Quantum algorithm for solving the advection equation using Hamiltonian simulation

Peter Brearley[*] and Sylvain Laizet

Department of Aeronautics, Imperial College London, London SW7 2BX, United Kingdom

(Received 18 December 2023; revised 25 April 2024; accepted 17 June 2024; published 9 July 2024)

A quantum algorithm for solving the advection equation by embedding the discrete time-marching operator into Hamiltonian simulations is presented. One-dimensional advection can be simulated directly since the central finite-difference operator for first-order derivatives is anti-Hermitian. Here this is extended to industrially relevant multidimensional flows with realistic boundary conditions and arbitrary finite-difference stencils. A single copy of the initial quantum state is required and the circuit depth grows linearly with the required number of time steps, the sparsity of the time-marching operator, and the inverse of the allowable error. State-vector simulations of a scalar transported in a two-dimensional channel flow and lid-driven cavity configuration are presented as a proof of concept of the proposed approach.

DOI: 10.1103/PhysRevA.110.012430

I. INTRODUCTION

Quantum computing is expected to bring a profound shift in our computational capability. Among the most promising applications is solving large-scale partial differential equations (PDEs) more efficiently than classical computers. Partial differential equations are ubiquitous across science and engineering and solving them currently occupies the majority of the world's high-performance computing resources. The development of efficient quantum algorithms is therefore of immense value and has attracted a large interdisciplinary research community, spurred on by the continuing advancements in quantum hardware [1,2].

The advection equation is a foundational linear PDE spanning multiple industries as it describes the transport of a scalar quantity in advection-dominated flows. It is given by

$$\frac{\partial \phi}{\partial t} + u_j \frac{\partial \phi}{\partial x_j} = 0, \tag{1}$$

where ϕ represents the scalar field (e.g., temperature or concentration) and u_j is the jth component of the advective velocity vector, where repeated indices invoke summation over all spatial dimensions. Applications include modeling the vast oceanic [3], atmospheric [4], and geological flows [5] used in climate studies, drug delivery systems in biomathematics [6], and heat exchangers for cooling oil refineries, chemical processing plants, and power stations [7]. When the advection equation is discretized in space using the finite-difference method, the homogeneous ordinary differential equation (ODE)

$$\frac{d\vec{\phi}}{dt} = M\vec{\phi} \tag{2}$$

is produced. The original motivation for quantum computing was to simulate quantum dynamics governed by the Schrödinger equation [8], which can be expressed in the form of Eq. (2) for an anti-Hermitian coefficient matrix M. The central finite-difference operator for first-order derivatives is anti-Hermitian, so for one-dimensional flows with periodic boundary conditions, advection can be simulated on a quantum computer using the unitary operator e^{Mt}. This implies that one-dimensional advection can be considered equivalent to quantum dynamics [9]. Indeed, advection in a divergence-free velocity field is inherently a norm-preserving process, making it well suited for simulation on a quantum computer. However, for industrially relevant multidimensional flows with realistic boundary conditions or noncentered finite-difference stencils, M ceases to be anti-Hermitian, so a different approach must be taken.

Quantum algorithms for solving PDEs can be delineated into two categories: fully quantum algorithms that implement quantum circuits to evolve the quantum state as described by the PDE of interest and quantum-classical hybrid algorithms where a quantum computer is used for a specific task in a larger, classical computation. Fully quantum approaches generally excel at solving linear PDEs because quantum operators act linearly on quantum superpositions, allowing algorithms based on the finite-difference method (FDM) [10–13], the finite-element method (FEM) [14,15], and spectral methods [16] to be effectively represented quantum mechanically. Encoding the solution from $N = 2^n$ grid points within the amplitudes α_j of an n-qubit quantum state $|\psi\rangle = \sum_{j=1}^{N} \alpha_j |j\rangle$ leads to an exponentially growing capacity to store information and an inherent quantum parallelism when processing it. Amplitude encoding does not allow for the inspection of the full solution as with classical methods, but rather the extraction of global statistics into the limited output space. This may

[*] Contact author: p.brearley@imperial.ac.uk

be adequate depending on the context and make previously intractable problems tractable. Fully quantum approaches are not limited to linear PDEs as techniques have been proposed based on the derivation of the nonlinear Schrödinger equation using mean-field techniques [17] and Carleman linearization [18] to tackle nonlinear PDEs, but these are generally limited to weakly nonlinear interactions. On the other hand, variational quantum algorithms (VQAs) [19] for solving optimization problems have been used as the basis for hybrid algorithms that have demonstrated a greater capability of tackling nonlinear PDEs [20–22]. Kyriienko *et al.* [21] used a machine learning strategy where differentiable quantum circuits were trained to solve nonlinear differential equations. Jaksch *et al.* [22] extended a quantum algorithm for solving nonlinear problems [20] to fluid dynamics, evaluating cost functions from matrix product state representations of the flow [23] to obtain a polynomial upper bound on the depth of the variational network. The VQA-based algorithms are of interest because of their potential to operate on near-term hardware [19], though a definitive quantum advantage is yet to be demonstrated.

Most quantum algorithms for solving linear PDEs have a quantum linear systems algorithm (QLSA) at their core [10,12–16,24–28] such as the Harrow-Hassidim-Lloyd (HHL) algorithm [29] or further optimizations thereof [30,31]. Clader *et al.* [14] developed a quantum algorithm using a QLSA to implement the FEM for solving Maxwell's equations, and this was further clarified and developed by Montanaro and Pallister [15]. Cao *et al.* [10], Wang *et al.* [12], and Childs *et al.* [13], in their respective studies, optimized a quantum algorithm based on the FDM to solve the Poisson equation by expressing the PDE as a system of linear equations and then solving with a QLSA. Algorithms for solving homogeneous time-independent ODEs [24,25,27,28] in the form of Eq. (2) can be applied to the spatially discretized advection equation. Berry [24] proposed an algorithm using a QLSA with a linear multistep method, where the error per time step is a high power of the time-step size. A different approach was later taken by Berry *et al.* [25] by encoding a truncated Taylor series expansion of e^{Mt} in a linear system of equations, achieving an exponentially improved dependence on precision. This work was later extended by Krovi [27] to nondiagonalizable and singular matrices while also achieving an exponential improvement over Berry *et al.* [25] for diagonalizable matrices with a bounded value of $\|e^{Mt}\|$. Berry and Costa [28] proposed encoding a truncated Dyson series into a system of linear equations, achieving a scaling that is linear in the evolution time T and the norm $\|M\|$ and polylogarithmic in the allowable error ϵ. The primary disadvantage of algorithms that employ a QLSA is that they require a large number of state initialization queries that grow linearly with the condition number of the matrix [29], which itself grows linearly with the desired simulation time [32]. This introduces a potentially prohibitive computational overhead given the challenges of state preparation. In addition, the runtime of the HHL algorithm has a dependence on $1/\epsilon$ arising from the quantum phase estimation (QPE) step, which may become prohibitive for algorithms requiring repeated applications. Subsequent works [30,31] have improved the ϵ dependence by applying the inverse of the matrix as a linear combination of unitaries (LCU) [30] or by the quantum adiabatic theorem [31], though with significantly more involved implementations.

Quantum algorithms for solving PDEs that do not depend on a QLSA have also been developed. Costa *et al.* [11] put forward a quantum algorithm that evolves a quantum state according to the wave equation using Hamiltonian simulation without the need for a QLSA, aside from in the proposed generation of the initial conditions. A practical implementation of this algorithm was then developed and analyzed by Suau *et al.* [33], confirming that the gate requirements agreed with the theoretical complexity. In a different study, Budinski [34] proposed a quantum algorithm for the advection-diffusion equation centered around the lattice Boltzmann method (LBM) that tracks the evolution of particle distribution functions on a grid. The main challenge of implementing the nonunitary collision step of the LBM was achieved with the LCU method. Simulating ODEs in the form of Eq. (2) as Hamiltonian simulations has received much recent attention [32,35–38]. An *et al.* [38] expressed the evolution as a linear combination of Hamiltonian simulation (LCHS) problems, which is a special case of the LCU method. The algorithm was later improved upon [32] by further generalizing the method and leading to the discovery of faster-decaying coefficients in the weighted sum, improving the ϵ dependence. Jin *et al.* [35–37] provided an alternative viewpoint on this problem by introducing a Schrödingerization method that maps linear PDEs to a higher-dimensional system of Schrödinger equations and then solving with Hamiltonian simulation. Both the Schrödingerization [35–37] and the LCHS [32,38] algorithms assume the Hermitian part of M to be negative semidefinite, applicable to problems and numerical schemes that maintain or dampen the solution, but do not amplify it. This applies to upwind and central finite-difference schemes for discretizing the advection equation, but not downwind schemes. A time-marching algorithm for the ODE problem was proposed by Fang *et al.* [39] by explicitly integrating the PDE over short time steps, as is a common practice in the classical numerical solution of differential equations. This operator is not unitary and so has a probability of failure in a block-encoding strategy, ordinarily leading to an exponentially decaying success probability in the simulation time T. This was overcome by applying a uniform singular value amplification at each time step, resulting in a runtime with a quadratic dependence on T. The absence of a QLSA in these methods generally leads to improved state preparation costs [32] and favorable polylogarithmic scaling in the allowable error per time step using sparse Hamiltonian simulation [40] or LCU [30] algorithms. Other methods of quantum matrix multiplication with the potential to be applied to solving PDEs using explicit time advancement were compared by Shao [41] based on the SWAP test, singular value estimation, and HHL algorithms. However, all of these methods utilize QPE as a subroutine, thus limiting their ϵ dependence and the number of state initialization queries required.

The algorithm presented here uses an explicit time-marching strategy for solving the advection equation by embedding numerical integrators into a series of Hamiltonian simulations. The algorithm achieves linear scaling in the required number of time steps N_T, requires a single copy of the initial quantum state, and applies to various boundary conditions with arbitrary finite-difference stencils. The mathematical description of the algorithm is provided next, followed by analyses of the errors and complexity in Secs. III and IV, respectively. State-vector simulations of a two-dimensional laminar channel flow and a lid-driven cavity problem are provided and analyzed in Sec. V. The paper is finalized with concluding remarks in Sec. VI.

II. ALGORITHM

The first step is to discretize the advection equation in space and time using the FDM. Central, forward, or backward schemes of any order of accuracy can be chosen for the spatial derivatives, and these can vary throughout the domain, e.g., by reducing the order of accuracy and transitioning to a one-sided scheme towards a wall. A second-order central scheme for a one-dimensional problem will be chosen to describe the algorithm and analyze its baseline properties, with different schemes being demonstrated later in the paper. When combined with forward Euler discretization in time, the advection equation in one dimension for a constant velocity u becomes

$$\frac{\phi_m^{t+1}-\phi_m^t}{\Delta t}+u\frac{\phi_{m+1}^t-\phi_{m-1}^t}{2\Delta x}=0 \tag{3}$$

for spatial grid point m and temporal location t. When solved classically, the forward-time central-space scheme in Eq. (3) is unstable for hyperbolic PDEs such as the advection equation, requiring upwind schemes for conditional stability [42]. The quantum representation of such forward-time schemes in the present algorithm is stable for arbitrary finite-difference stencils, as will be demonstrated later in this section. Equation (3) can be solved for ϕ_m^{t+1}, obtaining an equation to advance the solution in time. A vector $\vec{\phi}_t=[\phi_0^t,\phi_1^t,\dots,\phi_{N-1}^t]$ can be constructed from ϕ to write Eq. (3) as a matrix transformation

$$\vec{\phi}_{t+1}=A\vec{\phi}_t. \tag{4}$$

For the described one-dimensional advection equation discretized with a second-order central FDM, the matrix A takes the form

$$A=\begin{bmatrix} 1 & -\frac{r}{2} & & 0 & \frac{r}{2} \\ \frac{r}{2} & \ddots & \ddots & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & \ddots & \ddots & -\frac{r}{2} \\ -\frac{r}{2} & 0 & & \frac{r}{2} & 1 \end{bmatrix} \tag{5}$$

when considering simple periodic boundary treatment where the first and last grid points are adjacent, leading to entries in the top right and bottom left corners. The stability parameter $r=u\Delta t/\Delta x$ is the Courant-Friedrichs-Lewy (CFL) number and is related to the condition number κ. To ensure numerical stability, r must not exceed 1 [43], though in practice and especially for explicit schemes, much lower values are required. In D-dimensional space with k-order-accurate spatial discretization, the sparsity of the matrix $s=1+Dk$ with FDM coefficients determined by k.

While divergence-free advection is a norm-preserving process, the truncation errors from the discretization procedure in Eq. (3) lead to the matrix A being nonunitary, i.e., $A^\dagger A\neq I$ and $AA^\dagger\neq I$. Therefore, to enact A on a quantum state $|\phi_t\rangle$, the nonunitary Hamiltonian embedding procedure described by Gingrich and Williams [44] is used. A Hamiltonian H is constructed from A in the form

$$H=\begin{bmatrix} 0 & iA \\ -iA^\dagger & 0 \end{bmatrix}, \tag{6}$$

where i is the imaginary unit. A quantum state is evolved according to H by the unitary operator

$$\begin{aligned}\Omega&=\exp(-iH\theta)\\&=\exp\begin{bmatrix} 0 & A\theta \\ -A^\dagger\theta & 0 \end{bmatrix},\end{aligned} \tag{7}$$

where the symbol θ is the Hamiltonian evolution time for a single time step. The time step size is encoded in the Hamiltonian, so θ affects the accuracy of a time step and its probability of success rather than the evolution time itself. The exponential function produces a block matrix with the structure [44]

$$\begin{aligned}\exp\begin{bmatrix} 0 & X \\ Y & 0 \end{bmatrix}&=\begin{bmatrix} \cosh(\sqrt{XY}) & X\frac{\sinh(\sqrt{YX})}{\sqrt{YX}} \\ Y\frac{\sinh(\sqrt{XY})}{\sqrt{XY}} & \cosh(\sqrt{YX}) \end{bmatrix}\\&=\begin{bmatrix} \cos(\theta\sqrt{AA^\dagger}) & A\frac{\sin(\theta\sqrt{A^\dagger A})}{\sqrt{A^\dagger A}} \\ -A^\dagger\frac{\sin(\theta\sqrt{AA^\dagger})}{\sqrt{AA^\dagger}} & \cos(\theta\sqrt{A^\dagger A}) \end{bmatrix},\end{aligned} \tag{8}$$

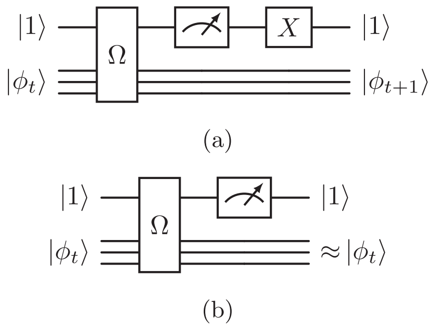
where the top right and top left blocks of Eq. (8) are termed $\tilde{A}$ and $\tilde{I}$, respectively. The $\tilde{A}$ matrix is closely proportional to A for small values of θ or when $\sqrt{A^\dagger A}\approx I$ since $\sin(I\theta)=I\sin(\theta)$, leading to $\tilde{A}\approx A\sin(\theta)$. Furthermore, the $\tilde{I}$ matrix is closely proportional to I for small values of θ or when $\sqrt{AA^\dagger}\approx I$, leading to $\tilde{I}\approx I\cos(\theta)$. For the advection equation evolution operator in Eq. (5), $\sqrt{A^\dagger A}$ and $\sqrt{AA^\dagger}$ can be approximated by $I+O(r^2)$ for small values of r. Equation (8) also reveals that the singular values $\sigma_i(\tilde{A})=\sin[\sigma_i(A)\theta]$ are bounded by 1, stabilizing the scheme regardless of the finite-difference stencil. Applying Ω to a solution register $|\phi_t\rangle$ supplemented by an ancilla qubit initialized as $|1\rangle$ produces the state

$$\Omega|1\rangle|\phi_t\rangle=\Omega\begin{bmatrix} 0 \\ |\phi_t\rangle \end{bmatrix}=\begin{bmatrix} \tilde{A}|\phi_t\rangle \\ \tilde{I}|\phi_t\rangle \end{bmatrix}\approx\begin{bmatrix} |\phi_{t+1}\rangle \\ |\phi_t\rangle \end{bmatrix}. \tag{9}$$

Postselecting the ancilla qubit in the state $|0\rangle$ collapses the solution register to $|\phi_{t+1}\rangle=\tilde{A}|\phi_t\rangle$, and if postselection fails, the state collapses to $\tilde{I}|\phi_t\rangle\approx|\phi_t\rangle$, allowing the state to be reused for further computation. The quantum circuit representations of a successful and unsuccessful time step are shown in Fig. 1.

III. ERROR ANALYSIS

An expression for the error upper bound can be derived by considering the errors at each time step associated with the application of $\tilde{A}$ and $\tilde{I}$ and combining them with the number of time steps required and the probability of postselection success. First, the errors per time step are derived in Sec. III A, followed by the probability of a successful time step in Sec. III B. These are combined to produce the overall



(a)

(b)

FIG. 1. Quantum circuit for (a) a successful time step measuring $|0\rangle$ and (b) an unsuccessful time step measuring $|1\rangle$.

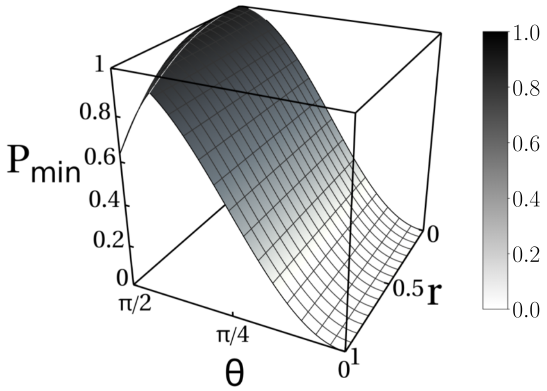


FIG. 2. Surface plot of the minimum probability of a successful time step defined in Eq. (16).

error bound in Sec. III C and compared to the errors when applied to a different PDE, the heat equation, in Sec. III D.

A. Error per time step

The matrix A in Eq. (5) is a Toeplitz matrix and so can be defined in terms of its diagonals, where the main diagonal $d_0 = 1$, the superdiagonal $d_1 = -r/2$, and the subdiagonal $d_{-1} = r/2$. Similarly, the corresponding terms in $\tilde{A}$ can be evaluated as a function of r and θ. Considering a 4×4 matrix to avoid unnecessary negligible terms, $\tilde{A}$ can be expressed as

$$\begin{aligned}\tilde{A} &= \frac{1}{2}\left(\sin(\theta) + \frac{\sin(\theta\sqrt{r^2+1})}{\sqrt{r^2+1}}\right) \quad \text{for } d_0\\ &= -\frac{r\sin(\theta\sqrt{r^2+1})}{2\sqrt{r^2+1}} \quad \text{for } d_1, -d_{-1}\\ &= \frac{1}{2}\left(\sin(\theta) - \frac{\sin(\theta\sqrt{r^2+1})}{\sqrt{r^2+1}}\right) \quad \text{for } d_2, d_{-2}. \qquad (10)\end{aligned}$$

The similarity between the matrices in Eqs. (5) and (10) can be compared by constructing an error matrix that quantifies the error of the terms relative to the main diagonal. The property that Eq. (10) must satisfy is $d_1, -d_{-1} = -rd_0/2$ with other elements equaling 0, i.e., the relative proportions of the matrix must be consistent. If Eq. (5) is scaled by $d_0^{[\text{Eq. (10)}]}$ to yield equal diagonal elements with Eq. (10), then subtracting Eq. (10) from the scaled Eq. (5) gives the error matrix E_A, given by

$$\begin{aligned}E_A &= 0 \quad \text{for } d_0\\ &= -\frac{r}{4}\left(\sin(\theta) - \frac{\sin(\theta\sqrt{r^2+1})}{\sqrt{r^2+1}}\right) \quad \text{for } d_1, -d_{-1}\\ &= -\frac{1}{2}\left(\sin(\theta) - \frac{\sin(\theta\sqrt{r^2+1})}{\sqrt{r^2+1}}\right) \quad \text{for } d_2, d_{-2}. \qquad (11)\end{aligned}$$

The maximum error for the application of $\tilde{A}$ can then be defined by taking the spectral norm of Eq. (11),

$$\|E_A\| = \tfrac{1}{2}[\sin(\theta)\sqrt{r^2+1} - \sin(\theta\sqrt{r^2+1})]. \qquad (12)$$

Following the same procedure, $\tilde{I}$ can be written as

$$\begin{aligned}\tilde{I} &= \tfrac{1}{2}[\cos(\theta) + \cos(\theta\sqrt{r^2+1})] \quad \text{for } d_0\\ &= 0 \quad \text{for } d_{-1}, -d_1\\ &= \tfrac{1}{2}[\cos(\theta) - \cos(\theta\sqrt{r^2+1})] \quad \text{for } d_2, d_{-2} \qquad (13)\end{aligned}$$

and an error matrix can be constructed that quantifies error relative to the identity matrix:

$$\begin{aligned}E_I &= 0 \quad \text{for } d_0\\ &= 0 \quad \text{for } d_1, d_{-1}\\ &= -\tfrac{1}{2}[\cos(\theta) - \cos(\theta\sqrt{r^2+1})] \quad \text{for } d_2, d_{-2}. \qquad (14)\end{aligned}$$

Taking the spectral norm of the error matrix gives

$$\|E_I\| = \tfrac{1}{2}[\cos(\theta) - \cos(\theta\sqrt{r^2+1})], \qquad (15)$$

which represents the maximum error for a failed postselection.

B. Time-step success probability

The probability of a successful time step $P = \|\tilde{A}|\phi\rangle\|^2$ can be studied by analyzing the contribution of $\tilde{I}$ to the state. The square of the spectral norm $\|\tilde{I}\|^2$ corresponds to the largest action of $\tilde{I}$ on a state vector squared and so can be used to find the worst-case probability of successful measurement $P_{\min} = 1 - \|\tilde{I}\|^2$. Using the definition of $\tilde{I}$ in Eq. (13), $P_{\min}$ is defined as

$$P_{\min} = \begin{cases}\sin^2(\theta) & \text{when } 0 < \theta \leqslant \frac{\pi}{1+\sqrt{r^2+1}}\\ \sin^2(\theta\sqrt{r^2+1}) & \text{when } \frac{\pi}{1+\sqrt{r^2+1}} < \theta \leqslant \frac{\pi}{2},\end{cases} \qquad (16)$$

which is visualized in Fig. 2. The figure shows that $P_{\min}$ is optimal for small values of r and when $\theta = \pi/(1 + \sqrt{r^2+1})$, approaching unity as r approaches zero. Using this value of θ for a typical case where $r = 0.1$, the minimum probability $P_{\min} = 99.9985\%$, corresponding to a worst case of 67 000 successful time steps per failed time step on average. It is shown in Sec. V that for a practical configuration, the probability of postselection success is mostly represented by $\sin^2(\theta)$, with the contribution from $\sin^2(\theta\sqrt{r^2+1})$ having a reduced role leading to $\theta = \pi/2$ being optimal in practice. For example, $r = 0.1$ and $\theta = \pi/2$ resulted in approximately 1×10^9 successful time steps for every failed time step.

C. Overall error bound

The overall error bound for the algorithm can be evaluated using $\epsilon = N_T\|E_A\| + N_F\|E_I\|$, where N_T is the required number of successful time steps and N_F is the expected number of failed time steps. Since the number of time steps required is inversely proportional to the time step, $N_T = T/r$, where T is the simulation time in nondimensional form and $N_F = N_T/P_{\min} - N_T$. Combining these expressions, the error per the simulation time evaluates to

$$\frac{\epsilon}{T} = \begin{cases} \frac{1}{2r}\{[\cos(\theta) - \cos(\theta\sqrt{r^2+1})]\cot^2(\theta) + \sin(\theta)\sqrt{r^2+1} - \sin(\theta\sqrt{r^2+1})\} & \text{when } 0 < \theta \leqslant \frac{\pi}{1+\sqrt{r^2+1}} \\ \frac{1}{2r}\{[\cos(\theta) - \cos(\theta\sqrt{r^2+1})]\cot^2(\theta\sqrt{r^2+1}) + \sin(\theta)\sqrt{r^2+1} - \sin(\theta\sqrt{r^2+1})\} & \text{when } \frac{\pi}{1+\sqrt{r^2+1}} < \theta \leqslant \frac{\pi}{2}. \end{cases} \tag{17}$$

This bound assumes that the number of failed time steps is relatively close to the expected value, which is reasonable as N_T becomes large. This expression is visualized in Fig. 3(a), which reveals that the overall error is mostly insensitive to the value of θ and grows linearly with r. This indicates that for the low values of r required for numerical stability, $\theta = \pi/2$ is the most efficient configuration as it minimizes the required circuit depth without incurring substantial additional errors. The linear growth of error in the quantum matrix representation indicates that the algorithm does not worsen the error complexity from the classical Euler method that underpins the algorithm and means there is little algorithmic benefit in pursuing higher-order time integrators.

D. Comparison with the heat equation

To demonstrate the advantageous properties of Eq. (17) for the advection equation, the error will be compared against the algorithm applied to a different PDE, the heat equation. The heat equation features a second derivative term on the right-hand side

$$\frac{\partial\phi}{\partial t} = D\frac{\partial^2\phi}{\partial x_j \partial x_j}, \tag{18}$$

where D is the diffusivity. Following the same discretization procedure for a one-dimensional problem, the matrix A takes the form $[d_{-1}, d_0, d_1] = [r_h, 1 - 2r_h, r_h]$ for the internal grid points, where $r_h = D\Delta t/(\Delta x)^2$ is the stability parameter with a theoretical maximum value of 0.5. The error bound for the heat equation evaluates to

$$\frac{\epsilon}{T} = \begin{cases} \frac{1}{2r_h}\left[\frac{(8r_h-3)\sin(\theta) + \sin(\theta-4r_h\theta) + 2\sin(\theta-2r_h\theta)}{2-4r_h} + |\sin(\theta-3r_h\theta) + 3\sin(\theta-r_h\theta)|\cot^2(\theta-4r_h\theta)\sin(r_h\theta)\right] & \text{when } 0 < r_h \leqslant \frac{1}{3} \\ \frac{1}{2r_h}\left[\frac{(1-4r_h)\sin(\theta) + (4r_h-3)\sin(\theta-4r_h\theta) + (2-8r_h)\sin(\theta-2r_h\theta)}{2-4r_h} + |\sin(\theta-3r_h\theta) + 3\sin(\theta-r_h\theta)|\cot^2(\theta-2r_h\theta)\sin(r_h\theta)\right] & \text{when } \frac{1}{3} < r_h \leqslant \frac{1}{2}, \end{cases} \tag{19}$$

which is visualized in Fig. 3(b). The expression and the figure reveal that the error asymptotically approaches infinity for $r_h \rightarrow 0.25$ and $r_h \rightarrow 0.5$. Furthermore, as $r_h \rightarrow 0$, the error bound does not approach zero as occurs for the advection equation but rather approaches approximately 2, thereby making it impossible to diminish the error by increasing the circuit depth. A local minimum value for the error occurs when $r_h = \frac{1}{3}$, where ϵ/T reduces to approximately 6 for $\theta = \pi/2$. The errors for the advection equation are analogous to classical computations since reducing r linearly reduces the error, with the error approaching zero as $r \rightarrow 0$. This is not the case when applied to the heat equation where the error is a convoluted discontinuous function of r_h without a clear trend. Even using the optimal value of $r_h = \frac{1}{3}$ or $r_h < 0.01$, the errors are still too large for useful computation, indicating that the heat equation evolution operator cannot be efficiently represented in this manner. Therefore, the algorithm cannot be considered a sufficiently general PDE solver.

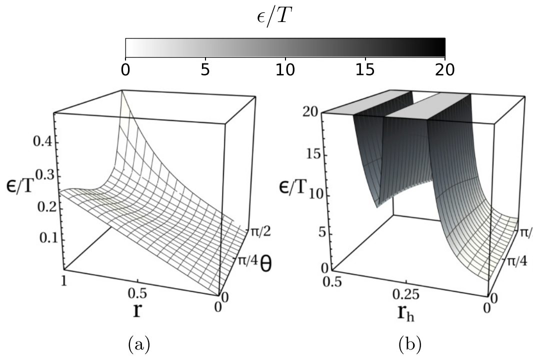
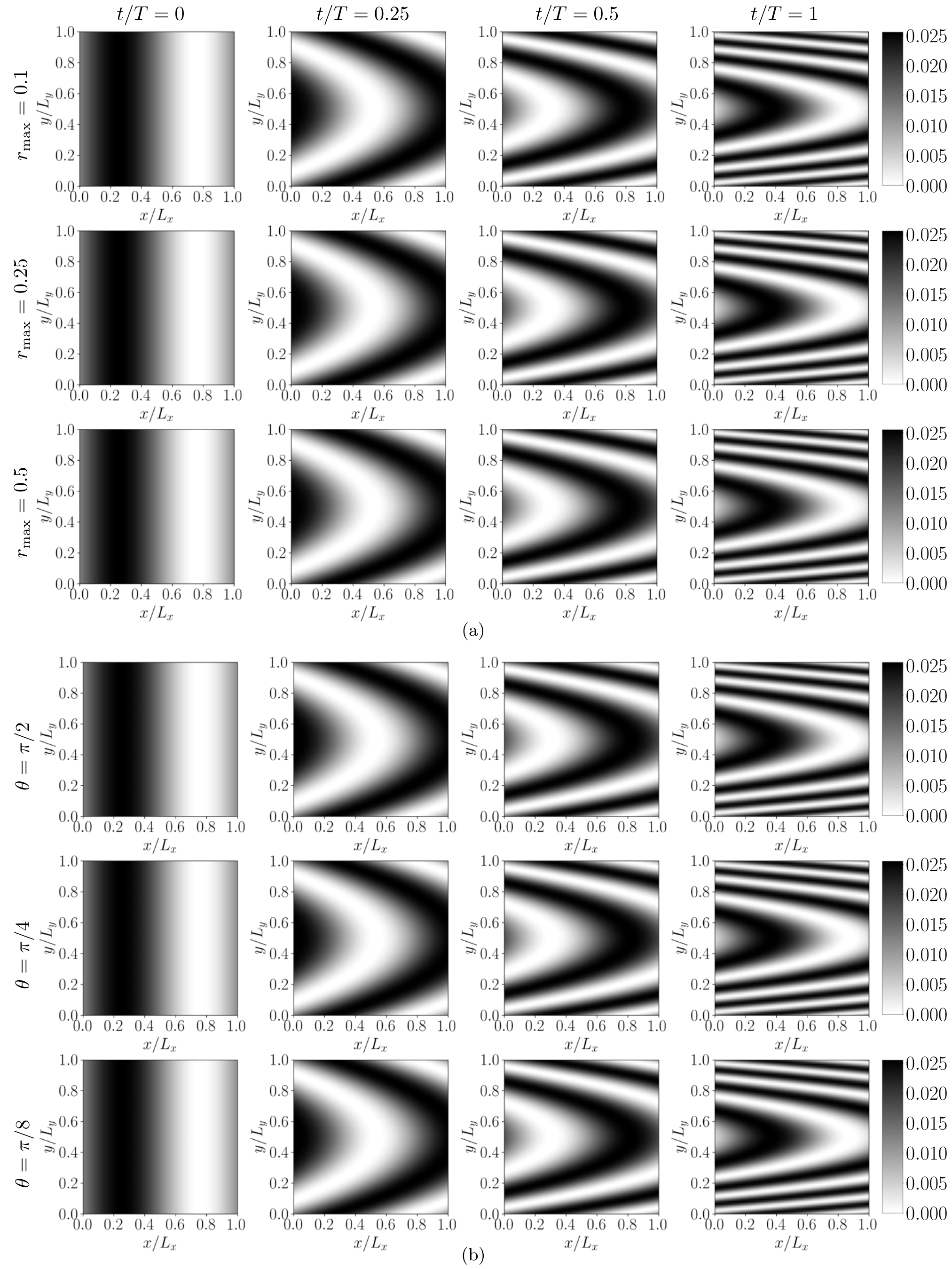


FIG. 3. Surface plot of the error bound per nondimensional simulation time as a function of r and θ for (a) the advection equation and (b) the heat equation. For the heat equation in (b), the surface has been truncated at $\epsilon/T = 20$ since ϵ asymptotically approaches infinity for $r_h \rightarrow 0.25$ and $r_h \rightarrow 0.5$.

IV. COMPLEXITY ANALYSIS

The number of qubits grows as $n = O(\log N)$ since the computational grid is compressed into the amplitudes of the exponentially growing computational basis states. The circuit depth grows linearly with the required number of time steps $N_T \propto TN^{1/D}$. The number of time steps growing linearly with the desired simulation time T is intuitive, but the $N^{1/D}$ factor with the number of grid points N requires a further breakdown. It is assumed that a D-dimensional domain of equal



(a)

(b)

FIG. 4. Evolution of the quantum amplitudes from state-vector simulations of a 2D laminar channel flow described by the advection equation at different time intervals for (a) $\theta = \pi/(1 + \sqrt{r_{\max}^2 + 1})$ while varying $r_{\max}$ and (b) $r_{\max} = 0.25$ while varying θ. The simulations use a second-order-accurate central finite-difference stencil.

side lengths is discretized by $N = N_x^D$ grid points, where $N_x = N_y = N_z$ for $D > 1$. The CFL number $r_{\max} = \max(u_j)\Delta t/\Delta x$ constrains the time step size to the grid spacing, where halving Δx requires halving Δt or, equivalently, doubling N_x requires doubling N_T. Given that $N_T \propto N_x$ and $N = N_x^D$, then $N \propto N_T^D$, or inversely $N_T \propto N^{1/D}$.

In terms of error, Fig. 3 shows that ϵ grows linearly with r, which is the same as the underpinning Euler method. Halving r halves ϵ, which requires doubling N_T and therefore doubling the circuit depth, so the circuit depth grows with $1/\epsilon$.

The remaining complexity considerations depend on the chosen implementation of the Hamiltonian simulations. As an example, the algorithm of Berry *et al.* [40] is considered due to its near-optimal properties. It combines the strategies of a Szegedy quantum walk [45,46] and fractional-query simulation [47]. The Szegedy quantum-walk approach scales optimally in the matrix sparsity but not in the allowable error, while the simulation of the fractional-query model scales optimally with the error but not with the sparsity. The fractional-query model is used to correct the phase more accurately than the QPE step in the quantum-walk approaches

[45,46], resulting in the favorable scaling in both sparsity and the allowable error. The overall algorithm implements Hamiltonian simulation with

$$O\left(\tau[n+\log^{5/2}(\tau/\epsilon)]\frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\right) \tag{20}$$

gates, where $\tau = s\|H\|_{\max}\theta$, s is the sparsity, and $\|H\|_{\max}\theta$ is the maximum value of the matrix $H\theta$ [40], which is $O(1)$ in this case. The complexity of the Hamiltonian simulation step can therefore be written as

$$O\left(s[\log(N)+\log^{5/2}(s/\epsilon)]\frac{\log(s/\epsilon)}{\log\log(s/\epsilon)}\right). \tag{21}$$

Suppressing the polylogarithmic terms for simplicity, this reduces to an almost linear dependence on the sparsity, $\tilde{O}(s)$. For the block-encoded Hamiltonian in Eq. (6), the sparsity of H is equal to the sparsity of A, which is $s = 1 + Dk$, where k is the order of the spatial discretization. Therefore, the complexity of the Hamiltonian simulation step can be written as $\tilde{O}(Dk)$. Since the error ϵ does not appear in this simplified expression, the dominant source of error arises from the encoding of the nonunitary operator and the explicit Euler method rather than the Hamiltonian simulation implementation. Combining all of these contributions, the circuit depth grows as $\tilde{O}(N_T s/\epsilon)$ or $\tilde{O}(TN^{1/D}Dk/\epsilon)$ for D-dimensional simulations.

Efficient classical simulations of the advection equation typically have a time complexity of $O(NN_T)$ [48], and given that $N_T \propto TN^{1/D}$, this can be written as $O(TN^{(1+D)/D})$. The quantum algorithm offers a significant polynomial improvement over the classical algorithm as their time complexities differ by a factor of N. The effectiveness of the quantum algorithm increases in higher-dimensional space as the number of grid points increases at a much faster rate than the required circuit depth, taking advantage of the exponentially growing Hilbert space.

V. SIMULATIONS

This section presents quantum state-vector simulations of the algorithm applied to a two-dimensional laminar channel flow in Sec. V A and a lid-driven cavity flow in Sec. V B. In Sec. V C the response of the algorithm to noise in the initial quantum state and the Hamiltonian embedding is demonstrated, and the errors are quantified for various finite-difference stencils with and without noise. In all simulations, $N = 64 \times 64 = 4096$ grid points have been used to discretize the problems in space, corresponding to 12 qubits representing the solution and the additional ancilla qubit required by the algorithm, totaling 13 qubits.

A. Laminar channel flow

The velocity field $\vec{u} = [u, v]$ is described by the analytical solution to the Navier-Stokes equations in this configuration, known as a plane Poiseuille flow, which has a parabolic profile of u leading to the nondimensional CFL parameter to be defined as

$$r(y) = \frac{u(y)\Delta t}{\Delta x} \tag{22}$$

$$= r_{\max}[4y(1-y)] \quad 0 \leqslant y \leqslant 1, \tag{23}$$

which varies from from 0 at the walls to $r_{\max}$ at the center of the domain, where y is the vertical normalized distance. The vertical component of the velocity $v = 0$ because the flow is laminar. The scalar ϕ is initialized as a sine wave in the horizontal direction as

$$\phi(x) = \sin(2\pi x) + 1, \quad 0 \leqslant x \leqslant 1, \tag{24}$$

where x is the corresponding normalized horizontal distance. The x boundaries are periodic, so fluid that flows out of the right boundary enters through the left boundary. The velocity reducing to zero at the walls (i.e., a no-slip wall) corresponds to Dirichlet boundary conditions, where the value of ϕ_{wall} is maintained according to Eq. (24). This boundary condition results in a single entry of 1 on the diagonal of A.

The evolution of the quantum amplitudes for (a) varying CFL parameter r and (b) varying Hamiltonian evolution time θ per time step is provided in Fig. 4, showing that a visually similar solution is reached regardless of the values of r or θ. The equivalent simulation time corresponds to 2000 successful time steps for $r_{\max} = 0.1$, 800 successful time steps for $r_{\max} = 0.25$, and 400 successful time steps for $r_{\max} = 0.5$. The simulations in Fig. 4(a) use $\theta = \pi/(1 + \sqrt{r_{\max}^2 + 1})$ calculated from Fig. 2 to optimize $P_{\min}$, i.e., the worst-case probability of measurement success. The simulations in Fig. 4(b) use a constant $r_{\max} = 0.25$ while varying θ to $\pi/2$, $\pi/4$, and $\pi/8$. Varying θ alters the probability of a successful time step $P = \|\tilde{A}|\phi_t\rangle\|^2$, and the variation of P with $r_{\max}$ and θ is presented in Figs. 5(a) and 5(b), respectively. The value of θ that optimizes the worst-case $P_{\min}$ is not found to be optimal in practice, with $\pi/2$ providing a significantly greater probability of postselection success. For example, $r_{\max} = 0.1$ and $\theta = \pi/(1 + \sqrt{r_{\max}^2 + 1})$ leads to a typical 67 000 successful time steps for every unsuccessful time step compared to approximately 1×10^9 when $\theta = \pi/2$. The prediction of $\sin^2(\theta)$ from Fig. 2 is very accurate for all values of θ as shown in Fig. 5(b). Although the probability of time step success is close to certain when $\theta = \pi/2$, there remains a small chance of failure so the algorithm cannot be considered entirely deterministic. However, this does not impact the ability of the algorithm to prepare the state $|\phi_T\rangle$; it just requires further attempted time steps. The $\theta = \pi/4$ and $\pi/8$ cases in Fig. 4(b) demonstrate the ability of the algorithm to withstand postselection failure, where approximately only 50% and 14.6% of the time steps succeed, respectively.

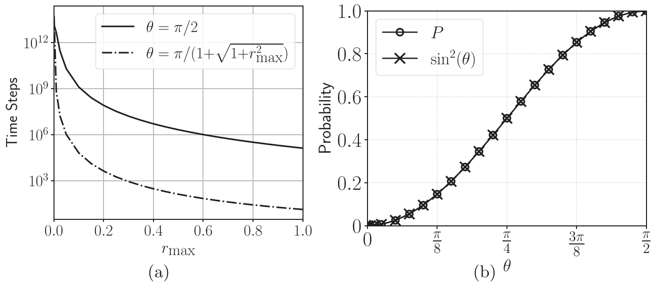
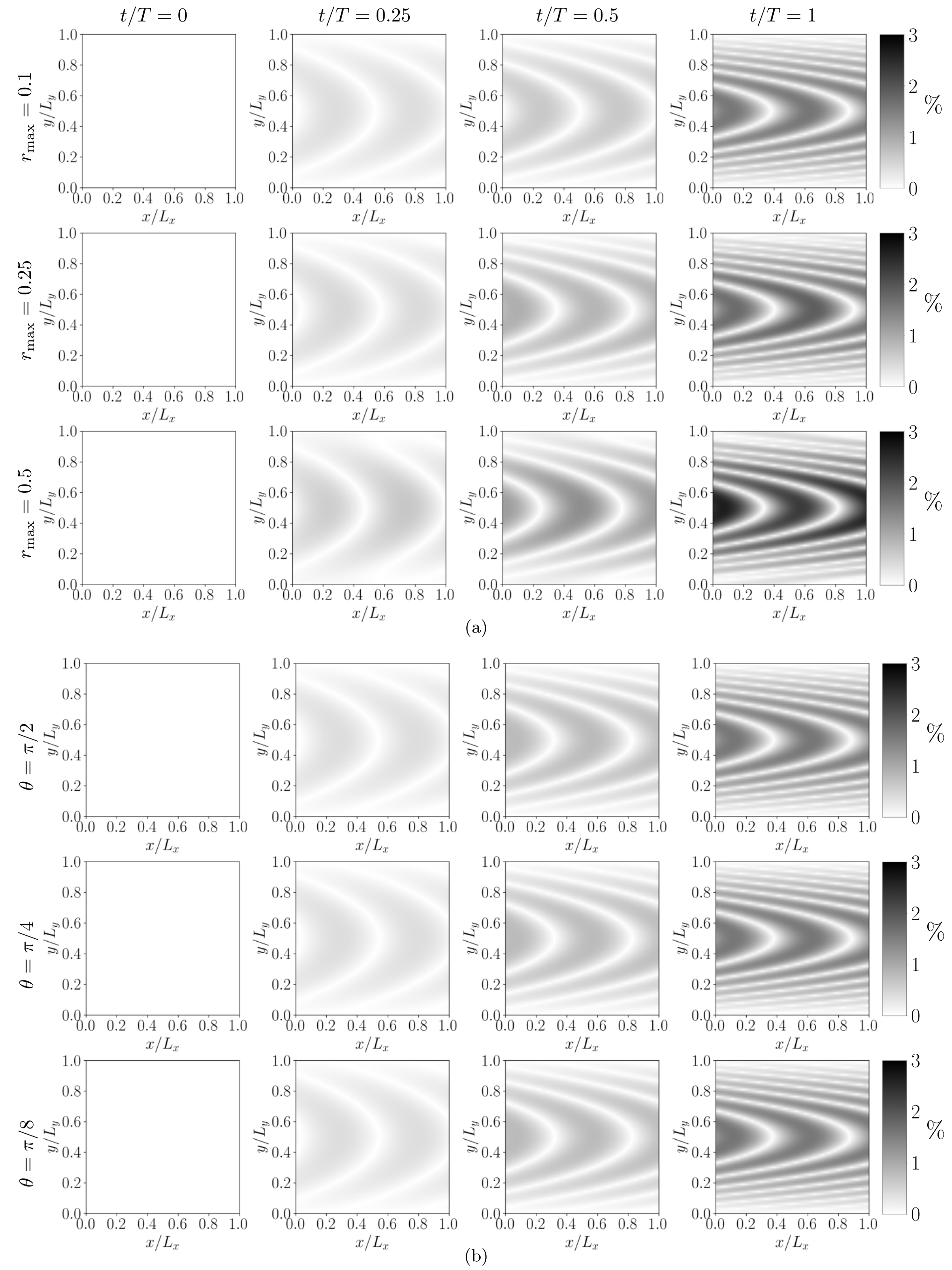


FIG. 5. (a) Typical number of successful time steps for every failed time step as a function of $r_{\max}$ for different values of θ and (b) probability of a successful time step $P = \|\tilde{A}|\phi_t\rangle\|^2$ and the prediction $\sin^2(\theta)$ for the $r_{\max} = 0.25$ case.



(a)

(b)

FIG. 6. Error contours for the channel flow simulations compared to the analytical solution as a percentage, defined as $100|\phi(x, y, t) - |\phi\rangle|/\max[\phi(x, y, t)]$, for (a) $\theta = \pi/(1 + \sqrt{r_{\max}^2 + 1})$ while varying $r_{\max}$ and (b) $r_{\max} = 0.25$ while varying θ.

The laminar channel flow configuration can be considered as an ensemble of one-dimensional advection problems and so has an analytical solution that can be calculated using the method of characteristics. Given the initial condition in Eq. (24), the analytical solution at time t is

$$\phi(x, y, t) = \frac{\sin\{2\pi[x - u(y)t]\} + 1}{\|\phi(x, y, 0)\|}, \tag{25}$$

where the denominator ensures a norm of 1 so the solutions can be compared like for like. Figure 6 compares the quantum solution for all cases against the analytical solution by plotting the local errors as a percentage, $100|\phi(x, y, t) - |\phi\rangle|/\max[\phi(x, y, t)]$. The errors remain within 3% for every case and appear to grow linearly with the local value of r, increasing towards the center of the domain where r is greatest and with increasing $r_{\max}$ as shown in Fig. 6(a). This is in agreement with the theoretical linear dependence of the algorithm with the error derived in Sec. IV. Figure 6(b) shows how the errors vary with θ under the influence of unsuccessful time steps, which shows that reducing θ does not decrease the overall accuracy of the solution, again agreeing with the theoretical expression visualized in Fig. 3. However, reducing θ does require more time steps to be attempted, thereby increasing the circuit depth and confirming the theoretical result

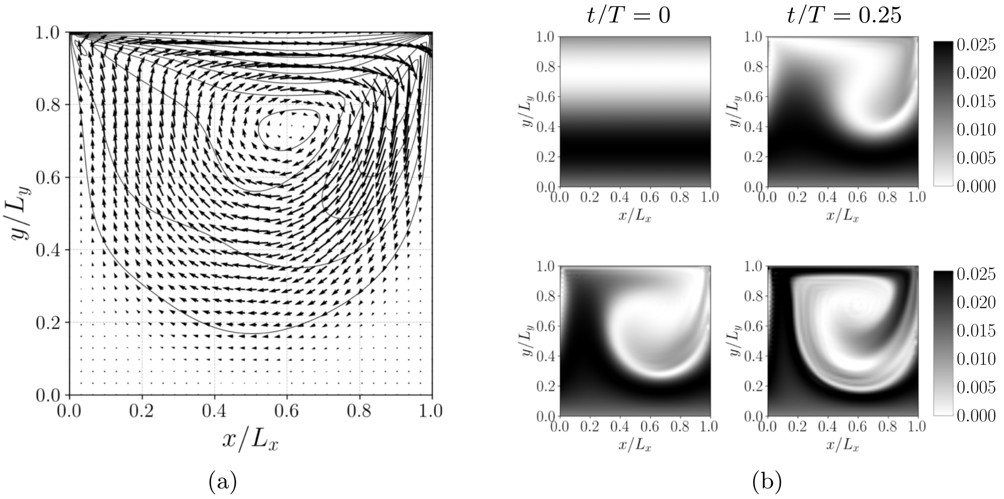


FIG. 7. (a) Velocity vectors overlaid with velocity magnitude contours for the lid-driven cavity simulations at Re = 100 and (b) contours showing the evolution of the quantum amplitudes for lid-driven cavity simulations discretized by a one-sided second-order-accurate upwind finite-difference stencil.

that $\theta = \pi/2$ is the most efficient configuration. In all cases, the errors smoothly display the parabolic profiles of the scalar field, indicating that discretization errors are consistently leading to propagation at a slightly different velocity to $u(y)$. The parabolas of zero error are false negatives, arising due to identical scalar values on either side of a local maximum or minimum masking the true error. The algorithm accurately represents the Dirichlet boundary where the scalar gradients are greatest, as demonstrated by the errors reducing to zero towards the y boundaries.

B. Lid-driven cavity

A scalar transported in a lid-driven cavity has been simulated to demonstrate the performance of the algorithm for a multidimensional problem where no straightforward analytical solution exists. The lid-driven cavity configuration involves a square cavity filled with a fluid, where the top wall (lid) moves horizontally at a constant velocity U_{wall}, while the other three walls remain stationary. The fluid directly adjacent to the moving wall acquires the same velocity as the wall due to the no-slip condition. This imparts momentum to the rest of the fluid, causing it to circulate within the cavity.

As no analytical solution is available for this configuration and for comparison, the velocity field has been generated by the commercial computational fluid dynamics software Ansys Fluent [49] for a Reynolds number $\text{Re} = U_{\text{wall}}L/\nu = 100$, where L is the side length of the cavity and ν is the kinematic viscosity. The domain was discretized with 64 × 64 cells and solved using the finite-volume method with a pressure-based coupled solver and a second-order upwind scheme for the spatial derivatives. The resulting velocity vectors are shown in Fig. 7(a), overlaid by the velocity magnitude. The vectors are drawn on a 32 × 32 grid for a less-cluttered graphic.

The scalar field is initialized as a sine wave in the y direction, $\phi(x, y) = \sin(2\pi y) + 1$ for $y = 0$ to 1. The quantum simulations use a second-order one-sided upwind finite-difference stencil to evaluate the matrix A, where the direction of the stencil is in the opposite direction to the local velocity component. The maximum CFL parameter $r_{\max} = 0.1$, the Hamiltonian evolution time per time step $\theta = \pi/2$, and the simulations have been carried out for $N_T = 2800$ time steps with each time step occurring successfully. The initial conditions and subsequent evolution of the scalar field are shown in Fig. 7(b), which shows the scalar field swirling and being distorted by the velocity field. The no-slip wall naturally leads to a Dirichlet condition at the boundary in pure advection problems, and Fig. 7(b) shows that the wall boundary values are effectively maintained. The solution is physically plausible and captures other qualitative features of the flow such as the vortex location and is in close quantitative agreement with corresponding classical simulations.

C. Effects of noise and the spatial scheme

Returning to the laminar channel flow where an analytical solution is available, Fig. 8 shows the effects of various types of noise on the simulations using $r_{\max} = 0.1$, $\theta = \pi/2$, and a fourth-order central scheme for the spatial derivatives. Figure 8(a) shows that, when subject to Gaussian noise with a standard deviation of 10% of mean(ϕ), the numerical methods in the algorithm are capable of handling the noise as the same qualitative solution is obtained as the noise-free simulations in Fig. 4. The initial noise is retained in the final solution and does not appear to accumulate or dampen. Figure 8(b) demonstrates the effects of noise in the Hamiltonian embedding procedure in Eq. (6). When the entries of A are subject to noise with a standard deviation of 1% of the true value, the correct qualitative solution is obtained with errors that appear to grow linearly with time and the local value of $u(y)$.

The growth of the mean absolute value of the error with no noise, initial state noise, and Hamiltonian embedding noise is shown in Fig. 9 for a fourth-order central scheme, a second-order central scheme, and a second-order one-sided upwind scheme. In all cases, the errors are calculated against the noise-free analytical solution in Eq. (25). Starting with the noise-free error evolution, Fig. 9 confirms that the mean error grows linearly with time for all spatial discretization schemes.

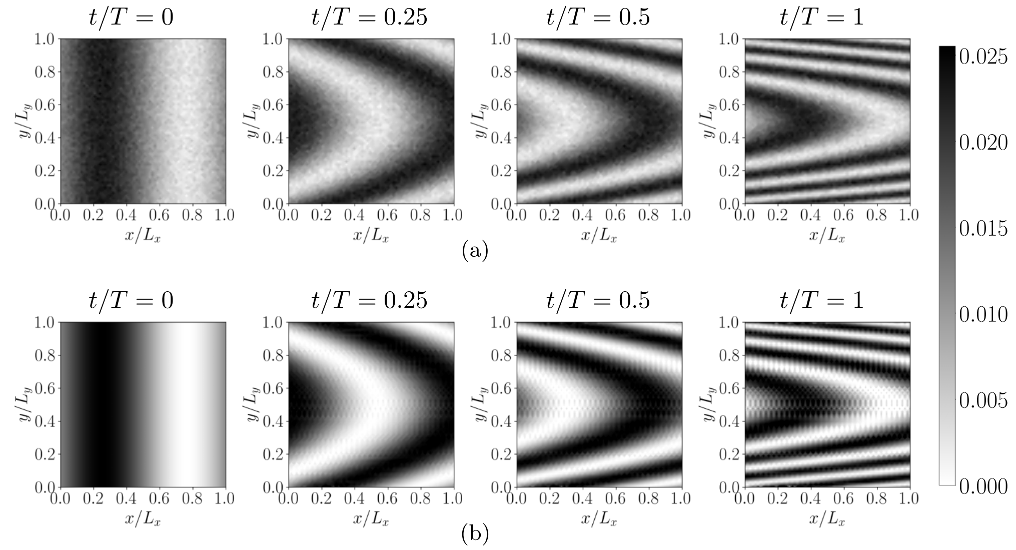


FIG. 8. Evolution of the quantum amplitudes from state-vector simulations of a noisy channel flow with (a) Gaussian noise in the initial state and (b) Gaussian noise in the Hamiltonian embedding. The simulations use a fourth-order-accurate central finite-difference stencil.

The second-order one-sided scheme is the least accurate, with mean errors reaching 1.3% after 2000 time steps, followed by the second-order and fourth-order central schemes with errors reaching 0.6% and 0.1%, respectively. In the presence of noise, the one-sided scheme performs best due to the greater numerical dissipation of the noise in the small scales while still capturing the derivative information in the large scales. For the initial state noise, the central schemes maintain the mean error at approximately 4% while the one-sided scheme sharply reduces the error before it stabilizes at below 3%. In the case of Hamiltonian embedding noise, errors grow linearly with the simulation time but with a much steeper gradient than the noise-free simulations. Again, the one-sided scheme significantly outperforms as the errors grow at a reduced rate.

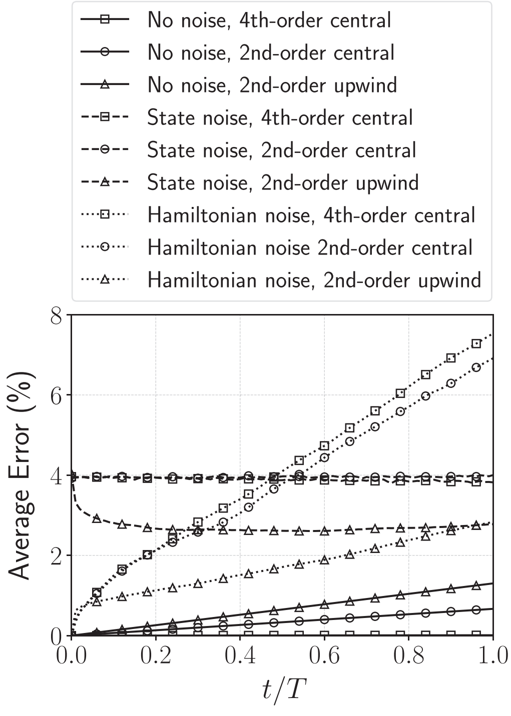


FIG. 9. Mean absolute value of the error with different noise and spatial discretization schemes.

VI. CONCLUSION

A quantum algorithm for solving the advection equation, a linear PDE prevalent in various scientific and engineering industries, was presented. The algorithm uses sparse Hamiltonian simulation to embed the discrete time-marching operator A into the Hamiltonian H, resulting in a unitary operator $\Omega = e^{-iH\theta}$ that encodes A to a high accuracy regardless of the values of θ. Postselection failure does not require further state initialization queries since the resulting operation closely approximates the identity matrix, having a minimal impact on the quantum state and allowing the computation to continue. The algorithm applies to multidimensional problems with arbitrary boundary conditions and finite-difference stencils.

From a resource utilization perspective, qubit requirements grow logarithmically with the number of grid points N and the circuit depth grows linearly with the desired number of time steps, the sparsity of the discrete time-marching operator, and the inverse of the desired error, $\tilde{O}(N_T s/\epsilon)$, when suppressing polylogarithmic terms. This represents a significant polynomial speedup in complexity compared to classical methods, which typically exhibit a scaling of $O(NN_T)$, leading to a polynomial improvement by a factor of N.

It was demonstrated mathematically that the methodology does not universally apply to all PDEs, using the heat equation as an example. The derived mathematical expressions for the effects of the r and θ parameters on the error have been validated numerically with state-vector simulations. In the channel flow configuration, the amplitudes closely agreed with the analytical solution for all parameters tested. It was shown that there is an advantage in using high-order spatial schemes when the state varies as a continuous function, but that low-order dissipative schemes may outperform in noisy environments.

The typical advantages of the proposed approach over other algorithms are that the runtime grows linearly with the simulation time, the state can be reused on postselection failure requiring a single copy of the initial state, any combination of finite-difference stencils can be used, and its simplicity of implementation. Other algorithms that can be applied to the advection equation are typically aimed at solving homogeneous ODEs in the form of Eq. (2), which the advection equation reduces to when discretized in space. In comparison to QLSA-based algorithms [24,25,27,28], the proposed approach excels due to the single copy of the initial quantum state required. The optimal QLSA [31] requires $O(\kappa \log(1/\epsilon))$ queries to the state preparation oracle, which can become prohibitive when pursuing a practical quantum advantage. The quantum time-marching algorithm [39] is conceptually the most similar to the algorithm presented here, although it results in a runtime with a quadratic dependence on T compared to a linear dependence in the present algorithm. Schrödingerization [35–37] and LCHS [32,38] algorithms both assume that the Hermitian part of the coefficient matrix M is negative semidefinite, which applies to central and upwind finite-difference schemes for the advection equation. Occasionally, however, the use of downwind schemes cannot be avoided, such as when resolving the flow near a computational boundary. Therefore, the flexibility of the proposed approach and its simplicity of implementation make it a preferred choice.

Finally, developing algorithms that evolve a quantum state by the PDE of interest is only a step towards achieving a practical quantum advantage, with methods for efficiently preparing the state and extracting useful global statistics being crucial for preparing scientific and engineering industries to be quantum ready.

ACKNOWLEDGMENT

This research was funded by EPSRC, Grant No. EP/W032643/1.

[1] S. Bravyi, O. Dial, J. M. Gambetta, D. Gil, and Z. Nazario, The future of quantum computing with superconducting qubits, J. Appl. Phys. **132**, 160902 (2022).

[2] L. J. Stephenson, D. P. Nadlinger, B. C. Nichol, S. An, P. Drmota, T. G. Ballance, K. Thirumalai, J. F. Goodwin, D. M. Lucas, and C. J. Ballance, High-rate, high-fidelity entanglement of qubits across an elementary quantum network, Phys. Rev. Lett. **124**, 110501 (2020).

[3] D. J. Webb, B. A. De Cuevas, and C. S. Richmond, Improved advection schemes for ocean models, J. Atmos. Ocean. Technol. **15**, 1171 (1998).

[4] R. B. Rood, Numerical advection algorithms and their role in atmospheric transport and chemistry models, Rev. Geophys. **25**, 71 (1987).

[5] H. Samuel and M. Evonuk, Modeling advection in geophysical flows with particle level sets, Geochem. Geophys. Geosyst. **11**, Q08020 (2010).

[6] Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. Hughes, $YZ\beta$ discontinuity capturing for advection-dominated processes with application to arterial drug delivery, Int. J. Numer. Methods Fluids **54**, 593 (2007).

[7] N. Diao, Q. Li, and Z. Fang, Heat transfer in ground heat exchangers with groundwater advection, Int. J. Therm. Sci. **43**, 1203 (2004).

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

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

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

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

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

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

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

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

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

[17] S. Lloyd, G. De Palma, C. Gokler, B. Kiani, Z.-W. Liu, M. Marvian, F. Tennie, and T. Palmer, Quantum algorithm for nonlinear differential equations, arXiv:2011.06571.

[18] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs, Efficient quantum algorithm for dissipative nonlinear differential equations, Proc. Natl. Acad. Sci. USA **118**, e2026805118 (2021).

[19] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. **5**, 4213 (2014).

[20] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, Variational quantum algorithms for nonlinear problems, Phys. Rev. A **101**, 010301 (2020).

[21] O. Kyriienko, A. E. Paine, and V. E. Elfving, Solving nonlinear differential equations with differentiable quantum circuits, Phys. Rev. A **103**, 052416 (2021).

[22] D. Jaksch, P. Givi, A. J. Daley, and T. Rung, Variational quantum algorithms for computational fluid dynamics, AIAA J. **61**, 1885 (2023).

[23] N. Gourianov, M. Lubasch, S. Dolgov, Q. Y. van den Berg, H. Babaee, P. Givi, M. Kiffner, and D. Jaksch, A quantum-inspired

approach to exploit turbulence structures, Nat. Comput. Sci. **2**, 30 (2022).

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

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

[26] J. M. Arrazola, T. Kalajdzievski, C. Weedbrook, and S. Lloyd, Quantum algorithm for nonhomogeneous linear partial differential equations, Phys. Rev. A **100**, 032306 (2019).

[27] H. Krovi, Improved quantum algorithms for linear and nonlinear differential equations, Quantum **7**, 913 (2023).

[28] D. W. Berry and P. C. S. Costa, Quantum algorithm for time-dependent differential equations using Dyson series, Quantum **8**, 1369 (2024).

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

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

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

[32] D. An, A. M. Childs, and L. Lin, Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters, arXiv:2312.03916.

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

[34] L. Budinski, Quantum algorithm for the advection–diffusion equation simulated with the lattice Boltzmann method, Quantum Inf. Process. **20**, 57 (2021).

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

[36] S. Jin, N. Liu, and Y. Yu, Quantum simulation of partial differential equations: Applications and detailed analysis, Phys. Rev. A **108**, 032603 (2023).

[37] S. Jin, X. Li, N. Liu, and Y. Yu, Quantum simulation for partial differential equations with physical boundary or interface conditions, J. Comput. Phys. **498**, 112707 (2024).

[38] D. An, J.-P. Liu, and L. Lin, Linear combination of hamiltonian simulation for nonunitary dynamics with optimal state preparation cost, Phys. Rev. Lett. **131**, 150603 (2023).

[39] D. Fang, L. Lin, and Y. Tong, Time-marching based quantum solvers for time-dependent linear differential equations, Quantum **7**, 955 (2023).

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

[41] C. Shao, Quantum algorithms to matrix multiplication, arXiv:1803.01601.

[42] A. R. Mitchell and D. F. Griffiths, *The Finite Difference Method in Partial Differential Equations* (Wiley, New York, 1980).

[43] R. Courant, K. Friedrichs, and H. Lewy, Über die partiellen differenzengleichungen der mathematischen physik, Math. Ann. **100**, 32 (1928).

[44] R. M. Gingrich and C. P. Williams, in *Proceedings of the Winter International Symposium on Information and Communication Technologies, Cancun, 2004* (Trinity College Dublin, Dublin, 2004).

[45] A. M. Childs, On the relationship between continuous-and discrete-time quantum walk, Commun. Math. Phys. **294**, 581 (2010).

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

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

[48] S. B. Pope, *Turbulent Flows* (Cambridge University Press, Cambridge, 2000).

[49] Ansys Fluent (Ansys, Inc., Canonsburg, 2024), available at https://www.ansys.com/products/fluids/ansys-fluent.