Hamiltonian simulation using the quantum singular-value transformation: Complexity analysis and application to the linearized Vlasov-Poisson equation

Kiichiro Toyoizumi,[1,*] Naoki Yamamoto,[1,2] and Kazuo Hoshino[1]

[1]*Department of Applied Physics and Physico-Informatics, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522, Japan*
[2]*Quantum Computing Center, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522, Japan*

(Received 24 April 2023; revised 13 October 2023; accepted 18 December 2023; published 25 January 2024)

Quantum computing can be used to speed up the simulation time (more precisely, the number of queries of the algorithm) for physical systems; one such promising approach is the Hamiltonian simulation (HS) algorithm. Recently, it was proven that the quantum singular-value transformation (QSVT) achieves the minimum simulation time for HS. An important subroutine of the QSVT-based HS algorithm is the amplitude amplification operation, which can be realized via the oblivious amplitude amplification or the fixed-point amplitude amplification in the QSVT framework. In this work we execute a detailed analysis of the error and number of queries of the QSVT-based HS and show that the oblivious method is better than the fixed-point one in the sense of simulation time. Based on this finding, we apply the QSVT-based HS to the one-dimensional linearized Vlasov-Poisson equation and demonstrate that the linear Landau damping can be successfully simulated.

DOI: 10.1103/PhysRevA.109.012430

I. INTRODUCTION

A. Background

Quantum computers are expected to outperform classical counterparts in some problems. Several quantum algorithms have obtained speedups over classical ones, such as the Grover search algorithm [1], Shor's algorithm for integer factorization [2], and the Harrow-Hassidim-Lloyd algorithm [3,4]. Quantum computing also gives us algorithms for solving physics problems. In particular, the algorithms [5,6] realize exponential speedup for the simulation of quantum systems. This seems natural since quantum computing is based on quantum mechanics. In recent years, some quantum algorithms for simulating classical physical systems have been developed, such as the Navier-Stokes equation [7,8], plasma equations [9–12], the Poisson equation [13,14], and the wave equation [15,16].

One of the quantum algorithms for simulating physical systems is a Hamiltonian simulation (HS) algorithm [17–26], which implements $U = \exp(-iHt)$, where H is a time-independent Hamiltonian and t is an evolution time. The optimal HS result was shown by Low and Chuang using quantum signal processing (QSP) [27,28]. This result has been generalized to the quantum singular-value transformation (QSVT) in Ref. [29]. The QSVT is a quantum algorithm for applying a polynomial transformation $P^{(\mathrm{SV})}(A)$ to the singular values of a given matrix A, called the singular-value transformation. Notably, the QSVT can formulate major quantum algorithms in a unified way, such as the Grover search, phase estimation [30], matrix inversion, quantum walks [31], and the HS algorithm. Therefore, the QSVT is called a grand unification of quantum algorithms in Ref. [32].

The HS algorithm using the QSVT has been proposed in Refs. [29,32]. The algorithm includes an amplitude amplification algorithm that can be implemented by the QSVT as a subroutine. There are two QSVT-based amplitude amplification algorithms proposed for HS: One is the oblivious amplitude amplification (OAA) algorithm [29] and the other is the fixed-point amplitude amplification (FPAA) algorithm [32]. However, there have been no discussions to compare those two schemes from either theoretical or numerical viewpoints. It would be helpful to clarify which one is preferable when the QSVT-based HS algorithm is applied to physical systems.

B. Contribution of this paper

We elaborate the QSVT-based HS using explicit quantum circuits and discuss the approximation error and query complexity. As a result, the number of queries for the OAA-based HS scales as $O(t + \log(1/\varepsilon))$, whereas the FPAA-based one scales as $O(t\log(1/\varepsilon) + \log^2(1/\varepsilon))$, where t is an evolution time and ε is an error tolerance. To support this fact, we perform numerical experiments: We plot the number of queries for a wide range of parameters t and ε; we curve fit the data to identify the constant factors and coefficients of the number of queries hidden behind the asymptotic scalings. Our findings indicate that the OAA-based method is both theoretically and numerically more advantageous than the FPAA-based method in the sense of the number of queries. Importantly, this advantage is consistent, independent of the type of the Hamiltonian.

To demonstrate the effectiveness of the QSVT-based HS combined with the above-described detailed analysis, we apply the OAA-based HS to the simulation of the linearized Vlasov-Poisson system. This system can be transformed into the same form of unitary time evolution as of quantum systems [9]. In addition to the simulation, we discuss several issues unaddressed in previous studies on quantum algorithms for plasma simulation [9–12]: We discuss the computational complexity of extending the evolution time using sequential

[*] toyoizumi@ppl.appi.keio.ac.jp

©2024 American Physical Society

short HS circuits, we propose an algorithm to obtain a quantity related to the distribution function, and we provide explicit quantum circuits for the higher-dimensional systems. Furthermore, we show a potential advantage of applying the HS to physical systems compared to the classical Euler method.

C. Comparison to prior work

The QSVT-based HS algorithms were introduced in Refs. [29,32]. The authors of Ref. [29] originally proposed the QSVT framework. Within the framework, they developed a method for implementing the exponential function and the OAA algorithm, which constructs the Chebyshev polynomial of the first kind. Combining these methods, they realized the QSVT-based HS and showed that its asymptotic complexity is consistent with the result of the HS by Low and Chuang [27,28], known to be optimal. The authors of Ref. [32] reviewed that several major quantum algorithms can be described in a unified way within the QSVT framework and suggested for the HS the use of FPAA, which constructs the approximate polynomial of the sign function. These authors independently proposed using OAA and FPAA for the HS, respectively, with rough analyses of the approximation error and query complexity. However, a theoretical or numerical comparison of these methods remains lacking.

To date, no investigation exists to compare OAA and FPAA in a non-QSVT framework. This is probably because these algorithms were originally developed for distinct purposes. The OAA algorithm was initially developed to simulate a sparse Hamiltonian evolution [21,23], achieving amplitude amplification without the reflection operator about an unknown initial state. It is also used to decompose single-qubit unitaries [33] and compute matrix products for nonunitary matrices [34]. On the other hand, the FPAA algorithm is an algorithm that ensures amplitude amplification regardless of an unknown amplitude. Notably, several non-QSVT-based FPAA algorithms have been developed, such as the $\pi/3$ algorithm [35], the measurement-based algorithm [36], and the FPAA technique by Yoder *et al.* [37].

Several studies have addressed quantum algorithms for plasma simulations. The authors of Ref. [10] conducted an extensive survey on applying quantum computers to plasma simulations. The authors of Ref. [9] introduced the quantum algorithm for calculating the time evolution of the one-dimensional linearized Vlasov-Poisson system using the HS algorithm by Low and Chuang [27,28]. They concluded that the algorithm achieves an exponential speedup for a velocity grid size. They also discussed an estimation of the electric field with the quantum amplitude estimation algorithm [38] and simulated its linear Landau damping. While the state's amplitude comprises the electric field and the distribution function, no method has been proposed to extract the latter's quantity from the state. They indicated that the above findings can be extended to systems with higher dimensions, yet without providing explicit circuits. The authors of Ref. [12] examined the computational complexity for a system size of a quantum algorithm for the one-dimensional Vlasov-Poisson system with collisions. They adopted the Hermite representation, reducing the equations to a linear ordinary differential equation problem, distinct from our work and Ref. [9]. The authors of Ref. [11] implemented the HS algorithm for one-dimensional cold plasma waves, dividing the HS circuit into shorter circuits to avoid a large evolution time t, yet without discussing its cost.

D. Organization of the paper

The rest of this paper is organized as follows. In Sec. II we present a brief description of the QSVT with application to the trigonometric functions; then we show the transformation from the linearized Vlasov-Poisson system to a form of the Schrödinger equation. In Sec. III we discuss the error and query complexity of the QSVT-based HS. The quantum algorithm for the linearized Vlasov-Poisson system is discussed in Sec. IV. We show the numerical results in Sec. V. The paper is summarized in Sec. VI.

II. PRELIMINARIES

A. Quantum singular-value transformation

The quantum singular-value transformation [29,32] is a quantum algorithm for applying a polynomial transformation $P^{(\mathrm{SV})}(A)$ to the singular values of a given matrix A. As mentioned above, the QSVT has been applied to many problems, including the HS. In these problems, a degree-d polynomial P_ε is used to ε approximate the corresponding objective function P. How much quantum speedup is obtained depends on the degree d. Recall that the QSVT generalizes the result of QSP [27,28,39,40]. We present a brief description of the derivation from QSP to QSVT in the Appendix.

We introduce the block encoding [29], which represents a matrix A as the upper left block of a unitary matrix U. Let A be a matrix acting on s qubits and U be a unitary matrix acting on $a+s$ qubits. Then, for $\alpha > 0$ and $\varepsilon > 0$, U is called an (α, a, ε) block encoding of A if

$$\|A - \alpha(\langle 0|_a \otimes I)U(|0\rangle_a \otimes I)\| \leqslant \varepsilon, \tag{1}$$

where $|0\rangle_a = |0\rangle^{\otimes a}$. Note that, since $\|U\| = 1$, we necessarily have $\|A\| \leqslant \alpha + \varepsilon$. If $\varepsilon = 0$, then we can represent A as the upper left block of U,

$$U = \begin{bmatrix} \frac{A}{\alpha} & \cdot \\ \cdot & \cdot \end{bmatrix}, \tag{2}$$

where the dot denotes a matrix with arbitrary elements.

Recall now that the singular-value decomposition of A, that is, any matrix $A \in \mathbb{C}^{m\times n}$, can be decomposed as

$$A = W\Sigma V, \tag{3}$$

where $W \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary matrices and Σ is diagonal and contains the set of non-negative real numbers $\{\sigma_k\}$, called the singular values of A. The matrix A is also expressed as

$$A = \sum_{k=1}^{r} \sigma_k |w_k\rangle \langle v_k|, \tag{4}$$

where $\{|w_k\rangle\}$ and $\{|v_k\rangle\}$ are right and left singular vectors and $r = \mathrm{rank}(A)$.

The singular-value transformation is defined from the singular-value decomposition as follows: For an odd

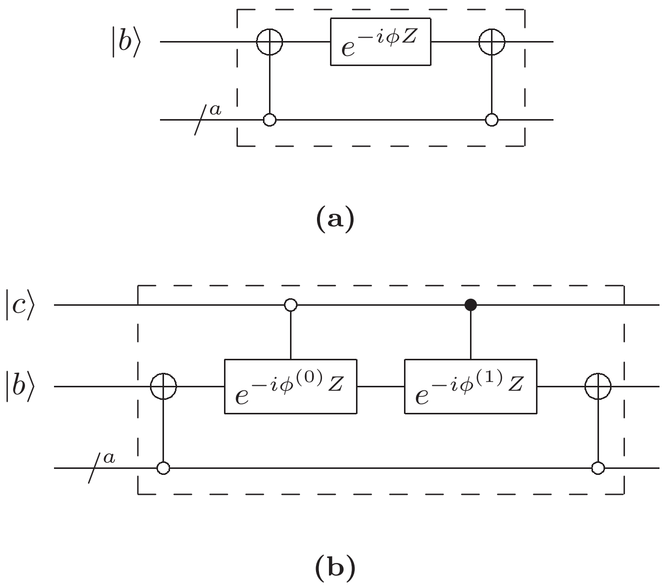

FIG. 1. Quantum circuits used to implement the unitary matrix $\exp(i\phi\Pi)$: (a) $|b\rangle\langle b| \otimes e^{(-1)^b i\phi\Pi}$ and (b) $|cb\rangle\langle cb| \otimes e^{(-1)^b i\phi^{(c)}\Pi}$. (a) A single phase ϕ is used and the series of gates surrounded by dashed lines is denoted by $S_1(\phi)$, which is used in Figs. 3 and 4. (b) Two phases $\phi^{(0)}$ and $\phi^{(1)}$ are used and the series of gates surrounded by dashed lines is denoted by $S_2(\phi^{(0)}, \phi^{(1)})$, which is used in Fig. 2.

polynomial $P \in \mathbb{C}$,

$$P^{(\mathrm{SV})}(A) \equiv \sum_k P(\sigma_k) |w_k\rangle \langle v_k| , \tag{5}$$

and for an even polynomial $P \in \mathbb{C}$,

$$P^{(\mathrm{SV})}(A) \equiv \sum_k P(\sigma_k) |v_k\rangle \langle v_k| . \tag{6}$$

If A is Hermitian and positive semidefinite, then $P^{(\mathrm{SV})}(A)$ is equal to the eigenvalue transformation $P(A)$.

Suppose that U is an $(\alpha, a, 0)$ block encoding of A as in Eq. (2). Then a unitary matrix U_Φ called the alternating phase modulation sequence in Ref. [29] is defined as follows: For odd d,

$$U_\Phi \equiv e^{i\phi_0\Pi} U e^{i\phi_1\Pi} \prod_{k=1}^{(d-1)/2} \left(U^\dagger e^{i\phi_{2k}\Pi} U e^{i\phi_{2k+1}\Pi}\right), \tag{7}$$

and for even d,

$$U_\Phi \equiv e^{i\phi_0\Pi} \prod_{k=1}^{d/2} \left(U^\dagger e^{i\phi_{2k-1}\Pi} U e^{i\phi_{2k}\Pi}\right), \tag{8}$$

where $\Phi = \{\phi_0, \phi_1, \dots, \phi_d\} \in \mathbb{R}^{d+1}$ is called the phases and $\Pi = 2|0\rangle_a \langle 0| - I$. The unitary matrix $\exp(i\phi\Pi)$ can be implemented as in Fig. 1. The phases are calculated efficiently from the degree-d polynomial P on a classical computer. The details of the calculation can be found in Refs. [29,41]. In this work we use the code provided in Ref. [42] to calculate the phases.

Given a degree-d polynomial $P \in \mathbb{C}$ and the corresponding phases $\Phi \in \mathbb{R}^{d+1}$, the unitary U_Φ can be represented as a $(1, a, 0)$ block encoding of $P^{(\mathrm{SV})}(A/\alpha)$:

$$U_\Phi = \begin{bmatrix} P^{(\mathrm{SV})}(\frac{A}{\alpha}) & \cdot \\ \cdot & \cdot \end{bmatrix},$$

$$P^{(\mathrm{SV})}\left(\frac{A}{\alpha}\right) = (\langle 0|_a \otimes I) U_\Phi (|0\rangle_a \otimes I). \tag{9}$$

Note that P is not arbitrary and has some constraints; P satisfies the following conditions [29,32]: (i) P has parity d mod 2, (ii) $|P(x)| \leqslant 1\,\forall\, x \in [-1, 1]$, (iii) $|P(x)| \geqslant 1\,\forall\, x \in (-\infty, -1] \cup [1, \infty)$, and (iv) if d is even, then $P(ix)P^*(ix) \geqslant 1\,\forall\, x \in \mathbb{R}$. These conditions are complicated, but taking the real part of P relaxes them, that is, $P_\Re \equiv \mathrm{Re}(P)$ satisfies the following conditions: (v) $P_\Re$ has parity d mod 2 and (vi) $|P_\Re(x)| \leqslant 1\,\forall\, x \in [-1, 1]$; the corresponding phases can be calculated as in [29,32].

B. Applying the QSVT to trigonometric functions

Let U be a $(1, a, 0)$ block encoding of H:

$$U = \begin{bmatrix} H & \cdot \\ \cdot & \cdot \end{bmatrix}, \quad H = (\langle 0|_a \otimes I) U (|0\rangle_a \otimes I), \tag{10}$$

where $\|H\| \leqslant 1$ is a Hermitian matrix that is positive semidefinite. We will discuss later the case in which H is negative and normalized by α. The goal of the HS is to construct a quantum circuit U_{HS} that is a block encoding of $\exp(-iHt)$ using the unitary U, where t is the evolution time. Note that U_{HS} cannot be realized by a single U_Φ using the QSVT, because $\exp(-ixt)$ has no definite parity. To avoid this problem, one can instead apply QSVT to two different functions: $\cos(xt)$ and $\sin(xt)$.

The functions $\cos(xt)$ and $\sin(xt)$ are given by the Jacobi-Anger expansion

$$\cos(xt) = J_0(t) + 2\sum_{k=1}^{\infty} (-1)^k J_{2k}(t) T_{2k}(x), \tag{11}$$

$$\sin(xt) = 2\sum_{k=0}^{\infty} (-1)^k J_{2k+1}(t) T_{2k+1}(x), \tag{12}$$

where $J_m(t)$ is the mth Bessel function of the first kind and $T_k(x)$ is the kth Chebyshev polynomial of the first kind. One can obtain the $\varepsilon_{\mathrm{tri}}$ approximation to $\cos(xt)$ and $\sin(xt)$ by truncating Eqs. (11) and (12) at an index R:

$$\left|\cos(xt) - J_0(t) - 2\sum_{k=1}^{R} (-1)^k J_{2k}(t) T_{2k}(x)\right| \leqslant \varepsilon_{\mathrm{tri}}, \tag{13}$$

$$\left|\sin(xt) - 2\sum_{k=0}^{R} (-1)^k J_{2k+1}(t) T_{2k+1}(x)\right| \leqslant \varepsilon_{\mathrm{tri}}, \tag{14}$$

where $0 < \varepsilon_{\mathrm{tri}} < 1/e$ and

$$R(t, \varepsilon_{\mathrm{tri}}) = \left\lfloor \frac{1}{2} r\left(\frac{et}{2}, \frac{5}{4}\varepsilon_{\mathrm{tri}}\right) \right\rfloor, \tag{15}$$

$$r(t, \varepsilon_{\mathrm{tri}}) = \Theta\left(t + \frac{\ln\left(\frac{1}{\varepsilon_{\mathrm{tri}}}\right)}{\ln\left(e + \ln\left(\frac{1}{\varepsilon_{\mathrm{tri}}}\right)/t\right)}\right)$$
$$\leqslant O(t + \log(1/\varepsilon_{\mathrm{tri}})). \tag{16}$$

For more details see Ref. [29].

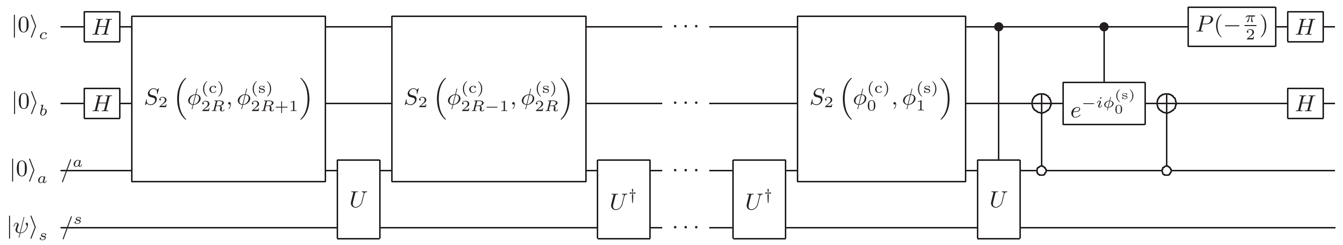
$|0\rangle_c$
$|0\rangle_b$
$|0\rangle_a$
$|\psi\rangle_s$
a
s
H
H
$S_2\left(\phi^{(c)}_{2R},\phi^{(s)}_{2R+1}\right)$
U
$S_2\left(\phi^{(c)}_{2R-1},\phi^{(s)}_{2R}\right)$
$U^\dagger$
$U^\dagger$
$S_2\left(\phi^{(c)}_{0},\phi^{(s)}_{1}\right)$
U
$e^{-i\phi^{(s)}_0}$
$P(-\frac{\pi}{2})$
H
H

FIG. 2. Quantum circuit U_{exp} that is a $(1, a+2, \kappa\varepsilon_{\text{tri}})$ block encoding of $\kappa\exp(-iHt)/2$. The gate S_2 is shown in Fig. 1(b), the unitary U is given by Eq. (10), and the phases $\Phi^{(c)} \in \mathbb{R}^{2R+1}$ and $\Phi^{(s)} \in \mathbb{R}^{2R+2}$ are calculated from the $(2R)$th polynomial $P^{\cos}_{\varepsilon_{\text{tri}},\kappa}(x)$ and the $(2R+1)$th polynomial $P^{\sin}_{\varepsilon_{\text{tri}},\kappa}(x)$ in Eq. (17), where R is given by Eq. (15).

We denote the approximate polynomials of Eqs. (13) and (14) by $P^{\cos}_{\varepsilon_{\text{tri}}}(x)$ and $P^{\sin}_{\varepsilon_{\text{tri}}}(x)$. Since cosine and sine are bounded in magnitude by 1, these polynomials obey $|P^{\cos}_{\varepsilon_{\text{tri}}}(x)|, |P^{\sin}_{\varepsilon_{\text{tri}}}(x)| \leqslant 1+\varepsilon_{\text{tri}}$. Therefore, condition (vi) is violated. Here we introduce rescaled polynomials

$$P^{\cos}_{\varepsilon_{\text{tri}},\kappa}(x) = \kappa P^{\cos}_{\varepsilon_{\text{tri}}}(x), \quad P^{\sin}_{\varepsilon_{\text{tri}},\kappa}(x) = \kappa P^{\sin}_{\varepsilon_{\text{tri}}}(x), \tag{17}$$

where $\kappa = 1/(1+\varepsilon_{\text{tri}})$. These polynomials satisfy the inequality

$$\begin{aligned}&\left|\frac{\kappa}{2}e^{-ixt} - \frac{P^{\cos}_{\varepsilon_{\text{tri}},\kappa}(x) - iP^{\sin}_{\varepsilon_{\text{tri}},\kappa}(x)}{2}\right| \\ &\quad\leqslant \frac{1}{2}\left|\kappa\cos(xt) - P^{\cos}_{\varepsilon_{\text{tri}},\kappa}(x)\right| + \frac{1}{2}\left|\kappa\sin(xt) - P^{\sin}_{\varepsilon_{\text{tri}},\kappa}(x)\right| \\ &\quad\leqslant \frac{\kappa\varepsilon_{\text{tri}}+\kappa\varepsilon_{\text{tri}}}{2} = \kappa\varepsilon_{\text{tri}},\end{aligned} \tag{18}$$

where in the first inequality we used the triangle inequality.

Suppose the phases $\Phi^{(c)} \in \mathbb{R}^{2R+1}$ and $\Phi^{(s)} \in \mathbb{R}^{2R+2}$ are calculated from the $(2R)$th polynomial $P^{\cos}_{\varepsilon_{\text{tri}},\kappa}(x)$ and the $(2R+1)$th polynomial $P^{\sin}_{\varepsilon_{\text{tri}},\kappa}(x)$. The quantum circuit U_{exp} using these phases is shown in Fig. 2. This circuit constructs the $(1, a+2, \kappa\varepsilon_{\text{tri}})$ block encoding of $\kappa e^{-iHt}/2$,

$$\left\|\frac{\kappa}{2}e^{-iHt} - (\langle 0|_{abc}\otimes I)U_{\text{exp}}(|0\rangle_{abc}\otimes I)\right\| \leqslant \kappa\varepsilon_{\text{tri}}, \tag{19}$$

with R uses of U and $U^\dagger$ and one use of the controlled U. Therefore, the query complexity of U_{exp} is

$$\begin{aligned}R+R+1 &= 2\left\lfloor \frac{1}{2}r\left(\frac{et}{2}, \frac{5}{4}\varepsilon_{\text{tri}}\right)\right\rfloor + 1 &(20)\\ &\leqslant O(t+\log(1/\varepsilon_{\text{tri}})). &(21)\end{aligned}$$

To obtain the block encoding of $\exp(-iHt)$, the amplitude amplification must be used. In Sec. III we discuss two types of QSVT-based amplitude amplification algorithms: the oblivious amplitude amplification [29] and fixed-point amplitude amplification [32].

C. Linearized Vlasov-Poisson system

The time evolution of the distribution function $f(\boldsymbol{x}, \boldsymbol{v}, t)$ for electrons with stationary ions and the electric field $\boldsymbol{E} = (E_x, E_y, E_z)$ governed by the Vlasov-Poisson system is described by the equations

$$\frac{\partial f}{\partial t} + \boldsymbol{v}\cdot\nabla f - \frac{e}{m}\boldsymbol{E}\cdot\frac{\partial f}{\partial \boldsymbol{v}} = 0, \tag{22}$$

$$\frac{\partial \boldsymbol{E}}{\partial t} = \frac{1}{\varepsilon_0}\int e\boldsymbol{v}f\,d\boldsymbol{v}, \tag{23}$$

where e is the absolute value of the electron charge, m is the electron mass, and ε_0 is the permittivity of the vacuum. The variables f and $\boldsymbol{E}$ are expanded into the equilibrium terms (labeled by 0) and perturbations (labeled by 1) to linearize Eq. (22):

$$\begin{aligned}f(\boldsymbol{x}, \boldsymbol{v}, t) &= f_0(\boldsymbol{v}) + f_1(\boldsymbol{x}, \boldsymbol{v}, t),\\ \boldsymbol{E} &= \boldsymbol{E}_1(\boldsymbol{x}, t).\end{aligned} \tag{24}$$

Note that we do not deal with the case when the nonzero electric field $\boldsymbol{E}_0$ increases the system's energy, i.e., $\boldsymbol{E}_0 = 0$.

We assume a Maxwellian background distribution $f_0 = f_{\text{M}}$ and apply the same transformations as in Ref. [9]: a Fourier transformation of the variables in space, change of variables, and discretization in velocity space with the dimensionless variables

$$\begin{aligned}&\hat{\boldsymbol{k}} = \lambda_{D_e}\boldsymbol{k}, \quad \hat{t} = \omega_{pe}t, \quad \hat{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\lambda_{D_e}\omega_{pe}},\\ &\hat{f} = \frac{(\lambda_{D_e}\omega_{pe})^3}{n_e}f, \quad \hat{\boldsymbol{E}} = \frac{e\lambda_{D_e}}{k_{\text{B}}T_e}\boldsymbol{E},\end{aligned} \tag{25}$$

where $\boldsymbol{k} = (k_x, k_y, k_z)$ is the wave vector for the Fourier transformation, λ_{D_e} is the Debye length with ions neglected, ω_{pe} is the electron plasma frequency, n_e is the electron number density, T_e is the electron temperature, and k_{B} is the Boltzmann constant. As a result, Eqs. (22) and (23) become

$$\begin{aligned}\frac{dF_j}{dt} &= -i(k_x v_{j_x} + k_y v_{j_y} + k_z v_{j_z})F_j\\ &\quad - i\mu_j(v_{j_x}E_x + v_{j_y}E_y + v_{j_z}E_z),\end{aligned} \tag{26}$$

$$\frac{dE_p}{dt} = -i\sum_j \mu_j v_{j_p}F_j \quad (p = x, y, z), \tag{27}$$

where the subscript 1 has been dropped,

$$\mu_{\boldsymbol{j}} = \mu\left(v_{j_x}, v_{j_y}, v_{j_z}\right) = \sqrt{\Delta v f_{\mathrm{M}}\left(v_{j_x}, v_{j_y}, v_{j_z}\right)}, \tag{28}$$

$$\begin{aligned} F_{\boldsymbol{j}} &= F\left(v_{j_x}, v_{j_y}, v_{j_z}, t\right) \\ &= i\sqrt{\frac{\Delta v}{f_{\mathrm{M}}\left(v_{j_x}, v_{j_y}, v_{j_z}\right)}} f\left(v_{j_x}, v_{j_y}, v_{j_z}, t\right), \end{aligned} \tag{29}$$

$$\sum_{\boldsymbol{j}} \cdot = \sum_{j_x=0}^{N_{v_x}-1} \sum_{j_y=0}^{N_{v_y}-1} \sum_{j_z=0}^{N_{v_z}-1}, \tag{30}$$

where $\Delta v = \Delta v_x \Delta v_y \Delta v_z$ is the product of the mesh sizes; $N_{v_x} = 2^{n_{v_x}}$, $N_{v_y} = 2^{n_{v_y}}$, and $N_{v_z} = 2^{n_{v_z}}$ are the grid sizes in velocity space; and the velocity space grid is represented by the index $\boldsymbol{j} = (j_x, j_y, j_z)$.

Equations (26) and (27) can be rewritten in the form of the Schrödinger equation

$$\frac{d\,|\psi(t)\rangle}{dt} = -iH\,|\psi(t)\rangle\,, \tag{31}$$

where H is a time-independent Hamiltonian and $|\psi(t)\rangle$ is a quantum state whose amplitudes are the variables, which is written in bra-ket notation as

$$\begin{aligned} |\psi(t)\rangle = \frac{1}{\eta}\Bigg(&\sum_{\boldsymbol{j}} F_{\boldsymbol{j}}\,|0\rangle_r\,|\boldsymbol{j}\rangle_v + E_x\,|1\rangle_r\,|\boldsymbol{0}\rangle_v \\ &+ E_y\,|2\rangle_r\,|\boldsymbol{0}\rangle_v + E_y\,|3\rangle_r\,|\boldsymbol{0}\rangle_v\Bigg), \end{aligned} \tag{32}$$

where $|\boldsymbol{j}\rangle_v = |j_x\rangle_{v_x}|j_y\rangle_{v_y}|j_z\rangle_{v_z}$, $|\boldsymbol{0}\rangle_v = |0\rangle_{v_x}|0\rangle_{v_y}|0\rangle_{v_z}$, and the normalization constant

$$\eta = \sqrt{\sum_{\boldsymbol{j}} |F_{\boldsymbol{j}}|^2 + |E_x|^2 + |E_y|^2 + |E_z|^2}. \tag{33}$$

Here $|\psi(t)\rangle$ has two registers labeled by r and v. The r register encodes the variable index: $|0\rangle_r \leftrightarrow F$, $|1\rangle_r \leftrightarrow E_x$, $|2\rangle_r \leftrightarrow E_y$, and $|3\rangle_r \leftrightarrow E_z$. The v register stores the velocity space dependence of F: $|0\rangle_r\,|\boldsymbol{j}\rangle_v \leftrightarrow F_{\boldsymbol{j}}$. The corresponding Hamiltonian H, which acts on these registers, is given by

$$\begin{aligned} H = \sum_{\boldsymbol{j}} \Big[&\left(k_x v_{j_x} + k_y v_{j_y} + k_z v_{j_z}\right)|0\rangle_r\,|\boldsymbol{j}\rangle_v\,\langle 0|_r\,\langle \boldsymbol{j}|_v \\ &+ \mu_{\boldsymbol{j}} v_{j_x}(|0\rangle_r\,|\boldsymbol{j}\rangle_v\,\langle 1|_r\,\langle \boldsymbol{0}|_v + |1\rangle_r\,|\boldsymbol{0}\rangle_v\,\langle 0|_r\,\langle \boldsymbol{j}|_v) \\ &+ \mu_{\boldsymbol{j}} v_{j_y}(|0\rangle_r\,|\boldsymbol{j}\rangle_v\,\langle 2|_r\,\langle \boldsymbol{0}|_v + |2\rangle_r\,|\boldsymbol{0}\rangle_v\,\langle 0|_r\,\langle \boldsymbol{j}|_v) \\ &+ \mu_{\boldsymbol{j}} v_{j_z}(|0\rangle_r\,|\boldsymbol{j}\rangle_v\,\langle 3|_r\,\langle \boldsymbol{0}|_v + |3\rangle_r\,|\boldsymbol{0}\rangle_v\,\langle 0|_r\,\langle \boldsymbol{j}|_v)\Big]. \end{aligned} \tag{34}$$

The solution of Eq. (31) is given by

$$|\psi(t)\rangle = e^{-iHt}\,|\psi(t=0)\rangle\,. \tag{35}$$

Therefore, the time evolution of Eqs. (26) and (27) can be computed by the HS.

III. QSVT-BASED HAMILTONIAN SIMULATION

A. Oblivious amplitude amplification

Oblivious amplitude amplification using the QSVT has been proposed in Ref. [29]. In this section we show the circuit of OAA and discuss the error and number of queries of the OAA-based HS. In the QSVT-based OAA, the dth Chebyshev polynomial of the first kind, defined by $T_d(x) = \cos[d\arccos(x)]$, is used as an objective function. For odd d, the corresponding phases $\Phi \in \mathbb{R}^{d+1}$ is given by

$$\begin{aligned} \phi_0 &= -\frac{d\pi}{2}, \\ \phi_k &= \frac{\pi}{2} \quad (k = 1, 2, \dots, d). \end{aligned} \tag{36}$$

From Eq. (18), the following inequality holds:

$$\begin{aligned} &\left|\frac{e^{-ixt}}{2} - \frac{P^{\cos}_{\varepsilon_{\mathrm{tri}},\kappa}(x) - iP^{\sin}_{\varepsilon_{\mathrm{tri}},\kappa}(x)}{2}\right| \\ &\leqslant \left|\frac{\kappa\varepsilon_{\mathrm{tri}}e^{-ixt}}{2}\right| + \left|\frac{\kappa e^{-ixt}}{2} - \frac{P^{\cos}_{\varepsilon_{\mathrm{tri}},\kappa}(x) - iP^{\sin}_{\varepsilon_{\mathrm{tri}},\kappa}(x)}{2}\right| \\ &\leqslant \frac{\kappa\varepsilon_{\mathrm{tri}}}{2} + \kappa\varepsilon_{\mathrm{tri}} \\ &= \frac{3\varepsilon_{\mathrm{tri}}}{2(1+\varepsilon_{\mathrm{tri}})} < \frac{3}{2}\varepsilon_{\mathrm{tri}}. \end{aligned} \tag{37}$$

In the first inequality we used the triangle inequality. Letting U be $U_{\exp}$ in Eq. (7) and using the phases

$$\begin{aligned} \phi^{(\mathrm{OAA})}_0 &= -\frac{3\pi}{2}, \\ \phi^{(\mathrm{OAA})}_k &= \frac{\pi}{2} \quad (k = 1, 2, 3), \end{aligned} \tag{38}$$

then one can get the block encoding of

$$T_3\left(\frac{e^{-iHt}}{2}\right) = T_3\left[\cos\left(\frac{\pi}{3}\right)\right]e^{-iHt} = -e^{-iHt}, \tag{39}$$

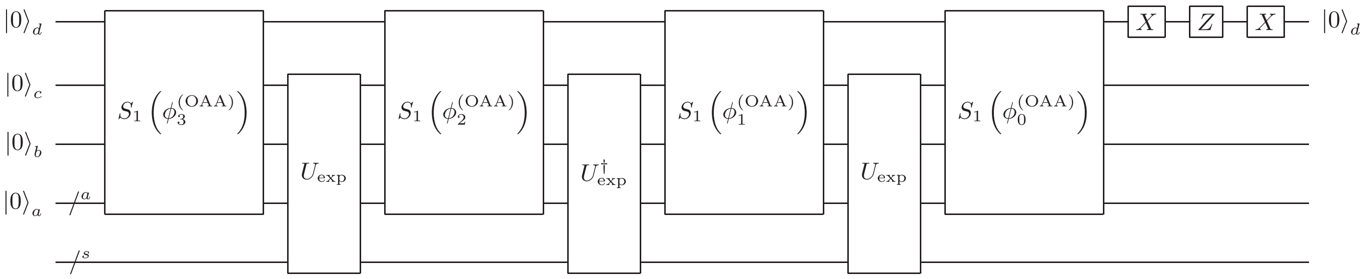
$|0\rangle_d$
$|0\rangle_c$
$|0\rangle_b$
$|0\rangle_a$
a
s
$S_1\left(\phi^{(\mathrm{OAA})}_3\right)$
$U_{\exp}$
$S_1\left(\phi^{(\mathrm{OAA})}_2\right)$
$U^{\dagger}_{\exp}$
$S_1\left(\phi^{(\mathrm{OAA})}_1\right)$
$U_{\exp}$
$S_1\left(\phi^{(\mathrm{OAA})}_0\right)$
X
Z
X
$|0\rangle_d$

FIG. 3. Quantum circuit U_{OAA} that is a $(1, a+2, \varepsilon)$ block encoding of $\exp(-iHt)$. The unitary $U_{\exp}$ is shown in Fig. 2, the gate S_1 is shown in Fig. 1(a), and the phase $\Phi^{(\mathrm{OAA})} \in \mathbb{R}^4$ is given by Eq. (38).

Algorithm 1. The OAA-based Hamiltonian simulation.

```
Input: A $(1, a, 0)$ block encoding of a Hamiltonian matrix $H$, an
    evolution time $t$, and an error tolerance $\varepsilon$.
Output: A $(1, a+2, \varepsilon)$ block encoding of $e^{-iHt}$.
Runtime: $Q_{\rm HS}^{\rm (OAA)}$ queries to the block encoding of $H$, where $Q_{\rm HS}^{\rm (OAA)}$
    is given by Eq. (41).
Procedure:
  1: Calculate the phases $\Phi^{\rm (c)} \in \mathbb{R}^{2R+1}$ and $\Phi^{\rm (s)} \in \mathbb{R}^{2R+2}$ on a
     classical computer from the $(2R)$th polynomial $P^{\cos}_{\varepsilon_{\rm tri},\kappa}(x)$ and
     $(2R+1)$th polynomial $P^{\sin}_{\varepsilon_{\rm tri},\kappa}(x)$ in Eq. (17), where
     $\kappa = 1/(1+\varepsilon_{\rm tri})$, $\varepsilon_{\rm tri} = \varepsilon/9$, and $R$ is given by Eq. (15).
  2: Construct the circuit $U_{\rm exp}$ in Fig. 2 using the phases $\Phi^{\rm (c)}$ and
     $\Phi^{\rm (s)}$, which is a $(1, a+2, \kappa\varepsilon_{\rm tri})$ block encoding of $\kappa e^{-iHt}/2$.
  3: Run the circuit $U_{\rm OAA}$ in Fig. 3 using the phases $\Phi^{\rm (OAA)} \in \mathbb{R}^4$ in
     Eq. (38).
```

where in the first equality we used that the singular value of a unitary matrix is 1. The OAA multiplies the error by a factor of $2d$ [29]. Therefore, the quantum circuit $U_{\rm OAA}$ in Fig. 3 using the phases $\Phi^{\rm (OAA)}$ in Eq. (38) constructs the $(1, a+2, 9\varepsilon_{\rm tri})$ block encoding of $\exp(-iHt)$,

$$\|e^{-iHt} - (\langle 0|_{abc} \otimes I)U_{\rm OAA}(|0\rangle_{abc} \otimes I)\| \leqslant 9\varepsilon_{\rm tri}, \tag{40}$$

with two uses of $U_{\rm exp}$ and one use of $U_{\rm exp}^{\dagger}$.

Given an error tolerance ε, the functions $\cos(xt)$ and $\sin(xt)$ should be $\frac{\varepsilon}{9}$ approximated. Therefore, the number of queries of the OAA-based HS is given by

$$Q_{\rm HS}^{\rm (OAA)} = 3\left[2R\left(t, \frac{\varepsilon}{9}\right) + 1\right] \leqslant O(t + \log(1/\varepsilon)). \tag{41}$$

We summarize the OAA-based HS in Algorithm 1.

B. Fixed-point amplitude amplification

Fixed-point amplitude amplification using the QSVT was proposed in Refs. [29,32]. In this section the error and number of queries of the FPAA-based HS are investigated in detail. In QSVT-based FPAA, the sign function

$$\operatorname{sgn}(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases} \tag{42}$$

is chosen as an objective function. The sign function can be estimated by a polynomial approximation to an error function $\operatorname{erf}(kx)$ for large enough k [32]. Let D be odd, $\Delta > 0$, and $\varepsilon_{\rm sgn} \in (0, \sqrt{2/e\pi}]$. The phases $\Phi^{\rm (FPAA)} \in \mathbb{R}^{D+1}$ can be calculated from the Dth polynomial $P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}$ [29,32] (the explicit form of $P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}$ was given in Ref. [43]):

$$\left|\operatorname{sgn}(x) - P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}(x)\right| \leqslant \varepsilon_{\rm sgn}$$
$$\text{for } x \in \left[-1, -\frac{\Delta}{2}\right] \cup \left[\frac{\Delta}{2}, 1\right]. \tag{43}$$

The degree D was given asymptotically in Refs. [29,32]. We give it explicitly using the result of Refs. [43,44]. If $k = \frac{\sqrt{2}}{\Delta}\ln^{1/2}(8/\pi\varepsilon_{\rm sgn}^2)$ and

$$\begin{aligned} D(k, \varepsilon_{\rm sgn}) &= 2\left\lceil \frac{16k}{\sqrt{\pi}\varepsilon_{\rm sgn}} \exp\left[-\frac{1}{2}W\left(\frac{512}{\pi\varepsilon_{\rm sgn}^2 e^2}\right)\right]\right\rceil + 1 \\ &= O\left(\frac{1}{\Delta}\log(1/\varepsilon_{\rm sgn})\right), \end{aligned} \tag{44}$$

where W is the Lambert W function, then $P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}$ is an $\varepsilon_{\rm sgn}$ approximation to the sign function in the region $[-1, -\frac{\Delta}{2}] \cup [\frac{\Delta}{2}, 1]$. We require that $\Delta/2 \leqslant \kappa/2$ because we desire that $\kappa/2$ be mapped to a value greater than $1 - \varepsilon_{\rm sgn}$, and then $1/\Delta \geqslant 1/\kappa$. If $1/\Delta$ increases, then D increases because of $D \propto k \propto 1/\Delta$. Therefore, we should choose $\Delta = \kappa$.

We discuss the upper bound of the error of the FPAA-based HS. Let us define $A = \kappa\exp(-iHt)/2$ and $\tilde{A} = (\langle 0|_{abc} \otimes I)U_{\rm exp}(|0\rangle_{abc} \otimes I)$. Equation (19) can be rewritten as

$$\|A - \tilde{A}\| \leqslant \kappa\varepsilon_{\rm tri}, \tag{45}$$

and the inequality

$$\begin{aligned} \|A + \tilde{A}\| &\leqslant \|A\| + \|A\| + \|\tilde{A} - A\| \\ &\leqslant \frac{\kappa}{2} + \frac{\kappa}{2} + \kappa\varepsilon_{\rm tri} \\ &= \kappa(1 + \varepsilon_{\rm tri}) = 1 \end{aligned} \tag{46}$$

holds, where in the first inequality we used the triangle inequality. Therefore, the matrices A and $\tilde{A}$ satisfy the inequality

$$\begin{aligned} \|A - \tilde{A}\| + \left\|\frac{A + \tilde{A}}{2}\right\|^2 &\leqslant \kappa\varepsilon_{\rm tri} + \frac{1}{4} \\ &= \frac{5}{4} - \frac{1}{1 + \varepsilon_{\rm tri}} \\ &< \frac{5}{4} - \frac{e}{1 + e} < 1, \end{aligned} \tag{47}$$

where in the penultimate inequality we used $\varepsilon_{\rm tri} < 1/e$. According to Lemma 23 in Ref. [29], we have that

$$\begin{aligned} &\left\|P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}(A) - P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}(\tilde{A})\right\| \\ &\quad\leqslant D\sqrt{\frac{2}{1 - \left\|\frac{A+\tilde{A}}{2}\right\|^2}}\|A - \tilde{A}\| \\ &\quad\leqslant \sqrt{\frac{8}{3}}D\frac{\varepsilon_{\rm tri}}{1 + \varepsilon_{\rm tri}} < \sqrt{3}D\varepsilon_{\rm tri}. \end{aligned} \tag{48}$$

Therefore, we have that

$$\begin{aligned} &\left\|\operatorname{sgn}(A) - P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}(\tilde{A})\right\| \\ &\quad\leqslant \left\|\operatorname{sgn}(A) - P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}(A)\right\| + \left\|P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}(A) - P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}(\tilde{A})\right\| \\ &\quad\leqslant \varepsilon_{\rm sgn} + \sqrt{3}D\varepsilon_{\rm tri} \equiv \varepsilon'. \end{aligned} \tag{49}$$

From the above inequality, we have that $\|P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}(\tilde{A})\| \leqslant 1 + \varepsilon'$, which violates condition (vi). Therefore, we must consider the rescaled polynomial $\frac{1}{1+\varepsilon'}P^{\rm sgn}_{\varepsilon_{\rm sgn},\Delta}(\tilde{A})$ such that

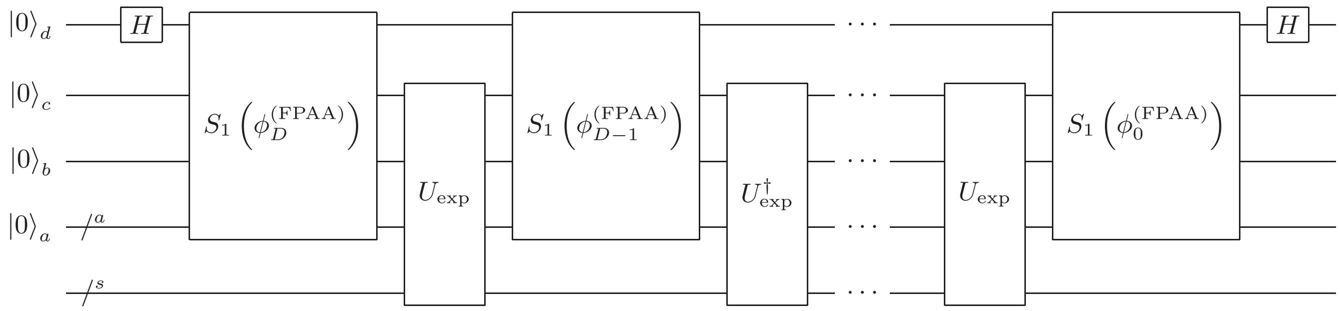
$|0\rangle_d$
$|0\rangle_c$
$|0\rangle_b$
$|0\rangle_a$
a
s
H
$S_1\left(\phi_D^{(\mathrm{FPAA})}\right)$
U_{exp}
$S_1\left(\phi_{D-1}^{(\mathrm{FPAA})}\right)$
$U_{\mathrm{exp}}^{\dagger}$
U_{exp}
$S_1\left(\phi_0^{(\mathrm{FPAA})}\right)$
H

FIG. 4. Quantum circuit U_{FPAA} that is a $(1, a+3, \varepsilon)$ block encoding of $\exp(-iHt)$. The unitary U_{exp} is shown in Fig. 2, the gate S_1 is shown in Fig. 1(a), and the phase $\Phi^{(\mathrm{FPAA})} \in \mathbb{R}^{D+1}$ is calculated from the Dth polynomial $P^{\mathrm{sgn}}_{\varepsilon_{\mathrm{sgn}},\Delta}$ that satisfies Eq. (43).

$\frac{1}{1+\varepsilon'}\|P^{\mathrm{sgn}}_{\varepsilon_{\mathrm{sgn}},\Delta}(\tilde{A})\| \leqslant 1$. Now we obtain the inequality

$$\begin{aligned}\left\|\mathrm{sgn}(A) - \frac{1}{1+\varepsilon'}P^{\mathrm{sgn}}_{\varepsilon_{\mathrm{sgn}},\Delta}(\tilde{A})\right\| \\ \leqslant \frac{1}{1+\varepsilon'}\left[\left\|\mathrm{sgn}(A) - P^{\mathrm{sgn}}_{\varepsilon_{\mathrm{sgn}},\Delta}(\tilde{A})\right\| + \|\varepsilon'\mathrm{sgn}(A)\|\right] \\ \leqslant \frac{\varepsilon'+\varepsilon'}{1+\varepsilon'} < 2\varepsilon' \qquad (50)\end{aligned}$$

and the error tolerance is defined as

$$\varepsilon \equiv 2\varepsilon' = 2(\varepsilon_{\mathrm{sgn}} + \sqrt{3}D\varepsilon_{\mathrm{tri}}). \qquad (51)$$

The quantum circuit U_{FPAA} using the phases $\Phi^{(\mathrm{FPAA})} \in \mathbb{R}^{D+1}$ in Fig. 4 constructs the $(1, a+3, \varepsilon)$ block encoding of $\exp(-iHt)$,

$$\|e^{-iHt} - (\langle 0|_{abcd} \otimes I)U_{\mathrm{FPAA}}(|0\rangle_{abcd} \otimes I)\| \leqslant \varepsilon, \qquad (52)$$

with $(D+1)/2$ uses of U_{exp} and $(D-1)/2$ uses of $U^{\dagger}_{\mathrm{exp}}$. Therefore, the number of queries of the FPAA-based HS is given by $D(2R+1)$. It varies depending on $\varepsilon_{\mathrm{tri}}$ and $\varepsilon_{\mathrm{sgn}}$ satisfying Eq. (51). The number of queries of the FPAA-based HS is defined as

$$Q^{(\mathrm{FPAA})}_{\mathrm{HS}} = \min_{\varepsilon_{\mathrm{tri}},\varepsilon_{\mathrm{sgn}}} D(\kappa, \varepsilon_{\mathrm{sgn}})[2R(t, \varepsilon_{\mathrm{tri}}) + 1]. \qquad (53)$$

This is asymptotically given by

$$\begin{aligned}Q^{(\mathrm{FPAA})}_{\mathrm{HS}} &= O(\log(1/\varepsilon_{\mathrm{sgn}})[t + \log(1/\varepsilon_{\mathrm{tri}})]) \\ &= O(t\log(1/\varepsilon) + \log^2(1/\varepsilon)). \qquad (54)\end{aligned}$$

We summarize the FPAA-based HS in Algorithm 2; also we summarize the comparison between OAA-based and FPAA-based HSs in Table I.

TABLE I. Comparison between the OAA-based and the FPAA-based HS. The asymptotic scalings of the number of queries are described in Eqs. (41) and (54).

Method	OAA-based HS	FPAA-based HS
polynomial used	$T_3(x)$	approximate polynomial of $\mathrm{sgn}(x)$
number of queries	$3\big(2R\big(t, \frac{\varepsilon}{9}\big) + 1\big)$	$D(\kappa, \varepsilon_{\mathrm{sgn}})[2R(t, \varepsilon_{\mathrm{tri}}) + 1]$
asymptotic query complexity	$O(t + \log(1/\varepsilon))$	$O(t\log(1/\varepsilon) + \log^2(1/\varepsilon))$

C. Hamiltonian simulation for the general Hermitian matrix and extension of evolution time

We describe the way to implement the HS for a general Hermitian matrix H that is not positive semidefinite and normalized by $\alpha > 0$, i.e., $\|H/\alpha\| \leqslant 1$. Suppose that U is an $(\alpha, a, 0)$ block encoding of H:

$$U = \begin{bmatrix} \frac{H}{\alpha} & \cdot \\ \cdot & \cdot \end{bmatrix}, \quad \frac{H}{\alpha} = (\langle 0|_a \otimes I)U(|0\rangle_a \otimes I). \qquad (55)$$

The authors of Ref. [32] proposed the unitary U' as in Fig. 5, which is a $(1, a+1, 0)$ block encoding of the positive-semidefinite Hermitian matrix $(H/\alpha + I)/2$. Instead of U, this unitary is used in the circuits U_{exp}, and U_{OAA} or U_{FPAA}, denoted by U_{HS}. Then U_{HS} becomes a $(1, a+3, \varepsilon)$ block encoding of $\exp(-i\frac{H/\alpha+I}{2}t)$:

$$\left\|e^{-i[(H/\alpha+I)/2]t} - (\langle 0|_{aa'bc} \otimes I)U_{\mathrm{HS}}(|0\rangle_{aa'bc} \otimes I)\right\| \leqslant \varepsilon. \qquad (56)$$

If the evolution time t is modified to $2\alpha t$, then U_{HS} becomes a $(1, a+3, \varepsilon)$ block encoding of e^{-iHt} up to a global phase. Therefore, the query complexity of the QSVT-based HS can be generally represented as

$$O(2\alpha t + \log(1/\varepsilon)). \qquad (57)$$

Note that the factor $\log(1/\varepsilon)$ is multiplied by the above complexity for the FPAA-based HS, but it is ignored for simplicity.

Larger t requires calculating a higher number of phases, which can be challenging. One can instead use N_t sequential U_{HS} for the smaller time step $\Delta t = t/N_t$ and extend the evolution time as follows:

$$e^{-iHt} \approx (\langle 0|_{aa'bc} \otimes I)U^{N_t}_{\mathrm{HS}}(\langle 0|_{aa'bc} \otimes I). \qquad (58)$$

This split is also found in Ref. [11]. We discuss the query complexity of the extension of the evolution time of the HS.

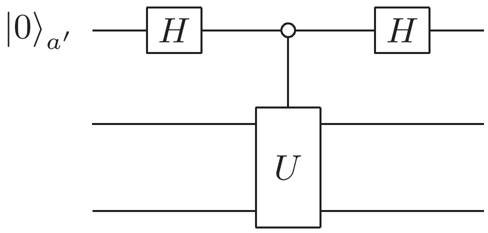

FIG. 5. Construction of the unitary U' that is a $(1, a+1, 0)$ block encoding of $(H/\alpha + I)/2$ from the unitary U, defined as in Eq. (55).

Algorithm 2. The FPAA-based Hamiltonian simulation.

```
Input: A $(1, a, 0)$ block encoding of a Hamiltonian matrix $H$, an evolution time $t$, and an error tolerance $\varepsilon$.
Output: A $(1, a+3, \varepsilon)$ block encoding of $e^{-iHt}$.
Runtime: $Q_{\rm HS}^{\rm (FPAA)}$ queries to the block encoding of $H$, where $Q_{\rm HS}^{\rm (FPAA)}$ is given by Eq. (53)
Procedure:
  1: Calculate $\varepsilon_{\rm tri}$ and $\varepsilon_{\rm sgn}$ which minimize $Q_{\rm HS}^{\rm (FPAA)}$, where $R$ and $D$ are given by Eqs. (15) and (44).
  2: Calculate the phases $\Phi^{\rm (c)} \in \mathbb{R}^{2R+1}$ and $\Phi^{\rm (s)} \in \mathbb{R}^{2R+2}$ on a classical computer from the $(2R)$th polynomial $P^{\cos}_{\varepsilon_{\rm tri},\kappa}(x)$ and the $(2R+1)$th polynomial $P^{\sin}_{\varepsilon_{\rm tri},\kappa}(x)$ in Eq. (17), where $\kappa = 1/(1+\varepsilon_{\rm tri})$.
  3: Construct the circuit $U_{\rm exp}$ in Fig. 2 using the phases $\Phi^{\rm (c)}$ and $\Phi^{\rm (s)}$, which is a $(1, a+2, \kappa\varepsilon_{\rm tri})$ block encoding of $\kappa e^{-iHt}/2$.
  4: Calculate the phases $\Phi^{\rm (FPAA)} \in \mathbb{R}^{D+1}$ on a classical computer from the $D$th polynomial $P^{\rm sgn}_{\varepsilon_{\rm sgn},\kappa}$ that satisfies Eq. (43).
  5: Run the circuit $U_{\rm FPAA}$ in Fig. 4 using the phases $\Phi^{\rm (FPAA)}$.
```

According to Lemma 53 in Ref. [29], the error of the product of two block-encoded matrices does not exceed the sum of each error. Let $U_{\rm HS}$ be a $(1, a+3, \delta)$ block encoding of $\exp(-iH\Delta t)$ and then we have that

$$\left\| e^{-iHt} - (\langle 0|_{aa'bc} \otimes I) U_{\rm HS}^{N_t} (|0\rangle_{aa'bc} \otimes I) \right\| \leqslant N_t \delta \equiv \varepsilon, \quad (59)$$

where ε is an error tolerance. The query complexity in Eq. (57) can be rewritten as follows:

$$N_t O\left(2\alpha\Delta t + \log\left(\frac{1}{\delta}\right)\right) = O\left(2\alpha t + N_t \log\left(\frac{N_t}{\varepsilon}\right)\right). \quad (60)$$

IV. QUANTUM ALGORITHM FOR THE LINEARIZED VLASOV-POISSON SYSTEM

Quantum algorithms for the linearized Vlasov-Poisson system have three steps: (1) initialization, (2) HS, and (3) extracting data. Initialization prepares a quantum state that represents initial physical conditions, HS implements the time evolution of Eq. (31), and physically meaningful data are extracted from a final state.

To encode data and construct the Hamiltonian H, we use a rotation gate called a variable rotation introduced in Ref. [9], defined as

$$R(x) \equiv \begin{cases} e^{-iY \arccos x} & (x \in \mathbb{R}) \\ e^{-iX \arccos {\rm Im}(x)} e^{iZ\pi/2} & (x \in \mathbb{C} \setminus \mathbb{R}), \end{cases} \quad (61)$$

where x is a rotation angle such that $|x| \leqslant 1$. This gate acts as $R(x)|0\rangle = x|0\rangle + \sqrt{1-|x|^2}|1\rangle$. We assume that the rotation angles can be calculated efficiently on temporary registers, and applying the rotations controlled on the angle qubits can be implemented efficiently. Then the cost to implement the variable rotations is $O({\rm poly}(n_v)) = O({\rm poly}(\log N_v))$ because the input register has n_v qubits.

A. One-dimensional linearized Vlasov-Poisson system

Efficiently preparing a quantum state that represents the physical data is a very difficult problem. This problem is related to amplitude encoding [45]. Assuming quantum random access memory [46], amplitude encoding is also implemented efficiently. Under this assumption, we desire that a quantum circuit for initialization $U_{\rm ini}^{\rm ideal}$ can prepare the initial state

$$\begin{aligned} &U_{\rm ini}^{\rm ideal} |0\rangle_r |0\rangle_v \\ &\quad = \frac{1}{\eta}\left(\sum_{j_x=0}^{Nv-1} F_{j_x}(t=0)|0\rangle_r |j_x\rangle_{v_x} + E(t=0)|1\rangle_r |0\rangle_{v_x} \right), \end{aligned} \quad (62)$$

where $\eta = \sqrt{\sum_{j_x} |F_{j_x}(t=0)|^2 + |E(t=0)|^2}$. We assume that the gate complexity of $U_{\rm ini}^{\rm ideal}$ is $O({\rm poly}(\log N_{v_x}))$.

To implement the HS for the linearized Vlasov-Poisson system, it is necessary to construct the corresponding unitary U that is an $(\alpha, a, 0)$ block encoding of H. The unitary for the one-dimensional system was proposed in Ref. [9], which consists of two unitaries $U_{\rm row}$ and $U_{\rm col}$. These unitaries are called state preparation unitaries in Ref. [29] and satisfy $U = U_{\rm row}^\dagger U_{\rm col}$. The gate complexity of U is given by $O({\rm poly}(\log N_{v_x}))$. According to Ref. [9], α satisfies the inequality

$$\frac{4\Lambda}{5} \leqslant \alpha \leqslant \Lambda, \quad (63)$$

where

$$\begin{aligned} \Lambda &= |k_x| v_{x,\max} + \sqrt{\Delta v_x N_{v_x} v_{x,\max} G_{\max}}, \\ v_{x,\max} &= \max_{j_x} |v_{j_x}|, \\ G_{\max} &= \max_{j_x} |v_{j_x} f_{\rm M}(v_{v_{j_x}})|. \end{aligned} \quad (64)$$

Since $\Delta v_x N_{v_x} = 2v_{x,\max} + \Delta v_x$, Λ does not increase with increasing N_{v_x}, i.e. $\alpha = \Theta(1)$. Therefore, the query complexity of the QSVT-based HS is given by $O(t + \log(1/\varepsilon))$. The gate complexity of the QSVT-based HS is the above equation multiplied by ${\rm poly}(\log N_{v_x})$.

One of the ways to obtain data from the final state is the quantum amplitude estimation (QAE) algorithm [38], which can produce an estimate $\tilde{p}$ of the probability p with an error bounded by

$$|\tilde{p} - p| \leqslant 2\pi \frac{\sqrt{p(1-p)}}{M} + \frac{\pi^2}{M^2}, \quad (65)$$

where M is the number of iterations. The authors of Ref. [9] used QAE to obtain an estimate of the magnitude $|E|$. They also proposed the algorithm for obtaining the real and imaginary parts of E and showed that its cost does not change asymptotically. We discuss the computational complexity of calculating the time evolution of E and the algorithm for obtaining quantities related to the distribution function f.

The state after the HS for the evolution time t is applied to the initial state in Eq. (62) becomes

$$|\psi(t)\rangle = \frac{1}{\eta}|0\rangle_a \left(\sum_{j_x=0}^{N_{v_x}-1} F_{j_x}(t)|0\rangle_r |j_x\rangle_{v_x} + E(t)|1\rangle_r |0\rangle_{v_x} \right), \quad (66)$$

where the label a is represented as all ancilla qubits including qubits labeled a, a', b, c, and d. Let $p = |E(t)|^2/\eta^2$ and its estimate be $\tilde{p}$. We introduce $a = |E(t)|^2$ and $\tilde{a} = \eta^2\tilde{p}$, which then satisfy the following inequality from Eq. (65):

$$\begin{aligned}|\tilde{a} - a| &\leqslant 2\pi\frac{\sqrt{a(\eta^2 - a)}}{M} + \frac{\pi^2\eta^2}{M^2}\\ &\leqslant \frac{2\pi\eta|E(t)|}{M} + \frac{\pi^2\eta^2}{M^2}.\end{aligned} \tag{67}$$

In the last inequality we used $\eta^2 - a \leqslant \eta^2$. Assuming we know the upper bound of $|E(t)|$ is E_u, we let $0 < \delta < 1$ and $M = \lceil\frac{(2E_u+1)\pi\eta}{\delta}\rceil$. The above inequality becomes

$$\begin{aligned}|\tilde{a} - a| &\leqslant \frac{2|E(t)|\delta}{2E_u + 1} + \frac{\delta^2}{(2E_u + 1)^2}\\ &\leqslant \frac{2|E(t)|\delta}{2E_u + 1} + \frac{\delta}{2E_u + 1} \leqslant \delta,\end{aligned} \tag{68}$$

where in the last inequality we used $2|E(t)| + 1 \leqslant 2E_u + 1$. The value of $E(t)$ can also be obtained asymptotically with the same complexity at an additional cost using the algorithm proposed in Ref. [9]. Since $M = O(1/\delta)$, the gate complexity of the whole algorithm for obtaining the estimate of $E(t)$ including the initialization and HS steps is given by

$$O\left(\frac{\text{poly}(\log N_v)}{\delta}[t + \log(1/\varepsilon)]\right). \tag{69}$$

Now we discuss the cost of calculating the time evolution of E. For simplicity, we assume the phases $\Phi^{(c)}$ and $\Phi^{(s)}$ for a large t, and a given error tolerance ε can be calculated. Let N_t be the number of time steps, $t_{\max}$ the maximum of the evolution time, and $\Delta t = t_{\max}/N_t$ the time step. The gate complexity of the algorithm for the evolution time $t_l = l\Delta t$ is given by Eq. (69) with t replaced by t_l, denoted by Q_l. Since

$$\sum_{l=1}^{N_t} t_l = \frac{\Delta t}{2}N_t(N_t + 1) = \frac{t_{\max}(N_t + 1)}{2}, \tag{70}$$

the gate complexity of calculating the time evolution of E is

$$\sum_{l=1}^{N_t} Q_l = O\left(\frac{\text{poly}(\log N_v)N_t}{\delta}[t_{\max} + \log(1/\varepsilon)]\right). \tag{71}$$

If we consider $t_{\max}$ to be a constant, then the cost of the quantum algorithm is asymptotically the same for the number of time steps N_t as that of a classical algorithm, which scales linearly with N_t.

We show the way to obtain the deviation from the Maxwell distribution

$$\begin{aligned}D_{\text{M}}(t) &\equiv \sum_{j_x=0}^{N_{v_x}-1} |f(v_{j_x}, t) - f_{\text{M}}(v_{j_x})|^2\Delta v_x\\ &= \sum_{j_x=0}^{N_{v_x}-1} |f_1(v_{j_x}, t)|^2\Delta v_x.\end{aligned} \tag{72}$$

Here we write the subscript 1 explicitly. This quantity can be used to know how well the fluid approximation is applied to the system. We add a single ancilla qubit and variable rotation gates with angles $\sqrt{f_{\text{M}}(v_{j_x})}$ and the state in Eq. (66) becomes

$$\frac{1}{\eta}\sum_{j_x=0}^{N_{v_x}-1} i f_1(v_{j_x}, t)\sqrt{\Delta v_x}\,|0\rangle_a|0\rangle_r|j_x\rangle_{v_x}|0\rangle + |\perp\rangle, \tag{73}$$

where $|\perp\rangle$ is the state of no interest. When the QAE is applied to the above state, we obtain the estimate

$$\tilde{p} \approx \frac{1}{\eta^2}\sum_{j_x=0}^{N_{v_x}-1} |f_1(v_{j_x}, t)|^2\Delta v_x = \frac{1}{\eta^2}D_{\text{M}}(t). \tag{74}$$

Similar to the discussion of QAE to obtain the estimate of $E(t)$, the number of iterations to obtain the estimate of D_{M} is also given by $M = O(1/\delta)$. Therefore, the gate complexity of the algorithm for calculating the deviation D_{M} is given by Eq. (69).

B. Higher-dimensional Vlasov-Poisson systems

The discussion of initialization and extracting data steps for the one-dimensional linearized Vlasov-Poisson system can easily be extended to the higher system. In this section we focus on the construction of unitaries $U = U_{\text{row}}^\dagger U_{\text{col}}$ that are $(\alpha, a, 0)$ block encoding of H for the higher systems.

We show the circuits U_{row} and U_{col} for the two-dimensional system in Figs. 6 and 7. The unitary U is an $(\alpha, 6, 0)$ block encoding of the Hamiltonian H,

$$\begin{aligned}\frac{H}{\alpha} = \sum_{\boldsymbol{j}}\Bigg[&c^2b_{\boldsymbol{j}}^2\,|0\rangle_r|\boldsymbol{j}\rangle_v\langle 0|_r\langle\boldsymbol{j}|_v + \frac{\sqrt{1-|c|^2}d_{\boldsymbol{j}}}{\sqrt{2N_v}}\\ &\times\,(p_{j_x}(|0\rangle_r|\boldsymbol{j}\rangle_v\langle 1|_r\langle\mathbf{0}|_v + |1\rangle_r|\mathbf{0}\rangle_v\langle 0|_r\langle\boldsymbol{j}|_v)\\ &+ q_{j_y}(|0\rangle_r|\boldsymbol{j}\rangle_v\langle 2|_r\langle\mathbf{0}|_v + |2\rangle_r|\mathbf{0}\rangle_v\langle 0|_r\langle\boldsymbol{j}|_v))\Bigg] + \hat{D}_2,\end{aligned} \tag{75}$$

where $\hat{D}_2$ is the unused subspace. Comparing Eq. (34) for the two-dimensional system with Eq. (75), we obtain

$$\begin{aligned}c^2b_{\boldsymbol{j}}^2 &= \frac{k_xv_{j_x} + k_yv_{j_y}}{\alpha},\\ \sqrt{1-|c|^2}\frac{p_{j_x}d_{\boldsymbol{j}}}{\sqrt{2N_v}} &= \frac{\mu_{\boldsymbol{j}}v_{j_x}}{\alpha},\\ \sqrt{1-|c|^2}\frac{q_{j_y}d_{\boldsymbol{j}}}{\sqrt{2N_v}} &= \frac{\mu_{\boldsymbol{j}}v_{j_y}}{\alpha},\end{aligned} \tag{76}$$

where $N_v = N_{v_x}N_{v_y}$. The angles $b_{\boldsymbol{j}}$, $d_{\boldsymbol{j}}$, p_{j_x}, and q_{j_y} are chosen as

$$\begin{aligned}b_{\boldsymbol{j}} &= \sqrt{\frac{k_xv_{j_x} + k_yv_{j_y}}{K_{\max}}}, \quad d_{\boldsymbol{j}} = \sqrt{\frac{f_{\text{M}}(v_{\boldsymbol{j}})}{g_{\max}}},\\ p_{j_x} &= \frac{v_{j_x}}{V_{\max}}, \quad q_{j_y} = \frac{v_{j_y}}{V_{\max}},\end{aligned} \tag{77}$$

where

$$\begin{aligned}K_{\max} &= \max_{\boldsymbol{j}}|k_xv_{j_x} + k_yv_{j_y}|,\\ g_{\max} &= \max_{\boldsymbol{j}} f_{\text{M}}(v_{j_x}, v_{j_y}),\\ V_{\max} &= \max_{j_x}|v_{j_x}|\max_{j_y}|v_{j_y}| = v_{x,\max}v_{y,\max}.\end{aligned} \tag{78}$$

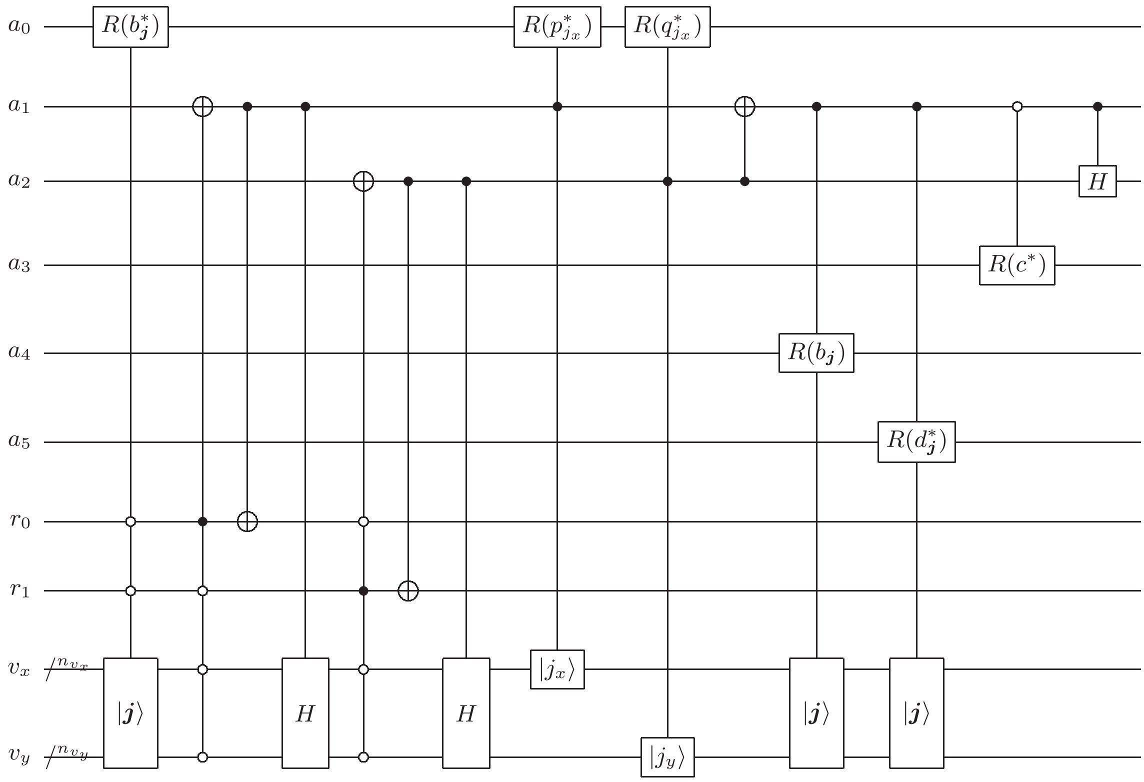

FIG. 6. Quantum circuit of the unitary U_{row} for the two-dimensional linearized Vlasov-Poisson system.

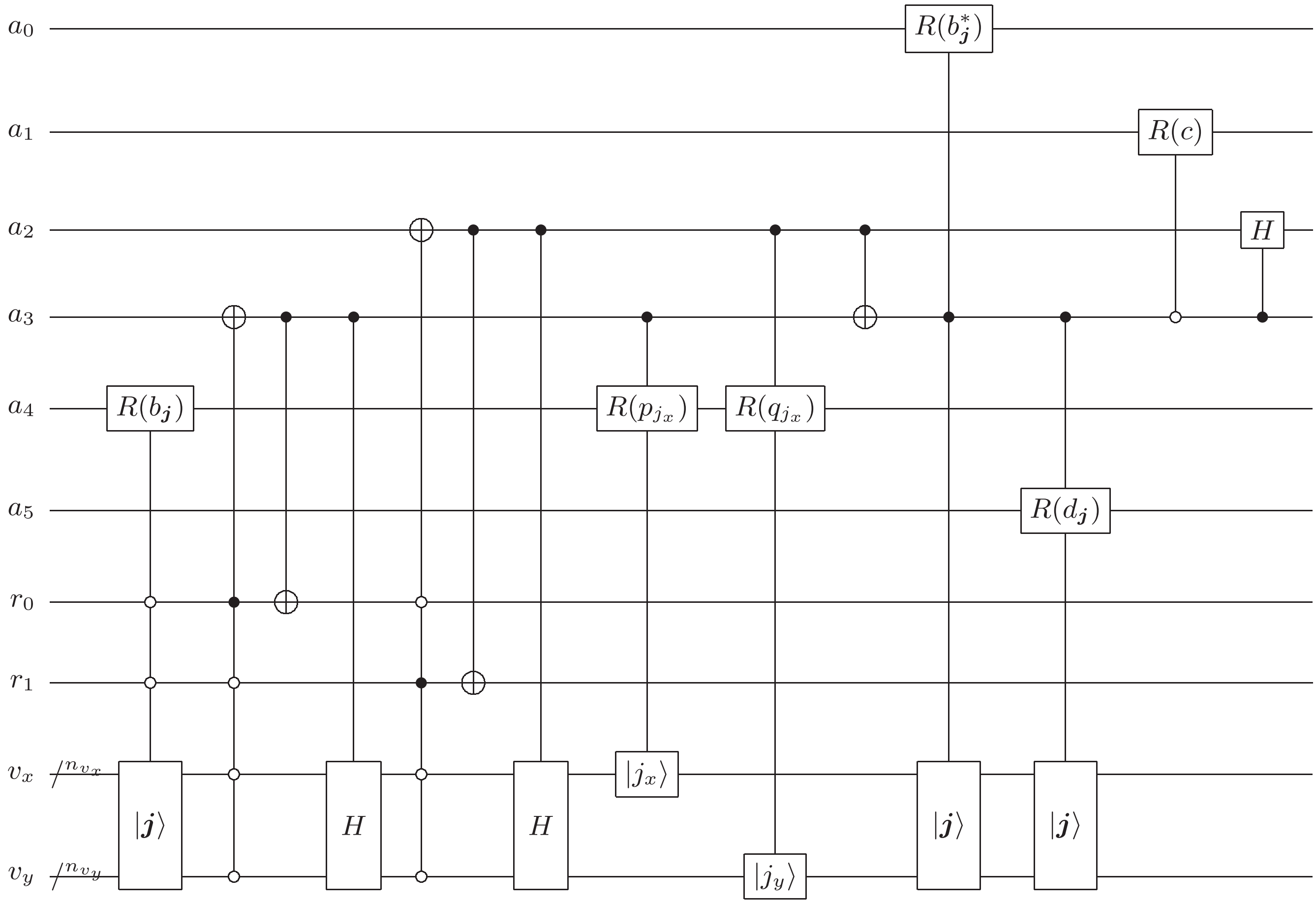

FIG. 7. Quantum circuit of the unitary U_{col} for the two-dimensional linearized Vlasov-Poisson system.

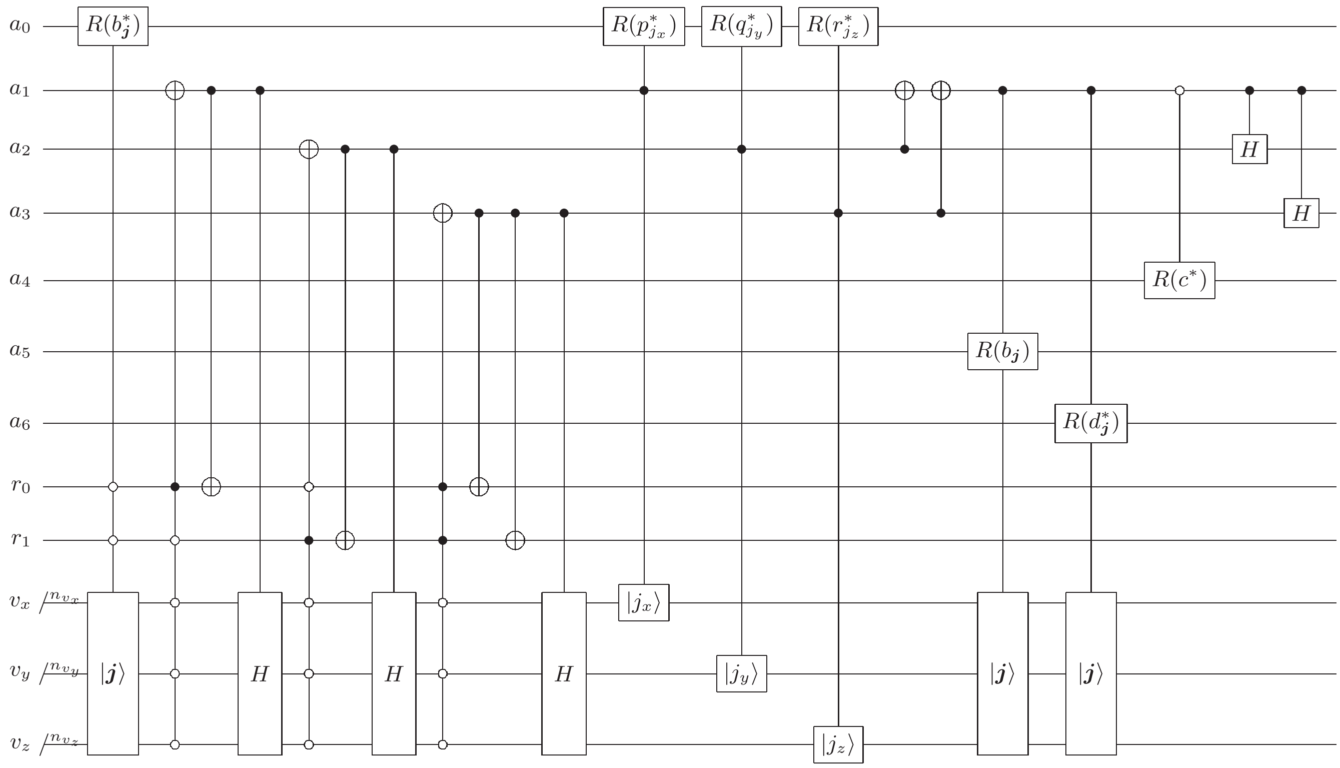

FIG. 8. Quantum circuit of the unitary U_{row} for the three-dimensional linearized Vlasov-Poisson system.

Note that we assume $v_{x,\max} \geqslant 1$ and $v_{y,\max} \geqslant 1$ to make $|p_{j_x}| \leqslant 1$ and $|q_{j_y}| \leqslant 1$ hold. From Eqs. (76) and (77), c and α are given by

$$c^2 = \frac{\Gamma}{2}\left(\sqrt{1+\frac{4}{\Gamma}}-1\right), \tag{79}$$

$$\alpha = \frac{K_{\max}}{c^2}, \tag{80}$$

where

$$\Gamma = \frac{K_{\max}^2}{2\Delta v N_v V_{\max}^2 g_{\max}}, \tag{81}$$

with $\Delta v = \Delta v_x \Delta v_y$. As for the one-dimensional system, α satisfies the inequality

$$\frac{4\Lambda}{5} \leqslant \alpha \leqslant \Lambda, \tag{82}$$

where

$$\Lambda = K_{\max} + \sqrt{2\Delta v N_v V_{\max}^2 g_{\max}.} \tag{83}$$

Since $\Delta v N_v = (2v_{x,\max} + \Delta v_x)(2v_{y,\max} + \Delta v_y)$, if N_v increases, then Λ does not increase, i.e. $\alpha = \Theta(1)$. Therefore, the query complexity also does not increase with increasing N_v. In addition, the gate complexity scales logarithmically with N_v because the input register of the variable rotations in Figs. 6 and 7 has $n_{v_x} + n_{v_y}$ qubits. These results are the same as for the one-dimensional system.

We show a unitary $U = U_{\text{row}}^\dagger U_{\text{col}}$ for the three-dimensional Vlasov-Poisson system and results similar to the ones for the lower-dimensional systems. The circuits U_{row} and U_{col} are shown in Figs. 8 and 9. The unitary U is an $(\alpha, 7, 0)$ block encoding of the Hamiltonian H,

$$\begin{aligned}\frac{H}{\alpha} = \sum_{\boldsymbol{j}} \Bigg(& c^2 b_{\boldsymbol{j}}^2 |0\rangle_r |\boldsymbol{j}\rangle_v \langle 0|_r \langle \boldsymbol{j}|_v + \frac{\sqrt{1-|c|^2} d_{\boldsymbol{j}}}{2\sqrt{N_v}} \big[p_{j_x} (|0\rangle_r |\boldsymbol{j}\rangle_v \langle 1|_r \langle \boldsymbol{0}|_v + |1\rangle_r |\boldsymbol{0}\rangle_v \langle 0|_r \langle \boldsymbol{j}|_v) \\ & + q_{j_y} (|0\rangle_r |\boldsymbol{j}\rangle_v \langle 2|_r \langle \boldsymbol{0}|_v + |2\rangle_r |\boldsymbol{0}\rangle_v \langle 0|_r \langle \boldsymbol{j}|_v) + r_{j_z} (|0\rangle_r |\boldsymbol{j}\rangle_v \langle 3|_r \langle \boldsymbol{0}|_v + |3\rangle_r |\boldsymbol{0}\rangle_v \langle 0|_r \langle \boldsymbol{j}|_v) \big] \Bigg) + \hat{D}_3, \end{aligned} \tag{84}$$

where $N_v = N_{v_x} N_{v_y} N_{v_z}$ and $\hat{D}_3$ is the unused subspace. The corresponding angles and α are

$$b_{\boldsymbol{j}} = \sqrt{\frac{k_x v_{j_x} + k_y v_{j_y} + k_z v_{j_z}}{K_{\max}}},$$

$$d_{\boldsymbol{j}} = \sqrt{\frac{f_{\text{M}}(v_{\boldsymbol{j}})}{g_{\max}}}, \quad p_{j_x} = \frac{v_{j_x}}{V_{\max}}, \tag{85}$$

$$q_{j_y} = \frac{v_{j_y}}{V_{\max}}, \quad r_{j_z} = \frac{v_{j_z}}{V_{\max}},$$

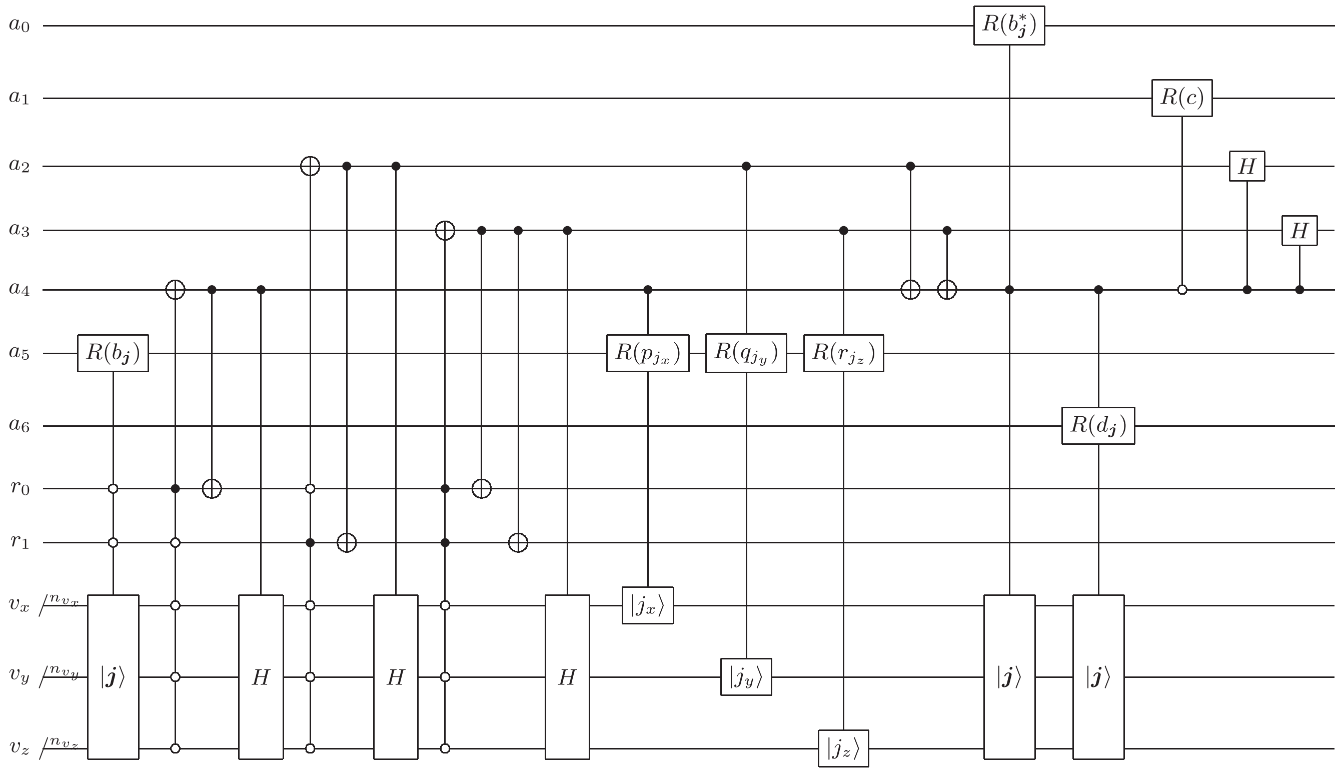

FIG. 9. Quantum circuit of the unitary U_{col} for the three-dimensional linearized Vlasov-Poisson system.

$$c^2 = \frac{\Gamma}{2}\left(\sqrt{1+\frac{4}{\Gamma}}-1\right), \tag{86}$$

$$\alpha = \frac{K_{\max}}{c^2}, \tag{87}$$

where

$$\begin{aligned} K_{\max} &= \max_{j} |k_x v_{j_x} + k_y v_{j_y} + k_z v_{j_z}|, \\ g_{\max} &= \max_{j} f_{\mathrm{M}}(v_{j_x}, v_{j_y}, v_{j_z}), \\ V_{\max} &= v_{x,\max} v_{y,\max} v_{z,\max}, \\ \Gamma &= \frac{K_{\max}^2}{4\Delta v N_v V_{\max}^2 g_{\max}}. \end{aligned} \tag{88}$$

As for the lower-dimensional system, α satisfies the inequality

$$\frac{4\Lambda}{5} \leqslant \alpha \leqslant \Lambda, \tag{89}$$

where

$$\Lambda = K_{\max} + \sqrt{4\Delta v N_v V_{\max}^2 g_{\max}}. \tag{90}$$

Since $\Delta v N_v = (2v_{x,\max} + \Delta v_x)(2v_{y,\max} + \Delta v_y)(2v_{z,\max} + \Delta v_z)$, if N_v increases, then Λ does not increase. Therefore, the same result is obtained for the query complexity and the gate complexity. We summarize the computational resources of the HS for the linearized Vlasov-Poisson system in Table II.

V. NUMERICAL RESULTS

A. QSVT-based Hamiltonian simulation

We compare the number of queries of the OAA-based and FPAA-based HS algorithms. The number of queries for some given error tolerances ε and evolution times t calculated from Eqs. (41) and (53) is shown in Figs. 10 and 11. Notably, the number of queries of the OAA-based HS is significantly smaller than that of the FPAA-based one for all parameters. These figures show that the number of queries $Q_{\mathrm{HS}}^{(\mathrm{OAA})}$ and $Q_{\mathrm{HS}}^{(\mathrm{FPAA})}$ scale linearly for t and linearly and quadratically for $\ln(1/\varepsilon)$, respectively. These results are consistent with the asymptotic scaling of Eqs. (41) and (54). To identify the constant factors and coefficients of the number of queries hidden behind the asymptotic scaling, we fit the curve for OAA with

$$Q_{\mathrm{HS}}^{(\mathrm{OAA})} = \alpha_0 + \alpha_1 t + \alpha_2 \ln(1/\varepsilon) \tag{91}$$

and the curve for FPAA with

$$\begin{aligned} Q_{\mathrm{HS}}^{(\mathrm{FPAA})} &= \alpha_0 + \alpha_1 t + \alpha_2 \ln(1/\varepsilon) \\ &\quad + \alpha_3 t \ln(1/\varepsilon) + \alpha_4 \ln^2(1/\varepsilon). \end{aligned} \tag{92}$$

The results are presented in Table III. Both values of FPAA are greater than those of OAA. Thus, OAA proves more effective for HS than FPAA in terms of the number of queries.

We emphasize that the advantage of OAA over FPAA in HS is a general result. The reasons for this can be explained as follows. The number of queries is calculated from Eqs. (41) and (53). These equations are derived under the general as-

TABLE II. Comparison of the Hamiltonian simulation for the linearized Vlasov-Poisson system. The system register is defined as a register labeled by r and v.

Dimension	Total grid size N_v	No. of system registers	No. of ancilla qubits of U	Gate complexity
1	N_{v_x}	$n_{v_x}+1$	4	$O(\mathrm{poly}(\log N_v)[t+\log(1/\varepsilon)])$
2	$N_{v_x}N_{v_y}$	$n_{v_x}+n_{v_y}+2$	6	
3	$N_{v_x}N_{v_y}N_{v_z}$	$n_{v_x}+n_{v_y}+n_{v_z}+2$	7	

sumption that the Hamiltonian is positive semidefinite and its norm is less than 1, that is, Eqs. (41) and (53) hold without respect to the type of Hamiltonian. Therefore, from the theoretical and numerical results of the number of queries, OAA is more advantageous than FPAA in Hamiltonian simulations of general systems.

To specify what degree of approximation of trigonometric functions or sign function dominates the number of queries of the FPAA-based HS, we fit the curves for R and D in Eq. (53) with

$$R=\alpha_0+\alpha_1 t+\alpha_2\ln(1/\varepsilon), \tag{93}$$

$$D=\alpha_0+\alpha_2\ln(1/\varepsilon) \tag{94}$$

for $0.1\leqslant t\leqslant 10$ and $10^{-5}\leqslant\varepsilon\leqslant 0.9$. Then we obtain the following results: For R,

$$\alpha_0=0.853,\quad \alpha_1=0.913,\quad \alpha_2=0.293, \tag{95}$$

and for D,

$$\alpha_0=21.8,\quad \alpha_2=10.1. \tag{96}$$

The constant factors and coefficients of D are larger than those of R. Thus, the large number of queries of the FPAA-based HS is caused by requiring a high degree of approximation of the sign function.

We now explain the intuitive reason why it is not appropriate to use a sign function for amplitude amplification in the

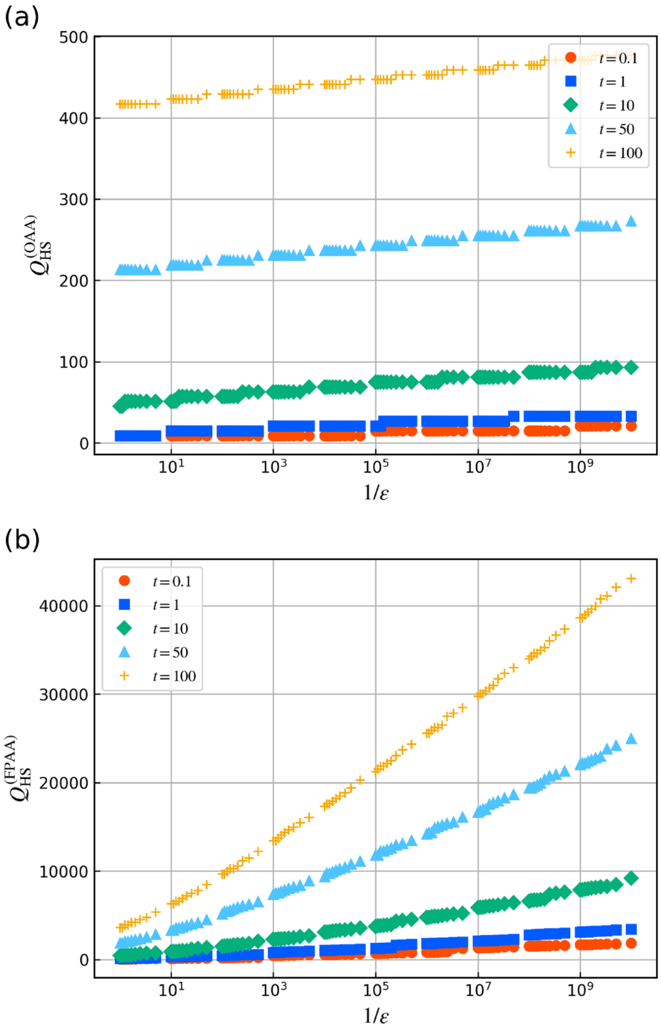

FIG. 10. Number of queries vs $1/\varepsilon$ for (a) the OAA-based HS and (b) the FPAA-based HS.

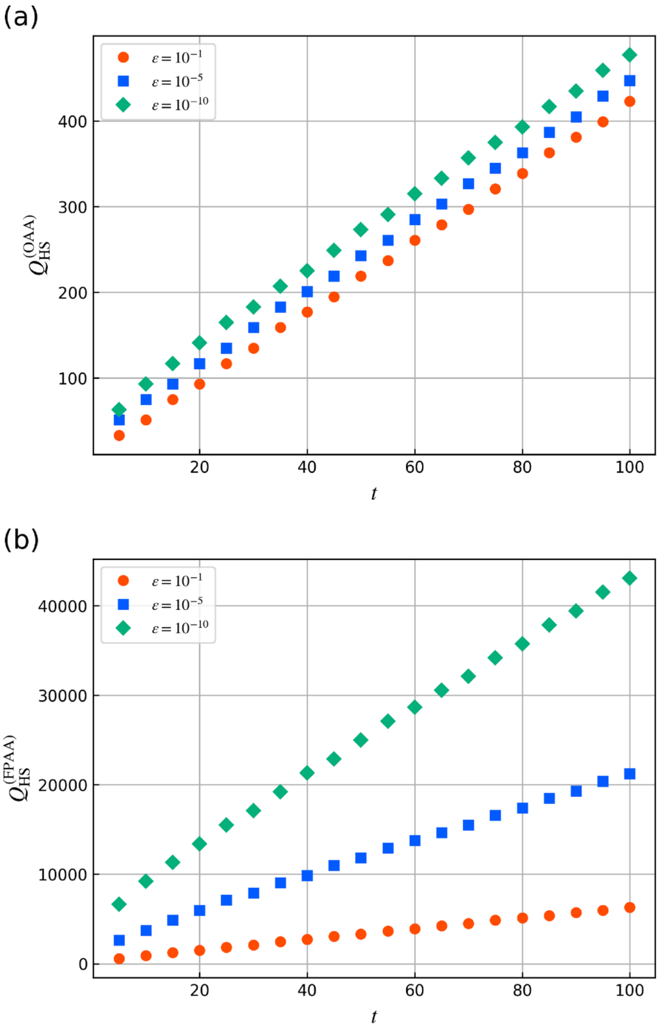

FIG. 11. Number of queries vs t for (a) the OAA-based HS and (b) the FPAA-based HS.

TABLE III. Constant factors and coefficients of Eqs. (41) and (53).

Method	Range of parameters	α_0	α_1	α_2	α_3	α_4
OAA	$0.1 \leqslant t \leqslant 10,\ 10^{-5} \leqslant \varepsilon \leqslant 0.9$	2.73	4.88	1.78		
OAA	$1 \leqslant t \leqslant 100,\ 10^{-10} \leqslant \varepsilon \leqslant 0.9$	2.77	4.18	2.45		
FPAA	$0.1 \leqslant t \leqslant 10,\ 10^{-5} \leqslant \varepsilon \leqslant 0.9$	142	28.3	11.9	20.9	7.26
FPAA	$1 \leqslant t \leqslant 100\, 10^{-10} \leqslant \varepsilon \leqslant 0.9$	490	21.7	−15.6	15.4	10.9

QSVT-based HS. As seen in Sec. II B, the specific amplitude value $\kappa/2 \approx \frac{1}{2}$ must be amplified in the QSVT-based HS. For the OAA-based HS, the third Chebyshev polynomial is precisely $|T_3(x)| = 1$ at $x = \frac{1}{2}$, whereas for the FPAA-based one the sign function is $\text{sgn}(x) = 1$ for $\frac{1}{2} \leqslant x \leqslant 1$. The OAA-based method amplifies the value exclusively at $x = \frac{1}{2}$. In contrast, the FPAA-based method aims to amplify the values for $\frac{1}{2} \leqslant x \leqslant 1$, leading to extra, unneeded effort for amplification in this range, as depicted in Fig. 12. This results in a high degree of approximation of the sign function and many queries for the FPAA-based HS.

B. Application to the linearized Vlasov-Poisson system

Here the OAA-based HS is applied to the simulation of the one-dimensional linearized Vlasov-Poisson system. The simulation is implemented on a classical emulator of a quantum computer using QISKIT [47], with Statevector Simulator as the backend. This backend gives us access to the whole output space at all moments, and we do not implement QAE directly to save the number of qubits. We compare the simulation results of the quantum algorithm using the HS with those of a classical algorithm which have been obtained by directly solving the Vlasov equation and the Poisson equation using the Euler method for the time. These simulations are performed using the following parameters for a given wave number k:

$$N_v = 32, \quad f_\text{M}(v_j) = \frac{1}{\sqrt{2\pi}} e^{-v_j^2/2},$$
$$v_\text{max} = 4.5, \quad f_1(v_j, t = 0) = 0.1 f_\text{M}(v_j),$$
$$E(t = 0) = \frac{i}{k} \sum_j f_1(v_j, t = 0)\Delta v. \tag{97}$$

We construct the unitary U' in Fig. 5 from the unitary $U = U_\text{row}^\dagger U_\text{cal}$ in Ref. [9], which is the $(\alpha, 4, 0)$ block encoding of H. Using the unitary U', we construct the circuit U_OAA for $t = 2$ and choose an error tolerance $\varepsilon = 10^{-3}$. Then U_OAA is a $(1, 7, 10^{-3})$ block encoding of $e^{-iH\Delta t}$, where $\Delta t = 1/\alpha$. We implement the HS for the evolution time $l\Delta t$ using l sequential U_HS because it is difficult to compute the phases $\Phi^{(c)}$ and $\Phi^{(s)}$ for a large t, as mentioned in Sec. III C.

Figure 13 shows the time evolution of the electric field E for $k = 0.4$ using the quantum algorithm. In this case, the normalization of the Hamiltonian becomes $1/\alpha = 0.238$. After a brief initial stage, the imaginary component of E is damped and oscillating. We fit the curve with the function $Ae^{-i\gamma(t-t_0)} \cos[\omega(t - t_0) - \rho] + E_0$ to obtain parameters of interest, i.e., the frequency ω and damping rate γ,

$$\omega = 1.285\,08, \quad \gamma = 0.066\,23, \tag{98}$$

where $t_0 = 5.23$. One can find precise values of $\omega = 1.285\,06$ and $\gamma = 0.066\,13$ from the linear Landau theory [48]. Figure 14 shows the comparison of the frequencies and damping rates obtained from the results of the quantum algorithm with

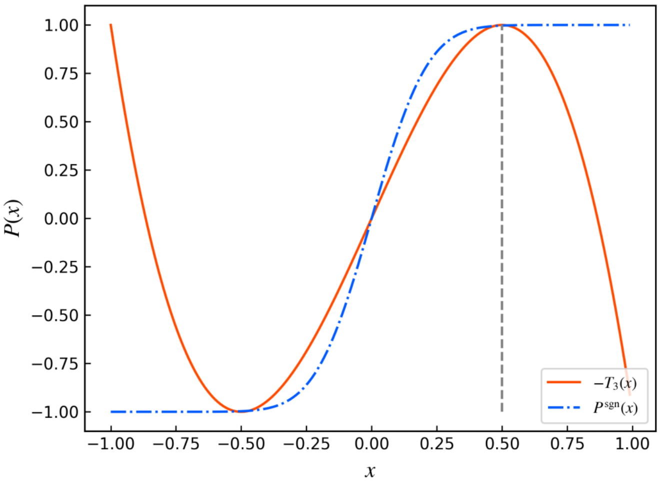

FIG. 12. Illustration of the difference between amplitude amplifications in the OAA-based and the FPAA-based HS. The red solid line represents the third Chebyshev polynomial with a negative sign. The blue dash-dotted line represents the polynomial approximating the sign function. The gray dashed line represents $x = \frac{1}{2}$.

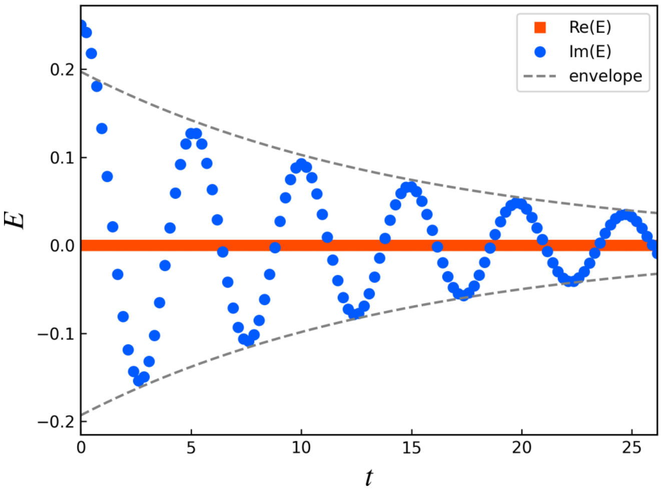

FIG. 13. Time evolution of the electric field E when using the quantum algorithm with $\Delta t = 0.238$ ($k = 0.4$). The envelope is a fitted exponential $\pm\exp(\gamma t)$.

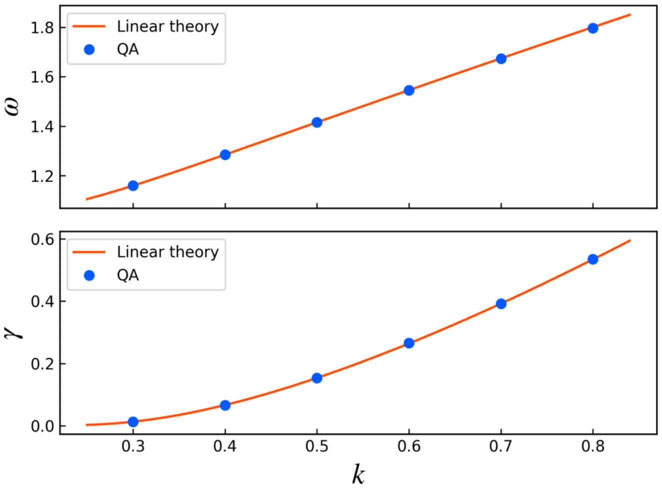

FIG. 14. Comparison of the frequencies and damping rates obtained from the results of the quantum algorithm (QA) for various wave numbers with those obtained from the linear Landau theory.

the linear theory for various wave numbers. The parameters obtained by fitting the curves agree well with the linear theory. These results indicate that our quantum algorithm accurately reproduces the linear Landau damping. Hereafter, the case $k = 0.4$ is discussed in both the quantum and classical algorithms.

Figure 15 shows the time evolution of E with the same time step $\Delta t = 0.238$ in the classical algorithm using the Euler method. Unlike the quantum algorithm, the imaginary component of E diverges numerically because of the long time step. Table IV shows the relative errors of ω and γ for the quantum and classical algorithms with different time steps. The classical algorithm requires a smaller time step Δt to obtain ω and γ with the same order of accuracy as in the quantum algorithm.

TABLE IV. Comparison of the frequencies and damping rates with different time steps obtained from the quantum algorithm (QA) and classical algorithm (CA) for $k = 0.4$.

	Relative error (%)	
Time step Δt	Frequency ω	Damping rate γ
QA		
2.38×10^{-1}	1.67×10^{-3}	1.60×10^{-1}
CA		
1.00×10^{-2}	6.25×10^{-2}	1.24×10^{1}
1.00×10^{-3}	8.77×10^{-3}	-1.12
5.00×10^{-4}	5.52×10^{-3}	-4.92×10^{-1}
1.00×10^{-4}	2.91×10^{-3}	6.51×10^{-3}

In the linearized Vlasov-Poisson system, the energy transfer between the particles and the electric field occurs and the distribution functions have a wavy structure. Figure 16 shows the velocity profiles of the distribution functions at different times. The distribution functions at $t = 8.32$, 16.65, and 24.97 have a wavy structure. At $t = 16.65$ and 24.97, the structure appears mainly around the phase velocity $v_\phi = \omega/k = 3.213$. These results are consistent with the linear Landau theory [48].

We validate the results of the distribution functions by comparing them with those of the classical algorithm. The error between different distribution functions is defined as

$$\delta(f, g) = \sum_j |f_j - g_j|^2 \Delta v, \tag{99}$$

where f and g are different distribution functions. We denote by f_{QA} and f_{CA} the distribution functions obtained from the quantum algorithm and the classical algorithm. The distribution function f_{CA} corresponds to the case when $\Delta t = 1 \times 10^{-4}$ and $k = 0.4$. We consider the result of this case as accurate because the time step Δt is sufficiently small. Table V shows that the errors between these distribution functions are sufficiently small. For reference, we show the error $\delta = 2.02 \times 10^{-5}$ between a Maxwellian distribution $f_{\mathrm{M}}(v)$

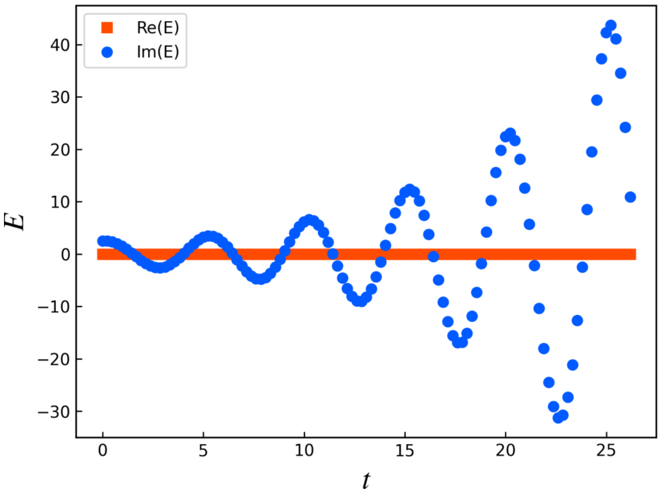

FIG. 15. Time evolution of the electric field E when using the classical algorithm using the Euler method with $\Delta t = 0.238$ ($k = 0.4$).

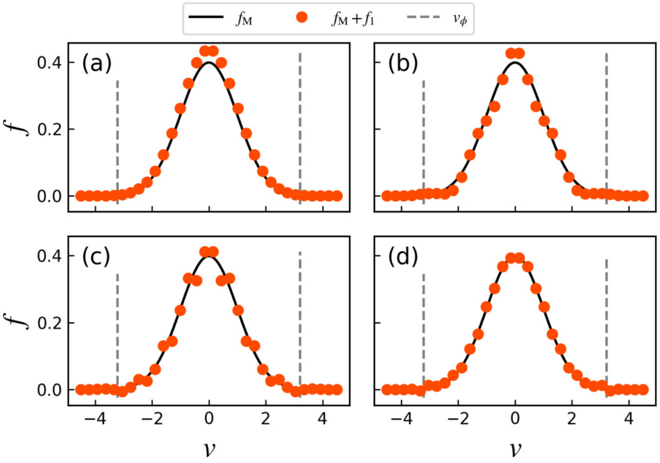

FIG. 16. Time evolution of the distribution function $f = f_{\mathrm{M}} + f_1$ ($k = 0.4$): (a) $t = 0.0$, (b) $t = 8.32$, (c) $t = 16.65$, and (d) $t = 24.97$. The black solid curve shows the Maxwellian $f = f_{\mathrm{M}}$ distribution function. The gray dashed lines show the theoretical phase velocity $v_\phi = 3.21$.

TABLE V. Errors between the distribution functions f_{QA} and f_{CA}, defined as in Eq. (99): f_{QA} is obtained from the quantum algorithm with $\Delta t = 0.238$ ($k = 0.4$); f_{CA} is obtained from the classical algorithm using the Euler method with $\Delta t = 1 \times 10^{-4}$ ($k = 0.4$).

t	δ
0	0.000
8.32	0.560
16.65	0.925
24.97	1.06

and a drift Maxwellian distribution $f_{\mathrm{M}}(v - 10^{-4})$ for $N_v = 32$. These results indicate that our quantum algorithm can reproduce precisely the structure of the distribution function.

VI. SUMMARY

In this study, we have shown how to apply the quantum singular-value transformation to the Hamiltonian simulation algorithm and discussed the error and query complexity of the HS using oblivious amplitude amplification and fixed-point amplitude amplification within the QSVT framework. As a result, the number of queries for the OAA-based HS scales as $O(t + \log(1/\varepsilon))$, whereas the FPAA-based one scales as $O(t \log(1/\varepsilon) + \log^2(1/\varepsilon))$, where t is an evolution time and ε is an error tolerance. In addition, we numerically compared the number of queries of these HS algorithms, showing that the number of queries of the OAA-based HS is smaller than that of the FPAA-based one, regardless of the parameters t and ε. Fitting the curve of the plotted data, we computed the constant factors and coefficients hidden behind the asymptotic scaling. Then we found that the values of the OAA-based HS are smaller than those of the FPAA-based HS. We also identified that the large number of queries of the FPAA-based HS is due to the high degree required to approximate the sign function. Therefore, the OAA method is more appropriate for the HS than the FPAA one.

Based on the above findings, by applying the OAA-based HS to the one-dimensional linearized Vlasov-Poisson system, we simulated the case of electrostatic Landau damping for various wave numbers on a classical emulator of a quantum computer using QISKIT [47]. The frequencies ω and damping rates γ obtained by curve fitting the time evolutions of the electric field E are in agreement with the linear Landau theory [48]. Moreover, the velocity profiles of the distribution function f that the quantum algorithm produces match the classical ones for the same velocity grid size N_v. These results show that the quantum algorithm can reproduce precisely the linear Landau damping with the structure of the distribution function.

We compared the results of the quantum algorithm using the HS with those of the classical algorithm using the Euler method for time. The classical algorithm with a large time step $\Delta t = 0.238$ causes numerical divergence. On the other hand, the quantum algorithm remains stable for the same Δt. This stability is because the state at the next time can be analytically determined by $U = \exp(-iHt)$, which is one of the features of the HS algorithm. The classical algorithm requires a smaller time step Δt to obtain ω and γ with the same order of accuracy as in the quantum algorithm. These results show that the HS algorithm has advantages in the time step over the classical algorithm using the Euler method.

We discussed the gate complexity of the algorithm for calculating the time evolution of the electric field E. The complexity scales logarithmically with the total grid size in velocity space N_v and linearly with the number of time steps N_t. We proposed an algorithm for obtaining the deviation from the Maxwell distribution. The gate complexity of this algorithm also scales logarithmically with N_v. Circuits of the unitaries which are block encodings of the Hamiltonian for the higher-dimensional systems were developed. The gate complexities of the HS using the circuits can be represented in the same form as the one-dimensional system and scale logarithmically with N_v. This result indicates that the quantum algorithm for the linearized Vlasov-Poisson system has exponential speedups over classical algorithms.

ACKNOWLEDGMENTS

This work was supported by MEXT Quantum Leap Flagship Program Grants No. JPMXS0118067285 and No. JPMXS0120319794, JSPS KAKENHI Grant No. 20H05966, and JST Grant No. JPMJPF2221.

APPENDIX: FROM QSP TO QSVT

Quantum singular-value transformation is based on the results of quantum signal processing [29]. Quantum signal processing is performed using a series of two gates W and S defined as

$$W(x) \equiv e^{i \arccos(x) X} = \begin{bmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{bmatrix} \tag{A1}$$

for $x \in [-1, 1]$ and

$$S(\phi) \equiv e^{i\phi Z} = \begin{bmatrix} e^{\phi} & 0 \\ 0 & e^{-\phi} \end{bmatrix}. \tag{A2}$$

These gates construct the gate sequence

$$W_{\Phi'} \equiv e^{i\phi'_0 Z} \prod_{k=1}^{d} W(x) e^{i\phi'_k Z}, \tag{A3}$$

where $\Phi' = (\phi'_0, \phi'_1, \ldots, \phi'_d) \in \mathbb{R}^{d+1}$. This convention is called the Wx convention in Ref. [32].

Another convention is the reflection convention, which uses a reflection gate R instead of W,

$$R(x) \equiv \begin{bmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{bmatrix}. \tag{A4}$$

The relationship between W and R is given by

$$W(x) = ie^{-i(\pi/4)Z} R(x) e^{-i(\pi/4)Z}. \tag{A5}$$

Therefore, Eq. (A3) is rewritten as

$$\begin{aligned} e^{i\phi'_0 Z} \prod_{k=1}^{d} W(x) e^{i\phi'_k Z} &= e^{i\phi_0 Z} \prod_{k=1}^{d} R(x) e^{i\phi_k Z} \\ &\equiv R_{\Phi}, \end{aligned} \tag{A6}$$

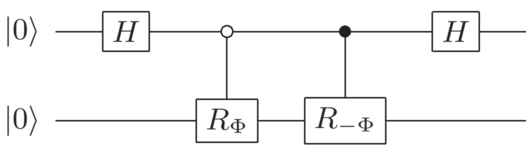

FIG. 17. Quantum circuit $U_{\mathrm{P}_\Re}$ that constructs a (1, 1, 0) block encoding of $P_\Re$.

where

$$\begin{aligned}\phi_0 &= \phi_0' + (2d-1)\frac{\pi}{4},\\ \phi_k &= \phi_k' - \frac{\pi}{2} \quad (k = 1, 2, \ldots, d-1),\\ \phi_d &= \phi_d' - \frac{\pi}{4}.\end{aligned} \tag{A7}$$

The phases $\Phi, \Phi' \in \mathbb{R}^{d+1}$ exist and the gate sequence constructs the (1, 1, 0) block encoding of a polynomial function $P \in \mathbb{C}$,

$$\langle 0|\, W_{\Phi'}\, |0\rangle = \langle 0|\, R_\Phi\, |0\rangle = P(x), \tag{A8}$$

if and only if conditions (i)–(iv) in Sec. III hold. If $P_\Re$ satisfies conditions (v) and (vi) in Sec. III, then there exists $P \in \mathbb{C}$ that satisfies $\mathrm{Re}(P) = P_\Re$ and the above conditions (i)–(iv).

Since $R^*(x) = R(x)$, if the complex conjugate of R_Φ is taken, we can get

$$R_\Phi^* = e^{-i\phi_0 Z} \prod_{k=1}^{d} R(x) e^{-i\phi_k Z} = \begin{bmatrix} P^*(x) & \cdot \\ \cdot & \cdot \end{bmatrix}, \tag{A9}$$

and R_Φ^* can be denoted by $R_{-\Phi}$. The quantum circuit in Fig. 17 constructs the (1, 2, 0) block encoding of $P_\Re$:

$$\begin{aligned}&\langle 0|\, \langle 0|\, U_{\mathrm{P}_\Re}\, |0\rangle\, |0\rangle\\ &\quad = \langle +|\, \langle 0|\, (|0\rangle\langle 0| \otimes R_\Phi + |1\rangle\langle 1| \otimes R_{-\Phi})\, |+\rangle\, |0\rangle\\ &\quad = \frac{P(x) + P^*(x)}{2}\\ &\quad = P_\Re(x).\end{aligned} \tag{A10}$$

Now we derive the result of the QSVT from that of QSP. Suppose that U is a $(1, a, 0)$ block encoding of a matrix A such that $A = \sum_{k=1}^{r} \sigma_k\, |w_k\rangle\langle v_k|$. The unitaries U and $e^{i\phi\Pi}$, where $\Pi = 2\,|0\rangle_a\langle 0| - I$, act on the two-dimensional invariant subspaces $\mathrm{Span}(|0\rangle_a |v_k\rangle, |\perp\rangle |v_k\rangle)$ and $\mathrm{Span}(|0\rangle_a |w_k\rangle, |\perp\rangle |w_k\rangle)$, where $|\perp\rangle$ satisfies $\langle \perp|0\rangle_a = 0$. The unitary U acts on these invariant subspaces as follows:

$$\begin{aligned}U\, |0\rangle_a |v_k\rangle &= \sigma_k\, |0\rangle_a |w_k\rangle + \sqrt{1-\sigma_k^2}\, |\perp\rangle |w_k\rangle,\\ U\, |\perp\rangle |v_k\rangle &= \sqrt{1-\sigma_k^2}\, |0\rangle_a |w_k\rangle - \sigma_k\, |\perp\rangle |w_k\rangle.\end{aligned} \tag{A11}$$

Therefore, U becomes

$$\begin{aligned}U &= \sum_k \begin{bmatrix} \sigma_k & \sqrt{1-\sigma_k^2} \\ \sqrt{1-\sigma_k^2} & -\sigma_k \end{bmatrix} \otimes |w_k\rangle\langle v_k|\\ &= \sum_k R(\sigma_k) \otimes |w_k\rangle\langle v_k|\end{aligned} \tag{A12}$$

and $U^\dagger$ becomes

$$\begin{aligned}U^\dagger &= \sum_k R^\dagger(\sigma_k) \otimes (|w_k\rangle\langle v_k|)^\dagger\\ &= \sum_k R(\sigma_k) \otimes |v_k\rangle\langle w_k|.\end{aligned} \tag{A13}$$

Moreover, Π acts on the invariant subspaces as follows:

$$\begin{aligned}\Pi \otimes I_s &= (|0\rangle_a\langle 0| - |\perp\rangle\langle\perp|) \otimes \sum_k |w_k\rangle\langle v_k|\\ &= \sum_k \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \otimes |w_k\rangle\langle v_k|\\ &= \sum_k Z \otimes |w_k\rangle\langle v_k|.\end{aligned} \tag{A14}$$

Therefore, $e^{i\phi\Pi}$ becomes

$$e^{i\phi\Pi} = \sum_k e^{i\phi Z} \otimes |w_k\rangle\langle v_k|. \tag{A15}$$

Now the alternating phase modulation sequence U_Φ defined in Eq. (7) becomes the $(1, a, 0)$ block encoding of $P^{(\mathrm{SV})}(A)$ as for odd d:

$$\begin{aligned}U_\Phi &= \sum_k e^{i\phi_0 Z} \prod_{j=1}^{d} [R(\sigma_k) e^{i\phi_j Z}] \otimes |w_k\rangle\langle v_k|\\ &= \begin{bmatrix} \sum_k P(\sigma_k)\, |w_k\rangle\langle v_k| & \cdot \\ \cdot & \cdot \end{bmatrix}\\ &= \begin{bmatrix} P^{(\mathrm{SV})}(A) & \cdot \\ \cdot & \cdot \end{bmatrix},\end{aligned} \tag{A16}$$

with a similar derivation for even d. We can construct the $(1, a+1, 0)$ block encoding of the real polynomial function $P_\Re^{(\mathrm{SV})}$ like QSP. Since $U_{-\Phi}$ is the $(1, a, 0)$ block encoding of $P^{*(\mathrm{SV})}$, we can construct a quantum circuit $U_{\mathrm{P}_\Re^{(\mathrm{SV})}}$:

$$\begin{aligned}&\langle 0|_b\, \langle 0|_a\, U_{\mathrm{P}_\Re^{(\mathrm{SV})}}\, |0\rangle_b\, |0\rangle_a\\ &\quad = ((\langle 0|_b H)\, \langle 0|_a\, (|0\rangle_b\langle 0| \otimes U_\Phi\\ &\qquad + |1\rangle_b\langle 1| \otimes U_{-\Phi})(H\, |0\rangle_b)\, |0\rangle_a\\ &\quad = \frac{\langle 0|_a\, (U_\Phi + U_{-\Phi})\, |0\rangle_a}{2}\\ &\quad = \frac{P^{(\mathrm{SV})}(A) + P^{*(\mathrm{SV})}(A)}{2}\\ &\quad = P_\Re^{(\mathrm{SV})}(A).\end{aligned} \tag{A17}$$

[1] L. K. Grover, in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (ACM, New York, 1996), pp. 212–219.

[2] P. W. Shor, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, Washington, DC, 1994), pp. 124–134.

[3] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett. **103**, 150502 (2009).

[4] 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).

[5] R. P. Feynman, Simulating physics with computers, Int. J. Theor. Phys. **21**, 467 (1982).

[6] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).

[7] F. Gaitan, Finding flows of a Navier–Stokes fluid through quantum computing, npj Quantum Inf. **6**, 61 (2020).

[8] F. Gaitan, Finding solutions of the Navier-Stokes equations through quantum computing—recent progress, a generalization, and next steps forward, Adv. Quantum Technol. **4**, 2100055 (2021).

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

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

[11] I. Novikau, E. A. Startsev, and I. Y. Dodin, Quantum signal processing for simulating cold plasma waves, Phys. Rev. A **105**, 062444 (2022).

[12] A. Ameri, E. Ye, P. Cappellaro, H. Krovi, and N. F. Loureiro, Quantum algorithm for the linear Vlasov equation with collisions, Phys. Rev. A **107**, 062412 (2023).

[13] 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).

[14] S. Wang, Z. Wang, W. Li, L. Fan, Z. Wei, and Y. Gu, Quantum fast Poisson solver: The algorithm and complete and modular circuit design, Quantum Inf. Process. **19**, 170 (2020).

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

[16] A. Suau, G. Staffelbach, and H. Calandra, Practical quantum computing: Solving the wave equation using a quantum approach, ACM Trans. Quantum Comput. **2**, 1 (2021).

[17] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, Commun. Math. Phys. **270**, 359 (2007).

[18] A. M. Childs and R. Kothari, Limitations on the simulation of non-sparse Hamiltonians, Quantum Inf. Comput. **10**, 669 (2010).

[19] D. W. Berry and A. M. Childs, Black-box Hamiltonian simulation and unitary implementation, Quantum Inf. Comput. **12**, 29 (2012).

[20] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, Quantum Inf. Comput. **12**, 901 (2012).

[21] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing* (ACM, New York, 2014), pp. 283–292.

[22] D. W. Berry, A. M. Childs, and R. Kothari, in *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, Washington DC, 2015), pp. 792–809.

[23] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian dynamics with a truncated taylor series, Phys. Rev. Lett. **114**, 090502 (2015).

[24] D. W. Berry and L. Novo, Corrected quantum walk for optimal Hamiltonian simulation, Quantum Inf. Comput. **16**, 1295 (2016).

[25] L. Novo and D. W. Berry, Improved Hamiltonian simulation via a truncated Taylor series and corrections, Quantum Inf. Comput. **17**, 623 (2017).

[26] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, Proc. Natl. Acad. Sci. USA **115**, 9456 (2018).

[27] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, Phys. Rev. Lett. **118**, 010501 (2017).

[28] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, Quantum **3**, 163 (2019).

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

[30] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (American Association of Physics Teachers, College Park, 2010).

[31] M. Szegedy, in *45th Annual IEEE Symposium on Foundations of Computer Science* (IEEE Computer Society, Washington, DC, 2004), pp. 32–41.

[32] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, Grand unification of quantum algorithms, PRX Quantum **2**, 040203 (2021).

[33] A. Paetznick and K. M. Svore, Repeat-until-success: Nondeterministic decomposition of single-qubit unitaries, Quantum Inf. Comput. **14**, 1277 (2014).

[34] A. Daskin and S. Kais, An ancilla-based quantum simulation framework for non-unitary matrices, Quantum Inf. Process. **16**, 33 (2017).

[35] L. K. Grover, Fixed-point quantum search, Phys. Rev. Lett. **95**, 150501 (2005).

[36] T. Tulsi, L. K. Grover, and A. Patel, A new algorithm for fixed point quantum search, Quantum Inf. Comput. **6**, 483 (2006).

[37] T. J. Yoder, G. H. Low, and I. L. Chuang, Fixed-point quantum search with an optimal number of queries, Phys. Rev. Lett. **113**, 210501 (2014).

[38] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, Quantum Comput. Inf. **305**, 53 (2002).

[39] G. H. Low, T. J. Yoder, and I. L. Chuang, Methodology of resonant equiangular composite quantum gates, Phys. Rev. X **6**, 041067 (2016).

[40] G. H. Low and I. L. Chuang, Hamiltonian simulation by uniform spectral amplification, arXiv:1707.05391.

[41] R. Chao, D. Ding, A. Gilyén, C. Huang, and M. Szegedy, Finding angles for quantum signal processing with machine precision, arXiv:2003.02831.

[42] R. Chao, D. Ding, A. Gilyén, C. Huang, and M. Szegedy, Finding angles for quantum signal processing with machine precision, arXiv:2003.02831; https://github.com/ichuang/pyqsp.

[43] G. H. Low, Quantum signal processing by single-qubit dynamics, Ph.D. thesis, Massachusetts Institute of Technology, 2017.

[44] K. Mitarai, K. Toyoizumi, and W. Mizukami, Perturbation theory with quantum signal processing, Quantum **7**, 1000 (2023).

[45] A. Prakash, Ph.D. thesis, Quantum algorithms for linear algebra and machine learning, University of California, Berkeley, 2014.

[46] V. Giovannetti, S. Lloyd, and L. Maccone, Architectures for a quantum random access memory, Phys. Rev. A **78**, 052310 (2008).

[47] M. S. Anis *et al.*, *Qiskit: An Open-Source Framework for Quantum Computing* (IBM, Yorktown Heights, 2021).

[48] F. F. Chen, *Introduction to Plasma Physics and Controlled Fusion* (Springer, New York, 1984), pp. 224–232.