

Quantum algorithm for the advection-diffusion equation by direct block encoding of the time-marching operator

Paul Over¹, Sergio Bengoechea¹, Peter Brearley², Sylvain Laizet², and Thomas Rung¹

¹*Institute for Fluid Dynamics and Ship Theory, Hamburg University of Technology, Hamburg D-21073, Germany*

²*Department of Aeronautics, Imperial College London, London SW7 2AZ, England, United Kingdom*



(Received 11 October 2024; accepted 4 June 2025; published 1 July 2025)

A quantum algorithm for simulating multidimensional scalar transport problems using a time-marching strategy is presented. A direct unitary block encoding of the explicit time-marching operator is constructed, resulting in the intrinsic success probability of the squared solution norm without the need for amplitude amplification, thereby retaining a linear dependence on the simulation time. The algorithm separates the explicit time-marching operator into an advection-like component and a corrective shift operator. The advection-like component is mapped to a Hamiltonian simulation and combined with the shift operator through the linear combination of unitaries algorithm. State-vector simulations of a scalar transported in a steady two-dimensional Taylor-Green vortex support the theoretical findings.

DOI: [10.1103/d8hb-fv93](https://doi.org/10.1103/d8hb-fv93)

Introduction. Partial differential equations (PDEs) provide a versatile mathematical framework for describing and simulating diverse phenomena, supporting technological progress with reduced costs compared to experimental methods. The computational demand for solving PDEs requires the extensive use of high-performance computing for many practical applications, leading to substantial investments in both hardware and algorithmic development. Despite these advancements, computational resources remain insufficient to address the scales of problems that scientists and engineers wish to simulate, often by several orders of magnitude [1]. Fault-tolerant quantum computing brings a paradigm shift in algorithmic performance by offering an exponential vector space for computation, although requiring the development of specialized algorithms to achieve this potential.

A prominent PDE in fluid dynamics is the advection-diffusion equation, given by

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = D \nabla^2 \phi, \quad (1)$$

that describes the transport of a scalar ϕ advected by an incompressible flow with velocity \mathbf{v} in a medium with diffusivity D . The prevailing strategy for solving such PDEs on quantum computers is by conversion to a system of ordinary differential equations

$$\frac{d\phi}{dt} = M\phi, \quad (2)$$

applying the finite difference method for spatial discretization [2–8]. This has the analytical solution of $\phi(t) = e^{Mt} \phi(0)$

for a matrix M describing the discretized physics. To solve this more general problem, Berry *et al.* [2] encoded a truncated Taylor series expansion of e^{Mt} into a system of linear equations, and applied a quantum linear systems algorithm [9,10]. This has steadily been improved in subsequent works with the extension to a wider range of matrices [3] and with improved dependence on the precision [4]. Other innovative approaches include the linear combination of Hamiltonian simulation method [5,6] and Schrödingerization [7,8], which map Eq. (2) to a dilated system of Schrödinger equations that can be solved using Hamiltonian simulation.

A less common strategy for solving problems in the form of Eq. (2) on a quantum computer is time marching, where e^{Mt} is approximated by its truncated Taylor series and evolved over a series of short time steps Δt where the approximation is accurate. For example, the forward Euler method is accurate to first-order terms and results in the system

$$\phi(t + \Delta t) = A\phi(t), \quad (3)$$

where $A = I + M\Delta t$ is the Euler time-marching matrix and I is the identity matrix. Time marching is often the most natural strategy for solving PDEs classically, so its limited use in quantum algorithms may be surprising. Since A is nonunitary, it cannot be implemented directly as a series of quantum gates, instead requiring a probabilistic implementation by block encoding. Considering the following unitary block encoding implementation of Eq. (3),

$$\begin{bmatrix} \frac{A}{\alpha} & * \\ * & * \end{bmatrix} \begin{bmatrix} |\phi_t\rangle \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{A}{\alpha} |\phi_t\rangle \\ * \end{bmatrix}, \quad (4)$$

A can only be encoded accurately up to a subnormalization factor $\alpha \geq 1$, where a subscript t refers to a time index and the asterisk indicates arbitrary, inconsequential blocks. Measuring $|0\rangle$ prepares the state $A/\alpha |\phi_t\rangle$ with probability $1/\alpha^2 \|A |\phi_t\rangle\|^2$. The constant factor of $1/\alpha^2$ induces an exponentially diminishing success probability with repeated time

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

steps for $\alpha > 1$ [11–13], limiting the usefulness of quantum time-marching algorithms to $\alpha = 1$.

Several algorithms capable of producing a general block encoding have been proposed [14–16]. For example, the linear combination of unitaries (LCU) algorithm [14] is based on the principle that any nonunitary operator can be written as a weighted sum of unitary operators $A = \sum_j \beta_j U_j$. However, finding an efficient decomposition is not straightforward. Any nonunitary matrix can be written as the limit of the linear combination of four unitary operators:

$$A = \lim_{\tau \rightarrow 0} \frac{1}{2\tau} (ie^{-i\tau H_1} - ie^{i\tau H_1} + e^{i\tau H_2} - e^{-i\tau H_2}), \quad (5)$$

where $H_1 = (A + A^\dagger)/2$ and $H_2 = (A - A^\dagger)/2i$ are Hermitian matrices such that $A = H_1 + iH_2$. This methodology was used by Bharadwaj and Sreenivasan [12] in the LCU framework, but has the disadvantage that as $\tau \rightarrow 0$, $\alpha \rightarrow \infty$ in Eq. (4), requiring a compromise between the accuracy and probability of success. Since the methodology cannot produce an accurate block encoding where $\alpha = 1$, time marching results in an exponentially decaying probability of success with the simulation time T . This problem can be overcome by the quantum singular value transformation [16], which unifies qubitization [15] and quantum signal processing [17] algorithms for performing arbitrary polynomial transformations on the singular values of a block-encoded matrix. This was the approach taken by Fang *et al.* [18] by applying a uniform singular value amplification [16] at each time step to bound the success probability, though at the cost of a quadratic dependence on the simulation time.

Here, a quantum algorithm for solving the advection-diffusion and heat equations using a direct, $\alpha = 1$ block encoding is presented. A quantum algorithm for solving the advection equation [19] is combined with corrective unitary operators in the LCU framework [14] to recover the advection-diffusion time-marching operator. The intrinsic success probability of the squared norm of the solution vector is achieved while retaining a linear simulation time dependence. This is beneficial since the success probability is solely dependent on the physical problem rather than the number of time steps. This improves over the general strategy in Eq. (5) [12] in terms of the probability of success, and quantum singular value transformation methods [16,18] in terms of the dependence on the simulation time.

The algorithm applies to multidimensional problems with spatially varying velocity fields. However, a one-dimensional periodic problem discretized using N_x grid points with an equal grid spacing Δx will briefly be considered for simplicity. The one-dimensional advection-diffusion equation discretized using the forward Euler method and a second-order central finite difference stencil in space is

$$\frac{\phi_{t+1}^m - \phi_t^m}{\Delta t} + v \frac{\phi_t^{m+1} - \phi_t^{m-1}}{2\Delta x} = D \frac{\phi_t^{m+1} - 2\phi_t^m + \phi_t^{m-1}}{(\Delta x)^2}, \quad (6)$$

for discrete space and time locations m and t , respectively. This produces the explicit time-marching equation

$$\phi_{t+1}^m = \phi_t^{m-1} \left(r_h + \frac{r_a}{2} \right) + \phi_t^m (1 - 2r_h) + \phi_t^{m+1} \left(r_h - \frac{r_a}{2} \right), \quad (7)$$

where $r_a = v\Delta t/\Delta x$ and $r_h = D\Delta t/(\Delta x)^2$ are the stability parameters for advection and diffusion, respectively. Equation (7) can be written in the form of Eq. (3) with the time-marching operator

$$A = \begin{bmatrix} 1-2r_h & r_h - \frac{r_a}{2} & 0 & r_h + \frac{r_a}{2} \\ r_h + \frac{r_a}{2} & \ddots & \ddots & 0 \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & r_h - \frac{r_a}{2} \\ r_h - \frac{r_a}{2} & 0 & r_h + \frac{r_a}{2} & 1-2r_h \end{bmatrix}. \quad (8)$$

Equation (3) is implemented efficiently by decomposing the matrix A into an advection-like component and a corrective shift operator. The advection-like component can be implemented with the quantum advection algorithm of Brearley and Laizet [19], which can then be linearly combined with the unitary shift operator using the LCU algorithm [14] to directly enact A on a subspace of the quantum state.

Matrix decomposition. The decomposition begins by temporarily scaling Eq. (8) to $A/(1 - 2r_h)$ such that the diagonal entries are 1, although this factor will later cancel out to recover the direct block encoding. Then, the scaled A can be written as the sum

$$\frac{A}{1 - 2r_h} = \hat{A} + \frac{2r_h}{1 - 2r_h} S, \quad (9)$$

where

$$\hat{A} = \begin{bmatrix} 1 & \frac{-r_h - r_a/2}{1 - 2r_h} & 0 & \frac{r_h + r_a/2}{1 - 2r_h} \\ \frac{r_h + r_a/2}{1 - 2r_h} & \ddots & \ddots & 0 \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & \frac{-r_h - r_a/2}{1 - 2r_h} \\ \frac{-r_h - r_a/2}{1 - 2r_h} & 0 & \frac{r_h + r_a/2}{1 - 2r_h} & 1 \end{bmatrix} \quad (10)$$

is the advection-like operator with the same mathematical structure as the time-marching operator for the discretized advection equation [19], not to be confused with the advection-diffusion time-marching operator A from Eq. (8), and

$$S = |N_x - 1\rangle \langle 0| + \sum_{j=1}^{N_x-1} |j-1\rangle \langle j| \quad (11)$$

is the unitary shift operator that shifts the quantum amplitudes to the preceding basis state.

Quantum advection algorithm. The constructed advection-like matrix \hat{A} is implemented through a block encoding by Hamiltonian simulation [19,20], which prepares a unitary $e^{-iH\theta}$ for a Hamiltonian evolution time per time step θ . The Hamiltonian is defined as

$$H = \begin{bmatrix} 0 & -i\hat{A}^\dagger \\ i\hat{A} & 0 \end{bmatrix}, \quad (12)$$

where $(\dots)^\dagger$ denotes the conjugate transpose, resulting in the unitary operator

$$e^{-iH\theta} = \exp \begin{bmatrix} 0 & -\hat{A}^\dagger \theta \\ \hat{A} \theta & 0 \end{bmatrix}. \quad (13)$$

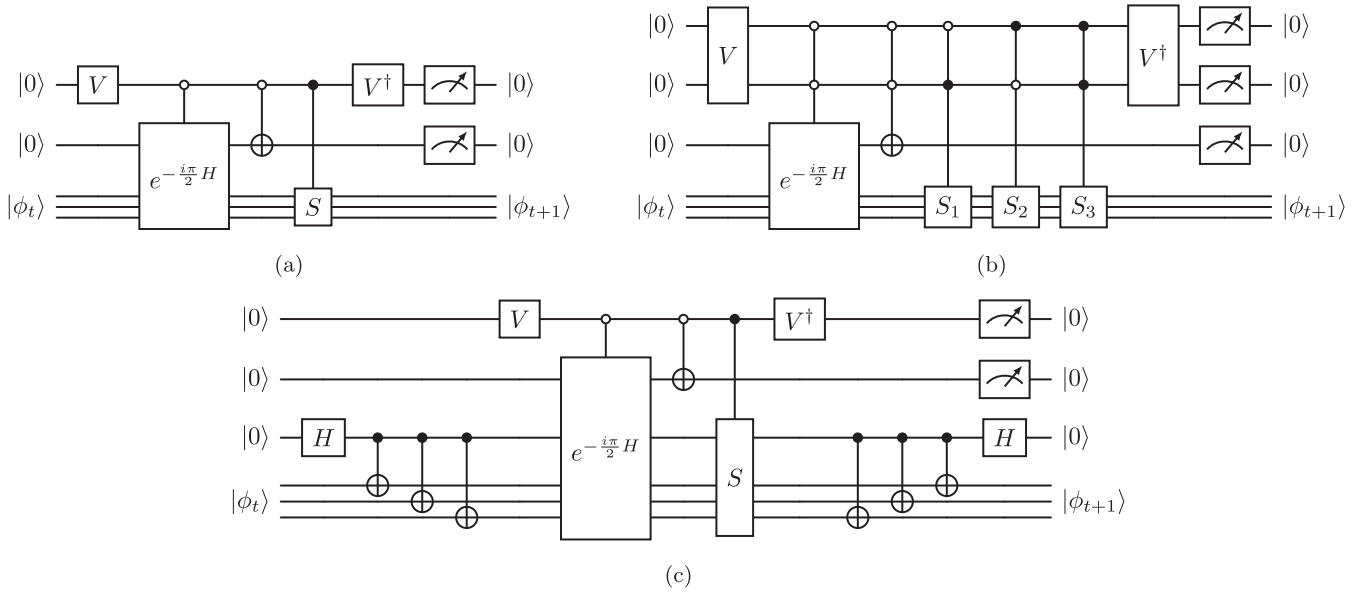


FIG. 1. Quantum circuit for implementing a single time step of the algorithm in (a) one dimension ($d = 1$), (b) three dimensions ($d = 3$), and (c) one dimension ($d = 1$) with homogeneous Neumann boundary conditions.

Brearley and Laizet [19] showed that Eq. (13) has the exact block matrix structure

$$e^{-iH\theta} = \begin{bmatrix} \cos(\sqrt{\hat{A}^\dagger \hat{A}} \theta) & -\hat{A}^\dagger \frac{\sin(\sqrt{\hat{A}^\dagger \hat{A}} \theta)}{\sqrt{\hat{A}^\dagger \hat{A}}} \\ \hat{A} \frac{\sin(\sqrt{\hat{A}^\dagger \hat{A}} \theta)}{\sqrt{\hat{A}^\dagger \hat{A}}} & \cos(\sqrt{\hat{A}^\dagger \hat{A}} \theta) \end{bmatrix}, \quad (14)$$

in terms of matrix trigonometric functions. The unitary operator in Eq. (14) can prepare the state $\hat{A}|\psi\rangle$ when $\theta = \pi/2$ with error $O([r_a + r_h]^2)$ [19], conditional on measuring an ancilla qubit initialized as $|0\rangle$ in the state $|1\rangle$. The error of the encoding arises from the equivalent advection stability parameter of \hat{A} being $(2r_h + r_a)/(1 - 2r_h)$, which has the asymptotic behavior of $O(r_a + r_h)$ in the limit of small r_h , and the advection algorithm can prepare a block encoding with an error that is the square of the equivalent advection stability parameter [19]. The encoding is direct when $\theta = \pi/2$ and postselection succeeds with near certainty [19].

Linear combination of unitaries. A single time step for solving the one-dimensional advection-diffusion equation can be implemented using the quantum circuit shown in Fig. 1(a), which uses the LCU algorithm [14] to combine the quantum advection algorithm [19] with the shift operator.

The unitary

$$V = \begin{bmatrix} \sqrt{\frac{\kappa_0}{\kappa_0 + \kappa_1}} & -\sqrt{\frac{\kappa_1}{\kappa_0 + \kappa_1}} \\ \sqrt{\frac{\kappa_1}{\kappa_0 + \kappa_1}} & \sqrt{\frac{\kappa_0}{\kappa_0 + \kappa_1}} \end{bmatrix} \quad (15)$$

encodes the square roots of the coefficients of the LCU sum in the first column corresponding to an ancilla measurement of $|0\rangle$, where $\kappa_0 = 1$ and $\kappa_1 = 2r_h/(1 - 2r_h)$ from Eq. (9) for the one-dimensional problem briefly being considered. Implementing the circuit in Fig. 1(a) maps

$$\begin{aligned} |00\rangle |\phi_t\rangle &\mapsto |00\rangle \frac{\kappa_0 \hat{A} + \kappa_1 S}{\kappa_0 + \kappa_1} |\phi_t\rangle + \dots \\ &\mapsto |00\rangle A |\phi_t\rangle + \dots, \end{aligned} \quad (16)$$

where \hat{A} is represented with error $O([r_a + r_h]^2)$ from the full bottom-left element of the matrix in Eq. (14). The mapping simplifies to the action of the original, unscaled time-marching operator A in Eq. (3), since the normalization of $\kappa_0 + \kappa_1 = 1/(1 - 2r_h)$ entirely cancels the original normalization of $1 - 2r_h$ in Eq. (9). This property generalizes to higher dimensions as will be shown in *Generalization to higher dimensions*, but is restricted to second-order central schemes in space.

The probability of success for a time step is

$$p_t = \langle \phi_t | A^\dagger A | \phi_t \rangle = \frac{\|\phi_{t+1}\|^2}{\|\phi_t\|^2}, \quad (17)$$

where vector notation denotes the unnormalized solution. This results in a cumulative probability of $\|\phi(T)\|^2/\|\phi(0)\|^2$ for the entire simulation time T . For a continuous variable $\phi(x, t)$ defined in $x \in \Omega$ by considering $\Delta x \rightarrow 0$, the probability of success becomes

$$\frac{\int_\Omega |\phi(x, T)|^2 dx}{\int_\Omega |\phi(x, 0)|^2 dx}, \quad (18)$$

which is intrinsic to the physical problem being solved rather than an artifact of numerical discretization.

The discussed probability of success can be improved to $O(\|\phi(T)\|/\|\phi(0)\|)$ with uniform singular value amplification [16] by removing the need for intermediate measurements [18]. However, this is at the cost of a quadratic gate complexity in T , so its inclusion is therefore case dependent.

Generalization to higher dimensions. Applying the methodology to d spatial dimensions requires a shift operator corresponding to each spatial dimension for a linear combination of $d + 1$ unitary operators. The quantum circuit for $d = 3$ is shown in Fig. 1(b). The normalization in Eq. (9) generalizes to $1 - 2dr_h$. Because of the need to encode $d + 1$ coefficients,

the first column of the unitary V now consists of

$$V_{i,0} = \sqrt{\frac{\kappa_i}{\sum_{j=0}^d \kappa_j}} \quad (19)$$

for rows $i = 0$ to d , where $\kappa_0 = 1$ corresponds to the advection term, and $\kappa_1, \dots, \kappa_d = 2r_h/(1 - 2dr_h)$ correspond to the d spatial dimensions. Any excess coefficients, κ_{d+1} onward, are set to zero with the corresponding shift operators excluded from the circuit. The remaining columns form an orthonormal basis such that V is unitary. In d dimensions, the quantum algorithm requires $n = \lceil \log_2(d+1) \rceil + 1$ ancilla qubits and maps

$$\begin{aligned} |0\rangle^{\otimes n} |\phi_t\rangle &\mapsto |0\rangle^{\otimes n} \frac{\kappa_0 \hat{A} + \sum_{j=1}^d \kappa_j S_j}{\sum_{j=0}^d \kappa_j} |\phi_t\rangle + \dots \\ &\mapsto |0\rangle^{\otimes n} A |\phi_t\rangle + \dots \end{aligned} \quad (20)$$

Evidently from the definitions of κ_i , the property of $\sum_{j=0}^d \kappa_j = 1/(1 - 2dr_h)$ holds for all positive integers of d , ensuring that the intrinsic probability of success in Eqs. (17) and (18) applies in every multidimensional setting.

Boundary conditions. The proposed algorithm can be extended to Neumann and Dirichlet boundary conditions by two distinct methodologies, each with their advantages and disadvantages. The first methodology is the simulation of an extended domain with even or odd symmetry at the boundaries, and the second methodology is the inclusion of additional unitaries into the LCU sum to modify the boundary values of A .

Neumann and Dirichlet boundary conditions can be simulated within a subspace of the periodic domain by reflection with even and odd symmetry at the subspace boundary, respectively. This method is widely used in the numerical simulation of PDEs using spectral methods, for example by the discrete sine transformation that naturally corresponds to a Dirichlet condition, or the discrete cosine transformation that naturally corresponds to a Neumann condition [21]. The operation is inherently unitary and requires one additional ancilla qubit per nonperiodic spatial dimension to perform the reflection, as shown by Sano and Hamamura [22]. At the end of the simulation, the symmetric subdomains can simply be discarded by uncomputation. For example, the quantum circuit for implementing homogeneous Neumann boundary conditions for the one-dimensional heat equation is shown in Fig. 1(c). The linear complexity in the simulation time is retained by this approach.

Alternatively, the methodology can be extended to nonperiodic boundary conditions by including additional pairs of unitary operators into the LCU sum that, with the appropriate coefficients, adjust the boundary rows of A to implement the desired condition. For example, implementing an insulated boundary corresponds to a homogeneous Neumann condition, and results in a first boundary row of A of $[1 - r_h, r_h, 0, \dots, 0]$ for the diffusion terms in one dimension. This can be implemented using two pairs of unitary operators. The first pair contains an identity operator and a negative identity operator with boundary values of 1, which sum to a zero matrix with boundary values of 2. Therefore, including the

weighting of $r_h/2$ in the LCU sum will add r_h to the boundary value, effectively converting it from $1 - 2r_h$ to $1 - r_h$. This must be repeated for the top-right and bottom-left corner elements of the matrix, by subtracting the r_h term that implements periodicity. The inclusion of such unitaries for Dirichlet and Neumann conditions using this methodology does not retain the direct encoding described in the previous section. This can be overcome with the methodology of Fang *et al.* [18] by applying a uniform singular value amplification [16] to remove the unwanted $\alpha > 1$ factor. Hence, the above methodology with nonperiodic boundary conditions can be considered as a specific implementation of the algorithm by Fang *et al.* [18]. This approach's tradeoff for nonperiodic problems is a quadratic scaling in simulation time. Since the method of reflection discussed in the previous paragraph retains the linear complexity in the simulation time, it will be superior for most applications.

Complexity. Consider the simulation of a d -dimensional spatial domain discretized with $N = N_x^d$ grid points. The simulation is evolved over a time interval $T = N_T \Delta t$ using a number of time steps N_T with a constant time step Δt . This requires $O(\log N + \log d)$ qubits, $\tilde{O}(TN^{2/d}d/\epsilon)$ two-qubit gates, and $O(\|\phi(0)\|^2/\|\phi(T)\|^2)$ attempts for a successful run, where \tilde{O} is the complexity with the suppression of polylogarithmic terms and $\phi(t)$ is the unnormalized solution at time t . The probability of success and qubits requirements were demonstrated in *Linear combination of unitaries* and *Generalization to higher dimensions* respectively, so now the two-qubit gate complexity will be derived. The algorithm requires the repeated application of the quantum circuits in Fig. 1 for the desired number of time steps, so the gate complexity grows linearly with the desired number of time steps $N_T = O(TN^{2/d}/\epsilon)$. The number of time steps growing linearly with the desired simulation time T is apparent. The $N^{2/d}$ component arises from the limiting stability condition r_h that constrains $\Delta t \propto (\Delta x)^2$ and $N_T \propto N_x^2$, and since $N = N_x^d$, then $N_T \propto N^{2/d}$. The $1/\epsilon$ component arises from the inverse relationship $N_T \propto 1/\Delta t$ and that $\epsilon \propto \Delta t$, resulting from the error ϵ of the underlying Euler method and the accuracy to which \hat{A} is block encoded using the advection algorithm [19]. Each time step requires the implementation of a Hamiltonian simulation, where optimal algorithms [15] require a number of two-qubit gates that grow linearly with the sparsity of the Hamiltonian H , the largest absolute element of H and the evolution time. The algorithms are at least polylogarithmic in their parameters, and these terms will be omitted through the \tilde{O} notation. In the context of the Hamiltonian simulation in *Quantum advection algorithm*, the sparsity is $s = 1 + 2d = O(d)$ from the finite difference method, the largest absolute element $\|H\|_{\max} = 1$ and the evolution time $\theta = \pi/2$, which are both $O(1)$. This completes the derivation of the overall two-qubit gate complexity of $\tilde{O}(TN^{2/d}d/\epsilon)$. This contrasts with the typical classical complexity of $O(NN_T) = O(TN^{(2+d)/d})$, where the classical error dependence has been omitted due to the prevalence of high-order numerical methods in classical computing [23]. Comparing the expressions, the quantum algorithm provides a polynomial speedup by reducing the exponent of N by a factor of $2/(2+d)$.

Simulations. In this section, the quantum algorithm is applied to state-vector simulations of passive scalar transport in

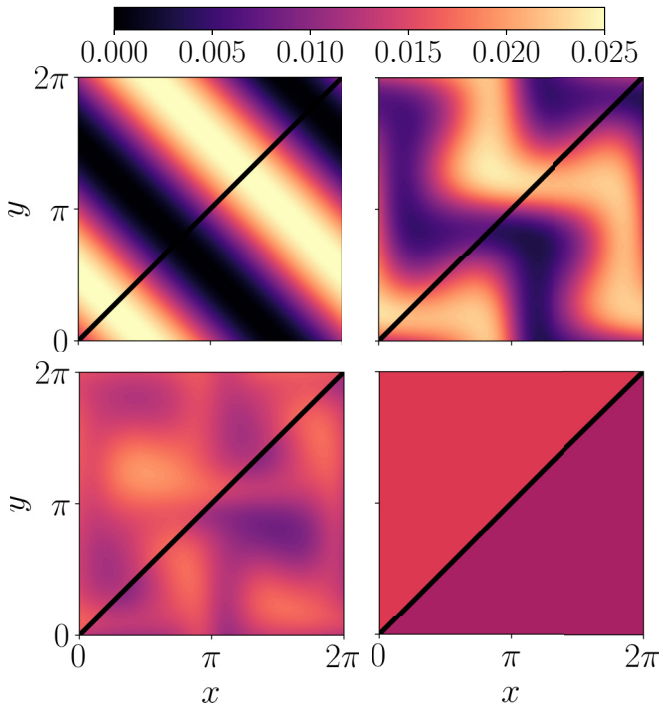


FIG. 2. Contours of the scalar evolving by state-vector simulation (normalized, top-left half) and the corresponding classical solution (unnormalized, bottom-right half) for $t/T = 0, 0.1, 0.2,$ and 1 in left-to-right, top-to-bottom order.

a steady two-dimensional flow field that has characteristics of a Taylor-Green vortex with velocity

$$\mathbf{v}(x, y) = \begin{bmatrix} \sin(x) \cos(y) \\ -\cos(x) \sin(y) \end{bmatrix} \quad (21)$$

in a periodic domain $x, y \in [0, 2\pi)$. The domain is discretized by $64 \times 64 = 4096$ grid points corresponding to 15 qubits including three ancilla qubits. The simulation time T is divided into $N_T = 1400$ time steps with the maximum advection stability parameter $\max(r_a) = \max(|v_x| + |v_y|)\Delta t/\Delta x = 0.1$ and the uniform diffusion stability parameter $r_h = 0.1$. This results in an adequate temporal resolution by ensuring at least ten time steps are taken for the fluid to traverse the spatial resolution Δx by advection, and a diffusion stability parameter that satisfies the condition of $r_h < 1/(2d)$, derived from Fourier–von Neumann stability analysis [24]. It was shown by Brearley and Laizet [19] that r_a is not subject to the conditional stability of explicit methods ($r_a \leq 1$) due to the bounding of the spectral radius in the block encoding of a nonunitary operator, although the condition is met regardless. The scalar $\phi(x, y)$ is initialized proportionally to $\sin(x + y) + 1$ with a norm of 1, defined as a non-negative function of both spatial coordinates to be an effective tracer of the two-dimensional velocity field, which may represent a physical quantity such as a species concentration. Figure 2 shows the evolution of $\phi(x, y)$ by the described quantum algorithm and by the classical action of A directly. The classical simulation converges to a constant steady state of $\text{mean}[\phi(0)]$, while the quantum simulation converges to $1/\sqrt{N}$ due to the requirements of norm preservation. The Péclet number,

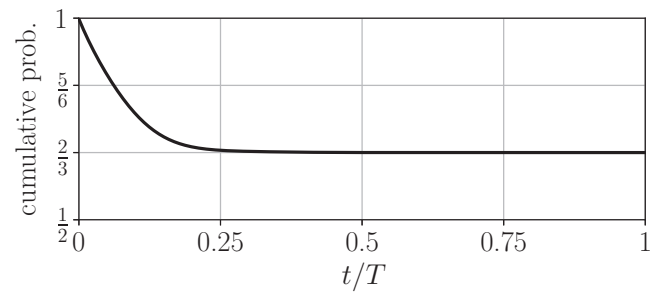


FIG. 3. Cumulative probability of algorithm success, equivalent to $\|\phi(t)\|^2/\|\phi(0)\|^2$, demonstrating convergence to the theoretical value of $2/3$.

defined as the ratio of the advective to diffusive transport, is $\text{Pe} = \mathbf{v}_{\text{rms}}\pi/D = 23$ for root-mean-square velocity $\mathbf{v}_{\text{rms}} = 1/\sqrt{2}$ and characteristic length π , indicating advection as the primary mode of scalar transport with the influence of diffusion occurring over longer timescales. The quantum algorithm closely approximates the action of A with a mean-squared error compared against the normalized classical solution not exceeding 0.5%, given as a percentage of $\max(|\phi_t|^2)$. This is comparable to other more general algorithms for solving the advection-diffusion equation [25]. Figure 3 shows the cumulative probability of measurement success for the simulation, demonstrating that the algorithm success probability approaches the intrinsic value of $\|\phi(T)\|^2/\|\phi(0)\|^2 = 2/3$ for these initial conditions.

Conclusion. The presented quantum algorithm simplifies the solution of diffusion and scalar transport problems by providing an accurate, direct block encoding of the explicit time-marching operator, achieving the intrinsic success probability of the squared solution norm.

For periodic computational domains, the algorithm achieves a linear simulation time dependence, improving over other algorithms such as the linear combination of Hamiltonian simulation [5] and time-marching algorithms using amplitude amplification [18], while retaining their advantageous nonvanishing success probabilities. The proposed algorithm offers an improvement over approaches based on quantum linear systems solvers, e.g., by Berry *et al.* [2], as there is no ϵ dependence in the number of queries to the state initialization oracle [10]. Simulating d -dimensional problems requires a linear combination of $d + 1$ unitary operators to recover the unscaled time-marching operator, improving over general methods requiring four unitaries that asymptotically approach the target matrix [12], with diminishing success probabilities over successive time steps. Moreover, the presented quantum algorithm is efficient in its usage of ancillary qubits, with requirements that increase logarithmically with the number of spatial dimensions and independently of the simulation time, unlike the other approaches [5, 18].

Nonperiodic boundaries can be implemented by including additional unitary operators into the LCU circuit (see *Boundary conditions*). The algorithm, combined with amplitude amplification [16], then becomes a specific implementation of the general time-marching algorithm [18] for advection-diffusion problems. The probability of success improves to $O[\|\phi(T)\|/\|\phi(0)\|]$, but at the cost of a quadratic $O(T^2)$ scaling in the evolution time. Alternatively, nonperiodic

conditions can be implemented by introducing symmetry along the computational boundary, with parity that is dependent on the specific boundary condition. This approach retains the linear $O(T)$ scaling in the evolution time.

The algorithm is limited to the forward Euler method with a second-order central discretization of the spatial derivatives, although the latter is not so restrictive given the capacity to simulate extremely large computational domains on quantum computers.

The general methodology of matrix decomposition, as presented in *Matrix decomposition*, may have varied applications extending beyond scalar transport problems. As a

potential candidate due to its similar mathematical structure, the Fokker-Planck equation opens up potential applications in stochastic fields such as financial modeling, statistical mechanics, and chemical kinetics. Exploring these diverse applications represents an interesting premise for future research.

Acknowledgments. P.O., S.B., and T.R. are supported by the European Union's Horizon Europe Research, HORIZON-CL4-2021-DIGITAL-EMERGING-02-10, Grant No. 101080085. P.B. and S.L. are supported by Engineering and Physical Sciences Research Council Grant No. EP/W032643/1.

-
- [1] J. P. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. J. Mavriplis, CFD vision 2030 study: A path to revolutionary computational aerosciences, Nasa Report No. NASA/CR-2014-218178, NF1676L-18332, <https://ntrs.nasa.gov/citations/20140003093>.
- [2] 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).
- [3] H. Krovi, Improved quantum algorithms for linear and nonlinear differential equations, *Quantum* **7**, 913 (2023).
- [4] D. W. Berry and P. C. Costa, Quantum algorithm for time-dependent differential equations using Dyson series, *Quantum* **8**, 1369 (2024).
- [5] 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).
- [6] I. Novikau and I. Joseph, Quantum algorithm for the advection-diffusion equation and the Koopman-von Neumann approach to nonlinear dynamical systems, *Comput. Phys. Commun.* **309**, 109498 (2025).
- [7] S. Jin, N. Liu, and Y. Yu, Quantum simulation of partial differential equations: Applications and detailed analysis, *Phys. Rev. A* **108**, 032603 (2023).
- [8] J. Hu, S. Jin, N. Liu, and L. Zhang, Quantum circuits for partial differential equations via Schrödingerisation, *Quantum* **8**, 1563 (2024).
- [9] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* **103**, 150502 (2009).
- [10] P. C. S. 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).
- [11] A. Mezzacapo, M. Sanz, L. Lamata, I. Egusquiza, S. Succi, and E. Solano, Quantum simulator for transport phenomena in fluid flows, *Sci. Rep.* **5**, 13153 (2015).
- [12] S. S. Bharadwaj and K. R. Sreenivasan, Compact quantum algorithms for time-dependent differential equations, *Phys. Rev. Res.* **7**, 023262 (2025).
- [13] C. Sanavio, E. Mauri, and S. Succi, Explicit quantum circuit for simulating the advection-diffusion-reaction dynamics, *IEEE Transactions on Quantum Engineering* **6**, 1 (2025).
- [14] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, *Quantum Inf. Comput.* **12**, 901 (2012).
- [15] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, *Quantum* **3**, 163 (2019).
- [16] 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)* (ACM Press, New York, NY, USA, 2019), pp. 193–204.
- [17] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, *Phys. Rev. Lett.* **118**, 010501 (2017).
- [18] D. Fang, L. Lin, and Y. Tong, Time-marching based quantum solvers for time-dependent linear differential equations, *Quantum* **7**, 955 (2023).
- [19] P. Brearley and S. Laizet, Quantum algorithm for solving the advection equation using hamiltonian simulation, *Phys. Rev. A* **110**, 012430 (2024).
- [20] R. M. Gingrich and C. P. Williams, Non-unitary probabilistic quantum computing, in *Proceedings of the Winter International Symposium on Information and Communication Technologies, Cancun, Mexico, January 05* (JPL Open Repository, Pasadena, CA, 2004).
- [21] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics* (Springer, New York, 2007).
- [22] Y. Sano and I. Hamamura, Quantum state preparation for probability distributions with mirror symmetry using matrix product states, [arXiv:2403.16729](https://arxiv.org/abs/2403.16729).
- [23] S. Laizet and E. Lamballais, High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy, *J. Comput. Phys.* **228**, 5989 (2009).
- [24] J. G. Charney, R. Fjørtoft, and J. von Neumann, Numerical integration of the barotropic vorticity equation, *Tellus A* **2**, 237 (1950).
- [25] J. Ingelmann, S. S. Bharadwaj, P. Pfeffer, K. R. Sreenivasan, and J. Schumacher, Two quantum algorithms for solving the one-dimensional advection–diffusion equation, *Computers and Fluids* **281**, 106369 (2024).