Quantum Signal Processing (QSP) for simulating cold plasma waves

I. Novikau,1 E. A. Startsev,1 and I. Y. Dodin1,2
1)Princeton Plasma Physics Laboratory, Princeton, New Jersey 08543, USA
2)Department of Astrophysical Sciences, Princeton University, Princeton, New Jersey 08544, USA
(*Electronic mail: inovikau@pppl.gov)

(Dated: 16 December 2021)

Numerical modeling of radio-frequency waves in plasma with sufficiently high spatial and temporal resolution remains challenging even with modern computers. However, such simulations can be sped up using quantum computers in the future. Here, we propose how to do such modeling for cold plasma waves, in particular, for an X wave propagating in an inhomogeneous one-dimensional plasma. The wave system is represented in the form of a vector Schrödinger equation with a Hermitian Hamiltonian. Block-encoding is used to represent the Hamiltonian through unitary operations that can be implemented on a quantum computer. To perform the modeling, we apply the so-called Quantum Signal Processing algorithm and construct the corresponding circuit. Quantum simulations with this circuit are emulated on a classical computer, and the results show agreement with traditional classical calculations. We also discuss how our quantum circuit scales with the resolution.

I. INTRODUCTION

Radio-frequency (RF) waves are widely used in experiments with magnetically confined fusion plasmas. At low power, these waves can serve as a diagnostic tool. At high power, they can help control fusion plasmas through RF heating2–4 and also current drive, which is mainly done with lower-hybrid5,6 and electron-cyclotron waves7 and can also help suppress instabilities8–10. Many of these RF techniques are practiced on existing fusion devices11,12, and current drive in particular is now envisioned to play a large role for achieving steady-state operation and for suppressing instabilities in future devices13–15.

Many RF codes have been developed in the last decades to facilitate these applications. However, precise simulations of certain RF processes, such as the energy deposition of waves at high-order cyclotron resonances16, mode conversions17, and even basic electron–cyclotron waves often require significant computational resources that remain beyond the reach of modern computers18,19, especially for three-dimensional simulations. Quantum Computing (QC) may be a solution to this problem. In particular, fluid linear RF problems are good candidates for the application of QC as discussed in Ref. [19].

In this work, we consider only the circuit model of quantum computation, where information is stored in a set of \( n_q \) quantum bits called qubits20,21. Being entangled, the qubits create a configuration space described by a \( 2^{n_q} \)-dimensional complex vector, and such an exponential scaling with \( n_q \) can be beneficial in large-dimensional problems. A quantum circuit consists of a sequence of so-called gates. Some of the gates operate in parallel, and the longest path between the input and output points of the circuit is called circuit depth, which is roughly proportional to the runtime. A gate that acts on \( m_q \) qubits can be represented by a \( 2^{m_q} \times 2^{m_q} \) unitary matrix, while the whole quantum circuit is described by a \( 2^{n_q} \times 2^{n_q} \) unitary matrix \( U \). This matrix is applied to an initial state \( \psi(0) \), generating an output state \( \psi(t) = U \psi(0) \), from which classical information is then extracted via a classical measurement. In particular, in this paper we focus on Quantum Hamiltonian Simulations (QHS) of systems with Hermitian Hamiltonians \( \mathcal{H} \) that are independent of time \( t \); then, \( U = \exp(-i\mathcal{H}t) \).

In recent years, there has been a significant development of QC applications to physics problems, including simulations of classical systems. Various methods have been proposed to solve the wave equation, Poisson’s equation, Maxwell’s equations, and first-order linear hyperbolic systems, the Navier–Stokes equation, the Boltzmann equation, and to simulate advection–diffusion processes. It was also shown that most of quantum algorithms can be considered as special cases of a more general algorithm called Quantum Singular Value Transform. This technique is based on the so-called Quantum Signal Processing (QSP) paradigm, which was originally developed for efficient QHS. The essence of the QSP is in encoding polynomials of given matrices into sequences of rotations. In the QHS, in particular, the QSP searches for a polynomial to approximate the exponential function of the system Hamiltonian, which is block-encoded into an auxiliary unitary. This state-of-the-art quantum method is now extensively studied in the QC community and holds promise as a universal numerical framework applicable to any linear Hamiltonian problem. For instance, it was recently applied to plasma zero-dimensional kinetic simulations.

Here, we show that QSP can be constructed efficiently for QHS of cold-plasma waves and present a working algorithm. We implement the QSP using QuEST computing toolkit. Specifically, we model X-wave propagation in a one-dimensional electron plasma with an inhomogeneous static magnetic field and inhomogeneous density. The corresponding quantum simulations are emulated on a classical computer and show agreement with classical simulations of the same problem. We discuss how the resulting QSP quantum circuit scales if one modifies the grid resolution, the precision of the QSP approximation and the simulated time interval.

Our paper is organized as follows. In Sec. III we discuss cold-wave dynamics within an electron-fluid model and present its Schrödinger form. In Sec. IV we construct the...
corresponding one-dimensional model and derive the Hamiltonian that is used later in our QHS. It is also shown in Sec. III how to encode the resulting plasma system into a quantum circuit and how to initialize the circuit. The QSP algorithm is explained in Sec. IV and the block-encoding of the wave Hamiltonian is constructed in Sec. V. The comparison of the QSP to classical simulations and the scaling of the QSP circuit are presented in Sec. VI. Finally, the advantages and challenges of applying QSP to classical plasma problems are discussed in Sec. VII. A reader not familiar with the quantum computing is encouraged to read a brief introduction into the field presented in Appendix A.

II. COLD-PLASMA WAVES

We assume a cold fluid electron plasma with density \( n(r) \) immersed into a background magnetic field \( B_0(r) \). Linear waves in such plasma can be described by the following equations:

\[
\begin{align*}
\partial_t \tilde{v} &= -\tilde{E} \times B_0 - \tilde{E}, \\
\partial_t \tilde{E} &= n \tilde{v} + \nabla \times \tilde{B}, \\
\partial_t \tilde{B} &= -\nabla \times \tilde{E},
\end{align*}
\]

where \( \tilde{v} \) is the electron fluid velocity, and \( \tilde{E} \) and \( \tilde{B} \) are the wave electric and magnetic fields, respectively. Time is measured in units of plasma frequency \( \omega_p \), where \( e \) is the absolute value of the electron charge, \( m \) is the electron mass, and \( n_0 \) is the maximum value of \( n(r) \). The velocity is normalized to the speed of light \( c \), while the fields are normalized to \( c \sqrt{4\pi n_0 e^2} \), and the space coordinate is normalized to \( \kappa_s = c \omega_p^{-1} \).

As shown in Ref. 13, one can also rewrite Eqs. (1) in a form of a Schrödinger equation

\[
\frac{\partial |\psi\rangle}{\partial t} = -i \mathcal{H} |\psi\rangle,
\]

where \( |\psi\rangle = (\sqrt{n} \tilde{v}, \tilde{E}, \tilde{B}) \), and \( \mathcal{H} \) serves as a time-independent Hamiltonian, which is Hermitian if the system has suitable boundary conditions (periodic or Dirichlet). The corresponding dynamics is described by

\[
|\psi(t)\rangle = e^{-i \mathcal{H} t} |\psi(0)\rangle,
\]

where \( \exp(-i \mathcal{H} t) \) is a unitary operator.

Equations (1) satisfy the Poynting’s theorem \( \int W dV + \int N \cdot dS = 0 \), where \( W = n|\tilde{v}|^2 + |\tilde{E}|^2 + |\tilde{B}|^2 \) is the system energy density, and \( N = \tilde{E} \times \tilde{B} \) is the Poynting vector. With appropriate boundary conditions (such as the Dirichlet boundary conditions), the integral over the surface \( S \) disappears and the total energy \( \int W dV \) is conserved.

The system (1) supports several waves, which are often identified as the O and X waves. Below, we focus on propagation of the X wave perpendicular to \( B_0 \). This wave has elliptic field polarization in the \((x, y)\) plane, where the \(x\)-axis goes along the gradients of the background field \( B_0 \) and density \( n \), and \( z \) is directed along \( B_0 \). It exhibits the so-called upper-hybrid (UH) resonance at frequency

\[
\omega_{\text{UH}} = \sqrt{B_0^2 + n}
\]

and the \( L \) cutoff at frequency

\[
\omega_L = \frac{1}{2} \left( B_0 + \sqrt{B_0^2 + 4n} \right),
\]

where all units are normalized as specified above (we do not consider R cutoff here).

III. ONE-DIMENSIONAL MODEL

A. Field configuration

We reduce the equations (1) to a one-dimensional system of size \( 2r_0 \). The corresponding space grid \( x \in [-r_0/\kappa_s, r_0/\kappa_s] \) has \( N_x \) points. For convenience, we define also a space grid \( s \) via

\[
s = x/\max(x)
\]

to have \( s \in [-1, 1] \). Because we focus on X-wave propagation, it is enough for us to keep only \( \tilde{v}_x, \tilde{v}_y, \tilde{E}_x, \tilde{E}_y, \) and \( \tilde{B}_z \), and we impose the Dirichlet boundary conditions

\[
\tilde{E}_y(-r_0) = \tilde{E}_y(r_0) = 0, \quad \tilde{B}_z(-r_0) = \tilde{B}_z(r_0) = 0.
\]

To mimic the effect of the antenna that launches the wave, we introduce an auxiliary oscillator \( Q \) with initial amplitude \( Q_0 \). Coupled to the magnetic field \( \tilde{B}_z \), the oscillator gradually transfers its energy to the X wave, while the energy of the whole system remains constant.

We seek to demonstrate wave tunneling through the L cutoff and localization of the wave energy at the UH resonance. Therefore, the case where the L cutoff and the UH resonance appear close enough to each other is considered. The Clemmow-Mullaly-Allis (CMA) diagram (Fig. 1a) shows the corresponding wave trajectory in the parameter space \( (\omega_L, \omega_p) \), where \( \omega_L = B_0 \) is the electron frequency and \( \omega_p = \sqrt{n} \) is the plasma frequency. The X wave is excited by a source placed at the center of the system, where the background density (represented by the plasma frequency) and magnetic field (represented by the cyclotron frequency) are low. Therefore, its trajectory starts at the bottom left corner of the CMA diagram. The wave propagates in both directions. The horizontal black arrow corresponds to the wave propagation toward the high-density cutoff–resonance pair (HCR). The X wave is partly reflected at the cutoff, partly tunnels through the narrow forbidden zone (gray area) beyond the cutoff (blue line), and accumulates at the UH resonance (red line), where its wavenumber ends up growing indefinitely. This configuration is of a particular interest since it allows to model the wave behaviour at the resonance, where high resolution of the spatial and time grids is required.

This demonstration can be impeded by wave reflection from the right boundary. To suppress such reflection, supplemental background profiles (Fig. 1b) are introduced, which ensures X-wave trapping inside an additional UH resonance.
More precisely, the additional profiles give rise to a cutoff–resonance pair at low density (LCR) and an isolated UH resonance (UHR) as shown on the CMA diagram (Fig. 1a). Due to the low density, the LCR pair does not have a forbidden zone. This allows the X wave to traverse the pair without reflection and reach the UHR. Figure 1d indicates the position of the cutoff–resonance pairs in space.

Classical simulations of this system (Figs. 2a-2b) show that the wave behaves in the manner described above, as desired. Specifically, the left-propagating wave reaches the HCR pair (white dashed line), where a part of its energy is reflected from the L cutoff, while the tunneled energy becomes localized at the corresponding UH resonance of the HCR. The right-propagating wave propagates without reflection towards the UHR (red dashed line). By trapping the X wave, the UHR restrains the wave propagation towards the right boundary.

B. Rescaled variables

In terms of the rescaled velocity \( \tilde{\xi} = \sqrt{\nu} \), our 1-D model can be written explicitly as

\[
\begin{align*}
\imath \partial_t \xi_n(x,t) &= -i B_0(x) \xi_n(x,t) - \imath \sqrt{n(x)} \tilde{E}_x(x,t), \\
\imath \partial_t \tilde{E}_x(x,t) &= i B_0(x) \tilde{\xi}_n(x,t) - \imath \sqrt{n(x)} \tilde{E}_y(x,t), \\
\imath \partial_t \tilde{E}_y(x,t) &= \imath \sqrt{n(x)} \tilde{\xi}_n(x,t), \\
\imath \partial_t \tilde{B}_x(x,t) &= \imath \sqrt{n(x)} \tilde{\xi}_n(x,t) - \imath \partial_t \tilde{E}_x(x,t), \\
\imath \partial_t \tilde{B}_y(x,t) &= \imath \sqrt{n(x)} \tilde{\xi}_n(x,t) - \imath \partial_t \tilde{E}_y(x,t), \\
\imath \partial_t \tilde{Q}(x,t) &= - \omega_s \tilde{Q}(x,t) - \beta \tilde{B}_z(x,t), \\
\tilde{Q}(x_1, 0) &= Q(x_1, 0) = Q_0,
\end{align*}
\]

where the source \( Q \) with a constant frequency \( \omega_s \) is coupled to the magnetic field \( \tilde{B}_z \) using an ad hoc coupling coefficient \( \beta \).

The 1-D model can be rewritten in the Hamiltonian form (Eq. 2) with

\[
\psi = (\tilde{\xi}_x, \tilde{\xi}_y, \tilde{E}_x, \tilde{E}_y, \tilde{B}_z, Q) \top,
\]

where \( \top \) denotes transposition. The energy density is represented now as

\[
\begin{align*}
W_{\text{tot}} &= \psi \psi^\top = W_e + W_{eb} + W_q, \\
W_e &= |\tilde{\xi}_x|^2 + |\tilde{\xi}_y|^2, \\
W_{eb} &= |\tilde{E}_x|^2 + |\tilde{E}_y|^2 + |\tilde{B}_z|^2, \\
W_q &= |Q|^2.
\end{align*}
\]

C. Discretization

The x-axis is represented by \( N_x = 2^n \) points with a step \( h \) (Fig. 3). Equations 9 are discretized in space using the central differencing scheme. The boundary conditions for the velocity are

\[
\begin{align*}
\imath \partial_t \tilde{\xi}_{x,0} &= i B_{0,0} \tilde{\xi}_{x,0}, \\
\imath \partial_t \tilde{\xi}_{x,N_x-1} &= i B_{0,N_x-1} \tilde{\xi}_{x,N_x-1},
\end{align*}
\]

for the wave electric field they are

\[
\begin{align*}
\imath \partial_t \tilde{E}_{y,0} &= 0, \\
\imath \partial_t \tilde{E}_{y,1} &= i \sqrt{n_{x,1}} \tilde{\xi}_{y,1} - \frac{i}{2h} \tilde{B}_{z,2}, \\
\imath \partial_t \tilde{E}_{y,N_x-2} &= i \sqrt{n_{x,N_x-2}} \tilde{\xi}_{y,N_x-2} + \frac{i}{2h} \tilde{B}_{z,N_x-3}, \\
\imath \partial_t \tilde{E}_{y,N_x-1} &= 0,
\end{align*}
\]
and for the magnetic field they are

\[ i \partial_t B_{z,0} = 0, \]
\[ i \partial_t B_{z,1} = -\frac{i}{2h} E_{y,2}, \]
\[ i \partial_t B_{z,N_z-2} = \frac{1}{2h} E_{y,N_z-3}, \]
\[ i \partial_t B_{z,N_z-1} = 0. \]

Since the source \( Q \) interacts with the wave magnetic field only at the center of the system (Eqs. [12],[13], Fig. 3), it does not enter the boundary conditions. The corresponding Hamiltonian can be expressed as

\[
\mathcal{H} = \begin{pmatrix}
0 & -iB_0 & -i\sqrt{\pi} & 0 & 0 & 0 \\
B_0 & 0 & 0 & -i\sqrt{\pi}\varepsilon & 0 & 0 \\
-i\sqrt{\pi} & 0 & 0 & 0 & 0 & 0 \\
0 & i\sqrt{\pi}\varepsilon & 0 & 0 & M_h & 0 \\
0 & 0 & 0 & M_h & 0 & M_\beta \\
0 & 0 & 0 & 0 & M_\beta & M_{\alpha_0} \\
\end{pmatrix}.
\] (15)

Here, \( \varepsilon = \text{diag}(0, 1, \ldots, 1, 0) \), \( M_h \) is the matrix representation of the operator \( \delta_x \):

\[
M_h = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & -\frac{1}{\Delta} & 0 \\
0 & \frac{\varepsilon}{\Delta} & 0 & -\frac{1}{\Delta} \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 \\
\end{pmatrix}.
\] (16)

The matrix \( M_\beta \) describes the source-wave coupling

\[
M_\beta = \begin{pmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \beta \\
0 & 0 & 0 & 0 & 0 & -\beta \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{pmatrix}.
\] (17)

while \( M_{\alpha_0} \) encodes the source frequency:

\[
M_{\alpha_0} = \begin{pmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & -\alpha_0 \\
0 & 0 & 0 & 0 & 0 & -\alpha_0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{pmatrix}.
\] (18)

D. Quantum encoding of plasma signals

To encode our discretized system into a quantum circuit, we map \( \psi \) on two registers: \( |d\rangle \) and \( |j\rangle \). The register \( |j\rangle \) has \( n_s = \log_2 N_s \) qubits and stores the space dependence of every variable. That is, \( |j\rangle \) contains the binary representation of the spatial-point indices in the \( x \)-grid. The register \( |d\rangle \) encodes the variable index:

\[
d = 0 \leftrightarrow \xi_x, \quad (19a)
\]
\[
d = 1 \leftrightarrow \xi_y, \quad (19b)
\]
\[
d = 2 \leftrightarrow E_x. \quad (19c)
\]
etc. Since we have six independent fields in \( \psi \), the register \(|d\rangle \) must have at least three qubits. Then,

\[
\psi = A_{d,j} |d⟩ |j⟩ \equiv A_{d,j} |d_2 d_1 d_0⟩ |j_{n_2-1} \ldots j_2 j_1 j_0⟩, \quad (20)
\]

where \( d_k \) and \( j_k \) take values of 0 or 1; \( A_{d,j} \) stores the amplitude of the variable with the index \( d \) at \( x_j \). For instance, the value of \( B_{2}(x=x_{5}) \) is stored in \( A_{d=4,j=5} \), which is the amplitude of the quantum state \(|100⟩_d |0\ldots0101⟩_j \). We assume that the rightmost qubit is the least-significant one (which is responsible for the parity), which is the bottom qubit in the quantum circuit. At the same time, once qubits are represented by a classical column vector, the amplitude of the least-significant qubit is stored in the first two top elements of the column vector as shown in Eq. \( \ref{eq:column_vector} \).

\[
\left( \begin{array}{c} 101 \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \\vdots \"}
is achieved by using the ancilla register \( |R\rangle \) in every row and column. The extension of the space \( \mathcal{H} \) is discussed in Section \( \text{V} \).

B. Block-encoding

The operator \( \mathcal{H} \) (Eq. 31) is discussed in Section \( \text{V} \). The corresponding circuit is shown in Fig. 6, which contains some oracle \( U_H \) and its complex-conjugate \( U_H^\dagger \) form the extended operator \( U_H' \) (Eq. 30). The left Pauli \( X \) gate implements the operator \( S \) (Eq. 29). The subcircuit within the blue box implements the reflector \( U_R \) (Eq. 32).

C. Qubitization

The purpose of the QSP is to build the desired polynomial \( f(\mathcal{H}) \). Because \( U_H \) is linear in \( \mathcal{H} \), multiple applications of \( U_H \) are required, where the block-encoding matrix acts as:

\[
U_H |0\rangle_a |\lambda\rangle_s = |0\rangle_a |\lambda\rangle_s + \sqrt{1 - \lambda^2} |\perp\rangle_{a,s}.
\]

Here, \( |\lambda\rangle \) are the eigenvalue and eigenvector of \( \mathcal{H} \), respectively; the state \( |\perp\rangle_{a,s} \) is created by the undefined part of \( U_H \) marked with dots in Eq. (24). To build higher powers of \( \mathcal{H} \), we need to stay within the space spanned by \( \langle 0|_a |\lambda\rangle_s, U_H |0\rangle_a |\lambda\rangle_s \). The problem is that in general, this space is not invariant under the action of \( U_H \). Next every application of \( U_H \) adds additional perpendicular vectors that are different from the original \( |\perp\rangle_{a,s} \). To overcome this issue, one can decompose (qubitize) the entire Hilbert space into two-dimensional orthogonal subspaces, \( \mathcal{P}_\lambda \), where each subspace \( \mathcal{P}_\lambda \) corresponds to a particular \( |\lambda\rangle \) of \( \mathcal{H} \). After that, one replaces \( U_H \) by a new operator \( W \) that performs rotation in each of these disjoint subspaces. As a result, the operator \( W \) block-encodes \( \mathcal{H} \) and by acting in an invariant space \( \mathcal{P}_\lambda = \langle 0|_a |\lambda\rangle_s, W |0\rangle_a |\lambda\rangle_s \) for each \( |\lambda\rangle \) can produce necessary moments of \( \mathcal{H} \) to construct the polynomial \( f(\mathcal{H}) \).

According to Lemma 10 from Ref. [35], \( W \) can always be constructed using the following procedure. First of all, one extends again Hilbert space with an ancilla \( |q\rangle \) initialized in the superposition state \( |+\rangle \) (Eq. (28)):

\[
|0\rangle_a \rightarrow |+\rangle_q |0\rangle_a.
\]

Then, one applies an \( X \) gate (Eq. (27)) to the ancilla \( |q\rangle \):

\[
S = X_q \otimes I_{a,s}
\]

and uses two copies of controlled \( U_H \):

\[
U_H' = |0\rangle_q \langle 0|_q \otimes U_H + |1\rangle_q \langle 1|_q \otimes U_H^\dagger.
\]

Combining the above operators, one constructs

\[
W = (U_R \otimes I_q) S U_H' R.
\]

where the so-called reflector \( U_R \):

\[
U_R = 2 |+\rangle_q \langle 0|_q |0\rangle_a \langle +|_q - I_q a,
\]

keeps unchanged the zero-ancilla state and inverts the sign of the perpendicular state. Here, \( I_q \) is the unit operator that acts on an ancilla \( |q\rangle \). The corresponding circuit is shown.
FIG. 8: General circuit of the oracle $O_F$. Its components are shown in Fig. 9. According to Table I, the block $E_x$ does not modify ancillae $|a_j⟩|a_d⟩$.

in Fig. 7 As it is proven in Lemma 8 and Lemma 10 from Ref. 35, the above construction guarantees that the matrix form of the operator $W$ is represented by the following direct sum:

$$W = \bigoplus_\lambda \left( \frac{\lambda}{\sqrt{1-\lambda^2}} - \sqrt{1-\lambda^2} \frac{\lambda}{\sqrt{1-\lambda^2}} \right),$$

(33)

which ensures the invariance of each subspace $P_\lambda$ under the action of $W$.

V. BLOCK-ENCODING OF THE WAVE HAMILTONIAN

A. General algorithm

To encode the wave Hamiltonian (Eq. 15), we apply the state-preparation technique. The standard procedure is described in Appendix B. In this algorithm, one finds positions of all nonzero elements within $H$ and then encodes values of these elements into the amplitudes of quantum states. First of all, for each row of $H$ one stores the column indices of nonzero matrix elements in an ancilla register. Being an integer, each index is encoded as a quantum state represented by a bit-string of qubits. After that, knowing the column and row indices, one finds the values of the corresponding nonzero elements. The values are encoded in the amplitudes of the quantum states. Since any element of the rescaled Hamiltonian $H/(\xi M)$ (Eq. 24) is less than (or equal to) unity by the absolute value, it can be represented as $\cos(\theta/2)$ of a certain angle $\theta$. The cosine can be generated by the rotation gate $R_y(\theta)$:

$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$$

(34)

acting on the zero state:

$$R_y(\theta) |0⟩ = \cos(\theta/2) |0⟩ + \sin(\theta/2) |1⟩.$$  

(35)

B. Normalization

To normalize the wave Hamiltonian (Eq. 15), we adopt:

$$\mathcal{H}_{qsp} = \beta_H \mathcal{H}, \quad t_{qsp} = 1/\beta_H, \quad \beta_H = \frac{1}{d_H M},$$

(36)
TABLE I: Action of the oracle \( O_F \). An input variable is encoded into the register \( |d\rangle \), and its space dependence is taken from the register \( |j\rangle \). The operator \( O_F \) returns the registers \( |a_1\rangle |d_a\rangle \) in the indicated output states, while the states of the registers \( |d\rangle \) remain unchanged.

| Variable | Input\(^a\) \( |d\rangle \) | Output \( |0\rangle_{a_d} |001\rangle_{a_d} + |000\rangle_{a_d} |100\rangle_{a_d} \) |
|----------|------------------|------------------|
| \( \xi_x \) | \( |000\rangle_d |0;N_x - 1\rangle \) | \( |00\rangle_{a_d} |000\rangle_{a_d} + |000\rangle_{a_d} |101\rangle_{a_d} \) |
| \( \xi_y \) | \( |001\rangle_d |1;N_x - 2\rangle \) | \( |00\rangle_{a_d} |000\rangle_{a_d} + |000\rangle_{a_d} |101\rangle_{a_d} \) |
| \( E_x \) | \( |010\rangle_d |0;N_x - 1\rangle \) | \( |00\rangle_{a_d} |000\rangle_{a_d} \) |
| \( E_y \) | \( |011\rangle_d |0;N_x - 1\rangle \) | \( |00\rangle_{a_d} |000\rangle_{a_d} + |00\rangle_{a_d} + |10\rangle_{a_d} |100\rangle_{a_d} \) |
| \( \hat{B}_z \) | \( |100\rangle_d |0;N_x - 1\rangle \) | \( |00\rangle_{a_d} |000\rangle_{a_d} + |00\rangle_{a_d} + |10\rangle_{a_d} |100\rangle_{a_d} \) |
| \( Q \) | \( |101\rangle_d |j \neq \bar{j}_Q\rangle \) | \( |00\rangle_{a_d} |010\rangle_{a_d} + |00\rangle_{a_d} + |00\rangle_{a_d} |101\rangle_{a_d} \) |

\(^a\) \( |0;N_x - 1\rangle \) denotes \( |0\rangle \) or \( |N_x - 1\rangle \) states; \( |1;N_x - 2\rangle \) denotes all states from \( |1\rangle \) up to \( |N_x - 2\rangle \); \( |j\rangle \) \( Q \) corresponds to the spatial position of the source \( Q \).

TABLE II: Action of the oracle \( O_{\sqrt{\Pi}} \). The operator is controlled by the indicated input registers. For various bit arrays encoded in these registers, it returns the indicated output amplitudes on the ancilla \( a_1 \) (or \( a_2 \)) as explained in Eq. 35.

| Input | Output amplitude |
|-------|------------------|
| \( |001\rangle_{a_d} |000\rangle_d \) | \( \sqrt{\beta_H}B_0 \) |
| \( |010\rangle_{a_d} |000\rangle_d \) | \( \sqrt{\beta_H B_{0_0}^{1/2}} \) |
| \( |011\rangle_{a_d} |001\rangle_d \) | \( \sqrt{\beta_H/(2\hbar)} \) |
| \( |00\rangle_{a_d} |010\rangle_d \) | \( \sqrt{\beta_H} \) |
| \( |00\rangle_{a_d} |001\rangle_d \) | \( \sqrt{\beta_H} \) |
| \( |10\rangle_{a_d} |100\rangle_d \) | \( \sqrt{\beta_H} \) |
| \( |10\rangle_{a_d} |101\rangle_d \) | \( \sqrt{\beta_H} \) |

or more explicitly,

\[ \beta_H = \frac{1}{d_H^2 \sqrt{\alpha_H^2 + \frac{1}{2\hbar} + B^2 + \alpha_d^2}} \]  

(37)

Here, \( \mathcal{H}_\text{qep} \) is the rescaled Hamiltonian, \( t_{\text{qep}} \) is the time interval to be simulated by the QSP circuit, \( \alpha_{\text{H,max}} = \sqrt{B_{0,\text{max}}^2 + n_{\text{max}}} \), and \( n_{\text{max}} \) and \( B_{0,\text{max}} \) are the maximum values of the background density and magnetic field, respectively. In our case, \( d_I^2 = 4 \), which is close to the sparsity \( \zeta = 3 \) of our wave Hamiltonian (Eq. 15), as it should be according to Eq. 24. By sparsity, we mean the number of nonzero elements in a matrix row maximized over the row.

![FIG. 10: Circuit of the control block \( JQ \) in a system with \( n_s = 4 \), where the source \( Q \) is placed at spatial points with indices \( j_{Q,1} = 8 \) and \( j_{Q,2} = 9 \).](image-url)
elements are encoded) has to be created from a single input state (where the row index is encoded). This can be done by applying the Hadamard gate (Eq. 12), and one needs at least $n_H = 2$ Hadamard gates to produce the superposition of $\zeta = 3$ states. However, the amplitude of each state in the resulting superposition has an additional multiplier $1/2^{n_H/2}$. Then, according to Eq. 35 and 42 we encode the square root of the absolute value of the matrix element $v$ as

$$\sqrt{\frac{|v|}{d_H^2 M}} = \frac{1}{2^{n_H/2}} \cos(\theta/2),$$

(38)

where $\cos(\theta/2) = \sqrt{|v|/M} \leq 1$, and therefore, $d_H = 2^{n_H/2}$.

Note that larger $t_{QSP}$ requires calculation of a higher number of QSP phases, which can be challenging.
doing that, we represent the QSP circuit with \( n \) sequential copies of a shorter circuit constructed for the time interval \( t_{\text{rep}}/n \). In this case, performing simulations on a digital emulator, one can also analyse intermediate quantum states.

C. Ancilla registers

To implement the state-preparation method (Appendix B), we introduce several ancilla registers (Fig. 8) to store positions of matrix nonzero elements. The register \( |a_d\rangle \) is responsible for the location of sub-blocks such as \( M_d \) or \( M_{db} \) in Eq. 15. Given a variable index \( |d\rangle \) (Eq. 20), the register \( |a_d\rangle \) stores the absolute column indices of all sub-blocks that contain nonzero elements. The maximum number of sub-blocks in a row (including zero sub-blocks) coincides with the number of variables in our plasma system. Thus, the size of \( |a_d\rangle \) is the same as that of the register \( |d\rangle \).

Then, we describe the location of nonzero elements within each sub-block. The ancilla register \( |a_j\rangle \) is introduced for this purpose. To decrease the total number of ancillae, this register stores not an absolute column index but the relative position of a nonzero element with respect to the sub-block diagonal:

\[
\begin{align*}
|00\rangle_{a_j} & \rightarrow i_x = i_r, \\
|01\rangle_{a_j} & \rightarrow i_x = i_r - 1, \\
|10\rangle_{a_j} & \rightarrow i_x = i_r + 1,
\end{align*}
\]

(39) \hspace{1cm} (40) \hspace{1cm} (41)

where \( i_r \) is the row index, which is the index of a given point on \( x \)-grid of a variable. The row index is stored in the register \( |j\rangle \) (Eq. 20). The integer \( i_x \) is the column index of the nonzero value within the sub-block \( |a_d\rangle \). For instance, \( |00\rangle_{a_j} \rightarrow i_x = i_r \) means that a nonzero element lies on the local diagonal of the sub-block. One should note that the size of the register \( |a_j\rangle \) does not depend on \( N_x \), which is not the case in the standard technique (Appendix B). This feature allows us to increase \( N_x \) without increasing the number of ancillae. The size of \( |a_j\rangle \) increases, however, with the discretization order and it may also depend on the boundary conditions.

Finally, two more single-qubit ancilla registers \( |a_1\rangle \) and \( |a_2\rangle \) are introduced. The rotation gates \( R_j(\theta_j) \) act on these qubits to store the nonzero elements of the Hamiltonian (Eq. 35). A thinner space grid has a smaller difference between neighboring Hamiltonian elements, and as a result, requires that the rotation angle \( \theta \) be calculated with a higher precision.

D. Block-encoding operator

Now, when we have introduced all necessary ancillae and know how they store the structure of \( \mathcal{H} \), we can represent the block-encoding as a product of several operators:

\[
U_H = O_F^{-1} O_{\sqrt{T}}^{-1} O_M O_{S,a1} O_{\sqrt{T},a1} O_F,
\]

(42)

where every operator is responsible for a particular part of the block-encoding procedure. The oracle \( O_F \) (Table I) and \( O_{\sqrt{T}} \) (Eq. 11) and Fig. 10 defines the location of nonzero elements for a given variable index stored in \( |d\rangle \) and for a local row index saved in \( |j\rangle \). This oracle writes the column indices of sub-blocks with nonzero elements into the register \( |a_d\rangle \) and local relative positions of these nonzero elements into the register \( |a_j\rangle \). To construct the quantum circuit of the oracle \( O_F \), we consider it as a sequence of sub-circuits for different variables (Fig. 8). Every sub-circuit corresponds to one block from Table I. If one assumes that multicontrolled gates are physically realizable, then the depth of the \( O_F \) circuit does not change with the system size \( N_x \). However, whether it will be possible to efficiently connect non-neighboring qubits in future quantum computers is yet to be seen.

According to Eq. 15, several rows of the Hamiltonian have only zero elements. In this case, to simplify the construction of the oracle \( O_F \), we output the column index equal to the row index, as one can see, for instance, for the input \( |101\rangle_{a_j} \rightarrow |0\rangle_{a_d} \) (in the block \( Q \)) in Table I. The presence of these diagonal elements does not affect the action of the oracles \( O_{\sqrt{T}} \) and \( O_S \).

The oracle \( O_{\sqrt{T}} \) (Table II, Fig. 11) reads the row and column indices and provides the square root of the absolute value of the corresponding nonzero element. It acts on the ancilla \( |a_1\rangle \) or ancilla \( |a_2\rangle \). The oracle \( O_{S,a1} \) (Table III, Fig. 12) describes whether an element is imaginary or real, as well as whether it is positive or negative.

To create the \( O_{\sqrt{T}} \) quantum circuit, we use gates that perform rotations \( R_j(\theta_j) \) conditioned on the register \( j \). Every combination of qubits in \( |j\rangle \) corresponds to a particular angle \( \theta_j \) expressing the space dependence of a given field on \( x \). For instance, to obtain the profile of the background magnetic field, one can use angles \( \theta_j = 2 \arccos \sqrt{B_0/j} \). For convenience, we denote the corresponding conditional gate as \( R(B) \) in our quantum circuit (Fig. 11).

The conditional-rotation gates must be expressed via arithmetic functions with a set of additional qubits to store the angles \( \theta_j \). In our work, every conditional gate is coded as a multi-qubit gate with inner sub-blocks as in Eq. 24 on the main diagonal. Every sub-block corresponds to one \( \theta_j \) for a particular \( j \). The circuit depth of \( O_{\sqrt{T}} \) may strongly depend on how efficient the implementation of the conditional gates is for given profiles and how they depend on the system size \( N_x \). However, the general tendency is that the depth of quantum arithmetic circuits scales as \( \mathcal{O}(\text{poly}(n_1)) \) (Refs. 24 and 47).

In the standard state-preparation algorithm, ancilla registers store the absolute column indices (Eqs. 14, 15). As a result, the index exchange between the ancilla and input registers can be implemented by a simple swap operator. In our case, the register \( |a_j\rangle \) works with relative indices. Because of that, we need to implement the mapping between the absolute and relative indices during the index exchange. This is provided by the oracle \( O_{\sqrt{T}} \) (Fig. 13) by using a subtractor and an adder by 1, which are depicted in Fig. 14. After the application of the oracle \( O_{\sqrt{T}} \), the register \( |j\rangle \) encodes the absolute column indices of nonzero elements within a sub-block, and the register \( |d\rangle \) contains sub-block column indices. On the other hand, the ancilla register \( |a_d\rangle \) encodes now a variable
index, while the register \( |a_i\rangle \) remains unchanged. The depth of the circuit \( O_M \) is proportional to \( \log_2(N_t) \) because of the adder and subtractor.

To sum up, taking the row index as an input in \( |d\rangle |j\rangle \) and the ancillae initialized in the zero state, the resulting oracle \( U_H \) outputs column indices encoded in \( |d\rangle |j\rangle \) as states with amplitudes equal to the corresponding Hamiltonian elements when all ancillae registers are returned in the zero state (Eq. 26).

VI. SIMULATION RESULTS

A. Comparison of classical and quantum simulations

We implement QHS using the circuit described above on a classical emulator of a quantum computer using QuEST toolkit\(^1\). Unlike in an actual quantum simulation, this gives us access to the whole output space at all moments of time. The results presented below are directly extracted from \( \psi \) (Eq. 20) without performing, or emulating, quantum measurements. Then, we compare our results with those of classical simulations, which have been obtained by directly solving Eqs. 8 using the central finite difference scheme for both space and time. As a reminder, the normalized background profiles are shown in Fig. 10. The size of the system is \( n_0[\text{cm}] = 20 \) with \( N_s = 1024 (n_s = 10) \) spatial points. The simulation time is \( t/\omega_p^{-1} = 300.5 \), which is split into \( n_t = 1200 \) time steps with duration \( \tau/\omega_p^{-1} = 0.2504 \). The corresponding Courant number is 0.76. The maximum background density is \( n_0[\text{cm}^{-3}] = 2 \times 10^{13} \), while the peak of the fictitious density profile is \( n_0, f[\text{cm}^{-3}] = 10^{12} \). The background magnetic field at the center is \( B_0[G] = 10^3 \), and the auxiliary magnetic field at \( s = 0.4 \) has \( B_0, f[G] = 7 \times 10^3 \). The source-field coupling coefficient is \( \beta = 0.1 \), and \( Q \) oscillates with the frequency \( \omega_p/\omega_p = 0.38 \). The corresponding wavenumber is \( k_r n_0 = 64.56 \).

With these plasma parameters, the normalization of the Hamiltonian (Eq. 37) becomes \( \beta_H = 0.102 \). Therefore, the QSP time step is \( \tau_{qsp} = \tau/\beta_H = 2.455 \) with the resulting time interval \( t_{qsp} = n_t \tau_{qsp} = 2946 \) to simulate. The QSP error is \( \epsilon_{qsp} = 10^{-6} \). As seen from Fig. 15a such \( \epsilon_{qsp} \) corresponds to \( \sim 10^{-4} \) error in the energy conservation. For this QSP error, the number of QSP angles is equal to 25 for the time interval \( \tau_{qsp} \).

According to Ref. 35, the query complexity (number of copies of the oracle \( U_H \)) of the QSP circuit is \( \mathcal{O}(t_{qsp} + \log_2(1/\epsilon_{qsp})) \). The asymptotic dependence is confirmed by our direct computation (Figs. 17a and 17b). There, the total number of queries is calculated as twice the number of the QSP phases, because each phase corresponds to two calls to the block-encoding oracle \( U_H \) (Figs. 16c and 16d). The number of the phases is found using the code from Ref. 45. We should remark here that in our particular case, where the whole QSP circuit is split into \( n_t \) subcircuits, the scaling is \( \mathcal{O}(n_t \tau_{qsp} + n_t \log_2(1/\epsilon_{qsp})) \).

We compare the time evolution of the energy components (Eqs. 11b and 11d) integrated in space. Figure 15b shows that both the classical and quantum simulations produce the same time histories of various energy components (Eqs. 11b and 11d). When the wave reaches the HCR pair (at \( t \approx 100 \), according to Fig. 26), the field energy converts partly into the kinetic plasma energy. Figures 16b and 16d show that the kinetic (field) energy has a similar spatial distribution in both the simulations, and the X wave accumulates in the UH resonance of the HCR pair. The wave also passes the LCR pair practically without interaction and deposits its energy at the UHR, as anticipated.

B. Oracle scaling

Assuming that multi-qubit controlled gates are realizable, the depths of the oracles \( O_F \) (Fig. 8) and \( O_S \) (Fig. 12) are independent of \( n_s = \log_2 N_s \). However, these depths can change if the discretization order is increased or if the source \( Q \) is distributed over multiple grid points. The depth of the oracle \( O_M \) increases linearly with \( n_t \) due to the adder and subtractor (Fig. 13).

The oracle \( U_H \) (Fig. 11) depends on \( n_t \) due to the conditional-rotation gates. Similar gates are considered in Ref. 40 (see supplemental materials there), where they are called multiplexed unitaries. These gates can be implemented via arithmetic functions that usually scale as \( \epsilon(n_s) \). For instance, as shown in Ref. 23, the depth of a general adder with one of the summands predefined scales as \( \epsilon(n_s) + 2n_s \). Generally, there is a trade between the number of ancillae used to store intermediate data and the depth of the resulting circuit. In our case, a conditional-rotation gate must implement a smooth function that depends on the space variable \( x \) encoded inside the register \( |j\rangle \). As explained in Ref. 47, a polynomial of order \( D \) can be evaluated by using the Horner scheme. A given polynomial \( y_D \) with coefficients \( a_i \) can be obtained by \( D \) subsequent iterations:

\[
\begin{align*}
y_1 &= a_D x + a_{D-1}, \\
y_2 &= y_1 x + a_{D-2}, \\
&\vdots \\
y_D &= y_{D-1} x + a_0.
\end{align*}
\]

The total number of gates necessary to implement the whole polynomial scales as \( \mathcal{O}(Dn_s^2) \).

C. Quantum measurements

In an actual quantum simulation, the output vector cannot be accessed directly. One can measure only the expectation value of a given operator on the output state. How to do this for potentially practical RF simulations is a problem separate from QSP that we discuss here, so it is left to future work. However, here is how at least one of the quantities of interest can be measured, namely, the wave energy within a given spatial volume.

The field components are encoded into the quantum state \( \psi \) as shown in Eq. 20. To compute the electric energy, one
FIG. 15: (a): space–integrated total system energy for different values of the QSP error $\varepsilon_{\text{qsp}}$. (b): comparison of the corresponding energy components (Eqs. [11b][11d]) in classical and quantum (with $\varepsilon_{\text{qsp}} = 10^{-6}$) simulations. Here, CL stands for classical simulations and QC stands for emulated quantum simulations.

FIG. 16: Comparison of classical (solid blue curves) and emulated quantum (red dashed curves) simulations. Real (a) and imaginary (c) components of the source $Q$. (b): kinetic energy $W_v$ (Eq. [11b]) at time $t_f$ as a function of $s$. (d): field energy $W_{eb}$ (Eq. [11c]) at time $t_f$ as a function of $s$. The vertical lines show the HCR pair (blue), the LCR pair (green), the UHR (red).

needs to sum the squares of the electric components in a desired spatial interval. As an example, a case with $n_x = 4$ is considered in Fig. 18. The QSP algorithm outputs $\psi_{\text{out}}$ that encodes plasma variables if the QSP ancillae $|a_{\text{qpp}}\rangle$ are all in the zero state. In this example, we sum up $W_e = E_x^2 + E_y^2$ over the spatial points with indices $j = [2, 3, 4, 5]$. The first controlled Pauli $X$ gate entangles the superposition of the amplitudes of both $E_x(j \in [2, 3])$ and $E_y(j \in [2, 3])$ with the state $|1\rangle_{m_0}$. The second $X$ gate entangles $E_x(j \in [4, 5])$ and $E_y(j \in [4, 5])$ with the state $|1\rangle_{c_1}$. The last controlled $X$ gate finds the conjunction (logical AND) of the above states, and as a result, stores $\sum_{j=2}^{5} W_e(x_j)$ as the probability of the state $|1\rangle_{m}$.

If the qubit $m$ has the state $|1\rangle$ with amplitude $a$, then it takes at least $O(|a|^{-2})$ repetitions of the whole QSP operator before the direct measurement returns $|m\rangle = |1\rangle$. How-
ever, there is a quantum Amplitude–Amplification (AA) algorithm that requires only $\Theta(|a|^{-1})$ iterations of the operator to measure the desired state with probability at least $\max(1 - |a|^2, |\bar{a}|^2)$. This method is the basis of so-called Amplitude–Estimation (AE) techniques that allow to find the state probability with a predefined absolute error $\delta$. The AA is based on the Grover-like rotation $R_{AA}$ in the quantum space spanned by a state of interest $|G\rangle (\langle m | = |1\rangle$ in our case) and a “garbage” state $|\bar{B}\rangle (\langle m | = |0\rangle)$ in such a way that the amplitude of the rotated state $|G\rangle$ becomes a sinusoidal function of the number of applications $n_{AA}$ of $R_{AA}$: $\sin((2n_{AA} + 1)\theta)$, where $\sin^2 \theta = |a|^2$ and $\theta \in [0, \pi/2]$. In our case, every rotation includes the whole QSP operator and its inverse.

The operator $R_{AA}$ has eigenvalues $\exp(\pm i2\theta)$. Therefore, one can calculate the probability $|a|^2$ from the estimation of the angle $\theta$ by constructing a superposition of states rotated with several $n_{AA}$, and by applying a subsequent Quantum Fourier Transform (QFT). That is the essence of the conventional AE algorithm described in Ref. 51. This method estimates the probability $|a|^2$ with an absolute error $\delta$ by applying $M = \Theta(1/\delta)$ queries (in our case, calls to the QSP) and by using $\log_2(M)$ ancilla qubits, while classically one would need $\Theta(1/\delta^2)$ queries due to the central limit theorem. In our case, the error $\delta$ corresponds to the absolute error of the measured space-integrated energy.

There are also state-of-the-art AE techniques with a similar asymptotic scaling but smaller number of ancillae and controlled gates. For instance, the algorithm proposed in Ref. 51 also uses a set of AA operators with a various number of rotations $R_{AA}$. However, instead of the QFT, it performs statistical post-processing of measurement results by implementing the maximum likelihood estimation of $|a|^2$.

The numerical implementation of the quantum measurements for classical RF systems is left to future work.

VII. DISCUSSION AND CONCLUSIONS

We have proposed how to apply the Quantum Signal Processing (QSP) technique to simulating cold-plasma waves and explicitly developed a quantum algorithm for modeling one-dimensional X-wave propagation in cold electron
plasma. We have demonstrated how to construct an oracle to encode the wave Hamiltonian into a quantum circuit. The number of the ancillae in this oracle does not depend on the spatial resolution, so one can use a resolution higher than in the case with the standard state-preparation method. Since oracle complexity scales as \( \mathcal{O}((\log_2 N_i)^n) \), the QSP simulation can provide a near-exponential speedup in comparison to classical simulations, which scale as \( \mathcal{O}(N_i) \). This approach can be particularly helpful in simulations with high spatial resolution, which is advantageous, for example, for modeling the wave dynamics near resonances.

Our quantum simulations have been performed on a digital emulator of the quantum circuit and have shown agreement with the corresponding classical modeling. For emulation, we used the QuEST numerical toolkit\(^{42} \) that operates with a whole \( 2^{n_q} \) state vector in a circuit with \( n_q \) qubits. For our one-dimensional QSP simulations, the emulator have shown efficient parallelization on GPUs. However, one might need a more advanced emulator for higher-dimensional simulations. One of the promising approaches in this regard is the model proposed in Ref.\(^{54} \). It uses the fact that a quantum state is mostly sparse, i.e., many elements of the state vector are zero. If only its nonzero elements were stored (e.g. in a form of a hash table), one could significantly reduce memory usage and the simulation runtime. Moreover, one could model oracles with conditional rotations implemented via actual arithmetic functions with a big number of ancillae, since the ancillae act only locally and are zeroed otherwise thus being removed from the memory. Yet, this model needs to be extended with a GPU parallelization and the corresponding implementation of dynamic hash tables\(^{55} \).

Based on our results, we can also assess the overall utility of the QSP technique in application to the linear problems. Being a universal algorithm with a clear hierarchical structure, the QSP can be easily coded as a set of subsequent subroutines where only the block-encoding module needs to be modified for different plasma problems. The QSP provides an optimal dependence of the query complexity (the number of calls to the oracle) on the simulation time and on the error tolerance (as was pointed out in Ref.\(^{33} \)). Also, the QSP requires only two ancillae in addition to the qubits used by the oracle. This reduces the circuit width. That said, the QSP circuits and our oracle in particular have many multi-controlled operators, where one gate is controlled by several controlled operators, where one gate is controlled by several

\[ t_{\text{qsp}} = \mathcal{O}\left( \frac{\log_2 N_i}{\delta} \left( t_{\text{qsp}} + \log_2(1/\delta) \right) \right), \] (43)

where we take the absolute error \( \delta \), which appears from measurements, equal to the QSP approximation error \( \varepsilon_{\text{qsp}} \). The numerical coefficient in the scaling is determined by the specific implementation of the conditional rotations and of the measurement algorithm. Studying these subjects is left to future work.

ACKNOWLEDGMENTS

The work was supported by the U.S. DOE through Contract No. DE-AC02-09CH11466. The authors also thank Alexander Engel for valuable discussions.

Appendix A: Basic notation and terminology of quantum computing

The elementary memory cell of a quantum computer is a qubit, whose state can be characterized by a two-dimensional vector. In the computational basis, the basis vectors are

\[ |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \] (A1)

A qubit can also be in a superposition state

\[ \alpha |0\rangle + \beta |1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \] (A2)

where \( \alpha \) and \( \beta \) are the complex amplitudes of the states \( |0\rangle \) and \( |1\rangle \), respectively, such that \( |\alpha|^2 + |\beta|^2 = 1 \). A quantum computer typically operates with \( n > 1 \) qubits, so its quantum state is characterized by a tensor product of the qubit states

\[ |k_{n-1}\rangle \otimes |k_{n-2}\rangle \otimes \ldots |k_1\rangle \otimes |k_0\rangle, \] (A3)

where \( k_i = 0, 1 \) in the computational basis. For convenience, one can organize qubits in quantum registers. A combination of \( n_r \) qubits in a register \( |r\rangle \) can be written either as a bit-string \( |k_{n_r-1}k_{n_r-2}...k_1k_0\rangle \), or as an integer \( |k\rangle \), where

\[ k = \sum_{i=0}^{n_r-1} k_i 2^i. \] (A4)

Several qubits can also form superposition states:

\[ |\psi\rangle = \sum_k \alpha_k |k\rangle, \] (A5)

where \( |\alpha_k|^2 \) is the probability amplitude of the state \( |k\rangle \), and \( \sum_k |\alpha_k|^2 = 1 \).

To modify a quantum state of a circuit, one applies gates, which are unitary operators acting on \( n_g \geq 1 \) qubits and can be represented by matrices of size \( 2^{n_r} \times 2^{n_r} \). The gates that are used in this work are mainly the X Pauli, Hadamard \( H \),
Phase $P$, and SWAP operators. The $X$ gate inverts the qubit value:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

$$X(\alpha |0\rangle + \beta |1\rangle) = \alpha |0\rangle + \beta |1\rangle.$$  \hfill (A6)

The Hadamard gate creates a superposition of states:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

$$H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),$$

$$H|1\rangle = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$  \hfill (A8)

The Phase gate modifies the phase of the state $|1\rangle$, and keeps the zero state $|0\rangle$ unchanged (Fig. 19):

$$P(\theta)(\alpha |0\rangle + \beta |1\rangle) = \alpha |0\rangle + e^{i\theta} \beta |1\rangle.$$  \hfill (A10)

The SWAP gate exchanges the qubit values (Fig. 19):

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\text{SWAP} |k\rangle |j\rangle = |j\rangle |k\rangle.$$  \hfill (A13)

A gate can be controlled by one or several other qubits (control nodes) as shown in Fig. 20. For instance, a 1-controlled $H$ gate can be written as

$$CH = |0\rangle \langle 0| \otimes I_t + |1\rangle \langle 1| \otimes H_t,$$  \hfill (A15)

where the Hadamard gate acts on the target qubit $t$ only if the control qubit $c$ is in the state $|1\rangle$.

Also note that a gate can be controlled by a whole register. In this case, the gate is executed only if each qubit in the register is in the $|0\rangle$ or $|1\rangle$ state, as illustrated in Fig. 21.

Appendix B: Block-encoding by state-preparation method

The block-encoding $U_H$ of a Hermitian operator $\mathcal{H}$ can be constructed by applying the state-preparation algorithm as a product of two unitary matrices:

$$U_H = T_2^2 T_1,$$  \hfill (B1)

FIG. 19: Top: SWAP gate of two qubits $|k\rangle$ and $|j\rangle$. Bottom: phase gates with an angle $\theta$. If the dot is hollow, then the phase gate acts on the zero state.

FIG. 20: CNOT gate, or controlled $X$ gate. The 1-controlled gate takes action only if the control qubit is in the state $|1\rangle$: $|\text{out}\rangle = \alpha |0\rangle_c |0\rangle_t + \beta |1\rangle_c |1\rangle_t$. A 0-controlled (in this case, the dot is hollow) gate takes action only if the control node is in the state $|0\rangle$. Here, the black dot is called 1-control node. If the dot is hollow, it is called 0-control node.

FIG. 21: Circuit containing a gate $G$ controlled by the register $|r\rangle$. The register has $n$ qubits. The gate $G$ modifies the target qubit $|t\rangle$ only if all $n$ qubits of the register $|r\rangle$ are in the state $|1\rangle$.

where

$$T_1 = \sum_j |\psi_j\rangle \langle j|_s,$$

$$T_2 = \sum_k |\chi_k\rangle \langle k|_s.$$  \hfill (B3)

Each $T_i$ involves the sum of states $|\psi\rangle$ and $|\chi\rangle$ defined as

$$|\psi_j\rangle = \sum_{p \in F_j} \frac{|p\rangle_{a_1}}{\sqrt{2}} |\mathcal{H}_{p,j}\rangle |0\rangle_{a_2} |j\rangle,$$

$$|\chi_k\rangle = \sum_{p \in F_k} \frac{|k\rangle_{a_1}}{\sqrt{2}} |\mathcal{H}_{k,p}\rangle |0\rangle_{a_2} |p\rangle,$$

where

$$|\mathcal{H}_{k,p}\rangle = \sqrt{2} \left( |\mathcal{H}_k\rangle |0\rangle_{a_2} + \frac{1}{\sqrt{M}} |\mathcal{H}_p\rangle |1\rangle_{a_2} \right).$$  \hfill (B6)

Using the oracle $O_T$ (Eq. 22), one finds a set of column indices $F_j$ of all nonzero elements on the row $j$. The operator $T_1$ reads the row index $j$ and saves the corresponding column indices to the ancilla register $|a_2\rangle$. After that, it rotates the ancilla qubit $|a_1\rangle$, so the necessary matrix element (its square root) becomes the amplitude of the zero state $|0\rangle_{a_2}$ (Eq. 26). This is implemented by the oracle $O_{T_1}$ in Fig. 42. The operator $T_2$ rotates the ancilla qubit $|a_2\rangle$ in a similar way by taking a row index from the ancilla register $|a_2\rangle$ and column indices from the register $|s\rangle$. Since the encoding of the row and column indices in the registers $a_1$ and $s$ in $|\chi_k\rangle$ is swapped in comparison with the state $|\psi_j\rangle$, we introduce the oracle $O_M$, which performs the corresponding index swapping in Eq. 42.

1. N. J. Fisch, “Theory of current drive in plasmas,” Reviews of Modern Physics 59, 175–234 (1987).
2. R. I. Pinsker, “Introduction to wave heating and current drive in magnetized plasmas,” Physics of Plasmas 8, 1219–1228 (2001).
3. R. Prater, “Heating and current drive by electron cyclotron waves,” Physics of Plasmas 11, 2349–2376 (2004).
N. J. Fisch, “Confining a tokamak plasma with RF-driven currents,” Physical Review Letters 121, 225001 (2018).

P. T. Bonoli and M. Porkolab, “Radiofrequency current generation by F. Gaitan, “Finding flows of a Navier–Stokes fluid through quantum com-

I. Y. Dodin and E. A. Starsev, “On applications of quantum computing to plasma simulations,” Physics of Plasmas 28, 092101 (2021).

https://doi.org/10.1063/5.0056974

M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum In-

E. G. Rieffel and W. H. Polak, Quantum Computing: A Gentle Introduc-

Circuit depth explanation,” https://qiskit.org/textbook/ch-

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 Information Processing 19, 170 (2020).

S. Sinha and P. Russer, “Quantum computing algorithms for electromagnetic field simulation,” Quantum Information Processing 9, 385–404 (2010).

A. Scherer, B. Valiron, S.-C. Mau, S. Alexander, E. van den Berg, and T. E. Chapuran, “Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2D target,” Quantum Information Processing 16, 60 (2017).

P. Fillion-Gourdeau and E. Lorin, “Simple digital quantum algorithm for symmetric first-order linear hyperbolic systems,” Numerical Algorithms 82, 1099–1105 (2019).

F. Gaitan, “Finding flows of a Navier–Stokes fluid through quantum computing,” npj Quantum Information 6, 61 (2020).

S. Badinski, “Quantum algorithm for the collisionless Boltzmann equation,” Journal of Computational Physics 409, 109347 (2020).

Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, “Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetic,” in Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing STOC 2019 (Association for Computing Machinery, New York, NY, USA, 2019) p. 193–204.

J. Haah, “Product decomposition of periodic functions in quantum signal processing,” Physical Review Letters 121, 020501 (2018).

R. Chao, D. Ding, A. Gilyen, C. Huang, and M. Szegedy, “Finding angles for quantum signal processing with machine precision,” (2020), arXiv:2003.02831 [quant-ph].

Y. Dong, X. Meng, K. B. Whaley, and L. Lin, “Effi-

Scientific Reports 9, 10736 (2019).

T. H. Stix, Waves in Plasmas (AIP Press, 1992).

G. H. Low, T. J. Yoder, and I. L. Chuang, “Methodology of resonant equiangular composite quantum gates,” Physical Review Research 3, 033055 (2021).

A. Engle, G. Smith, and S. E. Parker, “Quantum algorithm for the Vlasov equation,” Physical Review A 100, 062315 (2019).

T. Jones, A. Brown, I. Bush, and S. C. Benjamin, “QuEEST and high performance simulation of quantum computers,” Physical Review A 103, 042419 (2021).

V. von Burg, G. H. Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer, “Quantum computing enhanced computational catalysis,” Physical Review Research 3, 033055 (2021).

Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, “Quantum algorithm and circuit design solving the Poisson equation,” New Journal of Physics 15, 013021 (2013).

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 Information Processing 19, 170 (2020).

S. Sinha and P. Russer, “Quantum computing algorithm for electromagnetic field simulation,” Quantum Information Processing 9, 385–404 (2010).

A. Scherer, B. Valiron, S.-C. Mau, S. Alexander, E. van den Berg, and T. E. Chapuran, “Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2D target,” Quantum Information Processing 16, 60 (2017).

P. Fillion-Gourdeau and E. Lorin, “Simple digital quantum algorithm for symmetric first-order linear hyperbolic systems,” Numerical Algorithms 82, 1099–1105 (2019).

F. Gaitan, “Finding flows of a Navier–Stokes fluid through quantum computing,” npj Quantum Information 6, 61 (2020).

S. Badinski, “Quantum algorithm for the advection-diffusion equation simulated with the lattice Boltzmann method,” Quantum Information Processing 20, 57 (2021).

J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, “A grand unification of quantum algorithms,” (2021), arXiv:2105.02859 [quant-ph].

G. H. Low and I. L. Chuang, “Optimal Hamiltonian simulation by quantum signal processing,” Physical Review Letters 118, 010501 (2017).

G. H. Low and I. L. Chuang, “Hamiltonian simulation by qubitization,” Quantum 3, 163 (2019).

A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, “Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetic,” in Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing STOC 2019 (Association for Computing Machinery, New York, NY, USA, 2019) p. 193–204.

J. Haah, “Product decomposition of periodic functions in quantum signal processing,” Physical Review Letters 121, 020501 (2018).

R. Chao, D. Ding, A. Gilyen, C. Huang, and M. Szegedy, “Finding angles for quantum signal processing with machine precision,” (2020), arXiv:2003.02831 [quant-ph].

Y. Dong, X. Meng, K. B. Whaley, and L. Lin, “Efficient phase-factor evaluation in quantum signal processing,” Physical Review A 103, 042419 (2021).

V. von Burg, G. H. Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer, “Quantum computing enhanced computational catalysis,” Physical Review Research 3, 033055 (2021).

A. Engle, G. Smith, and S. E. Parker, “Quantum algorithm for the Vlasov equation,” Physical Review A 100, 062315 (2019).

T. Jones, A. Brown, I. Bush, and S. C. Benjamin, “QuEEST and high performance simulation of quantum computers,” Physical Review A 103, 042419 (2021).

V. von Burg, G. H. Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer, “Quantum computing enhanced computational catalysis,” Physical Review Research 3, 033055 (2021).

A. Engle, G. Smith, and S. E. Parker, “Quantum algorithm for the Vlasov equation,” Physical Review A 100, 062315 (2019).

G. H. Low, T. J. Yoder, and I. L. Chuang, “Methodology of resonant equiangular composite quantum gates,”
17

45. Computation of angles for quantum signal processing in F#, https://github.com/microsoft/Quantum-NC/tree/main/src/simulation/qsp (2020), accessed: 04-2021.

46. Efficient phase-factor evaluation in quantum signal processing: code, https://github.com/qsppack/QSPPACK (2021), accessed: 09-2021.

47. T. Häner, M. Roetteler, and K. M. Svore, “Optimizing quantum circuits for arithmetic,” (2018), arXiv:1805.12445 [quant-ph]

48. L. K. Grover, “Quantum mechanics helps in searching for a needle in a haystack,” Physical Review Letters 79, 325–328 (1997).

49. L. K. Grover, “Quantum computers can search rapidly by using almost any transformation,” Physical Review Letters 80, 4329–4332 (1998).

50. G. Brassard, P. Høyer, M. Mosca, and A. Tapp, “Quantum amplitude amplification and estimation,” Quantum Computation and Information 305, 53–74 (2002).

51. Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, “Amplitude estimation without phase estimation,” Quantum Information Processing 19 (2020), 10.1007/s11128-019-2565-2

52. S. Uno, Y. Suzuki, K. Hisanaga, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, “Modified Grover operator for quantum amplitude estimation,” New Journal of Physics 23, 083031 (2021).

53. D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, “Iterative quantum amplitude estimation,” npj Quantum Information 7 (2021), 10.1038/s41534-021-00379-1

54. S. Jaques and T. Häner, “Leveraging state sparsity for more efficient quantum simulations,” (2021), arXiv:2105.01533 [quant-ph]

55. Y. Li, Q. Zhu, Z. Lyu, Z. Huang, and J. Sun, “DyCuckoo: Dynamic hash tables on GPUs,” in 2021 IEEE 37th International Conference on Data Engineering (ICDE) (IEEE Computer Society, Los Alamitos, CA, USA, 2021) pp. 744–755.