Let $\boldsymbol{w}_j^n$ be the approximation to $\boldsymbol{w}(t_n, p_j)$. For the periodic boundary condition, the value $\boldsymbol{w}_0 = \boldsymbol{w}_N$ along the boundaries is unknown. We introduce $\boldsymbol{w}_N$ into the vector of grid values $\boldsymbol{W}^n = [\boldsymbol{w}_1^n; \ldots; \boldsymbol{w}_N^n]$, with the semicolon indicating the straightening of $\{\boldsymbol{w}_i\}_{i\geqslant 1}$ into a column vector. Since $\lambda_k \leqslant 0$, we adopt the upwind discretization on p and obtain

$$\frac{\boldsymbol{w}_j^{n+1} - \boldsymbol{w}_j^n}{\Delta t} + A\frac{\boldsymbol{w}_{j+1}^n - \boldsymbol{w}_j^n}{\Delta p} = \boldsymbol{0}, \quad j = 1, \ldots, N-1.$$

The above system is closed by assuming the discretization holds at $p = p_0$. In view of the periodicity, the additional equation can be written as

$$\frac{\boldsymbol{w}_N^{n+1} - \boldsymbol{w}_N^n}{\Delta t} + A\frac{\boldsymbol{w}_1^n - \boldsymbol{w}_N^n}{\Delta p} = \boldsymbol{0}, \quad j = N.$$

The final iterative system is

$$\boldsymbol{W}^{n+1} = B\boldsymbol{W}^n, \quad n = 0, 1, \ldots, N_t - 1,$$

with

$$\boldsymbol{B} = \begin{bmatrix} I + A_1 & -A_1 & & & \\ & I + A_1 & -A_1 & & \\ & & \ddots & \ddots & \\ & & & I + A_1 & -A_1 \\ -A_1 & & & & I + A_1 \end{bmatrix},$$

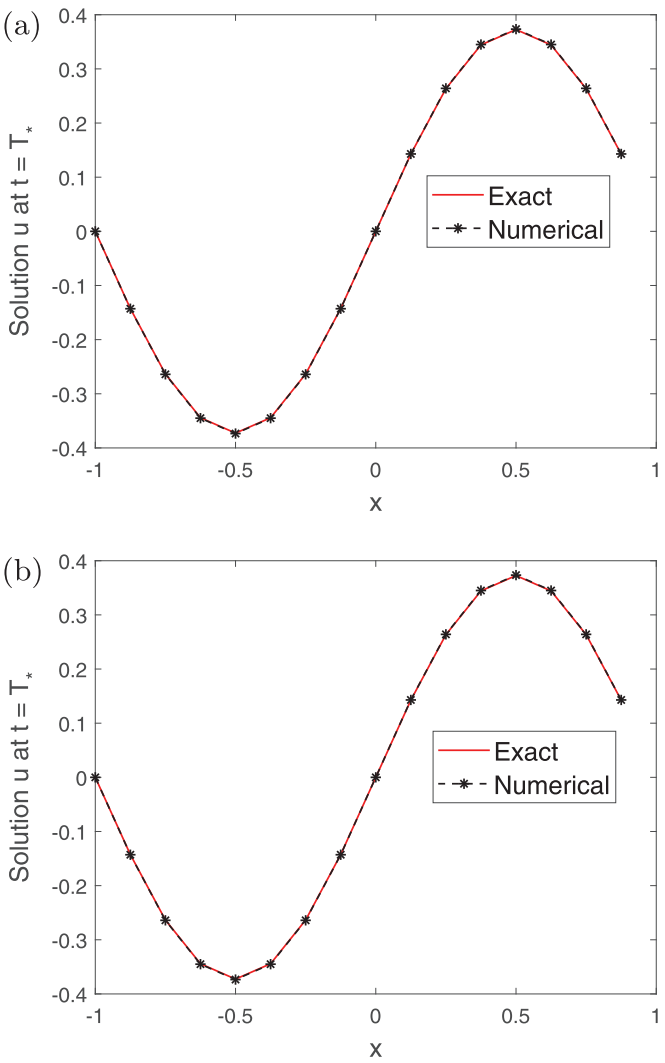

FIG. 7. Numerical and exact solutions of $\boldsymbol{u}(t = T_*)$ for the finite-difference method using (a) Eq. (3) and (b) Eq. (2).

$$A_1 = \frac{\Delta t}{\Delta p}A.$$

The numerical results are similar to that of the spectral method under the same settings, as shown in Fig. 7.

2. *Notation for the Fourier spectral discretization*

For the Fourier spectral discretization, we will use the notation given in [26]. For one-dimensional problems we choose a uniform spatial mesh size $\Delta x = (b-a)/M$ for $M = 2N = 2^m$, with m a positive integer, and time step Δt, and we let the grid points be

$$x_j = a + j\Delta x, \quad j = 0, 1, \ldots, M.$$

As an example we consider the periodic boundary conditions. For $x \in [a, b]$, the one-dimensional basis functions for the Fourier spectral method are usually chosen as

$$\phi_l(x) = e^{i\mu_l(x-a)}, \quad \mu_l = \frac{2\pi l}{b-a}, \quad l = -N, \ldots, N-1.$$

For convenience, we adjust the index as

$$\phi_l(x) = e^{i\mu_l(x-a)}, \quad \mu_l = \frac{2\pi(l-N-1)}{b-a},$$

where $1 \leqslant l \leqslant M = 2N$. The approximation to $u(x)$ in the one-dimensional space is

$$u(x) = \sum_{l=1}^{M} c_l \phi_l(x), \quad x = x_j,\ j = 0, 1, \ldots, M-1, \quad (14)$$

which can be written in vector form $\boldsymbol{u} = \Phi \boldsymbol{c}$, where $\boldsymbol{u} = [u(x_j)]_{M\times 1}$, $\boldsymbol{c} = (c_l)_{M\times 1}$, and $\Phi = (\phi_{jl})_{M\times M} = [\phi_l(x_j)]_{M\times M}$.

The d-dimensional grid points are given by $x_{\boldsymbol{j}} = (x_{j_1}, \ldots, x_{j_d})$, where $\boldsymbol{j} = (j_1, \ldots, j_d)$ and

$$x_{j_i} = a + j_i \Delta x, \quad j_i = 0, 1, \ldots, M-1,\ i = 1, \ldots, d.$$

The multidimensional basis functions are written as $\phi_{\boldsymbol{l}}(x) = \phi_{l_1}(x_1)\cdots\phi_{l_d}(x_d)$, where $\boldsymbol{l} = (l_1, \ldots, l_d)$ and $1 \leqslant l_i \leqslant M$. The corresponding approximate solution is $u(x) = \sum_{\boldsymbol{l}} c_{\boldsymbol{l}}\phi_{\boldsymbol{l}}(x)$, with the coefficients determined by the values at the grid or collocation points $x_{\boldsymbol{j}}$. These collocation values will be arranged as a column vector

$$\boldsymbol{u}(t) = \sum_{\boldsymbol{j}} u(t, x_{\boldsymbol{j}})|j_1\rangle \otimes \cdots \otimes |j_d\rangle,$$

that is, the $n_{\boldsymbol{j}}$th entry of $\boldsymbol{u}$ is $u(t, x_{\boldsymbol{j}})$, with the global index given by

$$n_{\boldsymbol{j}} := j_1 2^{d-1} + \cdots + j_d 2^0, \quad \boldsymbol{j} = (j_1, \ldots, j_d).$$

Similarly, $c_{\boldsymbol{l}}$ is written in a column vector as $\boldsymbol{c} = \sum_{\boldsymbol{l}} c_{\boldsymbol{l}}|l_1\rangle \otimes \cdots \otimes |l_d\rangle$. For convenience, let $c_{\boldsymbol{l}} = c_{l_1}\cdots c_{l_d}$. Then

$$u(t, x_{\boldsymbol{j}}) = \sum_{\boldsymbol{l}} c_{l_1}\cdots c_{l_d}\phi_{l_1}(x_{j_1})\cdots\phi_{l_d}(x_{j_d}) \quad (15)$$

and the transition between $\boldsymbol{u}$ and $\boldsymbol{c}$ is given by

$$\boldsymbol{u} = \boldsymbol{u}^{(1)} \otimes \cdots \otimes \boldsymbol{u}^{(d)} = (\Phi \boldsymbol{c}^{(1)}) \otimes \cdots \otimes (\Phi \boldsymbol{c}^{(d)}) = \Phi^{\otimes^d} \boldsymbol{c}, \quad (16)$$

where

$$\Phi^{\otimes^d} = \underbrace{\Phi \otimes \cdots \otimes \Phi}_{d \text{ matrices}}, \quad \boldsymbol{c}^{(i)} = (c_{l_i})_{M\times 1}, \quad \boldsymbol{u}^{(l)} = \Phi \boldsymbol{c}^{(l)},$$

$$\boldsymbol{c} = \boldsymbol{c}^{(1)} \otimes \cdots \otimes \boldsymbol{c}^{(d)} = \sum_{\boldsymbol{l}} c_{\boldsymbol{l}}|l_1\rangle \otimes \cdots \otimes |l_d\rangle. \quad (17)$$

For later use, we next determine the transitions between the position operator $\hat{x}_j$ and the momentum operator $\hat{P}_j = -i\frac{\partial}{\partial x_j}$ in discrete settings. Let $u(x)$ be a function in one dimension and $\boldsymbol{u} = [u(x_0), \ldots, u(x_{M-1})]^T$ be the mesh function with $M = 2N$. The discrete position operator $\hat{x}^d$ of $\hat{x}$ can be defined as

$$\boldsymbol{u} = u(x_i) \to \hat{x}^d \boldsymbol{u} = D_x \boldsymbol{u},$$

where $D_x = \mathrm{diag}(x_0, x_1, \ldots, x_{M-1})$ is the matrix representation of the position operator in x space. By the discrete Fourier expansion in (14), the momentum operator can be discretized as

$$\hat{P}u(x) \approx \hat{P}\sum_{l=1}^{M} c_l\phi_l(x) = \sum_{l=1}^{M} c_l\hat{P}\phi_l(x) = \sum_{l=1}^{M} c_l[-i\partial_x\phi_l(x)]$$

$$= \sum_{l=1}^{M} c_l\mu_l\phi_l(x), \quad \mu_l = 2\pi(l-N-1)$$

for $x = x_j$, $j = 0, 1, \ldots, M-1$, which is written in matrix form as

$$\hat{P}^d \boldsymbol{u} = \Phi D_\mu \Phi^{-1}\boldsymbol{u} =: P_\mu \boldsymbol{u}, \quad D_\mu = \mathrm{diag}(\mu_1, \ldots, \mu_M),$$

where $\hat{P}^d$ is the discrete momentum operator. The matrices D_μ and P_μ can be referred to as the matrix representation of the momentum operator in the momentum space and the position space, respectively, and are related by the discrete Fourier transform. For d dimensions, the discrete position operator $\hat{x}_l^d$ is defined as

$$\boldsymbol{u} = \boldsymbol{u}^{(1)} \otimes \cdots \otimes \boldsymbol{u}^{(d)} \to \boldsymbol{u}^{(1)} \otimes \cdots \otimes \tilde{\boldsymbol{u}}^{(l)} \otimes \cdots \otimes \boldsymbol{u}^{(d)},$$

where

$$\tilde{\boldsymbol{u}}^{(l)} := \left[x_{j_{l_i}} u^{(l)}\left(x_{j_{l_i}}\right)\right] = D_x \boldsymbol{u}^{(l)}.$$

Then

$$\hat{x}_l^d \boldsymbol{u} = (I^{\otimes^{l-1}} \otimes D_x \otimes I^{\otimes^{d-l}})\boldsymbol{u} =: \boldsymbol{D}_l \boldsymbol{u}.$$

Using the expansion in (15), one easily finds that

$$\hat{P}_l^d \boldsymbol{u} = (I^{\otimes^{l-1}} \otimes P_\mu \otimes I^{\otimes^{d-l}})\boldsymbol{u} =: \boldsymbol{P}_l \boldsymbol{u}.$$

Note that $\boldsymbol{D}_l$ and $\boldsymbol{P}_l$ are Hermitian matrices and

$$(\Phi^{\otimes^d})^{-1}\boldsymbol{P}_l\Phi^{\otimes^d} = I^{\otimes^{l-1}} \otimes D_\mu \otimes I^{\otimes^{d-l}} =: \boldsymbol{D}_l^\mu. \quad (18)$$

For convenience, we denote by $F_x = \Phi^{\otimes^d}$ the Fourier transformation matrix in d dimensions. The above denotations also apply to the variable p. One just needs to modify the subscript x to p; for example, F_p represents the Fourier transformation matrix for p. We will use the same denotations D_μ and P_μ for both x and p whenever no confusion will arise.

Remark 2. Given a set of numbers $x_0, x_1, \ldots, x_{M-1}$, the discrete Fourier transform and the inverse DFT are defined by

$$y_k = \frac{1}{\sqrt{M}}\sum_{j=0}^{M-1} e^{2\pi ijk/M}x_j, \quad k = 0, \ldots, M-1$$

and

$$x_j = \frac{1}{\sqrt{M}}\sum_{k=0}^{M-1} e^{-2\pi ijk/M}y_k, \quad j = 0, \ldots, M-1,$$

respectively. Denote the transformation matrix of the DFT by F. It is easy to find that the transformation matrix introduced above satisfies $\Phi = \sqrt{M}SF$, where $S = \mathrm{diag}([1, -1, \ldots, 1, -1]_{M\times 1})$ is a diagonal matrix.

3. The heat equation as a system of Schrödinger equations

Equation (5) clearly shows w is governed by the free Schrödinger equation in the Fourier space for p. Below, in the

discrete space and for the heat equation with a source term, we also show how the heat equation can be reformulated as a system of Schrödinger equations. Consider the linear heat equation with a source term

$$\begin{aligned} \partial_t u - \Delta u &= V(x)u, \\ u(0,x) &= u_0(x), \end{aligned} \tag{19}$$

where $V(x)$ is a scalar function. We now consider the spectral discretization of (1) with respect to the variable p. According to the above discussion, the term $\partial_p w$ in (1) can be discretized as

$$\partial_p w = i(-i\partial_p w) \longrightarrow iP_\mu \boldsymbol{w}(t,x),$$

where $\boldsymbol{w}(t,x)=[w(t,x,p_0), w(t,x,p_1),\ldots, w(t,x,p_{M-1})]^T$. Thus we have the semidiscrete system of partial differential equations

$$\begin{aligned} \partial_t \boldsymbol{w}(t,x) + i[\Delta_x + V(x)]P_\mu \boldsymbol{w}(t,x) &= \boldsymbol{0}, \\ \boldsymbol{w}(0,x) = \boldsymbol{w}_0(x) &:= [w_0(x,p_j)]_{M\times 1}. \end{aligned}$$

In the momentum basis, one has

$$\begin{aligned} \partial_t \boldsymbol{c}(t,x) + i[\Delta_x + V(x)]D_\mu \boldsymbol{c}(t,x) &= \boldsymbol{0}, \\ \boldsymbol{c}(0,x) = \boldsymbol{c}_0(x) &:= \Phi^{-1}\boldsymbol{w}_0(x), \end{aligned} \tag{20}$$

where $\boldsymbol{c} = F_p^{-1}\boldsymbol{w}$. The above PDEs can be viewed as a quantum system since we have introduced the imaginary number $i=\sqrt{-1}$ and $D_\mu = \mathrm{diag}(\mu_1,\ldots,\mu_M)$ is a real diagonal matrix.

In conclusion, we have converted the heat equation to a decoupled system of Schrödinger equations by using the idea in Sec. II A 1 and the discrete Fourier transform on p.

Remark 3. The Schrödinger equation is $i\partial_t\Psi = -\frac{\hbar}{2m}(\Delta - \frac{2m}{\hbar^2}\tilde{V})\Psi$. To match the form above, we can interpret $\mu_j \leftrightarrow -\hbar/2m$ and $V_j = -\tilde{V}/\hbar\mu_j$, which is the potential corresponding to the jth mode. However, this requires us to have the right sign $\mu_j < 0$; otherwise we need to interpret negative mass. This issue can be easily resolved by introducing the new variable $\hat{\boldsymbol{c}}_j(\hat{t},\cdot) = \boldsymbol{c}_j(t,\cdot)$, where $\hat{t} = -t$ if $\mu_j > 0$.

4. Quantum simulation of the heat equation

One can now directly simulate the (decoupled) system of equations (20). For clarity, we will consider (x,p) as a new variable and repeat the construction in the preceding section. Let

$$w(t,x,p) = \sum_{\boldsymbol{l}} c_{\boldsymbol{l}}(t)\phi_{\boldsymbol{l}}(x,p), \quad \boldsymbol{l} = (l_1,\ldots,l_d,l_p). \tag{21}$$

The collocation points are denoted by $(x_{\boldsymbol{j}}, p_{j_p})$ with $\boldsymbol{j} = (j_1,\ldots,j_d)$. As in (17), we define $\boldsymbol{c} = \boldsymbol{c}_x \otimes \boldsymbol{c}_p$, where $\boldsymbol{c}_x = \boldsymbol{c}^{(1)}\otimes\cdots\otimes\boldsymbol{c}^{(d)}$. We also introduce the notation $\boldsymbol{w} = \boldsymbol{w}_x \otimes \boldsymbol{w}_p$, where $\boldsymbol{w}_x = \boldsymbol{w}^{(1)}\otimes\cdots\otimes\boldsymbol{w}^{(d)}$, and $\boldsymbol{w}^{(l)} = \Phi\boldsymbol{c}^{(l)}$ can be viewed as the approximate solution of w in x_l direction. Following the discussion in Sec. II A 2, one has

$$\begin{aligned} \Delta_x w_p &= \sum_{l=1}^d \partial_{x_l}^2 \partial_p w = -i\sum_{l=1}^d (-i\partial_{x_l})^2(-i\partial_p)w \\ &= -i\sum_{l=1}^d \hat{P}_l^2\hat{P}_p w \to -i\sum_{l=1}^d (\hat{P}_l^d)^2\hat{P}_p^d \boldsymbol{w} \\ &= -i\sum_{l=1}^d (\boldsymbol{P}_l^2\otimes I)(I^{\otimes^d}\otimes P_\mu)(\boldsymbol{w}_x\otimes\boldsymbol{w}_p) \\ &= -i\sum_{l=1}^d (\boldsymbol{P}_l^2\otimes P_\mu)\boldsymbol{w}, \end{aligned}$$

where $\boldsymbol{P}_l = I^{\otimes^{l-1}}\otimes P_\mu \otimes I^{\otimes^{d-l}}$, and

$$\begin{aligned} V(x)w_p &= iV(x)(-i\partial_p)w \to iV(\hat{x}^d)\hat{P}_p^d\boldsymbol{w} \\ &= i(\boldsymbol{V}\otimes I)(I^{\otimes^d}\otimes P_\mu)(\boldsymbol{w}_x\otimes\boldsymbol{w}_p) = i(\boldsymbol{V}\otimes P_\mu)\boldsymbol{w}, \end{aligned}$$

where $\boldsymbol{V}$ is a diagonal matrix with

$$\boldsymbol{V}_{n_j,n_j} = V(x_{\boldsymbol{j}}), \quad n_{\boldsymbol{j}} = j_1 2^{d-1} + \cdots + j_d 2^0.$$

The resulting ODEs are

$$\frac{d}{dt}\boldsymbol{w}(t) - i\big[\big(P_1^2 + \cdots + \boldsymbol{P}_d^2 - V\big)\otimes P_\mu\big]\boldsymbol{w}(t) = \boldsymbol{0}, \tag{22}$$

which is an expected Hamiltonian system.

The system (22) can be solved by using, for example, the first-order time or Trotter splitting, as done for the standard Schrödinger equation in [26]. To do so, we first diagonalize the matrix with respect to p for convenience as in (20). Introducing a new variable $\tilde{\boldsymbol{w}} = (I^{\otimes^d}\otimes F_p^{-1})\boldsymbol{w}$, one gets

$$\frac{d}{dt}\tilde{\boldsymbol{w}}(t) = i\boldsymbol{H}\tilde{\boldsymbol{w}}(t), \quad \boldsymbol{H} = \big(P_1^2 + \cdots + \boldsymbol{P}_d^2 - V\big)\otimes D_\mu. \tag{23}$$

From time $t = t_n$ to time $t = t_{n+1}$, the system can be solved in two steps.

(i) One first solves

$$\begin{aligned} &\frac{d}{dt}\tilde{\boldsymbol{w}}(t) = i\big[\big(\boldsymbol{P}_1^2 + \cdots + \boldsymbol{P}_d^2\big)\otimes D_\mu\big]\tilde{\boldsymbol{w}}(t), \quad t_n < t < t_{n+1}, \\ &\tilde{\boldsymbol{w}}(t_n) = \tilde{\boldsymbol{w}}^n \end{aligned}$$

for one time step, where $\tilde{\boldsymbol{w}}^n$ is the numerical solution at $t = t_n$. Noting the relation (18), by letting $\tilde{\boldsymbol{c}}(t) = (F_x^{-1}\otimes I)\tilde{\boldsymbol{w}}(t)$, we instead solve

$$\begin{aligned} &\frac{d}{dt}\tilde{\boldsymbol{c}}(t) = i\boldsymbol{H}_D\tilde{\boldsymbol{c}}(t), \quad t_n < t < t_{n+1}, \\ &\tilde{\boldsymbol{c}}(t_n) = \tilde{\boldsymbol{c}}^n = \big(F_x^{-1}\otimes I\big)\tilde{\boldsymbol{c}}^n, \end{aligned}$$

where

$$\boldsymbol{H}_D = \big[\big(D_1^\mu\big)^2 + \cdots + \big(D_d^\mu\big)^2\big]\otimes D_\mu$$

is a diagonal matrix. The numerical solution will be denoted by $\tilde{\boldsymbol{c}}^*$.

(ii) Let $\tilde{\boldsymbol{w}}^* = (F_x\otimes I)\tilde{\boldsymbol{c}}^*$. The second step is to solve

$$\frac{d}{dt}\tilde{\boldsymbol{w}}(t) = -i(\boldsymbol{V}\otimes D_\mu)\tilde{\boldsymbol{w}}(t) =: -i\boldsymbol{H}_V\tilde{\boldsymbol{w}}(t)$$

for one time step, with $\tilde{\boldsymbol{w}}^*$ the initial data, where $\boldsymbol{H}_V$ is a diagonal matrix. This gives the updated numerical solution $\tilde{\boldsymbol{w}}^{n+1}$.

We denote by m the number of qubits per dimension. The total number of qubits is then given by $m_H = m_d + m_p$ for d-dimensional problems, where $m_d = dm$ and m_p are the number of qubits on x and p registers, respectively. Let Δx, Δp, and Δt be the step sizes for x, p, and t, respectively, where we assume the same spatial step along each dimension. Then $m \sim \log(1/\Delta x)$ and $m_p \sim \log(1/\Delta p)$.

For convenience we write the time complexity in terms of the steps sizes and the number of qubits throughout the paper. For the given error bound ε, the ε dependence of these quantities is determined by the particular scheme one wishes to use. For instance, the mesh strategy of the heat equation can be given by

$$\Delta x \sim \left(\frac{\varepsilon}{d}\right)^{1/\ell}, \quad \Delta t \sim \varepsilon, \quad \Delta p \sim \varepsilon.$$

Note that the initial condition, due to a lack of regularity, implies first-order accuracy on p.

Theorem 1. The solution to the heat equation can be simulated with gate complexity given by

$$N_{\text{gates}} = \frac{T}{\Delta t} O(dm \log m + m_p \log m_p),$$

where T is the evolution time.

Proof. Given the initial state of $\tilde{\boldsymbol{w}}^0$, applying the inverse QFT to the x register, one gets $\tilde{\boldsymbol{c}}^0$. At each time step, one needs to consider the procedure

$$\tilde{\boldsymbol{c}}^n \xrightarrow{e^{i\boldsymbol{H}_D \Delta t}} \tilde{\boldsymbol{c}}^* \xrightarrow{F_x \otimes I} \tilde{\boldsymbol{w}}^* \xrightarrow{e^{-i\boldsymbol{H}_V \Delta t}} \tilde{\boldsymbol{w}}^{n+1} \xrightarrow{F_x^{-1} \otimes I} \tilde{\boldsymbol{c}}^{n+1},$$

where $F_x = \Phi^{\otimes^d}$. It is known that the quantum Fourier transforms in one dimension can be implemented using $O(m \log m)$ gates. The diagonal unitary operators $e^{-i\boldsymbol{H}_V \Delta t}$ and $e^{i\boldsymbol{H}_D \Delta t}$ can be implemented using $O(m_H)$ gates [29,30]. Therefore, the gate complexity required to iterate to the nth step is

$$\begin{aligned} N_{\text{gates}} &= nO(dm \log m + m_p \log m_p + m_H) \\ &= \left(\frac{T}{\Delta t}\right) O(dm \log m + m_p \log m_p), \end{aligned}$$

where $m_p \log m_p$ results from the quantum Fourier transform (QFT) on the p register, which is only performed twice. ■

Remark 4. For the rest of the paper, we refer to the algorithm mentioned in [29,30] to implement e^{iHt} as Algorithm I, where H is a diagonal matrix.

B. Linear convection equation

In this section we provide a way to turn the linear convection equation in d dimensions with periodic boundary conditions into a system of Schrödinger equations

$$\partial_t u + \partial_{x_1} u + \partial_{x_2} u + \cdots + \partial_{x_d} u = 0, \tag{24}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in (-1, 1)^d$ and $u = u(t, x_1, x_2, \ldots, x_d)$.

1. Reformulation

Let

$$w = \sin(p)u,$$

where $p \in [-\pi, \pi]$, which is obviously periodic with respect to p. Then w satisfies

$$\partial_t w - \partial_{x_1, pp} u - \partial_{x_2, pp} u - \cdots - \partial_{x_d, pp} u = 0.$$

Considering the Fourier spectral discretization on x, one easily gets

$$\partial_t \boldsymbol{w}(t, p) - i \sum_{l=1}^{d} \boldsymbol{P}_l \partial_{pp} \boldsymbol{w}(t, p) = \boldsymbol{0},$$

where $\boldsymbol{w}(t, p) = \sum_j w(t, x_j, p)|\boldsymbol{j}\rangle$. Let $\boldsymbol{c}_x(t, p) = F_x^{-1} \boldsymbol{w}(t, p)$. We then get the system of (decoupled) free Schrödinger-type equations

$$\partial_t \boldsymbol{c}_x(t, p) - i \sum_{l=1}^{d} \boldsymbol{D}_l^{\mu} \partial_{pp} \boldsymbol{c}_x(t, p) = \boldsymbol{0} \tag{25}$$

in the momentum space, where $\boldsymbol{D}_l^{\mu}$ is a diagonal matrix defined by (18).

2. Quantum simulation

By further applying the Fourier transform on p, Eq. (25) becomes

$$\partial_t \tilde{\boldsymbol{c}}_x(t) + i \sum_{l=1}^{d} \left(D_l^{\mu} \otimes P_{\mu}^2\right) \tilde{\boldsymbol{c}}_x(t) = \boldsymbol{0},$$

where

$$\tilde{\boldsymbol{c}}_x(t) = [\tilde{\boldsymbol{c}}_{x,0}(t); \ldots; \tilde{\boldsymbol{c}}_{x,M-1}(t)], \tilde{\boldsymbol{c}}_{x,i}(t) = \sum_k \boldsymbol{c}_{x,i}(t, p_k)|k\rangle.$$

Let $\boldsymbol{c}(t) = (I^{\otimes^d} \otimes F_p^{-1}) \tilde{\boldsymbol{c}}_x$. Then

$$\partial_t \boldsymbol{c}(t) + i \sum_{l=1}^{d} \left(D_l^{\mu} \otimes D_{\mu}^2\right) \boldsymbol{c}(t) = \boldsymbol{0}, \tag{26}$$

where $\boldsymbol{c}$ is exactly the momentum variables of $\boldsymbol{w}$, i.e., $\boldsymbol{c} = (F_x^{-1} \otimes F_p^{-1}) \boldsymbol{w}$.

Theorem 2. The solution to the convection equation can be simulated with gate complexity

$$N_{\text{gates}} = O((d+1)m \log m),$$

where we have assumed the same number of qubits in every direction.

Proof. Given the initial state of $\boldsymbol{w}^0$, the implementation involves an application of an inverse QFT, followed by a multiplication of a diagonal unitary operator $\boldsymbol{H} = \sum_{l=1}^{d} (\boldsymbol{D}_l^{\mu} \otimes D_{\mu}^2)$ and a QFT. Hence the gate complexity (see Algorithm I in Remark 4) is

$$N_{\text{gates}} = O(m_{d+1} + 2(d+1)m \log m) = O((d+1)m \log m),$$

which completes the proof. ■

Remark 5. As was done for the Liouville equation in [11], one can apply the Fourier spectral discretization to the original

equation and get

$$\partial_t \boldsymbol{c}_x(t) + i\sum_{l=1}^{d} \boldsymbol{D}_l^{\mu} \boldsymbol{c}_x(t) = \boldsymbol{0},$$

where $\boldsymbol{c}_x = F_x^{-1}\boldsymbol{u}$ and $\boldsymbol{u}(t) = \sum_j u(t, x_j)|j\rangle$. The number of quantum gates required for its simulation is $O(dm \log m)$, which is comparable to that of the Schrödinger system. On the other hand, one can get $\boldsymbol{u}$ by projecting $\boldsymbol{w}$ onto the x register for our quantum simulation approach, since $w(t, x, p) = \sin(p)u(t, x)$ is separated in x and p, which means the costs of the computation of the observables are also comparable. Here we want to show that, for conceptual interest, the first-order hyperbolic equation can be transformed into a system of Schrödinger-type equations.

III. QUANTUM SIMULATION OF THE GENERAL LINEAR SYSTEM OF ODEs

Not every linear PDE can be transformed into the exact form of the Schrödinger equation, but we will see that they can certainly be transformed into Hamiltonian systems in the discrete setting. To do so, one only needs to show that a linear system of ordinary differential equations can be converted into a Hamiltonian system, since a PDE, after spatial discretization, becomes a system of ODEs.

A. General linear ODEs

Suppose one needs to solve the ODEs

$$\begin{aligned} \frac{d\boldsymbol{u}(t)}{dt} &= A\boldsymbol{u}(t) + \boldsymbol{b}(t), \\ \boldsymbol{u}(0) &= \boldsymbol{u}_0, \end{aligned} \tag{27}$$

where matrix A is independent of time and $A^{\dagger} \neq A$ in general. We remark that it suffices to assume $\boldsymbol{b}(t) = \boldsymbol{0}$. Otherwise one can instead consider the augmented system

$$\begin{aligned} \frac{d\boldsymbol{u}(t)}{dt} &= A\boldsymbol{u}(t) + \boldsymbol{b}(t)v, \quad \boldsymbol{u}(0) = \boldsymbol{u}_0, \\ v_t &= 0, \quad v(0) = 1, \end{aligned}$$

where the second equation gives $v(t) \equiv 1$, which leads to the original ODE system. The above ODEs can be written in the compact form

$$\begin{aligned} \frac{d\tilde{\boldsymbol{u}}(t)}{dt} &= \tilde{A}\tilde{\boldsymbol{u}}(t), \quad \tilde{\boldsymbol{u}}(0) = \tilde{\boldsymbol{u}}_0, \\ \tilde{\boldsymbol{u}} &= \begin{bmatrix} \boldsymbol{u} \\ v \end{bmatrix}, \quad \tilde{\boldsymbol{u}}_0 = \begin{bmatrix} \boldsymbol{u}_0 \\ 1 \end{bmatrix}, \end{aligned}$$

where

$$\tilde{A} = \begin{bmatrix} A & \boldsymbol{b}(t) \\ \boldsymbol{0}^T & 0 \end{bmatrix}$$

and the zero vector $\boldsymbol{0}$ has the same size as $\boldsymbol{b}$. For this reason, without loss of generality, we assume $\boldsymbol{b} = \boldsymbol{0}$ in the following.

Remark 6. Quantum simulations for time-dependent or -independent boundary-value problems are quite difficult because the ODE system resulting from spatial discretizations is not necessarily a Hamiltonian system. Spatial discretization of the boundary condition, for example, the Dirichlet boundary condition, could also give rise to $\boldsymbol{b} \neq \boldsymbol{0}$. Our approach combined with the above augmentation technique will resolve this problem in a generic and an efficient way, which, however, is beyond the scope of the present work.

One can always decompose A into a Hermitian term and an anti-Hermitian term

$$A = H_1 + iH_2,$$

where

$$H_1 = \frac{A + A^{\dagger}}{2} = H_1^{\dagger}, \quad H_2 = \frac{A - A^{\dagger}}{2i} = H_2^{\dagger}.$$

Apparently, $H_1H_2 = H_2H_1$ holds if and only if $A^{\dagger}A = AA^{\dagger}$. In view of the stability, it is natural to assume that H_1 is negative semidefinite (note that $\boldsymbol{x}^{\dagger}H_1\boldsymbol{x} = \boldsymbol{x}^{\dagger}A\boldsymbol{x}$). In addition, it is simple to see that

$$2s(H_1) \sim s(A) \sim 2s(H_2)$$

and

$$\|H_1\|_{\max}, \|H_2\|_{\max} \leqslant \|A\|_{\max},$$

where $s(A)$ is the sparsity of A and $\|A\|_{\max}$ denotes the largest entry of A in absolute value.

Using the warped phase transformation $\boldsymbol{v}(t, p) = e^{-p}\boldsymbol{u}(t)$ for $p > 0$ and symmetrically extending the initial data to $p < 0$ as in Sec. II A 1, the ODEs are then transformed into a system of linear convection equations

$$\begin{aligned} \frac{d}{dt}\boldsymbol{v}(t, p) &= A\boldsymbol{v}(t, p) = -H_1\partial_p\boldsymbol{v} + iH_2\boldsymbol{v}, \\ \boldsymbol{v}(0, p) &= e^{-|p|}\boldsymbol{u}_0. \end{aligned} \tag{28}$$

The solution $\boldsymbol{u}(t)$ can be restored by

$$\boldsymbol{u}(t) = \int_0^{\infty} \boldsymbol{v}(t, p)dp.$$

Apply the discrete Fourier transformation on p to get

$$\frac{d}{dt}\boldsymbol{w}(t) = -i(H_1 \otimes P_{\mu})\boldsymbol{w} + i(H_2 \otimes I)\boldsymbol{w}, \tag{29}$$

which is a Hamiltonian system as expected, where $\boldsymbol{w}$ collects all the grid values of $\boldsymbol{v}(t, p)$ and is defined by

$$\boldsymbol{w} = [\boldsymbol{w}_1; \boldsymbol{w}_2; \ldots; \boldsymbol{w}_n], \quad \boldsymbol{w}_i = \sum_k \boldsymbol{v}_i(t, p_k)|k\rangle,$$

with the semicolon indicating the straightening of $\{\boldsymbol{w}_i\}_{i\geqslant 1}$ into a column vector. By the change of variables $\tilde{\boldsymbol{w}} = (I_u \otimes F_p^{-1})\boldsymbol{w}$, one has

$$\frac{d}{dt}\tilde{\boldsymbol{w}}(t) = -i(H_1 \otimes D_{\mu})\tilde{\boldsymbol{w}} + i(H_2 \otimes I)\tilde{\boldsymbol{w}}. \tag{30}$$

In the following, we assume A is independent of time and apply the Hamiltonian simulation algorithm in [31] as described in the following lemma. For time-dependent Hamiltonians, we refer the reader to [32–34] for references. As in [31], we are concerned with the sparse access to the matrix, with the definition given below.

From now on we define s as the sparsity of A, and $\|A\|_{\max}$ as the largest entry of A in absolute value,

Definition 1. Let A be a Hermitian matrix with the (i, j)th entry denoted by A_{ij}. Sparse access to A is referred to as a 4-tuple $(s, \|A\|_{\max}, O_A, O_F)$. Here O_A is a unitary black box which can access the matrix elements A_{ij} such that

$$O_A|j\rangle|k\rangle|z\rangle = |j\rangle|k\rangle|z \oplus A_{jk}\rangle$$

for any $j, k \in \{1, 2, \cdots, N\} =: [N]$, where the third register holds a bit string representing of A_{jk}; O_F is a unitary black box which allows to perform the map

$$O_F|j\rangle|l\rangle = |j\rangle|F(j, l)\rangle$$

for any $j \in [N]$ and $l \in [s]$, where the function F outputs the column index of the lth nonzero elements in row j.

The quantum algorithm for general sparse Hamiltonian simulation with nearly optimal dependence on all parameters can be found in [31]. The run time is measured in terms of the query complexity or the number of queries made to the oracles $\mathcal{O}_A$ and $\mathcal{O}_F$.

Lemma 1 (Algorithm II and Theorems 1 and 2 in [31]). An s-sparse Hamiltonian H acting on m_H qubits can be simulated within error ε with

$$O\left(\tau \frac{\log(\tau/\varepsilon)}{\log\log(\tau/\varepsilon)}\right)$$

queries and

$$O\left(\tau[m_H + \log^{2.5}(\tau/\varepsilon)]\frac{\log(\tau/\varepsilon)}{\log\log(\tau/\varepsilon)}\right) = O(\tau m_H L_{\text{polylog}})$$

additional two-qubit gates, where $\tau = s\|H\|_{\max}t$ and t is the evolution time and

$$L_{\text{polylog}} \equiv [1 + \log^{2.5}(\tau/\varepsilon)]\frac{\log(\tau/\varepsilon)}{\log\log(\tau/\varepsilon)}.$$

This result is nearly optimal.

For convenience, we assume A arises from some discretization in d dimensions, which implies A is of order $N_A \sim 2^{m_d}$, where $m_d = dm$ and m is the number of qubits along each direction.

Theorem 3. Our quantum simulation method has gate complexity

$$N_{\text{gates,Schr}} = (m_d + m_p)\tilde{O}\left(\frac{s(A)\|A\|_{\max}T}{\Delta p}\right) + O(m_p \log m_p),$$

where N_A is the order of A and $\tilde{O}$ denotes the case where all logarithmic factors are suppressed. In particular, if H_1 can be diagonalized in the momentum basis and H_2 is a diagonal matrix, then our method can be implemented with

$$N_{\text{gates,Schr}} = \frac{T}{\Delta t}O(dm \log m + m_p \log m_p).$$

Proof. (i) The Hamiltonian in (30) can be simulated with complexity

$$\begin{aligned} &T(m_d+m_p)\tilde{O}(s(H_1 \otimes D_\mu - H_2 \otimes I)\|H_1 \otimes D_\mu - H_2 \otimes I\|_{\max}) \\ &\quad \leqslant T\tilde{O}([s(H_1 \otimes D_\mu) + s(H_2)](\|H_1 \otimes D_\mu\|_{\max} + \|H_2\|_{\max})) \\ &\quad \leqslant T\tilde{O}([s(H_1) + s(H_2)](\|H_1\|_{\max}/\Delta p + \|H_2\|_{\max})) \\ &\quad \leqslant \tilde{O}(s(A)\|A\|_{\max}T/\Delta p), \end{aligned}$$

where $\|D_\mu\|_{\max} \lesssim 1/\Delta p$ is used. (ii) For the special case, one can solve (30) by the first-order time-splitting scheme. The gate complexity can be obtained as in the proof of Theorem 1. ■

The difference between this method and our approach in Sec. II is that here we first discretize in space (hence A is the difference matrix) and then use the warped phase transformation later, but in Sec. II we used the warped phase transformation first and then discretized in space. Let us compare the two ways for the linear heat and convection problems.

(i) For the heat equation, when applying the discrete Fourier transform on the space variables, one obtains

$$\frac{d}{dt}\boldsymbol{u}(t) = A\boldsymbol{u}(t), \quad A = -\left(\boldsymbol{P}_1^2 + \cdots + \boldsymbol{P}_d^2\right) + \boldsymbol{V}.$$

It is easy to find that (29) coincides with (22) since $H_1 = A$ and $H_2 = 0$, that is, the two treatments are equivalent due to the fact that A is a Hermitian matrix.

(ii) For the convection equation, the discrete Fourier transform on the space variables gives

$$\frac{d}{dt}\boldsymbol{u}(t) - i\sum_{l=1}^{d}\boldsymbol{P}_l\boldsymbol{u}(t) = \boldsymbol{0},$$

which corresponds to a special case $H_1 = 0$. In this case, there is no need to apply the warped phase transformation since it is already a Hamiltonian system. Of course, one can still use the $\sin(p)$ transform to get a Schrödinger-type system.

This general method is simply a combination of what we already did in previous sections. The H_2 can be handled either directly or by using the approach with a warped phase transformation.

Remark 7. Our approach can also apply to PDEs of higher-order time derivatives. Let us consider the second-order case $\boldsymbol{u}_{tt} = A\boldsymbol{u}$. Let $\boldsymbol{v} = \boldsymbol{u}_t$. One immediately obtains

$$\boldsymbol{u}_t = \boldsymbol{v}, \quad \boldsymbol{v}_t = A\boldsymbol{u}$$

or

$$\frac{d}{dt}\begin{bmatrix}\boldsymbol{u}\\ \boldsymbol{v}\end{bmatrix} = \begin{bmatrix}0 & I\\ A & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{u}\\ \boldsymbol{v}\end{bmatrix},$$

which reduces to the situation under discussion. The extension to higher-order derivatives is straightforward.

B. Parity-dilating unitarization of linear ODEs

In addition to transforming ODEs into Hamiltonian systems, one can also translate the evolution operator into products of unitary operators that are suitable for quantum simulation. Here we present and extend the unitary dilation technique (for instance, in [35] applied to the Black-Scholes equation) to more general cases. In the following we still consider the ODE system (27) and assume that $\boldsymbol{u}(t)$ has been encoded as a state vector $|\psi(t)\rangle$.

The solution of $|\psi(t)\rangle$ can be approximated by

$$|\psi(t)\rangle = e^{(H_1+iH_2)t}|\psi(t_0)\rangle \approx \prod_{j=1}^{N_t}(e^{H_1\Delta t}e^{iH_2\Delta t})_j|\psi(t_0)\rangle, \quad (31)$$

with $N_t\Delta t = t$. Higher-order terms with commutation relations between H_1 and H_2 can be added later for higher accuracy. We can enlarge the matrix (or operator) to make

unitaries out of each $e^{H_1\Delta t}e^{iH_2\Delta t}$ term. Since $e^{H_1\Delta t} =: H_{\Delta t}$ is Hermitian, one can define a unitary dilation operator $\tilde{U}$ as

$$\tilde{U} := \begin{bmatrix} H_{\Delta t} & \sqrt{I - H_{\Delta t}^2} \\ \sqrt{I - H_{\Delta t}^2} & -H_{\Delta t} \end{bmatrix} = (\sigma_z \otimes I)e^{i\sigma_y \otimes \arccos(H_{\Delta t})}, \tag{32}$$

which is a particular instance of the blocking encoding of Hermitian matrices [36], where I is the identity operator and

$$\sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Note that when H_1 is negative semidefinite, $\|H_{\Delta t}\| \leqslant 1$ holds, which guarantees $\tilde{U}$ being well defined.

Remark 8. The equals sign in (32) can be understood as follows. Let

$$A(\lambda) = \begin{bmatrix} \lambda & \sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & -\lambda \end{bmatrix} = \begin{bmatrix} \cos t & \sin t \\ \sin t & -\cos t \end{bmatrix} =: B(t),$$

where $\lambda = \cos t \in [-1, 1]$. Then the column vectors of $B(t)$ are the solutions of

$$\frac{du_1(t)}{dt} = -u_2(t),$$
$$\frac{du_2(t)}{dt} = u_1(t),$$

whose evolution is given by

$$B(t) = \exp\left(\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} t\right) B(0) = e^{-i\sigma_y t}B(0) = e^{-i\sigma_y t}\sigma_z$$

or $B(t) = \sigma_z e^{i\sigma_y t}$. This leads to $A(\lambda) = \sigma_z e^{i\sigma_y \arccos\lambda}$ and hence the desired equality by letting $\lambda = H_{\Delta t}$.

Let $|0\rangle = [1, 0]^T$. Then we can define a new unitary operator U acting on $|0\rangle \otimes |\psi(t_0)\rangle = [|\psi(t_0)\rangle; \mathbf{0}]$ as

$$U\begin{bmatrix} |\psi(t_0)\rangle \\ 0 \end{bmatrix} := \tilde{U}(I \otimes e^{iH_2\Delta t})\begin{bmatrix} |\psi(t_0)\rangle \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} H_{\Delta t}e^{iH_2\Delta t}|\psi(t_0)\rangle \\ \sqrt{I - H_{\Delta t}^2}e^{iH_2\Delta t}|\psi(t_0)\rangle \end{bmatrix}, \tag{33}$$

where $\mathbf{0}$ has the same size with $|\psi(t_0)\rangle$. For convenience, we refer to the new operator U as the (first-order) evolutionary unitary dilation operator. Now if proceed to use this state directly to the next time step and multiply by another U, one does not recover a term $(H_{\Delta t}e^{iH_2\Delta t})^2|\psi(t_0)\rangle$ in the top entry, but rather it is augmented by another term $(\sqrt{I - H_{\Delta t}^2}e^{iH_2\Delta t})^2|\psi(t_0)\rangle$. This means we cannot postselect at the end, but only at every time step. For instance, in the second step the quantum state should be postselected as

$$\begin{bmatrix} H_{\Delta t}e^{iH_2\Delta t}|\psi(t_0)\rangle \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} |\psi(t_1)\rangle \\ \mathbf{0} \end{bmatrix} = |0\rangle \otimes |\psi(t_1)\rangle,$$

and hence we can repeat the procedure. However, if the success probability of every time step to obtain the top entry is P, then after N_t time steps, a single success requires $O(P^{-N_t})$ copies of $|\psi(t_0)\rangle$ at the beginning. A simple case is that H_1 commutes with H_2 as for the Black-Scholes equation [35]. In this situation

$$e^{At}|\psi(t_0)\rangle = e^{iH_2 t}e^{H_1 t}|\psi(t_0)\rangle$$

and one can first apply the unitary dilation technique to $e^{H_1 t}$ and perform one postselection at the end of the simulation of H_1.

However, H_1 and H_2 do not commute in general. We go for an alternative approach, where we instead begin with the state $|0\rangle^{\otimes N_t^*} \otimes |\psi(t_0)\rangle$, where $N_t^* = \log N_t$. In this case we only need to perform one postselection at the very end. Observing that

$$\begin{bmatrix} |\psi(t_0)\rangle \\ \mathbf{0} \end{bmatrix} \xrightarrow{U} \begin{bmatrix} |\psi(t_1)\rangle \\ * \end{bmatrix},$$

we can introduce the translations

$$\begin{bmatrix} |\psi(t_0)\rangle \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \xrightarrow{U_1} \begin{bmatrix} |\psi(t_1)\rangle \\ * \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \xrightarrow{U_2} \begin{bmatrix} |\psi(t_2)\rangle \\ * \\ * \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \xrightarrow{U_3} \cdots,$$

where the vector has N_t zero vectors and each unitary U_j is basically the evolutionary unitary dilation operator U in Eq. (33) but applied to different pairs of qubit systems. For this reason, we refer to it as the parity-dilating unitarization approach in this article. The costs are identical for all j and the explicit formula of U_j can be written as

$$\begin{aligned} U_j := {} & |0\rangle\langle 0| \otimes e^{iH_2\Delta t}H_{\Delta t} \\ & + (|j\rangle\langle 0| + |0\rangle\langle j|) \otimes e^{iH_2\Delta t}\sqrt{I - H_{\Delta t}^2} \\ & - |j\rangle\langle j| \otimes e^{iH_2\Delta t}H_{\Delta t} + \sum_{k\neq 0,\, j}^{N_t - 1} |k\rangle\langle k| \otimes I. \end{aligned}$$

This means that if assume access to U_j (as an oracle), then one needs to apply this oracle N_t times to the state $|0\rangle^{\otimes N_t^*} \otimes |\psi(t_0)\rangle$ and then just to perform a single postselection at the end. However, we note that this assumption is *not* equivalent to the usual assumption of unitary decomposition, where we decompose into single-qubit gates and controlled-NOT gates or some other combination.

Theorem 4. Assume sparse access to $\arccos(H_1\Delta t)$ with sufficient precision and suppose that the time step satisfies $\|A\|_1\Delta t \leqslant 1$. Then the first-order parity-dilation unitarization method has gate complexity

$$N_{\text{gates,unitary}} = \log N_A \times \tilde{O}\left(\frac{T}{\Delta t}s[\arccos(H_1\Delta t)] + s(A)\|A\|_{\max}\right).$$

In particular, if H_1 can be diagonalized in the momentum basis, H_2 is a diagonal matrix, and $\|H_1\|\Delta t \leqslant 1$, then the parity-dilation unitarization method can be simulated with

$$N_{\text{gates,unitary}} = \frac{T}{\Delta t}O(dm\log m).$$

Proof. (i) The first-order evolutionary unitary dilation operator is

$$U = (\sigma_z \otimes I)e^{i\sigma_y \otimes \arccos(H_1\Delta t)}(I \otimes e^{iH_2\Delta t}).$$

By Lemma 1, the operators $e^{i\sigma_y \otimes \arccos(H_1\Delta t)}$ and $e^{iH_2\Delta t}$ can be simulated with

$$\tilde{O}(m_d s[\arccos(H_1\Delta t)]\| \arccos(H_1\Delta t)\|_{\max}) + \tilde{O}(\Delta t m_d s(H_2)\|H_2\|_{\max})$$

two-qubit gates (note that Δt is not a factor in the first term), where $\arccos(H_1\Delta t)$ is defined according to the Taylor expansion

$$\arccos x = \tfrac{1}{2}\pi - x - \tfrac{1}{6}x^3 - \tfrac{3}{40}x^5 - \tfrac{5}{112}x^7 - \tfrac{35}{1152}x^9 - \cdots,$$

which is well defined when $|x| \leqslant 1$. One easily finds that $\arccos(H_1\Delta t)$ is well defined when the time step satisfies $\|H_1\|_1\Delta t \leqslant 1$. In fact, since H_1 is Hermitian, there exists a unitary matrix V such that $V^{-1}H_1V = \Lambda_1$, where Λ_1 is the diagonal matrix consisting of the eigenvalues of H_1. By definition,

$$\arccos(H_1\Delta t) = T \arccos(\Lambda_1\Delta t)T^{-1}, \tag{34}$$

where $\arccos(\Lambda_1\Delta t)$ is obviously well defined due to the fact that $|\lambda(H_1)|_{\max}\Delta t \leqslant \|H_1\|_1\Delta t \leqslant \|A\|_1\Delta t \leqslant 1$. Therefore,

$$\begin{aligned}\| \arccos(H_1\Delta t)\|_{\max} &\leqslant \| \arccos(H_1\Delta t)\|_1 \\ &\leqslant \tfrac{1}{2}\pi + \|H_1\Delta t\|_1 + \tfrac{1}{6}\|H_1\Delta t\|_1^3 + \cdots \\ &\leqslant \tfrac{1}{2}\pi + 1 + \tfrac{1}{6} + \tfrac{3}{40} + \cdots = \arccos(-1) = \pi.\end{aligned}$$

(ii) For the special case, the first-order evolutionary unitary dilation operator is

$$\begin{aligned}U =& (\sigma_z \otimes I)(I \otimes \Phi^{\otimes^d})e^{i\sigma_y \otimes \arccos(\Lambda_1\Delta t)} \\ &\times [I \otimes (\Phi^{\otimes^d})^{-1}](I \otimes e^{iH_2\Delta t}),\end{aligned} \tag{35}$$

where Λ_1 and H_2 are diagonal matrices and $\arccos(\Lambda_1\Delta t)$ is well defined if $\|H_1\|\Delta t \leqslant 1$. The gate complexity is obviously given by

$$\begin{aligned}N_{\text{gates,unitary}} &= n[2O(dm\log m) + 2O(dm)] \\ &= \frac{T}{\Delta t}O(dm\log m).\end{aligned}$$

This completes the proof. ∎

Remark 9. Note that for $\arccos(H_1\Delta t)$ to be well defined, one needs $\|H_1\|\Delta t \leqslant 1$. If H_1 is the discrete heat operator, then $\|H_1\| = O(\Delta x^2)$ in one dimension; hence one needs $\Delta t = O(\Delta x^2)$, which is the Courant-Friedrichs-Lewy stability condition for solving the heat equation using explicit time discretization. On the other hand, the time-splitting method is unconditionally stable; thus one can take $\Delta t = O(\Delta x)$ there. According to the special case in Theorems 3 and 4, the heat equation (19) can be simulated with

$$\begin{aligned}N_{\text{gates,Schr}} &= \frac{T}{\Delta t}O(dm\log m + m_p\log m_p) \\ &= \frac{T}{\Delta x}O(dm\log m + m_p\log m_p), \\ N_{\text{gates,unitary}} &= \frac{T}{\Delta t}O(dm\log m) \\ &= \frac{T}{\Delta x^2}O(dm\log m).\end{aligned}$$

The mesh strategy is $d\Delta x^\ell \sim \varepsilon$ and $\Delta p \sim \varepsilon$, which gives

$$\frac{N_{\text{gates,Schr}}}{N_{\text{gates,unitary}}} = \Delta x O\left(1 + \frac{\ell}{d}\frac{\log(1/\varepsilon)}{\log(d/\varepsilon)}\right).$$

This implies that the parity-dilating unitarization method requires more computational cost if ℓ is not large, that is, the solution u is not sufficiently smooth.

In the following, we focus on distinguishing the parity-dilating unitarization method with our approach.

For spectral discretizations, the coefficient matrix arising from the differential operators is always dense in the original variables due to the existence of the DFT matrix. It may be hard to get a sparse system when the equations have varying coefficients. For a dense matrix, the preparation of the matrix $\arccos(H_1\Delta t)$ seems to be rather involved. A candidate for the implementation is the Taylor expansion, which, however, cannot resolve the sparsity problem.

The finite-difference discretization usually yields sparse systems but with lower accuracy than the spectral method. The matrix $\arccos(H_1\Delta t)$ is often dense even if H_1 is very sparse. This is like the transformation in DFT, which makes the diagonal matrix a dense one [see (34)]. This means $s[\arccos(H_1\Delta t)]$ usually scales as $O(\Delta x^{-d})$, which can be far greater than $s(H_1)\|H_1\|_{\max}/\Delta p$ when d is large [the latter one may scale as $O(d^2/\Delta x^\alpha \Delta p)$]. On the other hand, the density makes the implementation of $\arccos(H_1\Delta t)$ difficult. It should be pointed out that for some cases, for instance, the central difference of the Laplacian, the matrix can be diagonalized in the discrete Fourier, discrete sine, or discrete cosine transformation matrix. In this case, the $\arccos(H_1\Delta t)$ can be efficiently implemented since the transformation matrices can be realized by the fast Fourier transform.

C. Block-encoding unitarization of linear ODEs

The parity-dilation method shows a particular example of a block-encoding strategy. More general results can be derived. In Sec. 4.1 of [20], the authors presented the block-encoding technique to solve (27) with negative-definite A and time-independent $\boldsymbol{b}$, where $\|A\| \leqslant 1$ is assumed for technical simplicity. Based on the evolutionary form

$$\begin{aligned}\boldsymbol{u}(t) &= e^{At}\boldsymbol{u}(0) + \int_0^T e^{A(T-s)}ds\,\boldsymbol{b} \\ &= e^{At}\boldsymbol{u}(0) + (e^{AT} - I)A^{-1}\boldsymbol{b},\end{aligned}$$

the algorithm first separately computes the homogeneous and the inhomogeneous parts and then combines them together using the technique of linear combination of quantum states.

For the construction of the linear combination, we refer the reader to Lemma 22 in [20] when given the block encodings of e^{At} and $\int_0^T e^{A(T-s)}ds$.

The query complexity of the block-encoding technique in [20] is based on access to the block encoding of matrix A, which can be difficult to realize. If we assume access to the block encoding of A, then the complexity is dependent of the condition number of A (i.e., the inverse of δ in Theorem 23 in [20]), which could give rise to a large factor. This is not the case for the Hamiltonian-simulation-based algorithms. However, if access to the block encoding of A^2 is assumed, this factor is no longer present.

D. Imaginary-time-evolution methods for linear ODEs

Imaginary-time-evolution methods refer to the application of a Wick rotation $\tau = -it$ to transform the heat equation $\partial_t u = \partial_{xx} u$ into a Schrödinger equation $-i\partial_\tau u(\tau, x) = \partial_{xx} u(\tau, x)$. This has been applied in heuristic schemes for quantum problems, for instance, in [37–39]. In this case, the state obeying the Schrödinger equation and its imaginary-time counterpart do not have the same evolution. This means that, except for the steady-state solution, extra resources are necessary to map between the solution in the unitarily evolving system to the other. Usually some heuristic techniques are used, like variational methods or classical optimization that requires input from quantum measurements on the quantum states themselves. These, however, have the benefit of being implementable on hybrid classical-quantum devices.

The imaginary-time-evolution approach can also be used to deal with the general linear ODE system (27) in principle. However, it may change the nature of the underlying PDEs. For example, if it is done for the heat equation, then one changes from a dissipative equation, which has the merit of converging fast (exponentially) to the ground state, to a Hamiltonian system that converges slowly to the ground or steady state.

This differs from our method, which does not rely on heuristic methods. Here we instead transform the heat equation to phase space by using the warped phase transformation. We can also apply this to ground-state or steady-state preparation [21].

IV. APPLICATIONS TO MORE PDEs

In this section we apply our method to more typical examples. For simplicity, we only provide the time complexity in terms of the number of qubits since the error estimates may be rather involved.

A. Black-Scholes equation

The Black-Scholes equation

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial V^2}{\partial S^2} = rV$$

is a PDE that evaluates the price of a financial derivative, where r and σ are constants. For a specific derivative contract, the problem is to determine its present price $V(t = 0, S)$ according to the terminal price $V(t = T, S)$ of the option [35]. The change of variables $S = e^x$, $-\infty < x < \infty$, leads to a backward parabolic equation

$$\frac{\partial V}{\partial t} + \left(r - \frac{\sigma^2}{2}\right)\frac{\partial V}{\partial x} + \frac{\sigma^2}{2}\frac{\partial V^2}{\partial x^2} = rV.$$

One can reverse time $t \to \tau = T - t$ to get a forward parabolic equation

$$\frac{\partial V}{\partial \tau} = \left(r - \frac{\sigma^2}{2}\right)\frac{\partial V}{\partial x} + \frac{\sigma^2}{2}\frac{\partial V^2}{\partial x^2} - rV. \tag{36}$$

This is a typical example, in which the underlying operators can be diagonalized in the momentum basis, which has been resolved by the unitarization approach in [35].

We first consider our method. By the warped phase transformation $W(t, x, p) = e^{-p}V(t, x)$ with periodic extension of the initial data, one gets

$$\partial_\tau W = \left(r - \frac{\sigma^2}{2}\right)\partial_x W + \left(\frac{\sigma^2}{2}\partial_{xx} - rI\right)(-\partial_p W).$$

By repeating the previous calculations, it is straightforward to derive a Hamiltonian system

$$\frac{d}{dt}\tilde{\boldsymbol{W}}(t) = i\boldsymbol{H}\tilde{\boldsymbol{W}}(t),$$
$$\boldsymbol{H} = \left(r - \frac{\sigma^2}{2}\right)(P_\mu \otimes I) + \left(\frac{\sigma^2}{2}P_\mu^2 + rI\right) \otimes D_\mu,$$

where $\tilde{\boldsymbol{W}} = (I \otimes F_p^{-1})\boldsymbol{W}$ and $\boldsymbol{W}(t) = \sum_{j,k} W(t, x_j, p_k)|j, k\rangle$. Performing the change of variables $\tilde{\boldsymbol{c}} = (F_x^{-1} \otimes I)\tilde{\boldsymbol{W}}$, where $F_x = \Phi$, one has

$$\frac{d}{dt}\tilde{\boldsymbol{c}}(t) = i\tilde{\boldsymbol{H}}\tilde{\boldsymbol{c}}(t),$$
$$\tilde{\boldsymbol{H}} = \left(r - \frac{\sigma^2}{2}\right)(D_\mu \otimes I) + \left(\frac{\sigma^2}{2}D_\mu^2 + rI\right) \otimes D_\mu.$$

Theorem 5. Our approach for the Black-Scholes equation can be simulated with gate complexity given by

$$N_{\text{gates,Schr}} = O(m \log m + m_p \log m_p).$$

Proof. The diagonal unitary operator $e^{-i\tilde{\boldsymbol{H}}t}$ can be simulated with

$$N_{\text{gates}}(e^{i\tilde{\boldsymbol{H}}t}) = O(m_H) = O(m + m_p)$$

two-qubit gates (see Algorithm I in Remark 4). The result follows by adding the number of gates for the QFT, which is only performed twice for both x and p. ■

For the unitarization method, we rewrite Eq. (36) as

$$\partial_\tau V = \hat{H}_{\text{BS}} V,$$

where $\hat{H}_{\text{BS}} = \hat{H}_1 + \hat{H}_2$, with

$$\hat{H}_1 V = -i\left(r - \frac{\sigma^2}{2}\right)\partial_x V, \quad \hat{H}_2 V = \frac{\sigma^2}{2}\partial_{xx} V - rV$$

representing the Hermitian and non-Hermitian parts, respectively. Using the DFT, the above equation can be written as

$$\partial_\tau \boldsymbol{V} = (\boldsymbol{H}_1 + i\boldsymbol{H}_2)\boldsymbol{V},$$

with

$$\boldsymbol{V}(t) = \sum_j V(t, x_j)|j\rangle, \quad \boldsymbol{H}_2 = \left(\frac{\sigma^2}{2} - r\right)P_\mu,$$
$$\boldsymbol{H}_1 = \frac{\sigma^2}{2}P_\mu^2 - rI.$$

Let $\tilde{\boldsymbol{V}} = F_x^{-1}\boldsymbol{V}$. The equation can be rewritten as the diagonalized form in the momentum space:

$$\partial_\tau \tilde{\boldsymbol{V}} = (\tilde{\boldsymbol{H}}_1 + i\tilde{\boldsymbol{H}}_2)\tilde{\boldsymbol{V}}, \quad \tilde{\boldsymbol{H}}_2 = \left(\frac{\sigma^2}{2} - r\right)D_\mu,$$
$$\tilde{\boldsymbol{H}}_1 = \frac{\sigma^2}{2}D_\mu^2 - rI.$$

Theorem 6. The parity-dilating unitarization approach for the Black-Scholes equation can be simulated with gate complexity given by

$$N_{\text{gates,unitary}} = O(m \log m).$$

Proof. Since $\tilde{\boldsymbol{H}}_1$ commutes with $\tilde{\boldsymbol{H}}_2$, the evolution of $\tilde{\boldsymbol{V}}$ can be simply written as

$$\tilde{\boldsymbol{V}}(\tau) = e^{(\tilde{\boldsymbol{H}}_1 + i\tilde{\boldsymbol{H}}_2)\tau}\tilde{\boldsymbol{V}}(0) = e^{\tilde{\boldsymbol{H}}_1\tau}e^{i\tilde{\boldsymbol{H}}_2\tau}\tilde{\boldsymbol{V}}(0).$$

Thus one just needs to perform the unitary dilation technique once. According to the proof of Theorem 4, the evolutionary unitary dilation operator is

$$U = (\sigma_z \otimes \mathrm{I})e^{i\sigma_y \otimes \arccos(\tilde{\boldsymbol{H}}_1\tau)}(\mathrm{I} \otimes e^{i\tilde{\boldsymbol{H}}_2\tau}),$$

where the diagonal unitary operators on the right-hand side can be simulated with $O(m+1)$ gates. The result follows by adding the number of gates for the QFT. ■

For this simple example, there is a slight overhead in time complexity for our approach, i.e.,

$$N_{\text{gates,Schr}} - N_{\text{gates,unitary}} = O(m_p \log m_p) = O(\log(1/\varepsilon))$$

since $m_p \sim \log(1/\Delta p)$ and $\Delta p \sim \varepsilon$. Such a conclusion is obviously valid for cases where the underlying operators can be diagonalized in the same basis (in this case, the arccos is not an issue as observed in the proof of Theorem 6), which are often encountered for the differential operators with constant coefficients.

B. Fokker-Planck equation

The Fokker-Planck equation describes the time evolution of the probability density function $f(t, x)$ of the velocity of a particle under the influence of drag forces and random forces [40]. It has the form

$$\partial_t f = -\nabla \cdot [\nabla V(x) f] + \sigma \Delta f, \tag{37}$$

where $V(x)$ is a scalar function and $\sigma > 0$ is a constant. The first term on the right-hand side is called the drifted term and the second term is the diffusion term generated by white noise. This equation has the steady-state solution $f = e^{-V(x)/\sigma}$. For convenience, we assume the periodic boundary conditions with $x = (x_1, \ldots, x_d) \in [-1, 1]^d$.

1. Conservation form

Equation (37) can also be written as

$$\partial_t f = \sigma \nabla \cdot [e^{-V/\sigma}\nabla(e^{V/\sigma} f)]. \tag{38}$$

As done for the heat equation, one can introduce the transformation $F(t, x, p) = e^{-p} f(t, x)$ and extend the initial data to $p < 0$ to obtain

$$\partial_t F = \sigma \nabla_x \cdot \{e^{-V/\sigma}\nabla_x[e^{V/\sigma}(-F_p)]\},$$
$$F(0, x, p) = e^{-|p|} f(t, x).$$

Apply the discrete Fourier transformation on both x and p to get

$$\begin{aligned}
&\nabla_x \cdot \{e^{-V/\sigma}\nabla_x[e^{V/\sigma}(-F_p)]\} \\
&= \sum_{l=1}^{d} \partial_{x_l}\{e^{-V/\sigma}\partial_{x_l}[e^{V/\sigma}(-F_p)]\} \\
&= -i\sum_{l=1}^{d}(-i\partial_{x_l})(e^{-V/\sigma}(-i\partial_{x_l})\{e^{V/\sigma}[-(-i\partial_p)]F\}) \\
&\to i\sum_{l=1}^{d}(\boldsymbol{P}_l \otimes I)(\boldsymbol{e}^{-\boldsymbol{V}/\sigma} \otimes I)(\boldsymbol{P}_l \otimes I) \\
&\quad \times (\boldsymbol{e}^{\boldsymbol{V}/\sigma} \otimes I)(I^{\otimes d} \otimes P_\mu)\boldsymbol{F} \\
&= i\sum_{l=1}^{d}(\boldsymbol{P}_l \boldsymbol{e}^{-\boldsymbol{V}/\sigma}\boldsymbol{P}_l \boldsymbol{e}^{\boldsymbol{V}/\sigma} \otimes P_\mu)\boldsymbol{F},
\end{aligned}$$

where $\boldsymbol{P}_l$ is the same as the one given for the heat equation and $\boldsymbol{e}^{\boldsymbol{V}} = \mathrm{diag}(\boldsymbol{g})$ is a diagonal matrix with the diagonal vector given by $\boldsymbol{g} = \sum_j e^{V(x_j)}|\boldsymbol{j}\rangle$. It should be pointed out that $\boldsymbol{e}^{\boldsymbol{V}}$ is not the matrix exponential $e^{\boldsymbol{V}}$, where we have used the bold symbol $\boldsymbol{e}$ to indicate the difference.

Let $\boldsymbol{A}_l = \boldsymbol{P}_l \boldsymbol{e}^{-\boldsymbol{V}/\sigma}\boldsymbol{P}_l$, which are Hermitian matrices. One has the ODE

$$\frac{d}{dt}\boldsymbol{F} = i\sigma\sum_{l=1}^{d}(\boldsymbol{A}_l \boldsymbol{e}^{\boldsymbol{V}/\sigma} \otimes P_\mu)\boldsymbol{F}.$$

Defining $\tilde{\boldsymbol{F}} = (\boldsymbol{e}^{\boldsymbol{V}/(2\sigma)} \otimes F_p^{-1})\boldsymbol{F}$, we get a Hamiltonian system

$$\frac{d}{dt}\tilde{\boldsymbol{F}} = i\boldsymbol{H}_1\tilde{\boldsymbol{F}},$$

where

$$\boldsymbol{H}_1 = (\boldsymbol{B}_1 + \cdots + \boldsymbol{B}_d) \otimes D_\mu,$$

$$\boldsymbol{B}_l = \sigma \boldsymbol{e}^{\boldsymbol{V}/(2\sigma)}\boldsymbol{A}_l \boldsymbol{e}^{\boldsymbol{V}/2\sigma}, \quad \boldsymbol{A}_l = \boldsymbol{P}_l \boldsymbol{e}^{-\boldsymbol{V}/\sigma}\boldsymbol{P}_l.$$

2. Heat equation form

Using the transformation $\psi(t, x) = e^{V/2\sigma} f$ (recall the definition of $\tilde{\boldsymbol{F}}$ in the conservation form), one easily gets that ψ satisfies the imaginary-time Schrödinger or heat equation [41]

$$\partial_t \psi = \sigma \Delta \psi - U(x)\psi, \tag{39}$$

where

$$U(x) := \frac{|\nabla V|^2}{4\sigma} - \frac{1}{2}\Delta V.$$

Since Eq. (39) has the same form of the heat equation in (19), one can introduce the transformation technique to get

$$\frac{d}{dt}\tilde{\boldsymbol{\Psi}}(t) = i\boldsymbol{H}_2\tilde{\boldsymbol{\Psi}}(t),$$

where

$$\boldsymbol{H}_2 = \left[\sigma\left(\boldsymbol{P}_1^2 + \cdots + \boldsymbol{P}_d^2\right) + \boldsymbol{U}\right] \otimes D_\mu$$

is a Hermitian matrix and $\boldsymbol{U}$ is defined as $\boldsymbol{V}$.

Remark 10. Obviously, it is more time consuming to perform the quantum simulation of the first matrix $\boldsymbol{H}_1$. In fact, $\boldsymbol{B}_l$ is not sparse in the x_l direction; hence the simulation of $\boldsymbol{H}_1$ is not sparse along x_l direction. However, $\boldsymbol{P}_l^2$ in $\boldsymbol{H}_2$ can be efficiently implemented in the frequency space by using the quantum Fourier transform.

C. Linear Boltzmann equation

In this section we consider the linear Boltzmann equation with isotropic scattering [42]

$$\partial_t f + \xi \cdot \nabla_x f = \frac{1}{|\Omega|}\int_\Omega f(t,x,\xi')d\xi' - f,$$
$$f(0,x,\xi) = f_0(x,\xi),$$

where $f = f(t,x,\xi)$, $x = (x_1,\ldots,x_d) \in [-1,1]^d$, and $\xi = (\xi_1,\ldots,\xi_{d-1})$ is a vector on the unit sphere $\mathbb{S}^{d-1}$ in $\mathbb{R}^d$. We assume the periodic boundary conditions are imposed.

1. Hamiltonian system

Proceeding as in the previous sections, we introduce the warped phase transformation

$$F(t,x,\xi,p) = e^{-p}f(t,x,\xi), \quad p > 0,$$

with the initial data symmetrically extended to $p < 0$, and find that F solves

$$\partial_t F + \xi \cdot \nabla_x F = -\left(\frac{1}{|\Omega|}\int_\Omega \partial_p F(t,x,\xi',p)d\xi' - \partial_p F\right),$$
$$F(0,x,\xi,p) = F_0(x,\xi,p) := e^{-|p|}f_0(x,\xi).$$

We use the discrete-ordinate method to discretize the integral. Let (w_k,ξ_k) be the quadrature weights and points, where $\xi_k = (\xi_{k1},\ldots,\xi_{kd})$ and $1 \leqslant k \leqslant N$. One has the semidiscrete system

$$\partial_t F_m + \xi_m \cdot \nabla_x F_m = -\left(\sum_k w_k \partial_p F_k - \partial_p F_m\right),$$

where $m = 1,2,\ldots,N$ and $F_m = F_m(t,x,p) := F(t,x,\xi_m,p)$. Let

$$\boldsymbol{F}_m = \sum_{\boldsymbol{j},j_p} \boldsymbol{F}_m(t,x_{\boldsymbol{j}},p_{j_p})|\boldsymbol{j},j_p\rangle, \quad \boldsymbol{j} = (j_1,\ldots,j_d),$$

which is a column vector of M^{d+1} entries. Taking the discrete Fourier transformation on both x and p yields

$$\frac{d}{dt}\boldsymbol{F}_m + i\sum_{l=1}^d \xi_{ml}(\boldsymbol{P}_l \otimes I)\boldsymbol{F}_m = -i(I^{\otimes^d} \otimes P_\mu)\left(\sum_{k=1}^N w_k \boldsymbol{F}_k - \boldsymbol{F}_m\right),$$

where $m = 1,\ldots,N$ and $\boldsymbol{P}_l = I^{\otimes^{l-1}} \otimes P_\mu \otimes I^{\otimes^{d-l}}$. The above system can be rewritten as

$$\frac{d}{dt}\boldsymbol{F} + i\sum_{l=1}^d (\Lambda_{\xi_l} \otimes \boldsymbol{P}_l \otimes I)\boldsymbol{F} = -i(W - I) \otimes I^{\otimes^d} \otimes P_\mu \boldsymbol{F}, \tag{40}$$

where $\boldsymbol{F} = [\boldsymbol{F}_1;\ldots;\boldsymbol{F}_N]$, with the semicolon indicating the straightening of $\{\boldsymbol{F}_i\}_{i\geqslant 1}$ into a column vector, $\Lambda_{\xi_l} = \mathrm{diag}(\xi_{1l},\xi_{2l},\ldots,\xi_{Nl})$, and $W = \Xi\Lambda_w$, with $\Xi = (\Xi_{ij})_{N\times N}$, $\Xi_{ij} \equiv 1$, $\Lambda_w = \mathrm{diag}(w_1,w_2,\ldots,w_N)$.

Define $\tilde{\boldsymbol{F}} = (\Lambda_w^{1/2} \otimes I^{\otimes^d} \otimes F_p^{-1})\boldsymbol{F}$. One has from (40) that

$$\frac{d}{dt}\tilde{\boldsymbol{F}} + i\sum_{l=1}^d \left(\Lambda_w^{1/2}\Lambda_{\xi_l}\Lambda_w^{-1/2} \otimes \boldsymbol{P}_l \otimes I\right)\tilde{\boldsymbol{F}} = -i\left(\Lambda_w^{1/2}\Xi\Lambda_w^{1/2} - I\right) \otimes I^{\otimes^d} \otimes D_\mu)\tilde{\boldsymbol{F}}.$$

Noting that $\Lambda_w^{1/2}\Lambda_{\xi_l}\Lambda_w^{-1/2} = \Lambda_{\xi_l}$, we finally derive a Hamiltonian system

$$\frac{d}{dt}\tilde{\boldsymbol{F}} = -i\boldsymbol{H}\tilde{\boldsymbol{F}}, \tag{41}$$

where

$$\boldsymbol{H} = \sum_{l=1}^d (\Lambda_{\xi_l} \otimes \boldsymbol{P}_l \otimes I) + \left(\Lambda_w^{1/2}\Xi\Lambda_w^{1/2} - I\right) \otimes I^{\otimes^d} \otimes D_\mu$$

is a Hermitian matrix.

2. Quantum simulation

For the Hamiltonian system (41), one can solve

$$\frac{d}{dt}\tilde{\boldsymbol{F}} = -i\boldsymbol{H}_\xi\tilde{\boldsymbol{F}}, \quad \boldsymbol{H}_\xi = \sum_{l=1}^d (\Lambda_{\xi_l} \otimes \boldsymbol{P}_l \otimes I)$$

for one time step, followed by solving

$$\frac{d}{dt}\tilde{\boldsymbol{F}} = -i\boldsymbol{H}_w\tilde{\boldsymbol{F}}, \quad \boldsymbol{H}_w = \left(\Lambda_w^{1/2}\Xi\Lambda_w^{1/2} - I\right) \otimes I^{\otimes^d} \otimes D_\mu$$

again for one time step. By introducing $\tilde{\boldsymbol{c}} = (I^{\otimes^d} \otimes F_x^{-1} \otimes I)\tilde{\boldsymbol{F}}$, where $F_x = \Phi^{\otimes^d}$, the system in the first step is transformed into

$$\frac{d}{dt}\tilde{\boldsymbol{c}} = -i\boldsymbol{H}_\xi\tilde{\boldsymbol{c}}, \quad \boldsymbol{H}_{D,\xi} = \sum_{l=1}^d \left(\Lambda_{\xi_l} \otimes \boldsymbol{D}_l^\mu \otimes I\right),$$

where $\boldsymbol{H}_{D,\xi}$ is a diagonal matrix.

Theorem 7. The solution to the Boltzmann equation can be simulated with gate complexity given by

$$N_{\text{gates}} = \tilde{O}\left(\frac{m_H N^2}{\Delta p} + \frac{m_H}{\Delta x}\right) + O\left(\frac{T}{\Delta t} dm \log m\right) + O(m_p \log m_p),$$

where N is the number of quadrature points and $m_H = dm + m_p$.

Proof. Given the initial state of $\tilde{\boldsymbol{F}}^0$ and applying the inverse QFT to the x register, one gets $\tilde{\boldsymbol{c}}^0$. At each time step, one needs to consider the procedure

$$\tilde{\boldsymbol{c}}^n \xrightarrow{e^{-i\boldsymbol{H}_{D,\xi}\Delta t}} \tilde{\boldsymbol{c}}^* \xrightarrow{I^{\otimes d}\otimes F_x\otimes I} \tilde{\boldsymbol{F}}^* \xrightarrow{e^{-i\boldsymbol{H}_w\Delta t}} \tilde{\boldsymbol{F}}^{n+1} \xrightarrow{I^{\otimes d}\otimes F_x^{-1}\otimes I} \tilde{\boldsymbol{c}}^{n+1}.$$

By Lemma 1, $e^{-i\boldsymbol{H}_w\Delta t}$ can be simulated with

$$N_{\text{gates}}(e^{-i\boldsymbol{H}_w\Delta t}) = \tilde{O}(\Delta t m_H s(\boldsymbol{H}_w)\|\boldsymbol{H}_w\|_{\max}) = \tilde{O}\left(\Delta t \frac{m_H N^2}{\Delta p}\right),$$

where $s(\boldsymbol{H}_w) = N$ and $\|\boldsymbol{H}_w\|_{\max} \lesssim NM_p$. Similarly, $e^{-i\boldsymbol{H}_{D,\xi}\Delta t}$ can be simulated with

$$N_{\text{gates}}(e^{-i\boldsymbol{H}_{D,\xi}\Delta t}) = \tilde{O}(\Delta t m_H s(\boldsymbol{H}_{D,\xi})\|\boldsymbol{H}_{D,\xi}\|_{\max}) = \tilde{O}\left(\frac{\Delta t m_H}{\Delta x}\right),$$

where $s(\boldsymbol{H}_{D,\xi}) = 1$ and $\|\boldsymbol{H}_{D,\xi}\|_{\max} \leqslant M = O(1/\Delta x)$.

It is known that the quantum Fourier transforms F_x can be implemented using $O(dm \log m)$ gates. Therefore, the gate complexity required to iterate to the nth step for the system (41) is

$$N_{\text{gates}} = n\left[\tilde{O}\left(\frac{\Delta t m_H N^2}{\Delta p} + \frac{\Delta t m_H}{\Delta x}\right) + O(dm \log m)\right] = \tilde{O}\left(\frac{m_H N^2}{\Delta p} + \frac{m_H}{\Delta x}\right) + O\left(\frac{T}{\Delta t} dm \log m\right).$$

The proof is complete. ∎

We remark that $\|H\|_{\max}$ in Algorithm II (see Lemma 1) is not significantly amplified by the auxiliary variable p. This is due to the presence of the first-order derivative with respect to x in the convection term. This is very different from our approach to the linear convection equation in Sec. II B, where the second-order derivative with respect to p is included, which leads to $1/\Delta p^2$ as the multiplicative factor in the time complexity if Algorithm II is used.

D. Vlasov-Fokker-Planck equation

We now present an example in which the first use of the warped phase transformation fails.

Consider $f = f(t, x, \xi) > 0$ satisfying the Vlasov-Fokker-Planck (VFK) equation (also called the Klein-Kramers-Chandrasekhar equation) [43]

$$\partial_t f + \xi \cdot \nabla_x f - \nabla V(x) \cdot \nabla_\xi f = \nabla_\xi \cdot (\xi f + \nabla_\xi f),$$

where $x, \xi \in \mathbb{R}^d$. Introducing

$$M = M(\xi) := e^{-|\xi|^2/2},$$

the VFK equation can be written as

$$\partial_t f + \xi \cdot \nabla_x f - \nabla V(x) \cdot \nabla_\xi f = \nabla_\xi \cdot [M \nabla_\xi (M^{-1} f)].$$

Let $W(\xi) = |\xi|^2/2$ and $\sigma = 1$. Then $M = e^{-W/\sigma}$ and the above equation can be rewritten as

$$\partial_t f + \xi \cdot \nabla_x f - \nabla V(x) \cdot \nabla_\xi f = \sigma \nabla_\xi \cdot [e^{-W/\sigma} \nabla_\xi (e^{W/\sigma} f)],$$

where the right-hand side has the same form as that of Eq. (38).

It seems that we can also try the conservation form and heat equation form as for the Fokker-Planck equation; however, the third term $\nabla V(x) \cdot \nabla_\xi f$ on the left-hand side makes the direct use of the warped phase transformation no longer work. Let us consider the conservation form as an example. Introduce the transformation $F(t, x, \xi, p) = e^{-p} f(t, x, \xi)$ with periodic extension of the initial data to get

$$\partial_t F + \xi \cdot \nabla_x F - \nabla V(x) \cdot \nabla_\xi f = \nabla_\xi \cdot \{e^{-W} \nabla_\xi [e^W (-F_p)]\}.$$

Applying the discrete Fourier transformation on all variables, one easily obtains

$$\xi \cdot \nabla_x F = \sum_{l=1}^d \xi_l \partial_{x_l} F \to i \sum_{l=1}^d (\boldsymbol{P}_l \otimes \boldsymbol{D}_l \otimes I)(\boldsymbol{F}_x \otimes \boldsymbol{F}_\xi \otimes \boldsymbol{F}_p),$$

$$\nabla V(x) \cdot \nabla_\xi F = \sum_{l=1}^d \partial_{x_l} V \partial_{\xi_l} F \to i \sum_{l=1}^d (\boldsymbol{V}_l \otimes \boldsymbol{P}_l \otimes I)(\boldsymbol{F}_x \otimes \boldsymbol{F}_\xi \otimes \boldsymbol{F}_p),$$

where $\boldsymbol{V}_l = \sum_{\boldsymbol{i}} \partial_{x_l} V(x_{\boldsymbol{i}}) |\boldsymbol{i}\rangle$ is a diagonal matrix and

$$\begin{aligned}
&\nabla_\xi \cdot \{e^{-W} \nabla_\xi [e^W (-F_p)]\} \\
&= \sum_{l=1}^d \partial_{\xi_l} \{e^{-W} \partial_{\xi_l} [e^W (-F_p)]\} \\
&= i \sum_{l=1}^d (-i\partial_{\xi_l}) \{e^{-W} (-i\partial_{\xi_l}) [e^W (-i\partial_p F)]\} \\
&\to i \sum_{l=1}^d (I^{\otimes d} \otimes \boldsymbol{P}_l \otimes I)(I^{\otimes d} \otimes \boldsymbol{e}^{-\boldsymbol{W}_\xi} \otimes I) \\
&\quad \times (I^{\otimes d} \otimes \boldsymbol{P}_l \otimes I)(I^{\otimes d} \otimes \boldsymbol{e}^{\boldsymbol{W}_\xi} \otimes I)(I^{\otimes d} \otimes I^{\otimes d} \otimes P_\mu)\boldsymbol{F} \\
&= i \sum_{l=1}^d (I^{\otimes d} \otimes \boldsymbol{P}_l \boldsymbol{e}^{-\boldsymbol{W}_\xi} \boldsymbol{P}_l \boldsymbol{e}^{\boldsymbol{W}_\xi} \otimes P_\mu)\boldsymbol{F},
\end{aligned}$$

where $\boldsymbol{e}^{-\boldsymbol{W}_\xi}$ is a diagonal matrix with the diagonal vector given by $\sum_j W(\xi_j)|\boldsymbol{j}\rangle$. Let $\boldsymbol{B}_l = \boldsymbol{P}_l \boldsymbol{e}^{-\boldsymbol{W}_\xi} \boldsymbol{P}_l$, which are Hermitian matrices. One gets the ODE

$$\frac{d}{dt}\boldsymbol{F} = i \sum_{l=1}^d (-\boldsymbol{P}_l \otimes \boldsymbol{D}_l \otimes I + \boldsymbol{V}_l \otimes \boldsymbol{P}_l \otimes I + I^{\otimes d} \otimes \boldsymbol{B}_l \boldsymbol{e}^{\boldsymbol{W}_\xi} \otimes P_\mu)\boldsymbol{F}.$$

For this system, it does not work by introducing the new variables $\tilde{\boldsymbol{F}} = \boldsymbol{I}^{\otimes^d} \otimes \boldsymbol{e}^{\boldsymbol{W}_\xi/2} \otimes F_p^{-1}$ since the first term on the right-hand side will change to $\boldsymbol{P}_l \otimes \boldsymbol{e}^{\boldsymbol{W}_\xi/2}\boldsymbol{D}_l\boldsymbol{e}^{-\boldsymbol{W}_\xi/2} \otimes \boldsymbol{I}$, which are not Hermitian matrices. To resolve this problem, one can first derive an ODE system resulting from the discretization of x and ξ variables and then apply the generalized approach in Sec. III A, with the details omitted.

E. Liouville representation for nonlinear ODEs

Consider the nonlinear ODEs

$$\frac{dq(t)}{dt} = F(q(t)), \quad q(0) = q_0, \quad q = [q_1, \ldots, q_d]^T. \tag{42}$$

For $x = (x_1, \ldots, x_d)$, let $\delta(x) = \Pi_{i=1}^d \delta(x_i)$ be the Dirac delta distribution. The Liouville equation corresponding to (42) can be derived by considering a function $\rho(t, x) : \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}$, defined by

$$\rho(t, x) = \delta(x - q(t)),$$

which represents the probability distribution in space x that corresponds to the solution $x = q$. By the properties of the δ function, one obtains the solution of (42) by taking the moment

$$q(t) = \int x\delta(x - q(t))dx = \int x\rho(t, x)dx. \tag{43}$$

To this end, we can characterize the dynamics of $\rho(t, x)$ and find the solution $q(t)$ via (43).

One can check that ρ satisfies, in the weak sense, the linear $(d + 1)$-dimensional PDEs

$$\partial_t \rho(t, x) + \nabla_x \cdot [F(x)\rho(t, x)] = 0,$$
$$\rho_0(x) := \rho(0, x) = \delta(x - q_0).$$

Since the initial data involve a δ function, we consider the following problem with the smoothed initial data [26]:

$$\partial_t u(t, x) + \nabla \cdot [F(x)u(t, x)] = 0,$$
$$u(0, x) = u_0(x) := \delta_\omega(x - q_0).$$

Let $u_1 = F_i(x)u(t, x)$. According to the notation for the Fourier spectral method, one has

$$-i\partial_{x_i} u_1 \to \hat{P}_i^d \boldsymbol{u}_1 = \boldsymbol{P}_i \boldsymbol{u}_1 = \boldsymbol{P}_i \Lambda_{F_i} \boldsymbol{u},$$

where $\Lambda_{F_i} = \mathrm{diag}(\boldsymbol{F}_i)$ is a diagonal matrix and $\boldsymbol{F}_i = \sum_j F_i(x_j)|\boldsymbol{j}\rangle$. The resulting system of ordinary differential equations is

$$\frac{d}{dt}\boldsymbol{u}(t) = -iA\boldsymbol{u}(t),$$
$$\boldsymbol{u}(0) = \boldsymbol{u}^0 = u_0(x_j), \tag{44}$$

where $A = \sum_{i=1}^d A_i$, with $A_i = \boldsymbol{P}_i \Lambda_{F_i}$. In Ref. [26] we proposed a quantum simulation method for the above problem by using the time-splitting approach. However, the simulation protocol there is different from the traditional time-marching Hamiltonian simulation since nonunitary procedures are involved at each time step, leading to exponential increase of the cost. By the generalized framework in Sec. III A, we are ready to recover a true Hamiltonian simulation.

Let us still consider the splitting in [26]. In fact, there is no need to apply the splitting since each $\boldsymbol{A}_i$ has the similar structure. Here we just indicate that our protocol also works for the splitting. The evolution of (44) can be written as

$$\boldsymbol{w}(t + \Delta t) = e^{-i(A_1 + \cdots + A_d)\Delta t}\boldsymbol{w}(t),$$

in which the evolutionary operator can be approximated by the first-order product formula

$$U_{\Delta t} = e^{-iA_d \Delta t} \cdots e^{-iA_1 \Delta t}. \tag{45}$$

Then the problem is reduced to the simulation of each A_j, where A_j is not necessarily symmetric. Consider the decomposition $-iA_j = H_1^j + iH_2^j$, where H_1^j and H_2^j are Hermitian matrices. Following the discussion in Sec. III A, one can construct a Hamiltonian system associated with the evolutionary operator $e^{-iA_j\Delta t}$,

$$\frac{d}{dt}\boldsymbol{w}(t) = i\big(-H_1^j \otimes P_\mu + H_2^j \otimes I\big)\boldsymbol{w}(t),$$

where $\boldsymbol{w}$ encodes all the grid values of the variable $\boldsymbol{v}$ defined by (28) [note that $\boldsymbol{v}$ corresponds to the original system (44)]. In contrast to the change of variables in [26], the transformations here are the same in every time step. We therefore restore the time-marching Hamiltonian simulation for the new variables.

V. SUMMARY

We provided technical details for a simple method for solving general linear PDEs using quantum simulation. The idea is to introduce a simple warped phase transformation that can translate the PDEs into a system of Schrödinger equations without employing more sophisticated methods. This enables quantum simulation to be applied to these PDEs (and also ODEs).

This approach was applied to several typical examples, including the heat, convection, Fokker-Planck, linear Boltzmann, and Black-Scholes equations. It can be extended to general linear partial differential equations including the Vlasov-Fokker-Planck equation and the Liouville representation equation for nonlinear ordinary differential equations. It has the potential to find a variety of applications in time-dependent or -independent boundary-value problems, artificial [23] and physical interface boundary conditions [22], and even linear algebra problems [24]. It is also naturally applicable in the continuous-variable framework [28].

ACKNOWLEDGMENTS

S.J. was partially supported by NSFC Grant No. 12031013, Shanghai Municipal Science and Technology Major Project No. 2021SHZDZX0102, and the Innovation Program of Shanghai Municipal Education Commission (Grant No. 2021-01-07-00-02-E00087). N.L. acknowledges funding from the Science and Technology Program of Shanghai, China (Grant No. 21JC1402900). Y.Y. was partially supported by China Postdoctoral Science Foundation (Grant No. 2022M712080). The authors were also supported by the Fundamental Research Funds for the Central Universities.

[1] P. Arrighi, G. Di Molfetta, I. Márquez-Martin, and A. Pérez, Dirac equation as a quantum walk over the honeycomb and triangular lattices, Phys. Rev. A **97**, 062111 (2018).

[2] P. Arrighi, G. Di Molfetta, and S. Facchini, Quantum walking in curved spacetime: Discrete metric, Quantum **2**, 84 (2018).

[3] P. Arrighi, V. Nesme, and M. Forets, The Dirac equation as a quantum walk: Higher dimensions, observational convergence, J. Phys. A: Math. Theor. **47**, 465302 (2014).

[4] D. W. Berry, High-order quantum algorithm for solving linear differential equations, J. Phys. A: Math. Theor. **47**, 105301 (2014).

[5] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, Quantum algorithm and circuit design solving the Poisson equation, New J. Phys. **15**, 013021 (2013).

[6] A. M. Childs, J. P. Liu, and A. Ostrander, High-precision quantum algorithms for partial differential equations, Quantum **5**, 574 (2021).

[7] A. W. Childs and J. Liu, Quantum spectral methods for differential equations, Commun. Math. Phys. **375**, 1427 (2020).

[8] P. C. S. Costa, S. Jordan, and A. Ostrander, Quantum algorithm for simulating the wave equation, Phys. Rev. A **99**, 012323 (2019).

[9] A. Engel, G. Smith, and S. E. Parker, Quantum algorithm for the Vlasov equation, Phys. Rev. A **100**, 062315 (2019).

[10] F. Golse, S. Jin, and N. Liu, Quantum algorithms for uncertainty quantification: Application to partial differential equations, arXiv:2209.11220.

[11] S. Jin and N. Liu, Quantum algorithms for computing observables of nonlinear partial differential equations, arXiv:2202.07834.

[12] S. Jin, N. Liu, and Y. Yu, Time complexity analysis of quantum difference methods for linear high dimensional and multiscale partial differential equations, J. Comput. Phys. **471**, 111641 (2022).

[13] N. Linden, A. Montanaro, and C. Shao, Quantum vs. classical algorithms for solving the heat equation, Commun math. Phys. **395**, 601 (2022).

[14] A. Montanaro and S. Pallister, Quantum algorithms and the finite element method, Phys. Rev. A **93**, 032324 (2016).

[15] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, Quantum algorithm for linear differential equations with exponentially improved dependence on precision, Commun. Math. Phys. **356**, 1057 (2017).

[16] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, SIAM J. Comput. **46**, 1920 (2017).

[17] P. C. S. Costa, D. An, Y. A. Sanders, Y. Su, R. Babbush, and D. W. Berry, Optimal scaling quantum linear systems solver via discrete adiabatic theorem, PRX Quantum **3**, 040303 (2022).

[18] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009).

[19] Y. Subaşi and R. D. Somma, Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing, Phys. Rev. Lett. **122**, 060504 (2019).

[20] D. An, J. Liu, D. Wang, and Q. Zhao, A theory of quantum differential equation solvers: Limitations and fast-forwarding, arXiv:2211.05246.

[21] S. Jin, N. Liu, and Y. Yu, Quantum simulation of partial differential equations via Schrödingerisation, arXiv:2212.13969.

[22] S. Jin, N. Liu, X. Li, and Y. Yu, Quantum simulation for partial differential equations with physical boundary or interface conditions, arXiv:2305.02710.

[23] S. Jin, N. Liu, X. Li, and Y. Yu, Quantum simulation for quantum dynamics with artificial boundary conditions, arXiv:2304.00667.

[24] S. Jin and N. Liu, Quantum simulation of discrete linear dynamical systems and simple iterative methods in linear algebra via Schrödingerisation, arXiv:2304.02865.

[25] I. Y. Dodin and E. A. Startsev, On applications of quantum computing to plasma simulations, Phys. Plasmas **28**, 092101 (2021).

[26] S. Jin, N. Liu, and Y. Yu, Time complexity analysis of quantum algorithms via linear representations for nonlinear ordinary and partial differential equations, J. Comput. Phys. **487**, 112149 (2023).

[27] I. Joseph, Koopman–von Neumann approach to quantum simulation of nonlinear classical dynamics, Phys. Rev. Res. **2**, 043102 (2020).

[28] S. Jin and N. Liu, Analog quantum simulation of partial differential equations, arXiv:2308.00646.

[29] S. Jin, X. Li, and N. Liu, Quantum simulation in the semiclassical regime, Quantum **6**, 739 (2022).

[30] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik, Polynomial-time quantum algorithm for the simulation of chemical dynamics, Proc. Natl. Acad. Sci. USA **105**, 18681 (2008).

[31] D. W. Berry, A. M. Childs, and R. Kothari, *2015 IEEE 56th Annual Symposium on Foundations of Computer Science, Berkeley, 2015* (IEEE, Piscataway, 2015), pp. 792–809.

[32] D. An, D. Fang, and L. Lin, Time-dependent unbounded Hamiltonian simulation with vector norm scaling, Quantum **5**, 459 (2021).

[33] D. An, D. Fang, and L. Lin, Time-dependent Hamiltonian simulation of highly oscillatory dynamics and superconvergence for Schrödinger equation, Quantum **6**, 690 (2022).

[34] D. W. Berry, A. M. Childs, Y. Su, X. Wang, and N. Wiebe, Time-dependent Hamiltonian simulation with l^1-norm scaling, Quantum **4**, 254 (2020).

[35] J. Gonzalez-Conde, A. Rodriguez-Rozas, E. Solano, and M. Sanz, Simulating option price dynamics with exponential quantum speedup, arXiv:2101.04023.

[36] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, Phoenix, 2019* (ACM, New York, 2019), pp. 193–204.

[37] 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).

[38] M. Motta, C. Sun, A. T. K. Tan, M. J. O'Rourke, E. Ye, A. J. Minnich, F. G. S. L. Brandão, and G. K.-L. Chan, Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution, Nat. Phys. **16**, 205 (2020).

[39] K. Seki and S. Yunoki, Quantum power method by a superposition of time-evolved states, PRX Quantum **2**, 010333 (2021).

[40] G. A. Pavliotis, *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*, Texts in Applied Mathematics Vol. 60 (Springer, New York, 2014).

[41] P. A. Markowich and C. Villani, *Proceedings of the Sixth Workshop on Partial Differential Equations, Rio de Janeiro, 1999, Part II* [Mat. Contemp. **19**, 1 (2000)].

[42] S. Chandrasekhar, *Radiative Transfer* (Dover, New York, 1960).

[43] H. P. McKean, Jr., A class of Markov processes associated with nonlinear parabolic equations, Proc. Natl. Acad. Sci. USA **56**, 1907 (1966).