Quantum simulation of wave optics in weakly inhomogeneous media using block encoding

Siavash Davani [1,2,*] Martin Gärttner [3,†] and Falk Eilenberger [4,1,2,‡]

[1] *Institute of Applied Physics, Abbe Center of Photonics, Friedrich Schiller University Jena, 07745 Jena, Germany*

[2] *Max Planck School of Photonics, 07745 Jena, Germany*

[3] *Institute of Condensed Matter Theory and Optics, Friedrich Schiller University Jena, 07743 Jena, Germany*

[4] *Fraunhofer Institute for Applied Optics and Precision Engineering IOF, 07745 Jena, Germany*

(Received 1 June 2026; accepted 1 October 2026; published 2 November 2026)

We propose a quantum algorithm that simulates the propagation of a light field through a weakly inhomogeneous medium. In the paraxial approximation, the wave equation in an inhomogeneous material takes the form of the Schrödinger equation with a time-dependent Hamiltonian. This reduction is used to simulate wave optical dynamics on a quantum computer. Beam propagator operators for a short propagation distance are constructed using an efficient and flexible block encoding that enables the simulation of various optical setups. The algorithm is showcased by simulating the propagation of a one-dimensional Gaussian beam through a lens of finite thickness, and the resulting spherical aberration is demonstrated.

DOI: 10.1103/snzq-fp39

I. INTRODUCTION

In recent years, quantum computers have received significant attention because of their potential to advance various fields and technologies including simulation [1,2], cryptography and communications [3], machine learning [4], and many more [5–9]. In particular, the simulation of physical systems was among the very first proposed applications of quantum computers [10] and has remained central to quantum computing research ever since [11,12].

Quantum algorithms have been introduced for solving linear systems [13] and differential equations [14]. In some cases, these algorithms can exponentially improve the performance of simulations compared to their classical counterparts [5], once a fault-tolerant quantum computer is available [15–18]. On the algorithmic side, much work has been done on fundamental primitives such as linear combination of unitaries [19], qubitization [20], quantum signal processing, and quantum singular value transformation [21]. These efforts have converged to a unified formalism [22], gathering many of the known quantum algorithms under one common framework using the notion of block encoding [20]. Quantum signal processing using block encoding has been shown to achieve optimal computational complexity in the time-independent Hamiltonian simulation problem [20,21].

Nevertheless, a robust quantum algorithm for simulating wave optics has not been proposed so far. The diverse applications of wave propagation algorithms range from designing optical waveguides, lasers, and integrated circuits to biomedical optics and imaging [23–27]. In modern optical engineering, there is an increasing need for more accurate simulations of beam propagation in complex structures where the precise control of light is essential. These simulations require significant memory and processing power; improving their performance is therefore highly desired [28–30]. Quantum computers are potential candidates to reduce these computational costs. Quantum registers store a field discretized at N spatial points in only $\log_2 N$ qubits. This efficiency in storage can potentially lead to exponential improvements in the performance as well, since quantum algorithms operate on a logarithmically smaller number of memory units compared to classical algorithms. Finding such efficient quantum algorithms for domain-specific applications is a key objective of quantum simulation research.

Previous works presented quantum algorithms for calculating the electromagnetic scattering cross section [31], performing quantum beam propagation in homogeneous media [32], and simulating quantum optical systems [33]. In this work, we simulate wave optics on quantum computers by solving the Helmholtz equation for a scalar field in a medium with a slowly varying refractive index. The simulator is implemented using an efficient and flexible block-encoding structure that can be tuned to simulate the effects of various optical elements on an incident light beam. The implementation uses an ancillary quantum register to construct block-encoded phase shifters acting on a target register. Using a split-step technique [34], these phase shifters are used along with quantum Fourier transform (QFT) subroutines [35] to simulate wave propagation in the paraxial approximation. The ancillary register only needs $\log_2 N$ qubits, and the gate complexity of the simulator up to an error ϵ defined using the spectral norm is quadratic in the simulation time $\mathcal{O}(T^2/\epsilon)$. Therefore, this algorithm establishes a significant

*Contact author: siavash.davani@uni-jena.de

†Contact author: martin.gaerttner@uni-jena.de

‡Contact author: falk.eilenberger@uni-jena.de

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

improvement in computational capacity for high-precision applications.

Section II discusses the reduction of the Helmholtz equation to a time-dependent Hamiltonian simulation problem. In Sec. III, the block encoding of phase propagators is introduced and its relation to Hamiltonian simulation is discussed. Finally, Sec. IV presents the simulation of a Gaussian beam incident on a plano-convex lens using the phase propagators.

II. PRELIMINARIES

A. Paraxial wave optics in weakly inhomogeneous media as a Hamiltonian simulation problem

The scalar wave equation for a field $u(\mathbf{r}, \omega)$ propagating in a weakly inhomogeneous medium is

$$\nabla^2 u(\mathbf{r}, \omega) + k^2(\mathbf{r}, \omega) u(\mathbf{r}, \omega) = 0, \tag{1}$$

known as the Helmholtz equation [36], where

$$k^2(\mathbf{r}, \omega) = \frac{\omega^2}{c^2} \varepsilon_r(\mathbf{r}, \omega) \tag{2}$$

is the squared wave number. Here, ω, ε_r, c, and ∇^2 are, respectively, the angular frequency of the monochromatic wave, the relative permittivity of the medium, the speed of light in vacuum, and the Laplacian operator. Equation (1) describes the propagation of a wave in an isotropic, dispersive, and inhomogeneous material as long as $\varepsilon_r(\mathbf{r}, \omega)$ varies slowly as a function of the spatial coordinate $\mathbf{r}$, that is,

$$\left| \frac{\nabla \varepsilon_r(\mathbf{r}, \omega)}{\varepsilon_r(\mathbf{r}, \omega)} \right| \ll \frac{1}{\lambda}, \tag{3}$$

where λ is the wavelength. In our discussion, we consider the case of a monochromatic wave and leave out the ω dependence of ε_r from now on. We also assume that the radiation is localized in the momentum space and apply the paraxial approximation. Without loss of generality, we choose the z coordinate axis to align with the optical axis. After these assumptions, the solution to Eq. (1) takes the form

$$u(\mathbf{r}) = v(\mathbf{r}) e^{i\tilde{k}z}, \tag{4}$$

where $\tilde{k} = \langle k(\mathbf{r}) \rangle$ is the wave number averaged over the volume of the simulation space. Inserting this ansatz into Eq. (1) leads to the following differential equation for $v(\mathbf{r})$:

$$i \frac{\partial}{\partial z} v(\mathbf{r}) + \frac{1}{2\tilde{k}} \nabla_\perp^2 v(\mathbf{r}) + \frac{k^2(\mathbf{r}) - \tilde{k}^2}{2\tilde{k}} v(\mathbf{r}) = 0, \tag{5}$$

where $\nabla_\perp^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the transverse Laplacian operator. Since we chose the z axis to match the direction of propagation, we can separate the transverse coordinates $\mathbf{r}_\perp = x\mathbf{e}_x + y\mathbf{e}_y$ and the propagation coordinate $z = ct$ to rewrite the differential equation as

$$i \frac{1}{c} \frac{\partial}{\partial t} v(t; \mathbf{r}_\perp) + \frac{1}{2\tilde{k}} \nabla_\perp^2 v(t; \mathbf{r}_\perp) + \frac{k^2(t; \mathbf{r}_\perp) - \tilde{k}^2}{2\tilde{k}} v(t; \mathbf{r}_\perp) = 0. \tag{6}$$

This equation is a first-order differential equation in time and resembles the form of the Schrödinger equation

$$i\hbar \frac{\partial}{\partial t} |v(t)\rangle = \hat{H}(t) |v(t)\rangle, \tag{7}$$

where

$$\langle \mathbf{r}_\perp | v(t) \rangle = v(t; \mathbf{r}_\perp), \tag{8}$$

and the Hamiltonian of the system is

$$\hat{H}(t) = \frac{c}{2\hbar\tilde{k}} (\hat{p}_x^2 + \hat{p}_y^2) - \frac{\hbar c}{2\tilde{k}} (k^2(t; \hat{x}, \hat{y}) - \tilde{k}^2) \tag{9}$$

$$= K(\hat{p}_x, \hat{p}_y) + V(t; \hat{x}, \hat{y}), \tag{10}$$

where K and V are the kinetic and potential energy terms. This shows that the problem of beam propagation in weakly inhomogeneous media reduces to a time-dependent Hamiltonian simulation problem and hence can be solved using a quantum computer. Note that the potential energy term in the Hamiltonian is time dependent (or equivalently z dependent in this case) because the propagating beam interacts with a varying refractive index during the propagation.

A simulation algorithm starts with a field distribution in the transverse plane $|v(0)\rangle$ at the initial time $t = 0$ and evolves the field after propagating for time T, or distance $z = cT$, in the direction of propagation to compute $|v(T)\rangle$. The time dependence of the Hamiltonian and the noncommuting kinetic and potential energy terms in Eq. (10) make this computation nontrivial. We use a split-step method [34] to perform the computation on a quantum computer by dividing the evolution time into a number of small time steps and executing the corresponding unitary steps sequentially.

Let us partition the total evolution time from $t = 0$ to $t = T$ into N_t equal intervals $\Delta t = T/N_t$ and define the time grid $t_j = j\Delta t$. Assuming that Δt is small and using the first-order Trotter-Suzuki formula [37], we can approximate the total time evolution operator (setting $\hbar = 1$) as a sequence of evolution operators for time Δt. For each time step, we use a midpoint approximation by choosing $t_{j+1/2}$ to evaluate the Hamiltonian in the time window $[t_j, t_{j+1}]$ that leads to

$$\hat{U}(T, 0) = \prod_{j=0}^{N_t - 1} e^{-i\hat{K}\Delta t} e^{-i\hat{V}(t_{j+1/2})\Delta t} + \mathcal{O}\left(\frac{T^2}{N_t}\right). \tag{11}$$

This means that, to execute the time evolution, we need to sequentially apply unitary propagation steps of the form $e^{-i\hat{K}\Delta t}$ and $e^{-i\hat{V}(t)\Delta t}$ to the state $|v\rangle$. The kinetic energy part $\hat{K}$ is not a function of time and is a simple quadratic function of the transverse frequency. Therefore, $e^{-i\hat{K}\Delta t}$ can be efficiently implemented using standard diagonal unitary synthesis and QFT subroutines [32]. However, the potential energy part $V(t; \hat{x}, \hat{y})$ is in general changing with propagation time and can be an arbitrary function of transverse coordinates. In Sec. III, we construct block-encoded operators that implement such unitary potential energy propagators and use them to realize $\hat{U}(T, 0)$. The block encoding is efficient for implementing a phase operator $\exp(if(\hat{x}))$ under two conditions: (1) the norm of the phase $\|f(x)\|_1$ should be small and (2) there should be an efficient state preparation unitary for the state $\phi(x)$ such that $f(x) \propto |\phi(x)|^2$. Condition (1) is naturally fulfilled in the wave propagation problem discussed here, because in order for the split-step approximation in Eq. (11) to be valid, the time step Δt needs to be small, leading to small local phases $f(x) = -V(t; x)\Delta t$ at each time step. Condition (2) remains as a requirement for efficiency, meaning that the

proposed simulator will be efficient if the geometry of the simulation problem leads to refractive index distribution profiles for which we can find efficient state preparation unitaries. Although state preparation is not efficient in general, there are large classes of states relevant to optics (e.g., smooth functions with bounded derivatives) that have efficient initializers [38–41]. Therefore, the block-encoding construction is suitable for implementing the evolution operator in Eq. (11). These conditions are discussed further in the following sections and the detailed analysis of the error scaling is included in Appendix C.

It is also worth mentioning that we have used the first-order Trotter formula in this discussion because of its simplicity. However, it is possible to use more advanced methods such as higher-order Trotterization [42] or Dyson series [43] to achieve a more favorable scaling of the time discretization error with Δt. Our main objective in this work, however, is to implement evolution operators for a small step size using block encoding, which then can also be used in other time discretization schemes.

B. Block encoding

Before presenting the main results, we briefly review the notion of block encoding. Block encoding is an intuitive and useful way to model nonunitary operations on quantum computers. The formal definition of block encoding is given in the following. The presentation follows [21].

Definition 1 (Block-encoding). A unitary operator U_A is called an (α, a, ϵ) block encoding of the (not necessarily unitary) operator A if

$$\|A - \alpha(\langle 0|^{\otimes a} \otimes I)U_A(|0\rangle^{\otimes a} \otimes I)\| \leqslant \epsilon. \tag{12}$$

As an intuitive example of the above definition, an exact block encoding of A ($\epsilon = 0$) takes the form

$$U_A = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}, \tag{13}$$

meaning that A occupies the top left block of U_A, and $*$ indicates that the exact values of the other blocks of U_A are not important as long as U_A is a unitary operator. If A were unitary and we knew how to construct it on a quantum computer, we could directly apply it to a register $|\psi\rangle$ to achieve the result

$$A|\psi\rangle. \tag{14}$$

However, since A can be nonunitary, $A|\psi\rangle$ is not necessarily normalized, and thus we can only hope for probabilistic access to it. The block-encoding U_A provides such probabilistic access. In this example, an a-qubit ancillary register is added to the primary register to form $|0\rangle^{\otimes a}|\psi\rangle$ and the block-encoding U_A acts on the combined register

$$U_A|0\rangle^{\otimes a}|\psi\rangle = |0\rangle^{\otimes a}\frac{A|\psi\rangle}{\alpha} + |\perp\rangle, \tag{15}$$

where $(\langle 0|^{\otimes a} \otimes I)|\perp\rangle = 0$. If we postselect on the a-qubit ancillary register being in the zero state, the primary register is projected onto a state proportional to $A|\psi\rangle$, meaning that the ancillary register flags the success case. The probability of success of the postselection is

$$P_{\text{success}} = \frac{1}{|\alpha|^2}\langle\psi|A^\dagger A|\psi\rangle. \tag{16}$$

Note that any unitary operator is trivially a (1,0,0) block encoding of itself; hence, using the block-encoding framework, we can extend the analysis of quantum algorithms beyond unitary operations by allowing for ancillary registers, probabilistic success, and flagging of the success case.

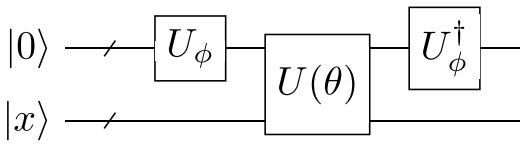


FIG. 1. Schematic circuit representation of the block-encoding unit for the phase protocol.

III. RESULTS

In order to implement potential energy propagators in the time evolution operator in Eq. (11), we first construct a block-encoding unit for a generic phase operator

$$e^{i\sum_x \Delta|\phi(x)|^2|x\rangle\langle x|} \tag{17}$$

for a small coefficient Δ and a normalized wavefunction $\phi(x)$. Then, we discuss under what conditions this block-encoding unit can be repeated to synthesize the phase propagator

$$e^{i\sum_x \alpha|\phi(x)|^2|x\rangle\langle x|} \tag{18}$$

for larger coefficients α with high accuracy. Because the unitary propagators $e^{-iV(t;\hat{x})\Delta t}$ from Eq. (11) are diagonal phase operators in the position basis and the phase $V(t;x)\Delta t$ is small at each time step, we expect this block-encoding structure to efficiently implement the propagator using only a small number of iterations by choosing appropriate Δ and $\phi(x)$ parameters. Hence, together with the kinetic propagators, the block-encoding unit realizes a single time step of the unitary evolution for a small step size, and this structure is then used in iterations to simulate arbitrarily long dynamics. The circuit representation of the block-encoding unit is presented in Fig. 1. Here, U_ϕ is an amplitude oracle unitary that prepares the state $|\phi\rangle$ in the n_q-qubit ancillary register

$$U_\phi|0\rangle = |\phi\rangle = \sum_{x=0}^{2^{n_q}-1}\phi(x)|x\rangle, \tag{19}$$

and $U(\theta)$ is defined as a conditional phase operator

$$U(\theta)|y\rangle|x\rangle = \begin{cases} e^{i\theta}|y\rangle|x\rangle, & \text{for } x = y, \\ |y\rangle|x\rangle, & \text{for } x \neq y. \end{cases} \tag{20}$$

The block encoding uses the ancillary state $|\phi\rangle$ to apply the $e^{i\Delta|\phi(x)|^2}$ phase to the n_q-qubit primary register $|x\rangle$.

As we will show in the following, the $U(\theta)$ operator is very efficient, requiring only $\mathcal{O}(n_q)$ gates for its implementation. Therefore, the efficiency of the block encoding is determined

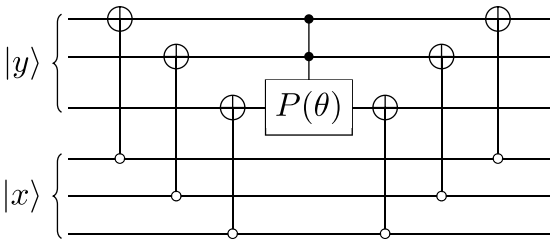


FIG. 2. Implementation of the conditional phase operator $U(\theta)$ for three-qubit input registers, using zero-controlled CNOT gates and a doubly controlled phase gate.

by the gate complexity of the U_ϕ operator. The main advantage of using this block-encoding structure is that it reduces the problem of diagonal phase operator construction to finding U_ϕ initializers for the corresponding phase distribution. Hence, the efficiency of our algorithm hinges upon efficient initializer circuits U_ϕ. However, the strength of this approach is that any phase profile, for which the corresponding state amplitudes can be efficiently prepared, is directly mapped to a unitary phase operator and can be treated efficiently by our algorithm to simulate wave propagation in a geometry with a given refractive index structure.

The construction of state preparation unitaries for specific states is a topic of ongoing research, and a full characterization of the efficiently preparable classes of functions remains elusive. In optics and many other scientific applications, the phase distributions usually belong to the class of smooth functions with bounded derivatives, and one can use techniques based on matrix product states [38,39] to prepare the states efficiently. Specific state initializers for particularly common states such as Gaussian distributions have also been reported in the literature [40]. Periodic distributions such as sine- or binary-type gratings can also be implemented by first preparing the Fourier spectrum of the function followed by an inverse QFT [41]. For an overview of the state preparation problem and the representation of data on a quantum computer, see Refs. [5,44]. These examples already show that many interesting real-world applications can be implemented using our algorithm, and the approach directly benefits from advances in the state preparation problem.

The role of the $U(\theta)$ operator is to establish the necessary entanglement between the registers to achieve the desired phase transformation. This operator is implemented using $\mathcal{O}(n_q)$ elementary gates. The general implementation of the operator for arbitrary n_q is discussed in Appendix A. However, as an example, the implementation for $n_q = 3$ is presented in Fig. 2, where $P(\theta) = \left[\begin{smallmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{smallmatrix}\right]$ is the elementary phase operator.

The efficient $\mathcal{O}(n_q)$ gate complexity means that, in most cases, the $U(\theta)$ operator will not be the limiting factor in the performance of the protocol, considering the usually higher gate complexity of the state preparation unitary U_ϕ. Also, when the protocol is used together with the QFT, e.g., in most simulation problems, the performance will be limited by the higher $\mathcal{O}(n_q^2)$ gate complexity of the QFT.

Direct calculation (Appendix B) of the output of the circuit in Fig. 1 shows that the circuit is a $(1, n_q, 0)$ block encoding of the operator

$$A(\theta) = I + 2ie^{i\frac{\theta}{2}} \sin\frac{\theta}{2} \sum_x |\phi(x)|^2 |x\rangle\langle x| \tag{21}$$

such that the following transformation is performed by the circuit:

$$U_A|0\rangle^{\otimes n_q}|\psi\rangle = |0\rangle^{\otimes n_q} A(\theta)|\psi\rangle + |\perp\rangle. \tag{22}$$

The success probability of the operation (namely, the probability of the ancillary register being $|0\rangle$) is

$$\begin{aligned} P_{\text{success}} &= \langle\psi|A^\dagger(\theta)A(\theta)|\psi\rangle \\ &\geqslant 1 - \sin^2\frac{\theta}{2}. \end{aligned} \tag{23}$$

If we take the phase to be small, $\theta = \Delta \ll 1$, then the operator becomes

$$\begin{aligned} A(\Delta) &= I + i\Delta \sum_x |\phi(x)|^2 |x\rangle\langle x| + \mathcal{O}(\Delta^2) \\ &= e^{i\Delta \sum_x |\phi(x)|^2 |x\rangle\langle x|} + \mathcal{O}(\Delta^2) \end{aligned} \tag{24}$$

in the first-order approximation in Δ, where the success probability of the operation is

$$P_{\text{success}} = 1 - \mathcal{O}(\Delta^2). \tag{25}$$

Starting with the primary register in the state $|\psi\rangle = \sum_x \psi(x)|x\rangle$ and applying the block encoding to it, we effectively imprint a phase onto the state as

$$|\psi\rangle = \sum_x \psi(x)|x\rangle \longrightarrow |\psi'\rangle \approx \sum_x \psi(x)\, e^{i\Delta|\phi(x)|^2}|x\rangle, \tag{26}$$

with arbitrarily high probability. The circuit in Fig. 1 is thus a $(1, n_q, \mathcal{O}(\Delta^2))$ block encoding of the operator

$$e^{i\Delta \sum_x |\phi(x)|^2 |x\rangle\langle x|}. \tag{27}$$

Hence, if we postselect the success case and repeatedly apply the circuit m times, we implement the operator

$$e^{i \sum_x \alpha |\phi(x)|^2 |x\rangle\langle x|}, \tag{28}$$

where $\alpha = m\Delta$, and the error of this implementation is $\epsilon = \mathcal{O}(m\Delta^2) = \mathcal{O}(\alpha^2/m)$. Therefore, the repeated application of the block-encoding unit is efficient when the phase coefficient α is not too large. Given a phase coefficient α, the number of iterations needed to achieve an error less than ϵ is $m = \mathcal{O}(\alpha^2/\epsilon)$.

This allows for the implementation of the time evolution operator in Eq. (11) as

$$\hat{U}(T,0) = \prod_{j=0}^{N_t-1} e^{-i\hat{K}\Delta t} e^{-i\hat{V}(t_{j+1/2})\Delta t} + \epsilon_{\text{tot}}, \tag{29}$$

where $\epsilon_{\text{tot}} = \mathcal{O}(T^2/N_t)$ is the total simulation error, and the $e^{-i\hat{V}(t)\Delta t}$ operators are implemented using the block-encoding construction by choosing suitable ancillary states $|\phi\rangle$ and phase coefficients α corresponding to the desired phase profiles $V(t;x)\Delta t$ at each time step. The requirement of the protocol on the Δ parameter to be small also fits well with the assumption of small Δt in the Trotter scheme used to simplify

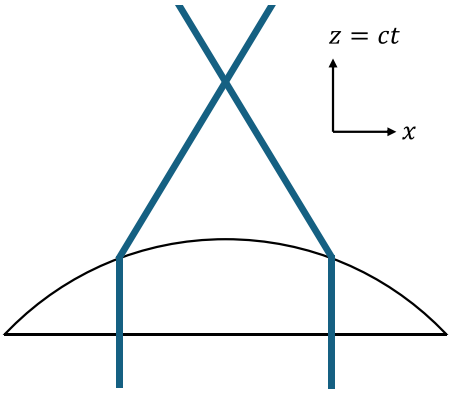


FIG. 3. Schematic visualization of the simulation setup. The beam propagates in the z direction.

the total time evolution unitary, which leads to no more than $m = \mathcal{O}(1)$ iterations of the block-encoding unit to realize the potential propagator at each time step. See Appendix C for a detailed analysis. This means that, in order to perform a time evolution for total time T with a maximum tolerated error ϵ_{tot}, we need $\mathcal{O}(T^2/\epsilon_{\text{tot}})$ iterations. Also note that the block encoding in Eq. (24) directly implements the first-order Taylor expansion of the unitary

$$e^{i\sum_x \Delta|\phi(x)|^2|x\rangle\langle x|} \tag{30}$$

and therefore does not require using advanced Hamiltonian simulation techniques such as quantum signal processing [21,45].

IV. SIMULATIONS

To showcase the algorithm, we simulated the propagation of a one-dimensional Gaussian beam in a two-dimensional setup passing through a spherical (plano-convex) lens [46,47]. We assume the paraxial approximation and align the z axis with the propagation direction. The lens is expected to cause the beam to focus at a certain propagation distance after the lens. The lens has a finite nonzero thickness, and due to the spherical shape of its surface, spherical aberration is expected to appear. The lens is also assumed to be lossless; therefore, it only imprints phases on the incident beam. The schematic plot of the setup is shown in Fig. 3.

Following the split-step procedure described in Eq. (11), in order to perform the simulation using the protocol, we cut the lens perpendicular to the propagation axis and approximate it as a stack of thin rectangular layers of material. This procedure is shown in Fig. 4. Note that the propagation distance z and the elapsed time are interchangeable according to $z = ct$.

This allows us to propagate the beam through the lens step by step by sequentially applying a rectangular phase profile to the field (in the real domain) and propagating it for the thickness of each layer (quadratic phase profile in the Fourier domain). For the phase corresponding to each layer of material in the real domain, the ancillary state $|\phi\rangle$ is chosen as a rectangular state whose width is calculated using the curvature of the lens surface, and the phase coefficient α is chosen such that the part of the field passing through the material picks up a phase equal to $\Delta\theta = (n-1)k_0 l$, where n, k_0, and l are, respectively, the refractive index of the material, the vacuum wave number, and the thickness of each slice (Fig. 5).This particular approximation is not the only possible choice and is used here for its simplicity. We note that the ability of the phase protocol to tune the applied phase profiles provides great flexibility to design and explore various possibilities for such simulations. This flexibility is showcased in Appendix D, where several other lens structures are simulated.

Moreover, the kinetic energy part of the Hamiltonian that simulates the propagation of the field along the z axis is implemented in three steps [32]: (1) applying a QFT to the state $|v\rangle$, (2) applying the paraxial-approximated transfer function (a quadratic phase) to the Fourier transform of the field as

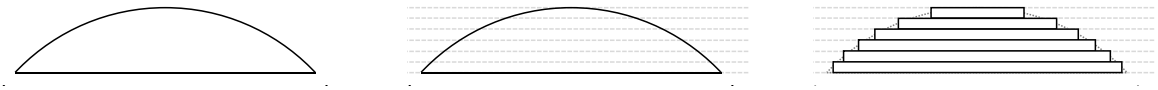
$$\mathcal{H}_f(k_\perp) = e^{-i\frac{k_\perp^2}{2k}z}, \tag{31}$$

where $k_\perp$ is the transverse spatial frequency and z is the propagation distance, and finally (3) applying an inverse QFT to retrieve the propagated field in the real domain.

We ran the simulation with the wavelength, initial Gaussian beam waist, refractive index, radius of curvature, and number of qubits given, respectively, by $\lambda = 1\,\mu\text{m}$, $w_0 = 25\,\mu\text{m}$, $n = 1.25$, $R = 50\,\mu\text{m}$, and $n_q = 7$. The result of this simulation is presented in Fig. 6.

Summarizing our observations from Fig. 6, our results reproduce the expected focusing of the beam, and we see the interference of out-of-phase refracted components of the wave before reaching the focus, which is an aberration expected from a thick lens with a spherical surface. Furthermore, it is also known that these aberration effects will decrease if we change the orientation of the lens such that the incident beam hits the convex surface first. To showcase the flexibility of this simulation protocol, we performed the simulation in the reverse orientation and several other cases, such as lenses with parabolic surface curvature, and the results are presented in Appendix D.

Next, we characterized the performance of the quantum simulation. The most important parameter for the performance is the maximum allowed phase step Δ_{max} in the block encodings in Eq. (24), which corresponds to the time step size in the simulation. For benchmarking the effect of Δ_{max}, we ran simulations keeping all the parameters fixed except for Δ_{max}, which was varied over the range $[0.001, 0.2]$, and analyzed two metrics: accuracy and probability of success. The expectation is that decreasing Δ_{max} leads to higher accuracy



(a) Original lens (b) Slicing (c) Rectangular approximation

FIG. 4. Slicing the lens and approximating it by a stack of rectangular layers with a fixed thickness in the propagation direction.

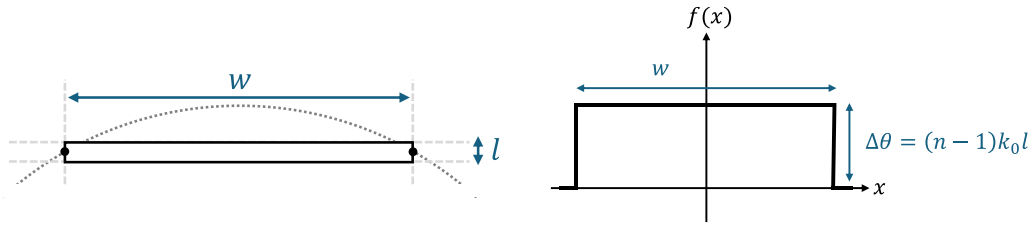


FIG. 5. The phase difference that each layer of the material imprints on the incident beam, depending on whether the corresponding part of the wave in the transverse axis passes through the material or propagates through the vacuum outside the material. (a) The geometry of a slice of the lens. (b) The corresponding relative phase $e^{if(x)}$ that each slice of the lens imprints on the incident beam in the real domain.

and also higher probability of success at the cost of more simulation steps, i.e., larger gate count and simulation time.

For measuring the accuracy, we calculated the fidelity between the output state of the quantum simulation and the state resulting from classical calculations at a fixed propagation distance $z \approx 200\,\mu$m. The classical simulation was done by directly multiplying the incident light field by the corresponding phases from the simulation scheme in Eq. (11) and using classical discrete Fourier transforms for the kinetic energy terms. The reason that the accuracy is defined in this way is to keep other errors such as the midpoint approximation of the time-ordered operator and the split-step error the same between the classical and quantum cases. With that, we isolated the effects of the errors explicitly resulting from the block-encoded phase operators of the quantum simulation protocol. The result is shown in Fig. 7(a).

From analytical calculations (Appendix C), we expect the error to scale as $\epsilon = \mathcal{O}(\Delta_{\max})$. Therefore, the fidelity as a function of $\Delta_{\max}$ behaves as

$$F = 1 - \mathcal{O}(\epsilon^2) = 1 - \mathcal{O}\left(\Delta_{\max}^2\right), \tag{32}$$

and this is confirmed by fitting a quadratic function to the fidelity in Fig. 7(a) for small values of $\Delta_{\max}$. Note that the fidelities are calculated in the case of successful postselections. To fully characterize the simulation protocol, we should also analyze the success probability. The total probability of success as a function of small $\Delta_{\max}$ is

$$P_{\text{success}} = 1 - \mathcal{O}(\epsilon) = 1 - \mathcal{O}(\Delta_{\max}), \tag{33}$$

meaning that for small $\Delta_{\max}$ we expect a linear behavior, which is seen in Fig. 7(b). We conclude the discussion by mentioning that, although it is not straightforward to calculate an exact practical value for $\Delta_{\max}$ to achieve high accuracy for a particular simulation problem, it can be said that as long as the probability of success is high, i.e., postselections are successful, we know that the accuracy will also be high. This makes the success or failure of the postselection process a measure of the accuracy of the simulation as well. In a real-world application, one can start with a reasonable value of $\Delta_{\max}$ and adapt by tuning it to smaller values if a simulation trial fails.

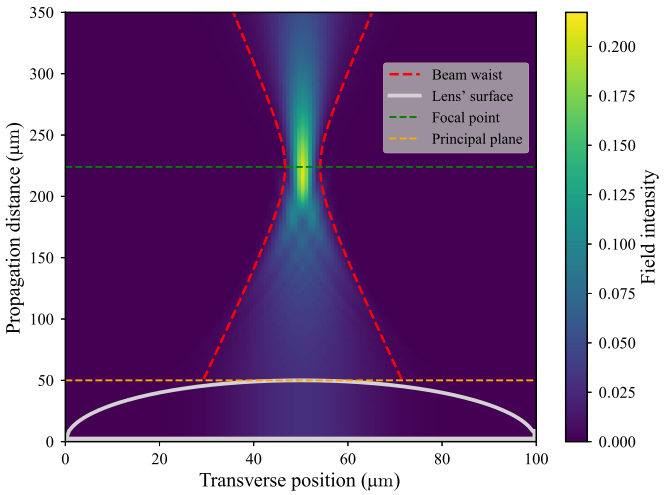


FIG. 6. The result of the quantum simulation of a Gaussian beam incident on a plano-convex lens. For this simulation, the maximum phase step was $\Delta_{\max} = 0.01$ and the lens was sliced into 10 000 layers.

V. CONCLUSION

In this work, we discussed how the problem of optical wave propagation in a weakly inhomogeneous medium in the paraxial approximation is reduced to a time-dependent Hamiltonian simulation problem. We then introduced a block-encoding structure that enables the implementation of unitary time evolutions accurate to the first-order approximation in propagation time. The method is highly flexible since the Hamiltonian is programmed in the quantum computer using an amplitude oracle that prepares the wavefunction corresponding to a given phase profile. The problem of unitary phase operator synthesis is therefore mapped to the state preparation problem. This modularity allows for the simulation of optical elements with various refractive index distributions, as long as the corresponding phase profiles can be prepared efficiently in a quantum register, and any improvement to the gate complexity of the state preparation part is directly transferred to the block-encoded phase operators.

The simulation algorithm is exponentially efficient in memory consumption, requiring $\mathcal{O}(\log N)$ qubits for a discretized light field with the grid size of N in the transverse plane. A similar improvement (polylogarithmic scaling with N) for the circuit depth is also possible for simulations involving refractive index distributions that are efficiently preparable in quantum registers. This condition is fulfilled by a large class of relevant scientific applications where the refractive index

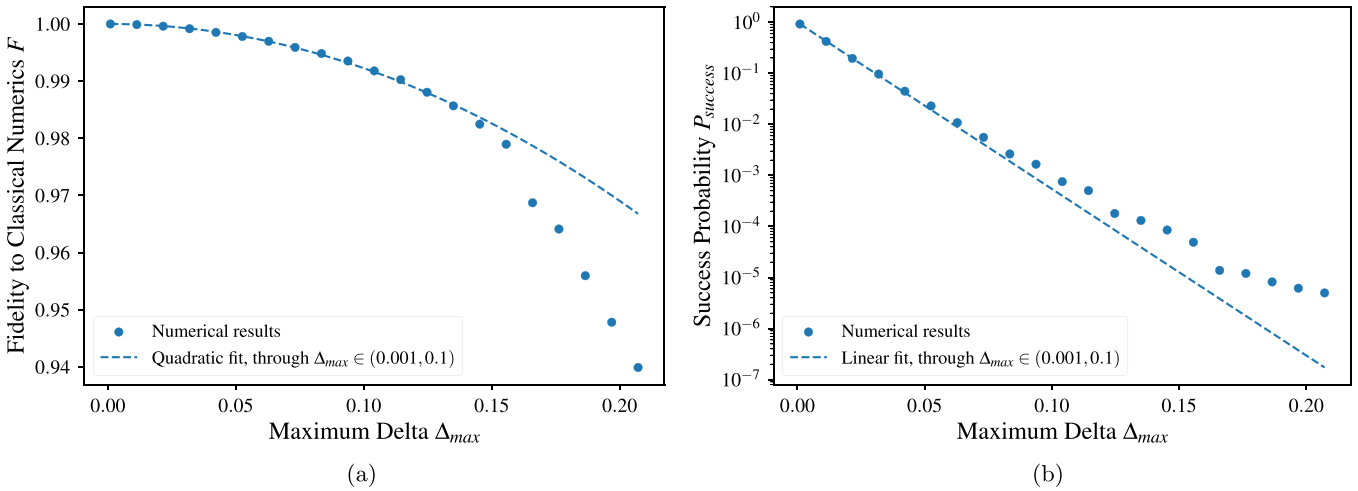


FIG. 7. Characterization of the error and the performance of the quantum simulation. (a) The fidelity between the resulting wavefunction from the quantum simulation and the classical numerical result obtained by directly multiplying the field with the phases (calculated at $z \approx 200\,\mu\text{m}$), postselecting successful outcomes. The quadratic fit is taken over the range $\Delta_{\max} \in (0.001, 0.1)$. (b) The probability of success of the quantum simulation, that is, the probability of all postselections leading to the desired outcome; the linear fit is taken over the same range. In these simulations, the number of qubits was $n_q = 6$.

distribution is smooth and bounded in its derivative. Notably, the number of times that the block-encoding construction is needed for the simulation grows only linearly with the number of time steps. Therefore, the block-encoded implementation does not add further computational complexity in terms of the simulation time beyond what time discretization already requires for achieving a desired accuracy. Meanwhile, the block-encoding subroutine at each time step can potentially be implemented using a number of gates polylogarithmic in the grid size N. This efficiency promises quantum advantage for a range of high-precision applications. We also showcased the algorithm by simulating a simple wave optics experiment, that is, the focusing of a Gaussian beam by a lens.

It must be pointed out that the simulation protocol described in this work will not be efficient if the objective is to extract the complete propagated field for a given propagation distance, because the output is computed as wavefunction amplitudes in a quantum register. Trying to extract the field intensity from this quantum register by direct measurement will inevitably lead to losing the quantum speedup. Reconstructing the output field via state tomography results in the same issue. Instead of direct measurements on the output register, one has to choose a particular observable and only extract information about that. This is, however, not a drawback specific to this protocol but a general consideration for many quantum numerical solvers. Nevertheless, focusing on one observable can still be very beneficial, for example, in an optical design application where the observable can be the coupling efficiency, mode overlap, or a Zernike coefficient that needs to be minimized or maximized for an application.

As an example of a concrete readout procedure, we discuss the estimation of the mode overlap of the final output field with a target mode, a common problem appearing in optical applications such as fiber coupling. We propose to estimate the mode overlap using the swap test [5]. The swap test takes the register containing the result of the quantum simulator $|v(t)\rangle$ as one of its inputs, while the other input is initialized in a reference state $|\psi_r\rangle$, which, for example, can be the fundamental mode of a fiber approximated as a Gaussian field. Measuring the test qubit $|q_h\rangle$ after the execution of the swap test subroutine (Fig. 8) results in the probabilities p_0 and p_1 of the qubit being in states $|0\rangle$ or $|1\rangle$, and $p_0 - p_1$, which is the expectation value of the Pauli-Z operator on the test qubit, is an estimate of the overlap of the two states $|\langle\psi_r|v(t)\rangle|^2$, which is exactly the coupling efficiency to the corresponding fiber mode. Note that since the estimation only needs measuring one qubit, the number of times needed to repeat the process to estimate the overlap to within an error ϵ is $N_r = \mathcal{O}(1/\epsilon^2)$, independent of the number of qubits in the original register $|v(t)\rangle$. In this way, state tomography is not performed but the relevant information is efficiently extracted. The ability to efficiently simulate the propagation of a light field through optical elements and to estimate design loss functions can greatly enhance iterative design processes and allow for rapid optimization of the design of optical systems.

While the simulation algorithm introduced here, in its current form, is limited to small numerical apertures and cannot yet treat polarization, it shows not only that quantum-enabled optical system design is feasible, but also that exponential speedup is a possibility. We expect our work to motivate the optics community to consider quantum computing as a promising platform for optical simulation and to further investigate the utility of recent developments in quantum algo-

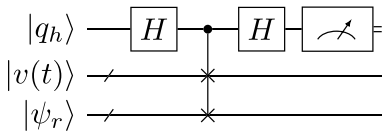


FIG. 8. The swap test for estimating the overlap of two quantum states. The qubit $|q_h\rangle$ starts in the $|0\rangle$ state, and the measurement is done in the computational basis.

rithms such as block encoding and quantum signal processing for advances in the field.

ACKNOWLEDGMENTS

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)-Project No. 398816777-SFB 1375. F.E. and S.D. are part of the Max Planck School of Photonics supported by the Dieter Schwarz Foundation, the German Federal Ministry of Research, Technology and Space (BMFTR), and the Max Planck Society.

DATA AVAILABILITY

Snapshots of the source code and the datasets generated and analyzed in this work are publicly available and permanently stored at, respectively, Refs. [46,47].

APPENDIX A: IMPLEMENTATION OF THE CONDITIONAL PHASE OPERATOR

Here, we provide the details of the implementation of the $U(\theta)$ operator introduced in Sec. III. The implementation needs $\mathcal{O}(n_q)$ gates, where n_q is the number of qubits of each of the input registers.

The operator is defined in terms of its action on the computational basis states as

$$U(\theta)|y\rangle|x\rangle = \begin{cases} e^{i\theta}|y\rangle|x\rangle, & \text{for } x = y, \\ |y\rangle|x\rangle, & \text{for } x \neq y. \end{cases} \tag{A1}$$

The goal is to implement this operator by decomposing it into elementary single- and two-qubit gates. The key to its implementation is that the condition $x = y$ in Eq. (A1) is realized when all corresponding qubits of the registers $|x\rangle$ and $|y\rangle$ are in the same state. If we represent the basis states of each register as the tensor product of the underlying qubits, we have

$$|x\rangle = |x_{n_q-1}\rangle \cdots |x_1\rangle|x_0\rangle, \tag{A2}$$

and similarly

$$|y\rangle = |y_{n_q-1}\rangle \cdots |y_1\rangle|y_0\rangle, \tag{A3}$$

where $x_j \in \{0, 1\}$ and $y_j \in \{0, 1\}$ are the bits of the binary representations of x and y. The condition $x = y$ is equivalent to $x_j = y_j$ for all j.

The equality of each qubit pair can be checked via elementary unitary operations. To check the condition $x_j = y_j$, we should calculate the flag bit $z_j = y_j \oplus \overline{x_j}$, where $\oplus$ is the XOR logical operator and the overline in $\overline{x_j}$ represents the logical NOT operation. The flag state $|z_j\rangle = |y_j \oplus \overline{x_j}\rangle$ can thus be computed using a CNOT gate acting on the $|y_j\rangle$ qubit with $|x_j\rangle$ as the control qubit and control state 0,

$$\text{CNOT}_0|x_j\rangle|y_j\rangle = |x_j\rangle|y_j \oplus \overline{x_j}\rangle = |x_j\rangle|z_j\rangle. \tag{A4}$$

Using n_q cnot$_0$ gates, we can compute all the flag bits z_j; a multicontrolled phase gate $P(\theta)$ with the phase parameter θ acting on the $|z_j\rangle$ qubits then applies the phase to the state if all the $|z_j\rangle$ qubits are $|1\rangle$, which is equivalent to the condition $x = y$. Finally, the temporarily computed $|z_j\rangle$ states are uncomputed by applying the same CNOT gates again.

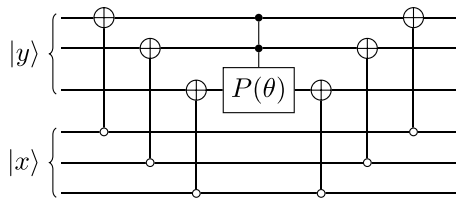


FIG. 9. Implementation of the conditional phase operator $U(\theta)$ for three-qubit input registers. This is the same circuit as Fig. 2, repeated here for convenience.

As an example, the implementation for the case of three-qubit quantum registers ($n_q = 3$) is presented in Fig. 9. Despite the appearance of the circuit structure, the $U(\theta)$ operator is in fact symmetric with respect to the exchange of its input registers, as is clear from Eq. (A1).

As we see in this implementation, we need $2n_q$ CNOT gates (n_q to compute the flag bits and n_q to uncompute them) and one multicontrolled phase gate acting on n_q qubits. Because the multicontrolled phase gate can be implemented using $\mathcal{O}(n_q)$ gates [35], the total gate complexity of the $U(\theta)$ operator is $\mathcal{O}(n_q)$.

APPENDIX B: DERIVATION OF THE BLOCK ENCODING

Here, we discuss the derivations of the results presented in Sec. III in detail. The discussion includes the block encoding of the operator

$$A(\theta) = I + 2i\, e^{i\frac{\theta}{2}} \sin\frac{\theta}{2} \sum_x |\phi(x)|^2 \Pi_x, \tag{B1}$$

its approximation for $\theta = \Delta \ll 1$:

$$A(\Delta) \approx e^{i\Delta \sum_x |\phi(x)|^2 |x\rangle\langle x|}, \tag{B2}$$

and the case of its iterative application.

Let us start by finding a useful representation of the conditional phase operator in Eq. (A1). The operator can be rewritten as

$$U(\theta) = \sum_x \left(\left[e^{i\theta}\Pi_x + \sum_{y\neq x} \Pi_y \right] \otimes \Pi_x \right), \tag{B3}$$

where $\Pi_x = |x\rangle\langle x|$ is the projection operator that projects vectors onto the $|x\rangle$ basis state. Using the completeness relation $\sum_y \Pi_y = I$, we have

$$U(\theta) = \sum_x \left[I + 2i\, e^{i\frac{\theta}{2}} \sin\frac{\theta}{2}\, \Pi_x \right] \otimes \Pi_x. \tag{B4}$$

To simplify the notation for the following derivations, we define $U_x = I + 2i\, e^{i\frac{\theta}{2}} \sin\frac{\theta}{2}\, \Pi_x$ and rewrite Eq. (B4) as

$$U(\theta) = \sum_x U_x \otimes \Pi_x. \tag{B5}$$

This operator does not contain any reference to the $\phi(x)$ state, and therefore it is not capable of applying x-dependent phase profiles. We need to involve the U_ϕ operator to form the complete block encoding. U_ϕ is defined such that it prepares the $|\phi\rangle$ state from the zero state, $|\phi\rangle = U_\phi|0\rangle$. Note that this

definition does not uniquely identify U_ϕ because its action on other computational basis states, $U_\phi|j\rangle$, is not specified. However, only the action on the zero state will be relevant in the overall block-encoding construction, and therefore any U_ϕ fulfilling $|\phi\rangle = U_\phi|0\rangle$ is sufficient for the implementation. Let us now construct the complete block-encoding unitary U including the U_ϕ and $U_\phi^\dagger$ operators acting on the ancillary register

$$\begin{aligned} U &= (U_\phi^\dagger \otimes I)(U(\theta))(U_\phi \otimes I) \\ &= \sum_x U_\phi^\dagger U_x U_\phi \otimes \Pi_x. \end{aligned} \tag{B6}$$

This means that the unitary U is a block encoding of the following operator:

$$A(\theta) = (\langle 0| \otimes I)U(|0\rangle \otimes I) \tag{B7}$$

$$= \sum_x (\langle 0|U_\phi^\dagger U_x U_\phi|0\rangle)\Pi_x \tag{B8}$$

$$= \sum_x \left(1 + 2i\, e^{i\frac{\theta}{2}} \sin\frac{\theta}{2}\, |\phi(x)|^2\right)\Pi_x \tag{B9}$$

$$= I + 2i\, e^{i\frac{\theta}{2}} \sin\frac{\theta}{2} \sum_x |\phi(x)|^2 \Pi_x, \tag{B10}$$

which is equal to the operator we wanted to construct [Eq. (B1)]. The probability of success of the postselection, assuming that the primary state was initially $|\psi\rangle = \sum_x \psi(x)|x\rangle$, is

$$P_{\text{success}} = \langle\psi|A^\dagger(\theta)A(\theta)|\psi\rangle$$

$$= \sum_x |\psi(x)|^2 \left(1 - 4\sin^2\frac{\theta}{2}\, |\phi(x)|^2(1 - |\phi(x)|^2)\right) \tag{B11}$$

$$= 1 - 4\Gamma^2 \sin^2\frac{\theta}{2} \tag{B12}$$

$$\geqslant 1 - \sin^2\frac{\theta}{2}, \tag{B13}$$

where

$$\Gamma^2 = \sum_x |\psi(x)|^2\, |\phi(x)|^2(1 - |\phi(x)|^2) \leqslant \frac{1}{4} \tag{B14}$$

is a constant that depends on the initial state and the applied phase profile. Furthermore, the wavefunction after postselection is

$$|\psi'\rangle = \frac{A(\theta)|\psi\rangle}{\sqrt{P_{\text{success}}}}. \tag{B15}$$

We can thus say that the operator U in Eq. (B6) is a $(1, n_q, 0)$ block encoding of $A(\theta)$ in Eq. (B1), which concludes the first part of our derivation. Next, we investigate the condition that brings the $A(\theta)$ operator into the form of a diagonal phase operator.

In the approximated case of $\theta = \Delta \ll 1$, the block-encoded operator is

$$A(\Delta) = e^{i\Delta \sum_x |\phi(x)|^2 |x\rangle\langle x|} + \mathcal{O}(\Delta^2); \tag{B16}$$

therefore, the same operator U is a $(1, n_q, \mathcal{O}(\Delta^2))$ block encoding of

$$e^{i\Delta \sum_x |\phi(x)|^2 |x\rangle\langle x|}. \tag{B17}$$

The error of this implementation, defined using the spectral norm, is

$$\epsilon = \left\| \frac{A(\Delta)}{\sqrt{P_{\text{success}}}} - e^{i\Delta \sum_x |\phi(x)|^2 |x\rangle\langle x|} \right\| = \mathcal{O}(\Delta^2) \tag{B18}$$

and the probability of success is

$$P_{\text{success}} = 1 - \mathcal{O}(\Delta^2) = 1 - \mathcal{O}(\epsilon), \tag{B19}$$

which concludes the second part of the derivations.

For the final part of the derivations, we investigate the computational complexity of realizing the phase operator

$$e^{i \sum_x f(x)|x\rangle\langle x|}, \tag{B20}$$

given an arbitrary phase profile $f(x)$, by repeated application of the block-encoding unit in Eq. (B16). In order to use the previous results, the first step is to write $f(x)$ in a form suitable for the block-encoding construction as

$$f(x) = \alpha|\phi(x)|^2 \tag{B21}$$

for some coefficient α and a normalized wavefunction $\phi(x)$. We assume that $f(x)$ is positive for now, but this requirement will be lifted later. Since $\phi(x)$ is normalized, the coefficient α is calculated as

$$\alpha = \sum_x f(x). \tag{B22}$$

The idea is then to write $\alpha = m\Delta$ and repeat the block-encoded operator in Eq. (B16) m times with the phase parameter Δ. Since the error at each step is $\mathcal{O}(\Delta^2)$ and the total worst-case error after m iterations is additive [35], we have the total error

$$\epsilon_{\text{tot}} = \mathcal{O}(m\Delta^2) = \mathcal{O}\left(\frac{\alpha^2}{m}\right) = \mathcal{O}(\alpha\Delta), \tag{B23}$$

and the operation that gets applied with the error ϵ_{tot} is

$$\left(e^{i\Delta \sum_x |\phi(x)|^2 |x\rangle\langle x|}\right)^m = e^{i\alpha \sum_x |\phi(x)|^2 |x\rangle\langle x|} = e^{i \sum_x f(x)|x\rangle\langle x|}. \tag{B24}$$

Furthermore, the total probability of success of m postselections is

$$P_{\text{success}} = [1 - \mathcal{O}(\Delta^2)]^m = 1 - \mathcal{O}(\epsilon_{\text{tot}}). \tag{B25}$$

Since we have

$$|\alpha| = \|f\|_1 = \sum_x |f(x)|, \tag{B26}$$

for a given f, the error as a function of the phase and the number of iterations used to realize it is

$$\epsilon_{\text{tot}} = \mathcal{O}\left(\frac{\|f\|_1^2}{m}\right). \tag{B27}$$

In the case of an arbitrary f without the positivity assumption, one can still write down the phase contributions as a sum of

positive and negative parts

$$\sum_x f(x)|x\rangle\langle x| = \sum_{f(x)>0} |f(x)||x\rangle\langle x| - \sum_{f(x)<0} |f(x)||x\rangle\langle x| \tag{B28}$$

and implement the two parts using phase steps $\Delta = \|f\|_1/m$ and $-\Delta$, respectively, which leads to the same total error scaling $\epsilon_{\text{tot}} = \mathcal{O}(\|f\|_1^2/m)$. The total number of gates to implement the iterative case is

$$\mathcal{O}(m(\log N + G)), \tag{B29}$$

where G is the gate complexity of the initializer U_ϕ (or $U_\phi^\dagger$).

APPENDIX C: ERROR ANALYSIS OF THE SIMULATION

The paraxial-approximated differential equation [Eq. (7)] governing the propagation of an optical wave v (setting $\hbar = c = 1$) is

$$i\frac{\partial}{\partial t}|v(t)\rangle = \hat{H}(t)|v(t)\rangle \tag{C1}$$

$$= \left[-\frac{\hat{\partial}_x^2}{2\tilde{k}} + V(t;\hat{x})\right]|v(t)\rangle, \tag{C2}$$

and the solution to this equation starting at $t = 0$ and propagating to time $t = T$ is formally written as

$$|v(T)\rangle = \hat{U}(T,0)|v(0)\rangle = \mathcal{T}\exp\left(-i\int_0^T \hat{H}(t)\,dt\right)|v(0)\rangle, \tag{C3}$$

where $\mathcal{T}$ denotes time ordering and the potential term $V(t;x)$ encodes the phases that the wave accumulates at each step of the propagation based on the geometry and the refractive index distribution of the space. We assume a one-dimensional transverse axis to match the simulation in Sec. IV, although most of the derivations are valid in the general case with minor modifications. There are several sources of error at different stages of transforming this formal solution into an expression suitable for a numerical solver, namely,

$$\begin{aligned}
\text{exact}: &\quad \mathcal{T}e^{-i\int \hat{H}(t)\,dt},\\
\text{time discretization}: &\quad \textstyle\prod_j e^{-i\hat{H}(t_{j+1/2})\Delta t},\\
\text{kinetic-potential split}: &\quad \textstyle\prod_j e^{-i\hat{K}\Delta t}e^{-i\hat{V}(t_{j+1/2})\Delta t},\\
\text{transverse space discretization}: &\quad \textstyle\prod_j e^{-i\hat{K}_h\Delta t}e^{-i\hat{V}_h(t_{j+1/2})\Delta t},\\
\text{quantum circuit implementation}: &\quad \textstyle\prod_j \hat{U}_{K_h}\hat{U}_{V_h(j)}.
\end{aligned}$$

Notably, except for the final implementation stage, the approximations at each stage are common between classical and quantum solvers. In the following, the norm of an operator $\|\cdot\|$ is defined as the spectral norm.

The first approximation needed is to discretize the time-ordered operator as a sequence of small steps in time by defining the total number of time steps N_t, the step size $\Delta t = T/N_t$, and the time grid points $t_j = j\Delta t$. We use a midpoint approximation [48,49], meaning that the Hamiltonian in the time window $[t_j, t_{j+1}]$ is approximated as $\hat{H}(t_{j+1/2})$. This leads to an approximation of the time evolution operator as

$$\hat{U}(T,0) = \mathcal{T}\exp\left(-i\int_0^T \hat{H}(t)\,dt\right) \tag{C4}$$

$$= \prod_{j=0}^{N_t-1} e^{-i\hat{H}(t_{j+1/2})\Delta t} + \epsilon_{\Delta t}, \tag{C5}$$

where the error is

$$\epsilon_{\Delta t} = \mathcal{O}\big(T\Delta t^2\big(\max_t \big\|\partial_t^2\hat{H}(t)\big\| + \max_t \|[\hat{H}(t), \partial_t\hat{H}(t)]\|\big)\big) \tag{C6}$$

$$= \mathcal{O}\big(T\Delta t^2\big(\max_t \big\|\partial_t^2\hat{V}(t)\big\| + \max_t \|[\hat{K}, \partial_t\hat{V}(t)]\|\big)\big). \tag{C7}$$

In a simulation, this error quantifies the lens-slicing error and is negligible as long as the material is weakly inhomogeneous.

The next error results from separating the noncommuting kinetic and potential energy parts of the Hamiltonian. Using first-order Trotter splitting [50]

$$e^{-i(\hat{A}+\hat{B})\Delta t} = e^{-i\hat{A}\Delta t}e^{-i\hat{B}\Delta t} + \mathcal{O}(\Delta t^2\,\|[\hat{A},\hat{B}]\|), \tag{C8}$$

we get

$$\hat{U}(T,0) = \mathcal{T}\exp\left(-i\int_0^T \hat{H}(t)\,dt\right) \tag{C9}$$

$$= \prod_{j=0}^{N_t-1} e^{-i\hat{K}\Delta t}e^{-i\hat{V}(t_{j+1/2})\Delta t} + \epsilon_s + \epsilon_{\Delta t}, \tag{C10}$$

where the accumulated splitting error for the whole propagation is

$$\epsilon_s = \mathcal{O}(T\Delta t \max_t \|[\hat{K}, \hat{V}(t)]\|). \tag{C11}$$

We then need to discretize the transverse axis. Discretization affects the field v as well as the kinetic and potential operators. Given a total transverse length L, we define the number of discretization points N_x, the grid size $\Delta x = L/N_x$, and the sample points $x_j = j\Delta x$. We sometimes use the substitute notation $h = \Delta x$ to keep the equations compact. After discretization, the action of the second-order partial derivative operator on the sampled field is approximated as

$$\partial_x^2|v\rangle \approx \frac{v_{j+1} - 2v_j + v_{j-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2), \tag{C12}$$

which means that the norm of the discretized kinetic operator scales as $\|K_h\| = \mathcal{O}(\Delta x^{-2})$, and the error resulting from the spatial discretization of the kinetic energy propagator is

$$\|e^{-i\hat{K}_h\Delta t} - e^{-i\hat{K}\Delta t}\| = \mathcal{O}(\Delta t\,\Delta x^2). \tag{C13}$$

A reasonable criterion for choosing temporal and spatial discretization grid sizes Δt and Δx is to balance the global error resulting from the spatial discretization $\mathcal{O}(T\Delta x^2)$ with the leading error term in the Trotter splitting $\epsilon_s = \mathcal{O}(T\Delta t)$, that is, to choose

$$\Delta t = C\Delta x^2 = \mathcal{O}(\Delta x^2), \tag{C14}$$

for a scalar C. Intuitively speaking, making the spatial grid finer will only result in more favorable total error scaling if

the time step size gets smaller accordingly. For this reason, based on the physical parameters of a simulation problem, the grid size Δx must not be chosen smaller than needed for capturing the largest spatial frequency components in the simulation, because it will unnecessarily add to the number of time steps without improving the error scaling. Therefore, in discussing the spatial discretization errors, the parameter $C = \Delta t/\Delta x^2$ is a more practically relevant parameter than Δt or Δx separately. Rewriting the kinetic-potential splitting error in terms of the new parameters after discretization, we have

$$\epsilon_s = \mathcal{O}(T\,\Delta t\|K_h\|\max_t\|V_h(t)\|) = \mathcal{O}\left(T\left(\frac{\Delta t}{\Delta x^2}\right)\max_t|V(t)|\right) = \mathcal{O}(TC\max_t|V(t)|). \tag{C15}$$

Since the kinetic energy propagator $e^{-i\hat{K}_h\Delta t}$ is now a discretized $N_x \times N_x$ operator that has a simple time-independent diagonal form in the Fourier basis, it can straightforwardly be implemented using a diagonal quadratic phase propagator and QFT subroutines with no extra error, which in total needs $\mathcal{O}(\log^2 N_x)$ quantum gates [32], and the potential energy propagator $e^{-i\hat{V}_h(t_{j+1/2})\Delta t}$ can be implemented using the block encoding described in Appendix B.

The potential propagator at each time step corresponds to the phase $f_h(t;x_j) = -V_h(t;x_j)\Delta t$, and the 1-norm of this phase is $\|f_h(t)\|_1 = \|V_h(t)\|_1\Delta t$. In order to analyze the implementation cost, we need to calculate how this norm scales with the discretization parameters. As we will argue in the following, the requirement that Δt scales as Δx^2 makes the norm of f_h small enough for the block-encoding implementation to be efficient. In the limit of large N_x, there is a relation between the average value of a continuous signal and its discretization:

$$\lim_{N_x\to\infty}\frac{1}{N_x}\sum_{j=0}^{N_x-1}|V_h(x_j)| = \frac{1}{L}\int_0^L|V(x)|\,dx; \tag{C16}$$

thus, the norms relate as $\|V_h(t)\|_1 = \mathcal{O}(\|V(t)\|_1/\Delta x)$. Combining this relation with the scaling of the time step with the grid size $\Delta t = C\Delta x^2$ to calculate the norm of the discretized phase function $\|f_h(t)\|_1 = \|V_h(t)\|_1\Delta t$, we have

$$\|f_h(t)\|_1 = \mathcal{O}(C\Delta x\,\|V(t)\|_1). \tag{C17}$$

Assuming at each time step we allow for m iterative applications of the block encoding to realize the phase, the implementation of the potential energy propagator introduces the local error

$$\epsilon_{b,\text{local}} = \mathcal{O}\left(\frac{C^2\Delta x^2\|V(t)\|_1^2}{m}\right) = \mathcal{O}\left(\frac{C\Delta t\|V(t)\|_1^2}{m}\right), \tag{C18}$$

which leads to the total error

$$\epsilon_b = \mathcal{O}\left(\frac{TC\max_t\|V(t)\|_1^2}{m}\right). \tag{C19}$$

Comparing this result to the splitting error $\epsilon_s = \mathcal{O}(TC\max_t|V(t)|)$, one sees that, after discretization, the requirement to keep the global kinetic-potential splitting error small leads to small phases for the potential energy propagator at each time step such that it makes the block encoding in Appendix B suitable for the implementation of the propagators. This implementation results in an additive global error that is no worse than the error introduced by discretization after choosing $m = \mathcal{O}(1)$ local iterations. In this case, we have the final result of

$$\hat{U}(T,0) = \mathcal{T}\exp\left(-i\int_0^T\hat{H}(t)\,dt\right) \tag{C20}$$

$$= \prod_{j=0}^{N_t-1}\hat{U}_{K_h}\hat{U}_{V_h(j)} + \epsilon_b + \epsilon_s + \epsilon_{\Delta t}, \tag{C21}$$

where $\epsilon_b = \mathcal{O}(TC\max_t\|V(t)\|_1^2)$ for $m = \mathcal{O}(1)$.

This concludes the analysis of the errors introduced at each stage of the quantum algorithm. The total numbers of gates for implementing the kinetic and potential energy propagators are, respectively, $\mathcal{O}(N_t\log^2 N_x)$ and

$$\mathcal{O}\left(N_t\log N_x + \sum_j G_j\right) = \mathcal{O}(N_t(\log N_x + \max_j G_j)), \tag{C22}$$

where G_j is the number of gates needed to implement the state initializer $U_{\phi(j)}$ corresponding to the phase profile $V(t_{j+1/2};x)$. Once all the simulation parameters such as T, L, $\hat{V}(t)$, and the number of sample points N_x are fixed, the total error scales as

$$\epsilon_{\text{tot}} = \epsilon_b + \epsilon_s + \epsilon_{\Delta t} = \mathcal{O}(\Delta t); \tag{C23}$$

thus, the final fidelity between the expected result and the outcome of the simulator is

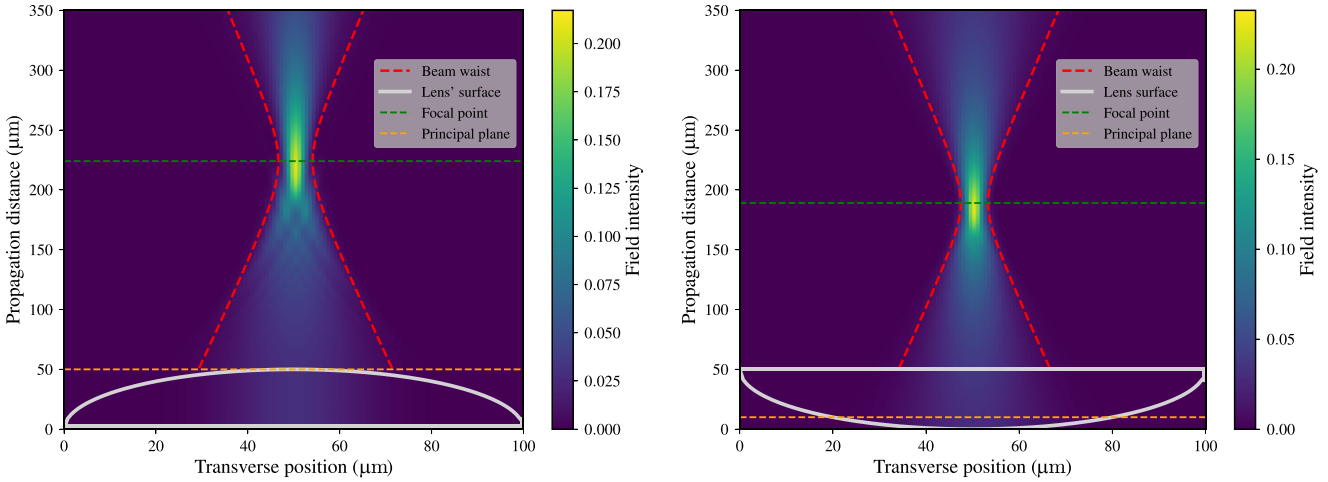
$$F = 1 - \mathcal{O}(\epsilon_{\text{tot}}^2) = 1 - \mathcal{O}(\Delta t^2), \tag{C24}$$

which is the relation used to analyze the performance of the simulator in Sec. IV. We conclude this Appendix by mentioning that the error analysis done here considers the worst-case scenario using the spectral norm. It is possible that the actual final error is much smaller than the worst case in specific applications.

APPENDIX D: SIMULATION OF VARIOUS CASES AND ORIENTATIONS OF LENSES

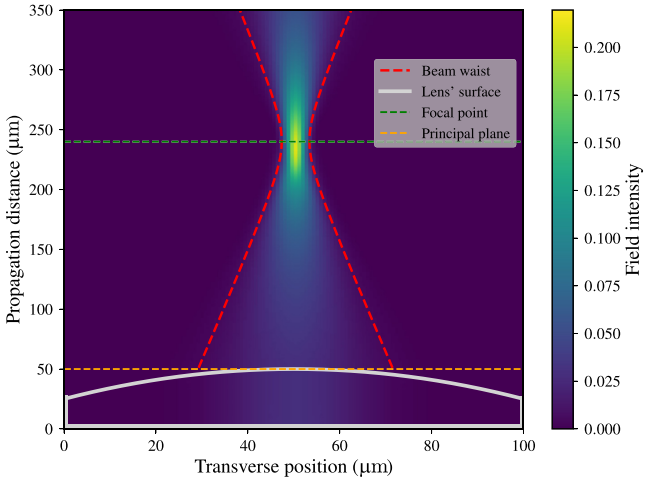
In this Appendix, we present additional lens simulations using the presented quantum algorithm to showcase the flexibility of the block-encoding technique for simulating different optical experiments. In wave optics [36], it is known that the focusing performance of a lens with a finite thickness depends on its shape and geometry, namely, the radii of curvature of its surfaces and the orientation of the lens.

To see these effects, we simulated four different cases. We started with a spherical lens in the orientation where the beam hits the planar surface first [Fig. 10(a)] and then simulated the same lens but in the reverse orientation [Fig. 10(b)]. Instead of the lens with a spherical surface, we then approximated the curved surface by a parabolic curvature and repeated the simulation in both orientations [Figs. 10(c) and 10(d)]. The parabolic approximation is chosen such that the curvature on the optical axis is the same for the spherical and parabolic

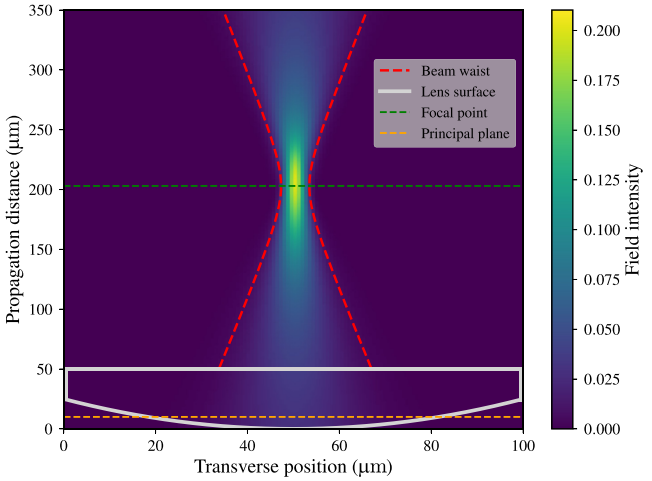


(a) Spherical lens, "bad" orientation

(b) Spherical lens, "good" orientation



(c) Parabolic lens, "bad" orientation



(d) Parabolic lens, "good" orientation

FIG. 10. Simulation results of focusing a Gaussian beam incident on various lenses. Panel (a) shows the same simulation as Fig. 6, repeated here for comparison. The simulation parameters are the wavelength, initial Gaussian beam waist, refractive index, radius of curvature (on the optical axis), and number of qubits, respectively, $\lambda = 1\,\mu\text{m}$, $w_0 = 25\,\mu\text{m}$, $n = 1.25$, $R = 50\,\mu\text{m}$, and $n_q = 7$.

lenses, which means that the parabolic lens generally has a smaller curvature at other points of its surface compared to the spherical lens. The general scheme of the simulations is exactly the same as that described in Sec. IV, and the only change is the different phases applied to the incident beam by the lens based on its shape and orientation.

Several effects are expected from and seen in these simulations. It is known [36] that the orientation of the lens where the beam hits the curved surface first is considered the "better" orientation because the focusing power is distributed over both surfaces, whereas in the other orientation, it is only the curved surface that contributes to the refraction. This leads to two effects. First, the "bad" orientation potentially leads to more prominent aberration effects because it amplifies the spherical aberration relatively more. Second, there will be a shift of the focal point because the principal plane (the approximate effective position of the focusing plane when the lens is treated as a thin lens) is located differently depending on the orientation. Comparing the spherical and parabolic lenses in a fixed orientation, we can also see that the focal distance is effectively larger in the parabolic case because of the smaller effective curvature of the parabolic approximation. A parabolic lens also reduces the spherical aberration for the Gaussian beam input. This shows that the simulation protocol presented in this work offers an intuitive way to run various optical simulations using a quantum computer.

[1] A. J. Daley, I. Bloch, C. Kokail, S. Flannigan, N. Pearson, M. Troyer, and P. Zoller, Practical quantum advantage in quantum simulation, Nature (London) **607**, 667 (2022).

[2] E. Altman, K. R. Brown, G. Carleo, L. D. Carr, E. Demler, C. Chin, B. DeMarco, S. E. Economou, M. A. Eriksson, K.-M. C. Fu, *et al.*, Quantum simulators: Architectures and opportunities, PRX Quantum **2**, 017003 (2021).

[3] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. L. Pereira, M. Razavi, J. S. Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden, Advances in quantum cryptography, Adv. Opt. Photonics **12**, 1012 (2020).

[4] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, Challenges and opportunities in quantum machine learning, Nat. Comput. Sci. **2**, 567 (2022).

[5] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, G. Salton, S. Wang, and F. G. S. L. Brandão, *Quantum Algorithms: A Survey of Applications and End-to-End Complexities* (Cambridge University Press, Cambridge, 2025).

[6] R. Santagati, A. Aspuru-Guzik, R. Babbush, M. Degroote, L. González, E. Kyoseva, N. Moll, M. Oppel, R. M. Parrish, N. C. Rubin, M. Streif, C. S. Tautermann, H. Weiss, N. Wiebe, and C. Utschig-Utschig, Drug design on quantum computers, Nat. Phys. **20**, 549 (2024).

[7] D. Herman, C. Googin, X. Liu, Y. Sun, A. Galda, I. Safro, M. Pistoia, and Y. Alexeev, Quantum computing for finance, Nat. Rev. Phys. **5**, 450 (2023).

[8] A. Di Meglio, K. Jansen, I. Tavernelli, C. Alexandrou, S. Arunachalam, C. W. Bauer, K. Borras, S. Carrazza, A. Crippa, V. Croft, *et al.*, Quantum computing for high-energy physics: State of the art and challenges, PRX Quantum **5**, 037001 (2024).

[9] G. Mazzola, Quantum computing for chemistry and physics applications from a Monte Carlo perspective, J. Chem. Phys. **160**, 010901 (2024).

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

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

[12] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, Rev. Mod. Phys. **86**, 153 (2014).

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

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

[15] P. Webster, M. Vasmer, T. R. Scruby, and S. D. Bartlett, Universal fault-tolerant quantum computing with stabilizer codes, Phys. Rev. Res. **4**, 013092 (2022).

[16] R. Toshio, Y. Akahoshi, J. Fujisaki, H. Oshima, S. Sato, and K. Fujii, Practical quantum advantage on partially fault-tolerant quantum computer, Phys. Rev. X **15**, 021057 (2025).

[17] F. Battistel, C. Chamberland, K. Johar, R. W. J. Overwater, F. Sebastiano, L. Skoric, Y. Ueno, and M. Usman, Real-time decoding for fault-tolerant quantum computing: Progress, challenges and outlook, Nano Futures **7**, 032003 (2023).

[18] E. T. Campbell, B. M. Terhal, and C. Vuillot, Roads towards fault-tolerant universal quantum computation, Nature (London) **549**, 172 (2017).

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

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

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

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

[23] J. D. Schmidt, J. A. Tellez, and G. J. Gbur, Semi-analytic simulation of optical wave propagation through turbulence, Appl. Opt. **61**, 9439 (2022).

[24] R. Heintzmann, L. Loetgering, and F. Wechsler, Scalable angular spectrum propagation, Optica **10**, 1407 (2023).

[25] M. Wende, J. Drozella, A. Toulouse, and A. M. Herkommer, Fast vector wave optical simulation methods for application on 3D-printed microoptics, J. Opt. Microsyst. **4**, 024501 (2024).

[26] Y. Liu, F. Guo, Z. Zhang, and R. Wu, Wave optical propagation in realistic lens systems through multi-slice decomposition with phase compensation, Appl. Opt. **63**, F18 (2024).

[27] Y. Sung, B. Nelson, and R. Gupta, Realistic wave-optics simulation of X-ray dark-field imaging at a human scale, Sci. Rep. **15**, 26748 (2025).

[28] J. Lu, D. Qu, J. Qu, R. Fong, G. H. Ahn, and J. Vuckovic, A systolic update scheme to overcome memory bandwidth limitations in GPU-accelerated FDTD simulations, arXiv:2502.20610.

[29] Z. Wu, Y. You, X. Zhou, and F. Zhang, Accelerated time-domain simulation of complex photonic structures with a data-aware Fourier neural operator, Opt. Commun. **604**, 132838 (2026).

[30] R. Shams and P. Sadeghi, On optimization of finite-difference time-domain (FDTD) computation on heterogeneous and GPU clusters, J. Parallel Distrib. Comput. **71**, 584 (2011).

[31] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, Preconditioned quantum linear system algorithm, Phys. Rev. Lett. **110**, 250504 (2013).

[32] C. Cholsuk, S. Davani, L. O. Conlon, T. Vogl, and F. Eilenberger, Efficient light propagation algorithm using quantum computers, Phys. Scr. **99**, 045110 (2024).

[33] J. S. Kottmann, M. Krenn, T. H. Kyaw, S. Alperin-Lea, and A. Aspuru-Guzik, Quantum computer-aided design of quantum optics hardware, Quantum Sci. Technol. **6**, 035010 (2021).

[34] J. M. Burzler, S. Hughes, and B. S. Wherrett, Split-step Fourier methods applied to model nonlinear refractive effects in optically thick media, Appl. Phys. B **62**, 389 (1996).

[35] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. (Cambridge University Press, Cambridge, New York, 2010).

[36] B. E. A. Saleh and M. C. Teich, *Fundamentals of Photonics*, 1st ed. (Wiley, New York, 1991).

[37] H. F. Trotter, On the product of semi-groups of operators, Proc. Am. Math. Soc. **10**, 545 (1959).

[38] S.-J. Ran, Encoding of matrix product states into quantum circuits of one- and two-qubit gates, Phys. Rev. A **101**, 032310 (2020).

[39] J. Iaconis, S. Johri, and E. Y. Zhu, Quantum state preparation of normal distributions using matrix product states, npj Quantum Inf. **10**, 15 (2024).

[40] M. Rosenkranz, E. Brunner, G. Marin-Sanchez, N. Fitzpatrick, S. Dilkes, Y. Tang, Y. Kikuchi, and M. Benedetti, Quantum state preparation for multivariate functions, Quantum **9**, 1703 (2025).

[41] M. Moosa, T. W. Watts, Y. Chen, A. Sarma, and P. L. McMahon, Linear-depth quantum circuits for loading Fourier approximations of arbitrary functions, Quantum Sci. Technol. **9**, 015002 (2024).

[42] X. Yang, X. Nie, Y. Ji, T. Xin, D. Lu, and J. Li, Improved quantum computing with higher-order Trotter decomposition, Phys. Rev. A **106**, 042401 (2022).

[43] M. Kieferová, A. Scherer, and D. W. Berry, Simulating the dynamics of time-dependent Hamiltonians with a truncated Dyson series, Phys. Rev. A **99**, 042314 (2019).

[44] M. Schuld and F. Petruccione, Representing data on a quantum computer, in *Machine Learning with Quantum Computers*, edited by M. Schuld and F. Petruccione (Springer International Publishing, Cham, 2021), pp. 147–176.

[45] A. Gilyén, S. Arunachalam, and N. Wiebe, Optimizing quantum optimization algorithms via faster quantum gradient computation, in *Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2019), pp. 1425–1444.

[46] S. Davani, Blackwild/qiu: Code repository snapshot 2026-03-30, Zenodo, 2026, https://doi.org/10.5281/zenodo.19333736.

[47] S. Davani, M. Gärttner, and F. Eilenberger, Dataset of the publication: Quantum simulation of wave optics in weakly inhomogeneous media using block-encoding, Zenodo, 2026, http://doi.org/10.5281/zenodo.19333783.

[48] S. Blanes, F. Casas, J. A. Oteo, and J. Ros, The Magnus expansion and some of its applications, Phys. Rep. **470**, 151 (2009).

[49] M. Hochbruck and C. Lubich, On Magnus integrators for time-dependent Schrödinger equations, SIAM J. Numer. Anal. **41**, 945 (2003).

[50] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Theory of Trotter error with commutator scaling, Phys. Rev. X **11**, 011020 (2021).