# Tensor-programmable quantum circuits for solving differential equations

Pia Siegl[1,2,*,†] Greta Sophie Reese[1,3,4,*,‡] Tomohiro Hashizume[1,4] Nis-Luca van Hülst[1] and Dieter Jaksch[1,4,5]

[1]*Institute for Quantum Physics, University of Hamburg, Luruper Chaussee 149, 22761 Hamburg, Germany*
[2]*Institute of Software Methods for Product Virtualization, German Aerospace Center (DLR), Nöthnitzer Straße 46b, 01187 Dresden, Germany*
[3]*Center for Optical Quantum Technologies, University of Hamburg, 22761 Hamburg, Germany*
[4]*The Hamburg Centre for Ultrafast Imaging, Hamburg, Germany*
[5]*Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, United Kingdom*

(Received 12 February 2025; accepted 8 December 2025; published 20 January 2026)

We present a quantum solver for partial differential equations based on a flexible matrix product operator representation. Utilizing midcircuit measurements and a state-dependent norm correction, this scheme overcomes the restriction of unitary operators. Hence, it allows for the direct implementation of a broad class of differential equations governing the dynamics of classical and quantum systems. The capabilities of the framework are demonstrated for linear and nonlinear partial differential equations using the example of the linearized Euler equations with absorbing boundaries and the nonlinear Burgers' equation. For a turbulence data set, we demonstrate potential advantages of the quantum-tensor scheme over its classical counterparts.

DOI: 10.1103/2qzh-yf49

## I. INTRODUCTION

Solving partial differential equations (PDEs) is a core task in many research and industry areas, ranging from the financial sector [1,2] and material science [3,4] to computational fluid dynamics [5–7]. Despite the enormous amount of resources nowadays available in classical computing, solving PDEs remains a challenge. One example is computational fluid dynamics, where resolving all relevant spatial scales quickly demands billions of data points [8], and approximations and the use of models become mandatory [9–13].

Quantum computers offer an efficient representation of classical data, as the number of qubits needed for amplitude encoding scales logarithmically with the number of data points [14–17]. To solve PDEs with quantum computers different approaches have been proposed: (1) algebraic quantum linear solvers as the Harrow-Hassidim-Lloyd (HHL) [18] algorithm and its extensions [19–22]; (2) specific PDEs were solved efficiently with discrete time-stepping schemes [23,24]; (3) variational quantum algorithms (VQAs) [25–27] that rely on a hybrid scheme combining parameterized quantum circuits and a classical parameter optimization.

Despite the rapid advancement in quantum hardware and error correction [28–30] and the promises for near-term devices with significant numbers of logical qubits [31,32], quantum linear solvers are expected to stay expensive or even unfeasible due to the large demand in resources [21] and their unfavorable scaling with the stiffness of the problem [18]. VQAs instead are characterized by shallow circuit structures, are predicted to exhibit beneficial scaling [15,21,33], and were successfully applied in various areas [26,33–35]. While noisy hardware can limit the accuracy of VQAs [36], noisy optimization is possible [37–39] and strategies like circuit recompilation [40,41] can significantly reduce the sensitivity to noise [42].

In addition to quantum computing, so called quantum inspired methods are under increasing attention as differential solvers. A prominent example is matrix product states (MPSs) [43,44] that exhibit a low-rank representation of many functions [45] and have proven successful in solving PDEs on classical hardware, showing potential to compete with conventional solvers and delivering remarkable results across various applications [15,46–52].

MPS methods show great promise when combined with VQAs. While the efficiency of MPS is limited to solutions with bounded entanglement [15,53], quantum circuits were shown to feature an exponential reduction in the number of variables parametrizing the solution [33] for certain use cases. Further, MPS algorithms scale at least polynomially better when ported to quantum computers [15,33], offering at least the same speedup as Grover's algorithm for unstructured search [54] and defining an upper bound of the circuit depths for state encoding. In combination with known methods to encode MPS with quantum circuits [55–57], transferring matrix product operators (MPOs) [58,59] is an important step to fully translate MPS-based algorithms onto quantum circuits.

Mapping classical PDEs on quantum computers demands mimicking the effect of nonlinear and nonunitary

[*]These authors contributed equally to this work.
[†]Contact author: pia.siegl@dlr.de
[‡]Contact author: greta.reese@uni-hamburg.de

dynamics by linear and unitary quantum operations. Lubasch *et al.* significantly advanced the field of quantum differential solvers (QDSs) by introducing a VQA that solves nonlinear PDEs [33] relevant in classical and quantum physics. This strategy offers an efficient classical parametrization of the solution, which allows to circumvent the read-out problem for each time step and enables the computation of different observables like velocity moments, coarse-grained solutions, or spatial means without rerunning the full time evolution [33,60,61]. While it was used to solve core fluid dynamic problems as the Burgers' equation [16,62] and was extended [63] to various boundary conditions [64] and space-time methods [37], it is limited to PDEs that directly map onto known quantum operators. As an additional drawback, cost functions are build up from numerous contributions where each requires a quantum circuit that needs to be measured individually. Furthermore, the cost function is not bounded, making it difficult to estimate the training progress and accuracy without comparing with a classical solution. Going to generic PDEs requires an entirely different approach that we will introduce in the following and apply to a linear and a nonlinear PDE.

In this article, we introduce a tensor-programmable variational quantum algorithm which utilizes the operator representation as MPO-based quantum circuits and has several advantages over previous approaches. First, it allows for the incorporation of nonunitary operators, extending the range of PDEs and solution techniques on quantum computers. Second, the number of quantum circuits $M$, required to build up the cost function, can be significantly reduced compared to previously introduced schemes. In general, all terms of the PDE can be summarized within one quantum circuit, allowing to infer the convergence of the training progress directly from one expectation value. Furthermore, it opens the path to a broadly applicable and modular scheme for solving problems in a wide range of scientific and industrial fields.

This paper is structured as follows. Section II introduces the tensor-programmable quantum scheme, starting with a general overview to then address the details on the operator mapping in Sec. II A and iterations steps in Sec. II B. Furthermore, it explains the necessary norm correction in Sec. II C and introduces an adaption of the Hadamard test and a convergence measure for the optimization procedure in Sec. II D. Section II E extends the introduced scheme to nonlinear differential equations. In Sec. III, we apply the introduced scheme to the linearized Euler equations (Sec. III A) and the nonlinear Burgers' equation (Sec. III B), while a third use case, the linear advection-diffusion equation is provided in Appendix D to show the successful application when using a larger number of qubits. Section IV presents a comparison of this scheme with classical MPS-based algorithms at the example of a turbulent dataset and considers the scaling of the operator. A conclusion is given in Sec. V .

## II. METHODS

In this section, we explain the tensor-programmable quantum scheme for linear and nonlinear PDEs. For simplicity, we first give a general overview of the method for linear PDEs, which is depicted in Fig. 1, together with the simulation results. We deepen the discussions in the following subsections and extend the scheme to nonlinear PDEs in Sec. II E.

The solution $\phi(\boldsymbol{x}, j)$ on a discretized grid at iteration step $j$ is amplitude encoded into a quantum register composed of $n$ qubits [65]. This state is generated by a quantum gate $\hat{U}(\boldsymbol{\theta}_j)$ [see Fig. 1(a)] that is classically parametrized by a real vector $\boldsymbol{\theta}_j$ and an additional real number $\theta_j^0$ setting the norm of $\phi(\boldsymbol{x}, j)$.

Here, we focus on uniform discretizations in one spatial dimension for simplicity. The extension to higher dimensions and nonuniform grids is conceptually straightforward. In one dimension, the state on the quantum register is given by $\theta_j^0 |\psi_j\rangle = \sum_{l=0}^{2^n-1} \phi(x, j) |x_b\rangle$, where $x_b$ is the binary form of $x$, with $|x_b\rangle$ representing the computational basis states of the $n$ qubit quantum register. Therefore, the vector $\boldsymbol{\theta}_j$ provides a classical representation of the solution, which is exponentially compressed for restricted circuit depths [33]. To encode the initial state into a quantum state, methods like the efficient approximate encoding of MPS via shallow quantum circuits [66] can be used if the initial state is not trivial.

The evolution of the system by one step is characterized by an operator $\hat{O}$ with $\theta_{j+1}^0 |\psi_{j+1}\rangle = \hat{O}\theta_j^0 |\psi_j\rangle$. We determine $\boldsymbol{\theta}_{j+1}$ by solving a problem-dependent cost function $\mathcal{C}$ that is proportional to the overlap $\mathcal{C} \propto -\langle 0| \hat{U}^\dagger(\boldsymbol{\theta}_{j+1})\hat{O} |\psi_j\rangle$. The overlap is measured via an adapted Hadamard test [Fig. 1(a), green box] by evaluating $\langle\sigma_z\rangle_{\text{anc}}$ of a global ancilla qubit at the end of the quantum circuit. The overlap is fed back to a classical computer that variationally updates the parameter vector $\boldsymbol{\theta}_{j+1}$ until a predefined convergence criterion is reached.

The operator $\hat{O}$ can summarize the terms of the PDE in one quantum circuit, resulting in a single-cost term or split them into several contributions. The most suited strategy should be chosen in dependence on the PDE and the available quantum hardware resources. The operator is in general nonunitary and implemented probabilistically [Fig. 1(a), purple box]. The success probability $\alpha_{\text{succ}}$ of the operator application is fed back to the classical computer to compute a norm correction, necessary to obtain the normalization constant $\theta_{j+1}^0$.

### A. Operator mapping

To realize the nonunitary operator $\hat{O}$, we utilize the operator representation in terms of MPOs. Many discretized differential operators exhibit a low-rank MPO representation with small bond dimension $\zeta$ [47,67,68], including derivatives of higher order accuracy and various boundary conditions. A collection of relevant MPOs for the considered PDEs is given in Appendix B . The classical MPO is translated into a set of unitaries $\hat{U}_{\text{MPO}}$ with an algorithm proposed by Termanova *et al.* [59], which is outlined in Appendix C. The algorithm introduces an MPO consisting out of isometric tensors to approximate the original operator up to a multiplicative constant $c_{\text{MPO}}$ and with relative approximation error $E$ [59] . This isometric MPO requires a larger bond dimension $Z > \zeta$, compensating for the reduced degrees of freedom due to the isometric constraints. This bond dimension defines an auxiliary qubit register of size $n_{\text{aux}} = \log_2(Z)$ [59]. The isometric MPO is converted into unitary gates $\hat{U}_{\text{MPO}}$, which are part of the quantum circuit in Fig. 1(a). Subsequent midcircuit measurements and postselection are employed to ensure that

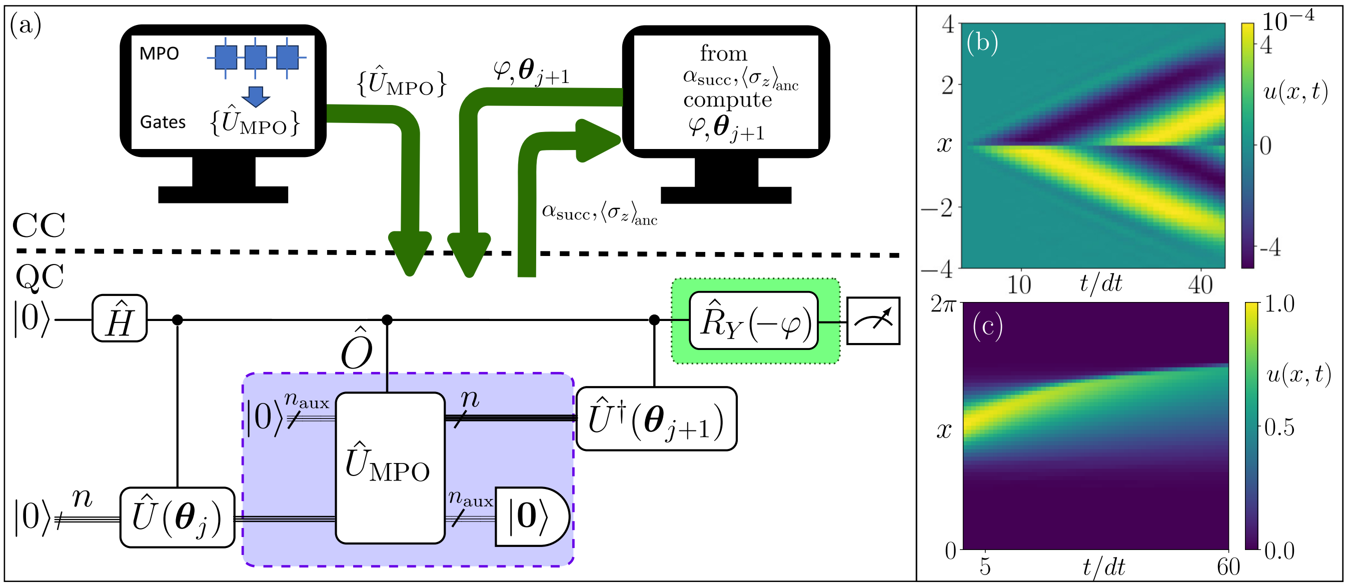
(a)
MPO
Gates
$\{\hat{U}_{\rm MPO}\}$
$\{\hat{U}_{\rm MPO}\}$
$\varphi, \boldsymbol{\theta}_{j+1}$
from
$\alpha_{\rm succ}, \langle\sigma_z\rangle_{\rm anc}$
compute
$\varphi, \boldsymbol{\theta}_{j+1}$
$\alpha_{\rm succ}, \langle\sigma_z\rangle_{\rm anc}$
CC
QC
$|0\rangle$
$\hat{H}$
$\hat{O}$
$\hat{R}_Y(-\varphi)$
$|0\rangle$ $n_{\rm aux}$
$n$
$\hat{U}_{\rm MPO}$
$\hat{U}^\dagger(\boldsymbol{\theta}_{j+1})$
$|0\rangle$ $n$
$\hat{U}(\boldsymbol{\theta}_j)$
$n_{\rm aux}$
$|\mathbf{0}\rangle$
(b)
$u(x,t)$
$10^{-4}$
4
0
-4
$x$
$t/dt$
(c)
$u(x,t)$
1.0
0.5
0.0
$2\pi$
0
5
60

FIG. 1. (a) Hybrid quantum-classical routine to solve linear PDEs iteratively in a variational manner. The computation of the unitary gates $\hat{U}_{\rm MPO}$ representing the operator $\hat{O}$ as well as the optimization routine take place on a classical computer (CC, upper part). The overlap $\langle 0|\,\hat{U}^\dagger(\boldsymbol{\theta}_{j+1})\hat{O}\hat{U}(\boldsymbol{\theta}_j)\,|0\rangle$ that determines the cost function, necessary to compute the solution at the next iteration step is computed on the quantum computer (QC, lower part) using an adapted Hadamard test. The angles $\boldsymbol{\theta}_j$ describe a previous iteration step, while $\boldsymbol{\theta}_{j+1}$ are to be determined during the classical optimization process. This procedure allows to map classical differential solvers to the quantum computer. (b) Solution of the Euler equations with our QDS for 45 time steps $dt$ encoded into six ansatz qubits. Depicted is the discretized velocity $u(x,t)$ evolution over time that arises, which is induced by a periodic pressure point source at $x = 0$. (c) Solution of the nonlinear Burgers' equations. Depicted is the discretized velocity field $u(x,t)$ over time, where initial Gauss peak evolves into a shock wave. A detailed description of all system and training parameters is given in Appendix A.

the operation corresponds to the actual MPO. Only those instances are kept where the auxiliary register is measured in the state $|\mathbf{0}\rangle_{\rm aux}$.

Tracking the number of successful and total runs determines the success probability $\alpha_{\rm succ}$ of the postselection during the cost function evaluation. No further quantum circuit is required. An alternative method to estimate $\alpha_{\rm succ}$ from the training parameter $\varphi$ is described later in this paper. Importantly, $\alpha_{\rm succ}$ was shown to have favorable magnitude and scaling for various examples [59], which can be extended to all relevant differential operators in this work as shown in Appendix E. In contrast to other approaches [69], there is no exponential decay of the overall success probability of the algorithm with the number of iteration steps $j$.

### B. Computing iteration steps

We identify the parameters defining the next iteration step $j+1$ by solving a problem-dependent cost function $\mathcal{C}$. Using the parameters from the previous iteration step as the initial guess for the optimization simplifies the process, as they are typically close to the solution of the current iteration step. This closeness leads to a significantly improved trainability even in the presence of shots and larger qubit numbers [70] (cf. Appendix F).

### C. Norm correction

As the operators are in general nonunitary, the computation of the normalization constant $\theta^0_{j+1}$ requires to incorporate several correction factors. One is the constant $c_{\rm MPO}$, computed in the creation of the isometric MPO. Furthermore, there is the state- and operator-dependent norm constant $f_{\hat{O},j}$ that accounts for the difference stemming from casting a non-norm-conserving operator into a norm-conserving form on the quantum computer. The necessary correction for each iteration step $j$ can be computed from $\alpha_{\rm succ}$ as

$$f_{\hat{O},j} = \frac{1+\alpha_{\rm succ}}{2\sin(\varphi) - \left(\sqrt{\alpha_{\rm succ}} - \frac{1}{\sqrt{\alpha_{\rm succ}}}\right)\cos(\varphi)}, \tag{1}$$

where $\varphi \in (0, \pi/2]$ is the rotation angle of the $\hat{R}_Y$ gate on the global ancilla qubit [cf. Fig. 1 (a)], which has an optimal value for each iteration step $j$. Then, the effect of the operator $\hat{O}$ can be computed using $\hat{O}\theta^0_j\,|\psi_j\rangle\,|\mathbf{0}\rangle_{\rm aux} = c_{\rm MPO}f_{\hat{O},j}P_{|0\rangle_{\rm aux}\langle 0|_{\rm aux}}\hat{U}_{\rm MPO}\theta^0_j\,|\psi_j\rangle\,|\mathbf{0}\rangle_{\rm aux}$, allowing for a correct estimation of $\theta^0_{j+1}$. Here, $|\rangle_{\rm aux}$ denotes the auxillary qubit register for the operator application and $P_{|0\rangle_{\rm aux}\langle 0|_{\rm aux}}$ denotes the projector of these qubits on $|0\rangle$.

The overlap $\langle 0|\,\hat{U}^\dagger(\boldsymbol{\theta}_{j+1})\hat{O}\,|\psi_j\rangle$ in the cost function $\mathcal{C}$ can be computed using the measurement result $\langle\sigma_z\rangle_{\rm anc}$ of a Hadamard test [33]. There, a global ancilla qubit controls the applications of the ansätze $\hat{U}^\dagger(\boldsymbol{\theta}_{j+1})$ and $\hat{U}(\boldsymbol{\theta}_j)$ and the operator $\hat{U}_{\rm MPO}$, and is measured in the computational bases at the end of the circuit [cf. Fig. 1(a)].

### D. Adapted Hadamard test and convergence measure

If the success probability $\alpha_{\rm succ}$ is smaller than one, the probabilistic application of the MPO has a negative impact

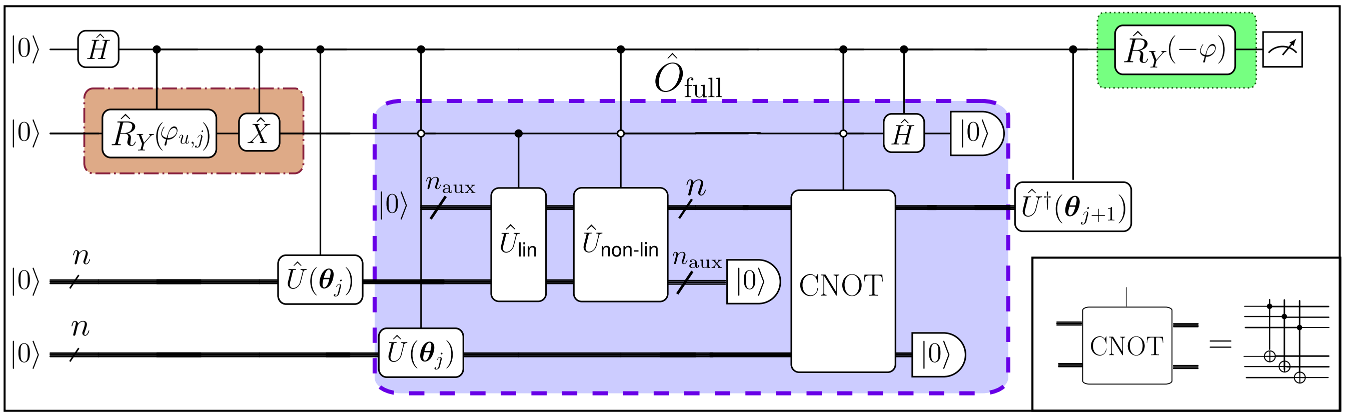
$|0\rangle$
$\hat{H}$
$\hat{R}_Y(\varphi_{u,j})$
$\hat{X}$
$\hat{O}_{\text{full}}$
$n_{\text{aux}}$
$\hat{U}_{\text{lin}}$
$\hat{U}_{\text{non-lin}}$
$n$
$\hat{U}(\boldsymbol{\theta}_j)$
CNOT
$\hat{U}^\dagger(\boldsymbol{\theta}_{j+1})$
$\hat{R}_Y(-\varphi)$

FIG. 2. Quantum circuit to encode one time step of a nonlinear PDE. Linear and nonlinear operators are separated into two MPOs and an additional auxillary qubit is introduced allowing to create a weighted superposition of the linear and nonlinear contributions. Double control of the global ancilla qubit and the extra auxiliary qubit can be avoided by applying CNOT gates before and after the nonlinear operators, where the global ancilla acts as control and the extra auxiliary qubit as target. Then, the control on the global auxillary qubit can be omitted. The concrete decomposition of the CNOT gate is shown for the example of $n = 3$.

on the Hadamard test. It stems from an increased contribution of $|0\rangle_{\text{anc}}$, which is always successful compared to $|1\rangle_{\text{anc}}$, where runs may be discarded, compared right after the application of the operator. This imbalance causes an increased variance of $\langle\sigma_z\rangle_{\text{anc}}$, raising the number of shots required to determine the cost term with a given accuracy. This nonoptimal behavior can be mitigated with an adaption of the standard Hadamard test as shown in Fig. 1(a) (green box). We substitute the second Hadamard gate by an angle-dependent rotation gate $\hat{R}_Y(-\varphi)$. The rotation angle that maximizes $\langle\sigma_z\rangle_{\text{anc}}$ depends on $\alpha_{\text{succ}}$ (cf. Appendix G). For $\alpha_{\text{succ}} = 1$, the optimal angle $\varphi = \pi/2$ restores the Hadamard gate.

When all operators of the PDE are summarized within one quantum circuit and the parameters $\boldsymbol{\theta}_{j+1}$ are well trained, there is an optimal

$$\varphi_{\text{opt}} = 2\arctan(\sqrt{\alpha_{\text{succ}}}), \tag{2}$$

which results in $\langle\sigma_z\rangle_{\text{anc}} = 1$. While optimizing $\boldsymbol{\theta}_{j+1}$, the training progress can be tracked with the fidelity $\mathcal{F} = ||\langle\mathbf{0}|\hat{U}^\dagger(\boldsymbol{\theta}_{j+1})P_{|0\rangle_{\text{aux}}\langle 0|_{\text{aux}}}\hat{U}_{\text{MPO}}|\psi_j\rangle||^2$. Combining all operators in one quantum circuit allows to infer the fidelity directly from $\langle\sigma_z\rangle_{\text{anc}}$ according to (cf. Appendix G for derivation details)

$$\mathcal{F} = \frac{(\alpha_{\text{succ}}\langle\sigma_z\rangle_{\text{anc}} + \alpha_{\text{succ}}\cos(\varphi_{\text{opt}}) + \langle\sigma_z\rangle_{\text{anc}} - \cos(\varphi_{\text{opt}}))^2}{4\alpha_{\text{succ}}\sin^2(\varphi_{\text{opt}})}. \tag{3}$$

Then, the fidelity can act as a convergence measure that reflects the status of the training without the need for verification with a classical solution. This is a significant advantage compared with previous approaches [33,64], where each term is treated individually and this knowledge on the solutions convergence cannot be easily inferred. The parameter $\varphi_{\text{opt}}$ either can be computed from the estimated success probability, by counting the number of discarded shots during the training process, or can be trained itself. If $\varphi$ is trained and the problem is described by one quantum circuit, $\alpha_{\text{succ}}$ can be computed from Eq. (2). As all trainable parameters are placed after the postselection procedure, the parameter-shift rule [71,72] or coherent gradient approximations [26,73] apply.

### E. Extension to nonlinear differential equations

To make the quantum-tensor scheme applicable beyond linear PDEs, we consider differential equations of the form

$$u^{j+1} = (\hat{O}_{\text{lin}} + u^j\hat{O}_{\text{nonlin}})u^j = \hat{O}_{\text{full}}u^j, \tag{4}$$

where $\hat{O}_{\text{lin}}$ and $\hat{O}_{\text{nonlin}}$ are both linear operators. Prominent examples of such PDEs are the Burgers' equation and the Navier-Stokes equations. We cannot summarize $\hat{O}_{\text{lin}}$ and $\hat{O}_{\text{nonlin}}$ into one MPO, as the nonlinear term has a direct dependence on $u^j$, which would require to compute unitaries for each iteration step. To preserve the advantages of the combined circuit for nonlinear problems, we introduce an extended circuit shown in Fig. 2. A second weighting ancilla qubit allows the simultaneous application of the linear and nonlinear terms. For the nonlinear term, we cast $\hat{O}_{\text{nonlin}}$ in the unitary $\hat{U}_{\text{nonlin}}$, while the nonlinearity itself is computed with CNOT gates as introduced by Lubasch *et al.* in Ref. [33]. The angle $\varphi_u$ in the rotation gate $\hat{R}_Y(\varphi_u)$ on the weighting ancilla qubit is responsible for the correct weighting of the linear and the nonlinear term. For explicit Euler time stepping, where the leading linear term is the identity matrix,

$$\varphi_{u,j} = 2\arctan\left(\frac{\theta_j^0|c_{\text{MPO,nonlin}}|}{|c_{\text{MPO,lin}}|}\right), \tag{5}$$

where $c_{\text{MPO,lin}}$ and $c_{\text{MPO,nonlin}}$ are the multiplicative factors from the unitary creation of $\hat{U}_{\text{lin}}$ and $\hat{U}_{\text{nonlin}}$ respectively. To compute the norm of the next time step, we compute $f_{\hat{O},j}$ as before. Then, the effect of the total operator $\hat{O}_{\text{full}}$ can be computed as $\hat{O}_{\text{full}}\theta_j^0|\psi_j\rangle|\mathbf{0}\rangle_{\text{aux}} = \sqrt{2}\,c_{\text{MPO,lin}}f_{\hat{O},j}r_uP_{|0\rangle_{\text{aux}}\langle 0|_{\text{aux}}}\hat{U}_{\text{full}}\theta_j^0|\psi_j\rangle|\mathbf{0}\rangle_{\text{aux}}$, where $\hat{U}_{\text{full}}$ summarizes the linear and nonlinear operator actions and $r_u = k_u/k_u^{QC}$ stems from the difference between the real ratio $k_u = \theta_j^0|c_{\text{MPO,nonlin}}|/|c_{\text{MPO,lin}}|$ and its actual application $k_u^{QC} = \sin(1/2\varphi_{u,j})$ on the quantum computer.

## III. APPLICATION TO DIFFERENTIAL EQUATIONS

In the following section, we show the successful application of this approach to two use cases. We will first discuss

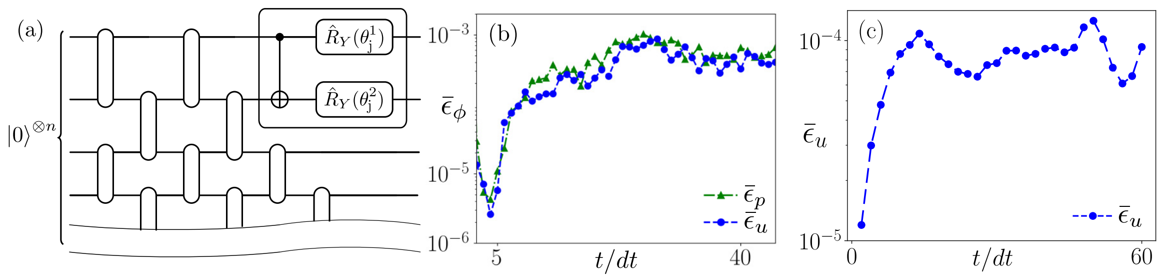

FIG. 3. (a) Brickwall ansatz with variational parameters. Each 2-qubit block is composed of two $\hat{R}_Y(\theta_i)$ gates and one CNOT gate. The ansatz can be used for a variable number of layers $L$, with each layer consisting of one column of 2-qubit blocks. (b) Evolution of the relative error $\bar{\epsilon}_\phi(t)$ [cf. Eq. (7)] for the Euler's equation. Here, $\phi(x,t)$ corresponds to the discretized solutions $u(x,t)$ and $p(x,t)$ of the Euler equation. To represent the solutions, a brickwall ansatz with 6 qubits and 14 layers is used. (c) Evolution of the relative error $\bar{\epsilon}_u(t)$ for the Burgers' equation. To represent the solutions, a brickwall ansatz with 6 qubits and 10 layers is used.

the linearized Euler equation to demonstrate the efficient incorporation of complex operators and time-stepping schemes. Next, we consider the nonlinear Burgers' equation to demonstrate the quantum circuit for nonlinear problems and the working principle of the convergence estimation from the expectation value. A third use case of the advection-diffusion equation demonstrating the usability with larger qubit numbers is presented in Appendix D. For both use cases, the ansatz functions are encoded with a brickwall ansatz for the circuit $\hat{U}(\boldsymbol{\theta}_j)$ as depicted in Fig. 3(a) and all operators are casted into highly accurate unitary approximations with bond dimension $Z = 16$ and relative approximation error below $E = 5 \times 10^{-10}$. For all use cases, the reinitialization of the training weights from the previous iteration step leads to a well pronounced training landscape, that is close to optimal for each training parameter even in the presence of shot noise and hence facilitates the optimization of the generally nonconvex cost function. The gradients around this initialization point remain pronounced for increasing system size, ensuring the trainability even for large qubit number. Details are given in Appendix F.

### A. Linear Euler equations

The 1D linear Euler equations describe the evolution of velocity $u(x,t)$ and pressure $p(x,t)$ of an inviscid flow [74] over time $t$. To this aim, we employ a noise-free quantum computing simulator and the explicit fourth-order Runge-Kutta (RK4) method to compute the solution of the next time step $dt$. We consider the particular situation of a periodic pressure point source with constant amplitude $A_0$ and angular frequency $\omega$ in the center of the domain $f(x,t) = A_0\delta(x)\sin(\omega t)$. Furthermore, nonreflective boundary conditions are applied using the sponge layer method [75], where a damping zone near the boundary is introduced via the sponge function $\gamma(x)$ (cf. Appendix B).

The coupled system of equations reads

$$\begin{aligned}\frac{\partial p}{\partial t} &= -\bar{\rho}c^2\left(\frac{\partial u}{\partial x}\right) + f(x,t) - \gamma(x)p,\\ \frac{\partial u}{\partial t} &= -\frac{1}{\bar{\rho}}\frac{\partial p}{\partial x} - \gamma(x)u,\end{aligned} \tag{6}$$

with the constants density $\bar{\rho}$ and the speed of sound $c$. We implement eighth-order accurate finite differences with Dirichlet boundary conditions and a staggered grid to avoid checkerboard oscillations [76]. For computational simplicity, we separate the fields into two ansatz circuits. Our scheme still significantly reduces qubit and circuit count compared to previously introduced methods [33,63,64], where additional circuits for each order of accuracy and the boundary implementations are needed. Instead, we can represent all operators acting on one field with one circuit of bounded depth. Furthermore, the sponge operator does not require an additional qubit register. The resulting cost functions, using RK4, are given in Appendix H II. Initially, the velocity and the pressure field are zero in the whole domain. Due to the pressure point source, an increasing pressure peak forms during the first time steps, leading to a nonzero contribution in the velocity field. This peak propagates towards the boundaries while additional peaks are formed by the source as shown in Fig. 1(b). To assess the quality of the solution, we use the normalized fidelity $\mathcal{F}^n = |(\phi(x,t)^{QDS}, \phi(x,t)^{cl})|^2/\|\phi(x,t)^{QDS}\|^2\|\phi(x,t)^{cl}\|^2$ [77] as a measure of closeness between the solution computed with our QDS ($\phi(x,t)^{QDS}$) and a classical solver ($\phi(x,t)^{cl}$), with $(\cdot,\cdot)$ being the inner product. The relative error is defined as

$$\bar{\epsilon}_\phi(t) = 1 - \mathcal{F}^n \tag{7}$$

and depicted in Fig. 3(b). During the time evolution, the maximal relative error is 0.1%, which is better than the error that would be introduced by current quantum hardware.

### B. Nonlinear Burgers' equation

The nonlinear weakly viscous Burgers' equation is defined as

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2}, \tag{8}$$

where $u$ is the velocity and $\nu$ the viscosity of the fluid. We implement a first-order accurate backward derivative $\frac{\partial}{\partial x}$ and a second-order accurate central derivative $\frac{\partial^2}{\partial x^2}$ with periodic boundary conditions with $x \in [0, 2\pi)$.

We consider the particular situation of a Gauss peak as initial conditions $u(x,t=0) = \exp(\frac{-(x-\pi)^2}{\sigma})$ with $\sigma = 0.5$ and choose $\nu$ and $dt$, such that a shock evolves within a reasonable simulation time. The detailed simulation parameters and the used cost function are listed in Appendixes A and H I. The velocity evolution is shown in Fig. 1(c), where the initial Gauss peak evolves into a shock wave. When compared to the classical solution, the relative error $\bar{\epsilon}_u$ reaches values around $10^{-4}$, showing a good agreement over the whole time evolution. We observe that the scaling of $\alpha_{\text{succ}}$ of the nonlinear multiplication over system size behaves similarly as reported by Lubasch *et al.* [33] for the expectation value in their nonlinear use case. More details are given in Appendix E.

## IV. SCALING ANALYSIS

In addition to the simulation itself, we are interested in the scaling behavior of the quantum ansatz. To this aim we first focus on a comparison with MPS-based algorithms, which are the classical counterpart of the tensor-programmable quantum scheme [59,78]. Second, we consider the cost arising from the operator application.

### A. Comparison with MPS methods

For the scaling comparison, we focus on the cost caused by the field encoding, which is normally dominant, as the operators feature a low-rank MPO representation (cf. Appendix B). We are mainly interested in two performance indicators, namely the memory consumption and the computational complexity. The memory consumption is dominated by number of parameters necessary to represent the field per iteration step, which scales as $O(n\chi^2)$ for the classical MPS and as $O(\eta_{\text{params}})$ for the quantum ansatz, where $\eta_{\text{params}}$ is the number of parameters $\boldsymbol{\theta}_j$ in the brickwall ansatz.

Regarding the computational complexity, it is well known that MPS algorithms scale at least as $n\chi^3$ [15] for linear operations. For the nonlinear Hadamard product, stable simulations with a scaling of $n\chi^4$ were reported [15]. The quantum algorithm scales mainly with the number of parameters necessary to represent the solution [78] as $O(\eta_{\text{params}})$. Additionally, sampling the expectation value $\langle\sigma_z\rangle_{\text{anc}}$ adds an additional cost $O(1/\epsilon^2)$ depending on the allowed sampling error $\epsilon$. There are methods that can reduce this cost to values close to $O(1/\epsilon)$ [79,80]. When all operators are combined in one quantum circuit and the expectation value approaches $\langle\sigma\rangle_{\text{anc}} = 1$, the sampling cost approaches $1/\epsilon$ even without additional modifications. Given the large shot-overhead caused by the scaling of the sampling error, the quantum-tensor-network method can clearly not outperform the classical algorithms for small $\chi$. However, for large-scale simulation where larger $\chi$ are required, this constant shot-overhead decreases in importance, and the first factor, the number of required parameters becomes decisive.

We consider the highly relevant use case of a three-dimensional turbulent flow field, namely a velocity field of an isotropic turbulent flow on $N_{\text{tot}} = 1024^3$ grid points provided by the Johns Hopkins turbulence database at a Taylor-scale Reynolds number of $Re_\tau \approx 433$ [81–83]. To reduce the problem into numerically feasible sizes, we consider a three-dimensional box of size $N = 2^{3n_{\text{d}}}$ at the center of the domain, where each spatial direction is resolved using $n_{\text{d}}$ qubits. Increasing $n_{\text{d}}$ allows to study the scaling behavior with $N$ and to provide an estimate for the full flow field.

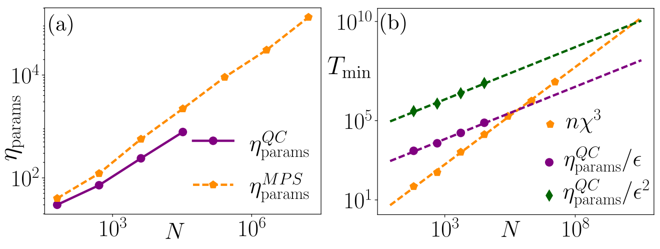

FIG. 4. Representation capabilities of the brickwall and the MPS ansatz for a three-dimensional turbulent flow defined on $N$ grid points. (a) Number of parameters to represent an increasing section of an isotropic flow field of total size $N_{\text{tot}} = 1024^3$ approximated up to an accuracy of $\bar{\epsilon}_v = 0.01$. (b) Estimate of the minimal required cost $T_{\min}$ computed for the given number of parameters from panel (a) and for an allowed maximal, the sampling error $\epsilon = 0.01$. The two minimal costs for the quantum scheme give the best (purple, circle) and the worst (green, diamond) case scenario for the cost contribution of the sampling error. The dashed lines correspond to a linear interpolation in the log-log scale.

Figure 4(a) depicts the number of parameters necessary for the MPS and the quantum circuit to represent the flow field of different box sizes, as well as their ratio. For the MPS, we consider the maximal bond dimension $\chi$ required to represent the $y$ component of the velocity field with a fidelity $\bar{\epsilon}_v < 0.01$. For the quantum circuit, we train a brickwall ansatz to represent the field with increasing layer number until $\bar{\epsilon}_v < 0.01$. For this, we have replaced the CNOT gates in the brickwall ansatz with CZ gates, as this allows to initialize additional low-depth layers as an identity. The reduction in parameters when comparing the quantum circuit and the MPS representation increases with the system size and the bond dimension. We expect this advantage to stem from the better entangling properties of the quantum circuit. Importantly, while the shown MPS representation was created using successive singular value decompositions [84] and is hence already optimal with respect to the $l_2$ norm, the quantum circuit representation does not possess this optimality. Different circuit architectures [78] and techniques as incremental structural learning [40,41] can potentially be exploited to reduce the number of parameters further, which would also improve the scaling properties. Additionally, Ref. [85] has provided a theoretical upper bound when representing each unitary as a $\kappa$ design, where the circuit depth scales only polynomially in the number of qubits and the chosen $\kappa$ adds a (potentially large) constant factor.

In Fig. 4(b), we look at the minimal cost $T_{\min}$ of the MPS ($T^{\text{MPS}}_{\min} = n\chi^3$) and the quantum approach ($T^{\text{QC}}_{\min} = \eta^{\text{QC}}_{\text{params}}/\epsilon^{(2)}$), given the computed number of parameters and assuming a maximal sampling error of the size of the target fidelity $\epsilon = 0.01$. While $T_{\min}$ is lower for the MPS approach for small qubit numbers, the quantum scheme shows its potential for larger scale simulations.

Choosing a suitable training strategy for the quantum use case is mandatory to preserve the potential advantage. Gradient-based methods, which rely on the exact computation of each gradient, add an additional scaling factor $O(\eta_{\text{params}})$, which would annihilate any advantage. However, strategies to estimate the gradient coherently, as the simultaneous perturbation stochastic approximation (SPSA) can fix the number of circuit evaluations to a number independent of the number of trainable parameters [26,73] and is still capable to train the solution to optimal parameters (cf. Appendix F).

Clearly, this is a simplified scaling analysis. It neglects the potential additional cost of $\chi^4$ of the MPS scheme in the presence of nonlinear operations and the impact of the success probability of the quantum circuit. Both depend on the chosen differential equation and its implementation details. For small success probabilities, the discussed expected theoretical advantage might be shifted to larger system sizes or might require additional strategies as, e.g., amplitude amplification, phase estimation or their combination as utilized by Goswami *et al.* [86] to be maintained. Standard amplitude amplification techniques would require repeated state encoding leading to an increased circuit depth with an additional factor of $O(1/\sqrt{\alpha_{\text{succ}}})$ contributing to the cost given above. Furthermore, we have assumed that both the classical and the quantum-tensor network algorithm involve a comparable scaling of the number of training steps with system size. Importantly, both approaches provide the fidelity $\mathcal{F}$ as convergence measure, which allows to adapt the training hyperparameters during the training procedure and improving the convergence. A significant part of the cost in the quantum algorithm is caused by the number of required shots. As this process is inherently parallelizable, practical benefits might occur even before theoretical advantages are reached.

### B. Scaling of the operator

In previous schemes [33,64], the required number of auxillary qubits used to encode potentials and finite difference derivatives scales linear with the number of ansatz qubits $n$. In comparison, in our approach the number of auxillary qubits depends solely on $Z$, which is expected to be independent of $n$. Additionally, the number of quantum circuits $M$ is considerably reduced. This beneficial scaling in qubit and circuit number is achieved with the same scaling of the circuit depth. The scaling of the number of 2-qubit gates for the operator application, $N_{2\text{q,op}}$, is upper bounded [87] by $N_{2\text{q,op}} = knZ^2$, with a universal proportionality constant $k$.

## V. CONCLUSION

This paper demonstrates how generic VQA can be programed using MPOs, allowing for the seamless integration of nonunitary operators. Importantly, higher order differential operators and various boundary conditions can be incorporated with little or no additional cost. Tensor network algorithms were effectively implemented for many large-scale simulations but can face challenges in the presence of complex data. Using the example of a turbulent flow field, we can show the additional compression capabilities of the quantum circuit. This leads us to expect successful scaling of the tensor-programmable quantum circuits, particularly when exploring additional parameter reduction techniques [40,41]. This potential is further enhanced by employing advanced optimization strategies, such as multigrid and local optimization, which have proven extremely successful in classical-tensor network schemes [15,88] and are also accessible for VQAs [37,89].

Being applicable to linear and nonlinear PDEs, the presented scheme contains all building blocks for solving PDEs critical for science and industry, e.g., the Navier-Stokes equations. This will become especially valuable once quantum hardware reaches the required capacity for industry relevant use cases, a milestone that, according to projections from companies like IBM and QuEra, could be achieved by 2029 [31,32]. Furthermore, the flexible operator representation of this scheme would enable interfacing between quantum algorithms and existing classical software packages.

The tensor-programmable quantum algorithms allow to encode all operators of a PDE into a single quantum circuit. This bears two main advantages. First, it increases the expectation value of the global ancilla qubit. Second, this expectation value is directly connected to the fidelity between the trained and the correct solution and can hence act as a global measure of convergence. This allows to estimate the quality of the solution without expensive comparisons to classical solution and yields the potential for adaptive optimization strategies, where hyperparameters and circuit depths are improved within the training loop. Finally, these improvements can potentially be augmented by utilizing the potential of phase estimation techniques [90,91] to improve the precision of the cost function measurements. In the presence of low success probabilities, additional techniques as amplitude amplification [86] should be considered to decrease the number of required shots. These questions will be investigated in future studies.

## ACKNOWLEDGMENTS

We thank Theofanis Panagos for helpful discussions on the use case and M. Lautaro Hickmann for helpful discussions on circuit optimization routines. P.S. acknowledges financial support by the DLR-Quantum-Fellowship Program. G.S.R., N.-L.v.H., and D.J. acknowledge funding from the European Union's Horizon Europe Research and Innovation Program (HORIZON-CL4-2021-DIGITAL-EMERGING-02-10) under Grant Agreement No. 101080085 QCFD. G.S.R. was supported by the Cluster of Excellence "CUI: Advanced Imaging of Matter" of the Deutsche Forschungsgemeinschaft (DFG)-EXC 2056—Project ID 390715994. T.H. and D.J. acknowledge funding by the Cluster of Excellence "Advanced Imaging of Matter" of the Deutsche Forschungsgemeinschaft (DFG)–EXC 2056 - Project ID 390715994. D.J. acknowledges support from the Hamburg Quantum Computing Initiative (HQIC) project EFRE. The project is co-financed by ERDF of the European Union and by "Fonds of the Hamburg Ministry of Science, Research, Equalities and Districts (BWFGB)." Further, the authors would like to thank Airbus and the BMW Group for providing the use case in the context of the Airbus-BMW Group Quantum Computing Challenge 2024.

## DATA AVAILABILITY

The data that support the findings of this article are openly available [92].

## APPENDIXES

First, we present the parameters employed for our two use cases, namely the linearized Euler equations and the Burgers' equation in Appendix A. Next, we explain the construction of the MPOs from the differential operators and providing especially details on the differential operators and the sponge MPO, which facilitates nonreflective boundary conditions in Appendix B. In Appendix C, we explain how we compile the necessary quantum operations from the MPOs. Next, we show the performance of the algorithm for large qubit numbers for the advection-diffusion equation in Appendix D. In Appendix E, we show the scaling of the success probability for different qubit numbers. Next, in Appendix F we analyze the trainability and the sensitivity of the loss landscape to shot noise. Next, the computation of the normalization constant $f_{\hat{O},j}$, the derivation of the angle $\varphi$ used in the adapted Hadamard test, and the derivation of the convergence measure, i.e., the fidelity, are given in Appendix G. Next, we present in Appendix H I the cost functions used for Euler time stepping of the Burgers' and the advection-diffusion equation. Finally, we present the cost functions and quantum circuits obtained using the fourth- order Runge-Kutta time stepping scheme in Appendix H II. We use the same notation and definitions as in the main text.

## APPENDIX A: SYSTEM AND TRAINING DETAILS

Here, we shortly outline the specific parameters that describe the system as well as the choose circuit sizes and training details.

*Euler equation.* The specific parameters used in our example of the linear Euler equation with a periodic point source are density $\bar{\rho} = 1.225$ kg/m$^3$, the frequency and amplitude of the point source $\omega = 100$ Hz and $A_0 = 0.4c$, and the sound of speed $c = 340.2$ m/s. We study a spatial domain of size $x \in [-4, 4]$. The ansatz is encoded into six qubits, corresponding to a discretization of the domain into $N_x = 2^6 = 64$ data points. This domain includes the inner zone $x_{\text{inner}} \in [-2, 2]$ of unperturbed spatial evolution as well as the outer zones $x_{\text{outer}} \in [-4, -2]$ and $\in [2, 4]$, where the sponge damps the signal to implement nonreflective boundary conditions. We discretize the space with first-order finite differences and use a fourth-order Runga-Kutta time stepping scheme with a stepsize of $ydt = 2.5 \times 10^{-4}$ s. The corresponding cost function are detailed in Appendix H II. We use the expression of the bounded sponge operator explained in Appendix B, with $\kappa = 0.13$, $\tilde{n} = 4$, and $\gamma_{\max} = 1500$.

We perform the optimization using an Adam Optimizer, followed by additional training epochs with a limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (LBFGS). For the simulation of the results shown in Figs. 1(b) and 1(c), we used a brickwall ansatz with 14 layers. We trained it using a learning rate of $\text{lr}_{\text{Adam}} = 0.05$ and $\text{lr}_{\text{LBFGS}} = 0.5$ and a number of epochs of $n_{\text{epochs,Adam}} = 751$ and $n_{\text{epochs,LBFGS}} = 75$. All runs are performed using the quantum computing software framework PennyLane [93] together with pyTorch for the parameter optimization [94].

*Burgers' equation.* For the Burgers' equation we consider a spatial domain of size $x \in [0, 2\pi]$ and discretize it on a uniform grid with $N_x = 2^6 = 64$ lattice points with lattice spacing $dx = 2\pi/N_x$. We choose $\nu = 0.001$ and $dt = 0.5dx$, such that a shock evolves within the first 60 Euler time steps. The choice of such a large $dt$ leads to a small time evolution error, but which is smaller than the introduced grid error. For the simulation of the results, we use a circuit with ten layers, a learning rate of $\text{lr} = 0.005$ and 1000 training epochs using the Adam optimizer. All runs are performed using the quantum computing software framework PennyLane [93] together with Jax [95] for the parameter optimization. The weights that encode the initial conditions are trained prior to the the application of the tensor-programmable quantum scheme by minimizing the relative error $\epsilon_u(t = 0) \approx 1.5 \times 10^{-6}$. In principle, it is also possible to start from a delta peak, and evolve this with the diffusion equation until the correct Gauss width is reached.

## APPENDIX B: MATRIX PRODUCT OPERATOR REPRESENTATION OF DIFFERENTIAL OPERATORS

We introduce the MPO of matrix $O$ in its generic form [84]:

$$\mathcal{O} = \sum_{\boldsymbol{\zeta},\boldsymbol{\sigma},\boldsymbol{\sigma}'} O[1]^{\sigma_1,\sigma_1'}_{\zeta_0,\zeta_1} \cdots O[n]^{\sigma_n,\sigma_n'}_{\zeta_{n-1},\zeta_n} \left|\boldsymbol{\sigma}\right\rangle \left\langle\boldsymbol{\sigma}'\right|, \tag{B1}$$

where $\boldsymbol{\zeta}$ ($\boldsymbol{\sigma}$) denotes the virtual (physical) indices, respectively, and $O[\cdot]$ the MPO cores, which are four-dimensional tensors. We assume $\zeta_0 = \zeta_n = 1$ and define the maximal bond dimension $\zeta_{\mathcal{O}} = \max(\dim(\zeta_j))$, with $j = 0, \ldots, n$. Whenever $O$ admits a sum of simple tensor products, it can be written in a factorized "matrix-of-operators" form $O = A[1] \bowtie \cdots \bowtie A[n]$, where each block entry of $A[j]$ is a local $2 \times 2$ operator $B_{\sigma_j,\sigma_j'}$. The symbol $\bowtie$ indicates that block multiplication is a tensor product on the physical legs and an ordinary matrix product on the virtual legs. We use the elementary $2 \times 2$ projectors $I_1, I_2$, and the shift $J$ and $J^T$ as a convenient local operator basis:

$$I_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad I_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},$$
$$J = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad J^T = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}. \tag{B2}$$

These generators yield compact MPOs for common gates and for banded matrices while keeping the MPO bond dimension small. For two qubits, CNOT decomposes into a sum of two product terms and thus admits an MPO with bond $\zeta_{\mathcal{O}} = 2$. The first core selects the control state via $(I_1, I_2)$, the second core applies either $I$ or $\sigma_x$ on the target conditioned on the virtual index, now written as

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

$$= \underset{\zeta_0}{\{}\,[t]\overbrace{(I_1,\ I_2)}^{\zeta_1} \bowtie \underset{\zeta_1}{\left\{\right.}[t]\overbrace{\begin{pmatrix} I \\ \sigma_x \end{pmatrix}}^{\zeta_2} = I_1 \otimes I + I_2 \otimes \sigma_x, \tag{B3}$$

with Pauli matrix $\sigma_x$. Similarly, tridiagonal Toeplitz matrices $\text{TriDiag}(\alpha, \beta, \gamma)$ have a standard three-block MPO with constant bond $\zeta_{\mathcal{O}} = 3$ [67]

$$\begin{aligned}\mathrm{TriDiag}(\alpha,\beta,\gamma) &= \begin{pmatrix}\alpha & \beta & & & \\ \gamma & \alpha & \beta & & \\ & \gamma & \alpha & \beta & \\ & & \ddots & \ddots & \ddots \\ & & & \gamma & \alpha\end{pmatrix}\\ &= (\alpha I + \beta J + \gamma J^T, \gamma J, \beta J^T)\\ &\quad \bowtie \begin{pmatrix} I & 0 & 0\\ J^T & J & 0\\ J & 0 & J^T\end{pmatrix}^{n-2} \bowtie \begin{pmatrix} I\\ J^T\\ J\end{pmatrix}.\end{aligned} \tag{B4}$$

In this MPO form, standard finite-difference operators with at most nearest-neighbor coupling admit compact representations. For instance, the central, second-order accurate first-derivative stencil on a uniform grid with spacing $\Delta x$ corresponds to the tridiagonal Toeplitz choice,

$$\alpha = 0, \quad \beta = \frac{1}{2\Delta x}, \quad \gamma = -\frac{1}{2\Delta x},$$

i.e., $\mathrm{TriDiag}(\alpha = 0, \beta = \frac{1}{2\Delta x}, \gamma = -\frac{1}{2\Delta x})$. The block structure above directly encodes operators under Dirichlet boundary conditions. If different boundary conditions are required (e.g., periodic or Neumann), the needed corrections can be added as low-rank modifications to the base $\mathrm{TriDiag}(\alpha,\beta,\gamma)$. Each added nonzero matrix element (such as a periodic link between the first and last grid point) increases the MPO bond dimension by at most one, so boundary corrections preserve a small bond dimension [47].

The respective decomposition for a bounded 1D sponge operator, with maximal factor of 1, are given by

$$A[j] = \frac{1}{e^{(2^{\tilde{n}}-1)\kappa} - 1}(I_2,\ I_1\, I_2,\ I_1), \quad \text{for } j = 1, \tag{B5}$$

$$A[j] = \begin{pmatrix} I_2 & & & \\ & I_1 & & \\ & & I_2 & \\ & & & I_1\end{pmatrix} 2 \leqslant j \leqslant n - \tilde{n}, \tag{B6}$$

$$A[j] = \begin{pmatrix} J_1(j) & & & \\ & J_1^{\mathrm{t}}(j) & & \\ & & I & \\ & & & I\end{pmatrix}, \quad \text{for } n - \tilde{n} < j \leqslant n-1, \tag{B7}$$

$$A[j] = \begin{pmatrix} J_1(j)\\ J_1^{\mathrm{t}}(j)\\ I\\ I\end{pmatrix}, \quad \text{for } j = n, \tag{B8}$$

where t refers to a mirroring with respect to the antidiagonal.

## APPENDIX C: MATRIX PRODUCT OPERATORS TO QUANTUM GATES

The algorithm for determining quantum gates that prepare an arbitrary MPS is well known [55–57]. This approach yields an exact encoding and provides an upper bound on the circuit depth for generating a certain amount of entanglement [33]. Recently, also the translation of MPOs into quantum gates has been reported [58,59]. The latter work by Termanova *et al.* will be outlined in the following. Its reduced requirements in qubit numbers made it a promising candidate for integration into the VQA framework. Let us start by introducing the MPO as [84]

$$\mathcal{O} = \sum_{\boldsymbol{\zeta},\boldsymbol{\sigma},\boldsymbol{\sigma}'} O[1]^{\sigma_1,\sigma_1'}_{\zeta_0,\zeta_1} \cdots O[n]^{\sigma_n,\sigma_n'}_{\zeta_{n-1},\zeta_n} |\boldsymbol{\sigma}\rangle\langle\boldsymbol{\sigma}'|, \tag{C1}$$

where $\boldsymbol{\zeta}$ ($\boldsymbol{\sigma}$) denotes the virtual (physical) indices, respectively, and $O[\cdot]$ the MPO cores, which are four-dimensional tensors. We assume $\zeta_0 = \zeta_n = 1$ and define the maximal bond dimension $\zeta_{\mathcal{O}} = \max(\dim(\zeta_j))$, with $j = 0, \ldots, n$.

We introduce the MPO $\mathcal{Q}$, which shall approximate target MPO $\mathcal{M}$ while satisfying isometric constraints. To account for the limitations on dimensionality and degrees of freedom imposed by these constraints, we expand the search for $\mathcal{Q}$ to encompass a larger Hilbert space. This is done implicitly by setting its bond dimension $Z_{\mathcal{Q}} = 2^{\ell}$, where $\ell$ is a positive integer and $Z_{\mathcal{Q}} > \zeta_{\mathcal{M}}$.

Following this, the search procedure is then formulated as a constrained optimization problem, which reads as [59]

$$\begin{aligned} C = &\min_{c_{\mathrm{MPO}},\hat{\mathcal{Q}}} \|c_{\mathrm{MPO}}\mathcal{Q} - \mathcal{M}\|^2\\ &\text{subject to } Q[1]^{\dagger} \in \mathrm{St}(r,s)\\ &\qquad\qquad Q[j] \ \in \mathrm{St}(r,s) \quad \forall j = 2,\ldots,n,\end{aligned} \tag{C2}$$

where $c_{\mathrm{MPO}}$ is a normalization constant, $\|\cdot\|$ the Frobenius norm, and $\mathrm{St}(r,s)$ the Stiefel manifold, which is the set of all $r \times s$ matrices with orthonormal columns, where $r \geqslant s$ [97]. Here, we introduced $Q[j] := Q[j]_{(Z_{\mathcal{Q}},\sigma_j),(\sigma_j',Z_{\mathcal{Q}})}$ as the reshaped, isometric cores for $1 < j < n$, and $Q[1] := Q[1]_{\sigma_j,(\sigma_j',Z_{\mathcal{Q}})}$ and $Q[n] := Q[n]_{(Z_{\mathcal{Q}},\sigma_j),\sigma_j'}$, respectively. The normalization constant $c_{\mathrm{MPO}}$ can be determined via [59]

$$c_{\mathrm{MPO}} = \mathrm{Re}\frac{\mathrm{tr}[\mathcal{Q}^{\dagger}\mathcal{M}]}{\|\mathcal{M}\|^2}. \tag{C3}$$

With all of this in place, we briefly outline the constraint minimization of Eq. (C2), graphically depicted in Fig. 5: (1) initialize the isometric cores of $\mathcal{Q}$, (2) compute normalization $c_{\mathrm{MPO}}$, (3) perform a single Riemannian gradient step on all tensors $Q_j$ [97], and (4) repeat steps (2) and (3) in an alternating manner until the error measure $\epsilon = \frac{\|c_{\mathrm{MPO}}\mathcal{Q} - \mathcal{M}\|^2}{\|\mathcal{M}\|^2}$ reaches the set tolerance. In step (3), the gradient $g = \frac{\partial C}{\partial Q[j]^*}$ is projected onto the tangent space of the core, resulting in $G$. A retraction is then performed in this direction, scaled by the learning rate, i.e., $-\mu G$. This retraction can be performed, for example, by a QR decomposition or a Cayley transformation [96,97]. The QR decomposition allows to rewrite a matrix $A$ as $A = QR$, where $Q$ is an orthogonal and $R$ is an upper triagonal matrix.

As soon as the algorithm has reached the set convergence criterion, the boundary isometric cores $Q[1]$ and $Q[n]$ with the shapes $2 \times (2Z_{\mathcal{Q}})$ and $(2Z_{\mathcal{Q}}) \times 2$, respectively, need to be raised to unitaries. To do so, the remaining columns (rows) are filled using Gram-Schmidt orthonormalization procedure [98], respectively. This results in the target matrix only being applied probabilistically, as the padding also enables a trajectory within the nullspace.

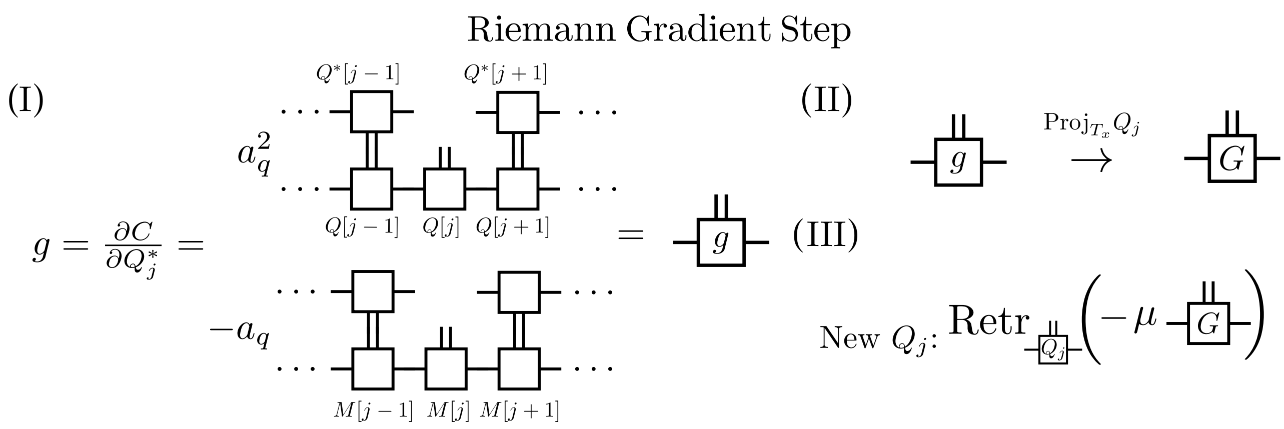

FIG. 5. Sketch of the execution of a single Riemannian gradient step on the tensors of a unitary MPO $\mathcal{Q}$ that approximates a target MPO $\mathcal{M}$. It can be divided into three sub-steps: (I) for each core $Q_j$ the gradient is computed by deriving the cost function $C$ with respect to its complex conjugate $Q_j^*$, we denote the result by $g$, (II) the gradient $g$ is projected onto the tangent space of $Q_j$ via $g - \frac{1}{2}Q_j(Q_j^T g + g^T Q_j) := G$ [96], where we have defined the Riemannian gradient $G$, (III) the $Q_j$ is found by a retraction antiparallel to the Riemannian gradient, where the magnitude of the update step is controlled by the learning rate $\mu$. Using the QR decomposition as a retraction map, this last step has the form $\mathrm{Retr}^{QR}_{Q_j}(-\mu G) = QR(Q_j - \mu G)$. We note that for the core $Q_1$, which is not in the Stiefel manifold, special measures must be taken. After step (I), $Q_j$ and the gradient $g$ must be transposed, then steps (II) and (III) are carried out, and the transposed result then gives the different $Q_j$.

## APPENDIX D: ADVECTION-DIFFUSION EQUATION

Next to the examples given in the main text, we implemented the advection-diffusion equation

$$\frac{\partial \phi}{\partial t} = \nu \Delta \phi - c \nabla \phi, \tag{D1}$$

with $\nu = 0.1$, $c = 20$, and periodic boundary conditions using the proposed quantum-tensor scheme. Its simple structure and reduced qubit requirements compared to the nonlinear use case makes it a suitable candidate for larger-scale simulations and the analysis of the cost function given in Appendix F. We consider the same initial condition as for the Burgers' equation, a Gauss peak with $\sigma = 0.5$ resolved with $N_x = 2^{10} = 1024$ lattice points, a lattice spacing of $dx = 2\pi/N_x$ and using a circuit with 15 layers. Figure 6 depicts the field evolution and the relative error over time, showing a good agreement with the classical solution.

## APPENDIX E: SCALING OF THE SUCCESS PROBABILITY WITH SYSTEM SIZE

The value of the success probability $\alpha_{\mathrm{succ}}$ plays a crucial role in the determination of the norm correction $f_{\hat{O},j}$. As it depends on both the operator and the field it acts on, it is a use-case-dependent quantity. Here, we look at the average success probability of all differential operators implemented for this work, following the procedure introduced by Termanova *et al.* [59]. They recognize that the average success probability $\bar{\alpha}_{\mathrm{succ}}$ of a matrix $M$ is directly connected to its Frobenius norm as

$$\bar{\alpha}_{\mathrm{succ}} = \frac{\|A\|}{2^n}, \tag{E1}$$

where $A = \frac{1}{c_{\mathrm{MPO}}} M$, and $c_{\mathrm{MPO}}$ is the multiplicative factor from the mapping of the MPO to a unitary. To study the scaling of the success probability over the system size, we consider the optimal $c$, being the leading singular value of $M$. Then, we find constant or converging average success probabilities for all differential operators used within this work as shown in Fig. 7(a).

Next, we are interested in the scaling of the success probability of the nonlinear multiplication implemented by the CNOT gates. While for random vectors, the point-wise multiplication might result in an exponential decay of the success probability with system size, for the discussed use case of the Burgers' equation the situation is more subtle. Figure 7(b) shows $\alpha_{\mathrm{succ}}$ over the system size for different widths of the Gauss peak. We observe that its value remains constant until the peak is resolved, and only then decays with system size. This is a behavior also observed by Lubasch *et al.* [33] for their example of the nonlinear Schrödinger equation. The implications of that get clearer in Fig. 7(c), where, we consider the success probability over system size, considering different Gauss widths $\sigma$ and choosing the number of required qubits by restricting the grid error below a threshold: Considering a constant grid error, the quantum circuit is capable to resolve increasingly narrow peaks with an increasingly better resolution without gaining any loss in the success probability. Furthermore, we expect techniques as phase estimation and amplitude amplification to mitigate the challenges that arise

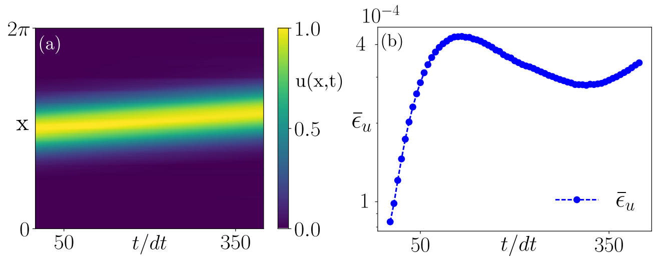

FIG. 6. Time evolution of the field according to the advection-diffusion equation. (a) Evolution of the initial Gauss peak $u(x, t = 0)$ over 400 evolution steps and (b) the relative error $\bar{\epsilon}_u$ over time.

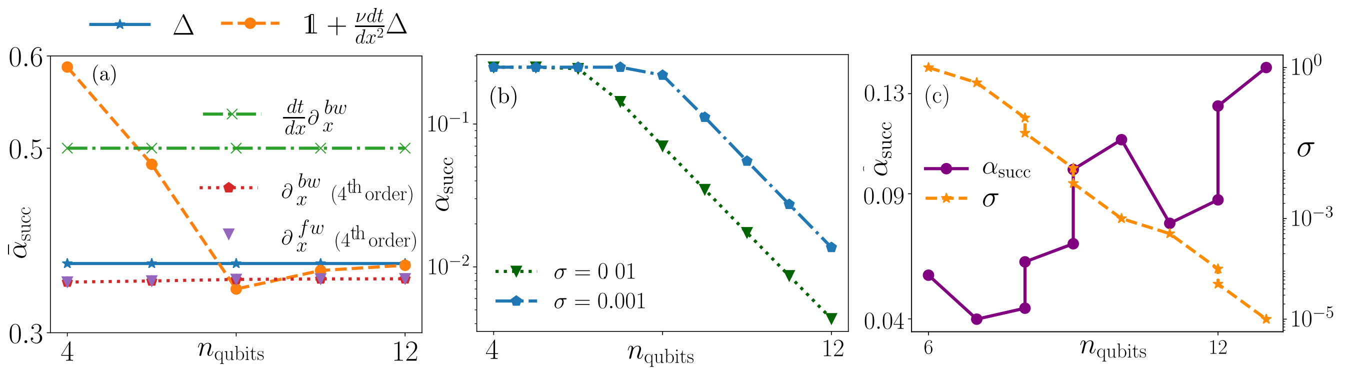

FIG. 7. (a) Scaling of the maximal average success probability $\bar{\alpha}_{\text{succ}}$ for all differential operators used throughout this work. The success probability of the sponge operator highly depends on its width compared to the system size. (b) Scaling of the success probability $\alpha_{\text{succ}}$, the point-wise multiplication $u\nabla u$, as present in the Burgers' equation for a Gauss peak of width $\sigma$. (c) Scaling of $\alpha_{\text{succ}}$ when considering different Gauss peak widths and choosing the number of qubits according to a maximal allowed root-mean-square error <0.001 between the solution and the next better resolution.

for use cases that result in low $\alpha_{\text{succ}}$ and will investigate their potential in future works.

## APPENDIX F: ANALYSIS OF THE TRAINABILITY

In this section, we study the shape of the cost landscape of the training parameters, its sensitivity to shot noise and the variance of gradients. We consider the first time step of the advection-diffusion equation, as it allows us to go to larger qubit number as the nonlinear use case with significantly reduced resource requirements. Please note that we observe qualitatively the same cost landscapes for the other use cases. We were mostly concerned with the following questions: First, considering initializations of the weights from the previous time step, how pronounced and optimal is the cost landscape for the single parameters. Second, how is this cost landscape affected by noise. Third, how does this cost landscape and its gradients change with system size. The results in Fig. 8(a.I) show clearly that weight reinitialization leads to a well pronounced loss landscape, especially when compared to a random initialization for each parameter. A comparison with the optimal cost landscape (all parameters well trained and one varied) shows that weight reinitialization leads to a close-to-optimal cost landscape, featuring a maximal difference of $10^{-4}$, further indicating close to optimal trainability. While the cost landscape roughens slightly in the presence of shots, the impact of shots on the loss landscape seems to remain constant over the system size as shown in Fig. 8(a.II).

Furthermore, we consider the variance of the gradients of the cost functions for increasing system size. When this variance decreases exponentially with system size, it is a strong indicator for barren plateaus [99,100], which makes training quickly unfeasible for larger systems. However, gradients also vanish close to the optimum, hence low variances at a certain parameter set, do not necessarily connect to a barren plateau. To allow for meaningful statistics, we therefore consider both the variances of the gradient [cf. Fig. 8(b.I)] and the mean of the corresponding cost [cf. Fig. 8(b.II)]. To allow for statistic in the case of the weight reinitialization (often referred to as warm-start [100]), we vary each reinitialized parameter

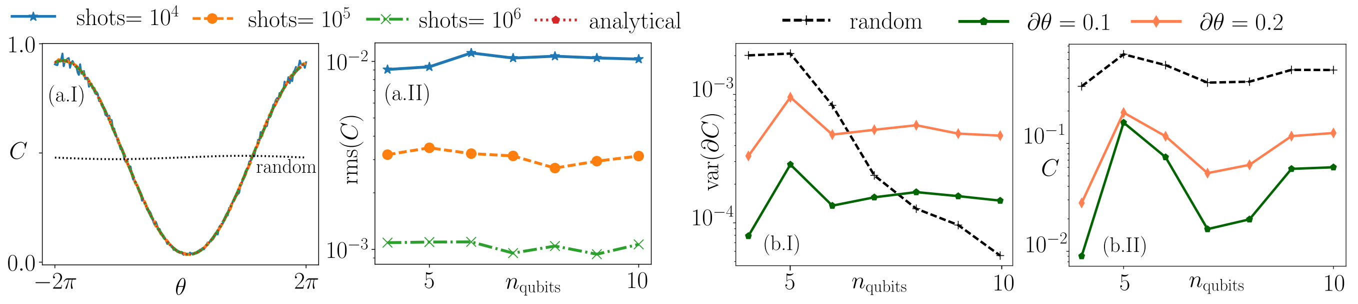

FIG. 8. Analysis of the cost landscape $C(\theta_{j+1}^i) = 1 - \langle\sigma_z\rangle_{\text{anc}}$ of one representative training parameter $i$ for different shot numbers and the shot-free analytical solution. (a.I) Cost landscape of a Gaussian peak with $\sigma = 0.5$ evolved according to the advection-diffusion equation and resolved with ten qubits when using weight initialization from the previous time step. We note that the biggest deviations in the cost landscape for small shot numbers appear far from the optimum. The black dotted line shows the cost landscape for random weight initialization. (a.II) Root-mean-square error between the shot based and the analytical cost landscape over the system size. (b.I) Variances of the gradients of the cost landscape over the system size. We consider a completely random initialization of weights (black dashed line, crosses) and a warm-start using the previous time step as the initial guess, but changing each parameter slightly by random numbers in the the range $[-\partial\theta, \partial\theta]$. (b.II) Mean of $C$ for the examples used in panel (b.I). To compute variance and mean, we considered 1000 random samples. All quantities were computed with $dx = 2\pi/N_x$ and a time step of $0.1\nu dx$, such that a stable simulation is possible. All cost landscapes are plotted using $\varphi = 0.6126$.

randomly within a certain threshold $\partial\theta$. The variance decreases roughly exponentially with system size for random weight initialization, while the cost remains large. We do not observe the same effect in the presence of warm starts. There, the variance seems to remain close to constant. Additionally, we initialize the parameters already close to the optimum, which is reflected by an increase in both the variance and the cost when increasing $\partial\theta$.

We tested the trainability of the next time step for the advection-diffusion equation with shots using the exact gradient computed with parameter-shift rule or a gradient approximation obtained from the SPSA algorithm [26,73] combined with an Adam optimization. With both strategies, we reach the best possible resolved loss for different shot numbers.

## APPENDIX G: DERIVING THE NORM CONSTANT $f_{\hat{O},j}$, THE OPTIMAL ANGLE $\varphi_{\text{opt}}$, AND THE CONVERGENCE MEASURE $\mathcal{F}$

In the following Appendix, we present the calculation of the norm constant $f_{\hat{O},j}$ for the standard and adapted Hadamard test. Furthermore, we determine the optimal rotation angle $\varphi$ and derive the convergence measure accessible via the adapted Hadamard test.

As explained in the main text, applying the operator $\hat{O}$ with the help of the unitaries $\hat{U}_{\text{MPO}}$ on the quantum computer implements the correct operation up to the factor $f_{\hat{O},j}c_{\text{MPO}}$:

$$\begin{aligned}\langle\hat{O}\rangle &= c_{\text{MPO}}\cdot\text{Re}\,\langle\Psi|\,P_{|0\rangle_{\text{aux}}\langle 0|_{\text{aux}}}\hat{U}_Q\,|\Psi\rangle\\ &= c_{\text{MPO}}\cdot f_{\hat{O},j}\cdot\langle\sigma_z\rangle_{\text{anc}}\,.\end{aligned} \tag{G1}$$

Here, the operator $\hat{U}_Q$ summarizes all controlled unitaries, $|0\rangle_{\text{anc}}|\Psi\rangle$ is the initial state. For linear differential equations, we have $\hat{U}_Q=\hat{U}(\boldsymbol{\theta}^{\dagger}_{j+1})\hat{U}_{\text{MPO}}\hat{U}(\boldsymbol{\theta}_j)$ and $|0\rangle_{\text{anc}}|\Psi\rangle = |0\rangle_{\text{anc}}|\mathbf{0}\rangle$. While $c_{\text{MPO}}$ is known from the algorithm that determines the unitaries from the initial MPO, $f_{\hat{O},j}$ needs to be determined from the success probability.

The quantum circuit of the adapted Hadamard test produces, prior to measurement, the state

$$\begin{aligned}&\hat{R}_Y(-\varphi)P_{|0\rangle_{\text{aux}}\langle 0|_{\text{aux}}}\hat{U}_Q\hat{H}\,|1\rangle_{\text{anc}}|\mathbf{0}\rangle\\ &\quad=\frac{1}{\sqrt{2}}\frac{1}{1+\alpha_{\text{succ}}}P_{|0\rangle_{\text{aux}}\langle 0|_{\text{aux}}}\left[|0\rangle_{\text{anc}}\left(-\cos\left(\frac{\varphi}{2}\right)|\mathbf{0}\rangle+\sin\left(\frac{\varphi}{2}\right)\hat{U}_Q|\mathbf{0}\rangle\right)+|1\rangle_{\text{anc}}\left(\sin\left(\frac{\varphi}{2}\right)|\mathbf{0}\rangle+\cos\left(\frac{\varphi}{2}\right)\hat{U}_Q|\mathbf{0}\rangle\right)\right],\end{aligned} \tag{G2}$$

For optimally trained $\boldsymbol{\theta}_{j+1}$, measuring the global ancilla qubit yields the expectation value

$$\begin{aligned}\langle\sigma_z\rangle_{\text{anc}} &= \frac{2\sin(\varphi)-\left(\sqrt{\alpha_{\text{succ}}}-\frac{1}{\sqrt{\alpha_{\text{succ}}}}\right)\cos(\varphi)}{1+\alpha_{\text{succ}}}\text{Re}\,\langle\mathbf{0}|\,\hat{U}_Q\,|\mathbf{0}\rangle\\ &= \frac{1}{f_{\hat{O},j}}\text{Re}\,\langle\mathbf{0}|\,\hat{U}_Q\,|\mathbf{0}\rangle\,.\end{aligned} \tag{G3}$$

This yields

$$f_{\hat{O},j}=\frac{1+\alpha_{\text{succ}}}{2\sin(\varphi)-\left(\sqrt{\alpha_{\text{succ}}}-\frac{1}{\sqrt{\alpha_{\text{succ}}}}\right)\cos(\varphi)} \tag{G4}$$

for the norm constant $f_{\hat{O},j}$ of the adapted Hadamard test with the special case of the standard Hadamard test ($\varphi=\pi/2$):

$$f_{\hat{O},j}(\varphi=\pi/2)=\frac{1+\alpha_{\text{succ}}}{2}. \tag{G5}$$

With this norm constant $f_{\hat{O},j}$, which depends on the success probability $\alpha_{\text{succ}}$ and the rotation angle $\varphi$ of the adapted Hadamard test, $\langle\hat{O}\rangle$ can then be calculated according to Eq. (G1).

If all operators of a differential equation are summarized within $\hat{O}$ and assuming optimally trained $\boldsymbol{\theta}_{j+1}$, there is an optimal angle $\varphi_{\text{opt}}=2\arctan(\sqrt{\alpha_{\text{succ}}})$ that leads to $\langle\sigma_z\rangle_{\text{anc}}=1$ and $f_{\hat{O},j}=\sqrt{\alpha}$.

Furthermore, we can derive a measure of convergence from Eq. (G2), which allows to compute the fidelity $\mathcal{F}=||\,\langle\mathbf{0}|P_{|0\rangle_{\text{aux}}\langle 0|_{\text{aux}}}\hat{U}_Q|\mathbf{0}\rangle\,||^2$ between the normalized trained solution and the real solution from $\langle\sigma_z\rangle_{\text{anc}}$. For simplicity, we assume optimal $\varphi=\varphi_{\text{opt}}$. For general $\boldsymbol{\theta}_{j+1}$, $\|\,\langle\mathbf{0}|\,\hat{U}_Q\,|\mathbf{0}\rangle\,\|^2=\alpha_{\text{succ}}\mathcal{F}$ and the global expectation value can be computed from Eq. (G2), reading

$$\langle\sigma_z\rangle_{\text{anc}}=\frac{2\sqrt{\alpha_{\text{succ}}\mathcal{F}}\sin(\varphi_{\text{opt}})-(\alpha_{\text{succ}}-1)\cos(\varphi_{\text{opt}})}{1+\alpha_{\text{succ}}}, \tag{G6}$$

which allows us to infer the fidelity of the trained solution during the training as

$$\mathcal{F}=\frac{(\alpha_{\text{succ}}\langle\sigma_z\rangle_{\text{anc}}+\alpha_{\text{succ}}\cos(\varphi_{\text{opt}})+\langle\sigma_z\rangle_{\text{anc}}-\cos(\varphi_{\text{opt}}))^2}{4\alpha_{\text{succ}}\sin^2(\varphi_{\text{opt}})}. \tag{G7}$$

## APPENDIX H: COST FUNCTIONS

In this Appendix, we introduce the utilized cost functions for the Euler time stepping and the fourth-order Runge Kutta scheme.

### 1. Euler time stepping

To simulate the time evolution of the Burgers' equation and the advection-diffusion equation, we employ the explicit Euler time stepping and summarize both PDEs within one quantum circuit. Hence, the equation we need to solve can be summarized as

$$\left(\mathbf{1}+dt\frac{\partial}{\partial t}\right)\phi(x,t)=\hat{O}\phi(x,t). \tag{H1}$$

When the solution at time step $t_j = n \cdot dt$ is known, $\phi(t_{j+1})$ is computed according to

$$\phi(x, t_{j+1}) = (\mathbf{1} + dt\frac{\partial}{\partial t})\phi(x, t_j) = \hat{O}_{\text{full}}\phi(x, t_j),$$
$$t_{j+1} = t_j + dt. \tag{H2}$$

This results in a cost function

$$C(\theta_{j+1}) = \| \, |\phi_{t_{j+1}}\rangle - \hat{O}_{\text{full}} \, |\phi_{t_j}\rangle \, \|^2$$
$$\propto -\Re \, \langle \phi_{t_{j+1}} | \, \hat{O}_{\text{full}} \, |\phi_{t_j}\rangle + \text{constant}. \tag{H3}$$

### 2. Fourth-order Runge Kutta time stepping

To perform the time evolution of the linearized Eulers' equation shown in the main text, we implemented a fourth-order Runge Kutta (RK4) scheme. Consider a differential equation of form

$$\frac{\partial \phi(x,t)}{\partial t} = g(t, \phi(x, t_j)). \tag{H4}$$

When the solution at time step $t_j = n \cdot dt$ is known, $\phi(t_{j+1})$ is computed according to

$$\phi(x, t_{j+1}) = \phi(x, t_j) + \frac{dt}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$
$$t_{j+1} = t_j + dt, \tag{H5}$$

where

$$k_1 = g(t_j, \phi(x, t_j)),$$
$$k_2 = g\Big(t_j + \frac{dt}{2}, \phi(x, t_j) + \frac{dt}{2}k_1\Big),$$
$$k_3 = g\Big(t_j + \frac{dt}{2}, \phi(x, t_j) + \frac{dt}{2}k_2\Big),$$
$$k_4 = g\big(t_j + dt, \phi(x, t_j) + dt k_3\big). \tag{H6}$$

For the quantum solver, we need to translate this procedure into cost functions that are to be minimized. According to the five equations required to compute the final $\phi(x, t_{j,m+1})$, we define five different optimization steps. The steps do not compute the $k_m$ directly, but instead the sum of $\phi^*_{j,m} = c^{st}_m \phi_j + c^{rk}_m k_m$, choosing $c^{st}_m$ and $c^{rk}_m$ such that $\phi^*$ corresponds to the right-hand side of Eq. (H6) for each RK4 step:

$$C^\phi_1 = \big\| |\phi^*_1\rangle - |\phi_j\rangle - \tfrac{1}{2}dt \, |k_1\rangle \big\|^2,$$
$$C^\phi_2 = \big\| |\phi^*_2\rangle - |\phi_j\rangle - \tfrac{1}{2}dt \, |k_2\rangle \big\|^2,$$
$$C^\phi_3 = \big\| |\phi^*_3\rangle - |\phi_j\rangle - dt \, |k_3\rangle \big\|^2,$$
$$C^\phi_4 = \big\| |\phi^*_4\rangle + \tfrac{1}{3} |\phi_n\rangle - \tfrac{1}{6}dt \, |k_4\rangle \big\|^2,$$
$$C^\phi_5 = \big\| |\phi_{\text{final}}\rangle - \tfrac{1}{3} |\phi^*_1\rangle - \tfrac{2}{3} |\phi^*_2\rangle - \tfrac{2}{3} |\phi^*_3\rangle - |\phi^*_4\rangle \big\|^2, \tag{H7}$$

where $|\phi^*_m\rangle = \theta^0_{j,m}\hat{U}(\boldsymbol{\theta}_{j,m})\,|0\rangle$ and $|\phi_{\text{final}}\rangle = \theta^0_{j+1}\hat{U}(\boldsymbol{\theta}_{j+1})\,|0\rangle$. The indices $j$ and $m$ refer to the index of the time step and the Runge Kutta step, respectively. In the considered example, the linear Euler equation, pressure and velocity are coupled. Hence, the right-hand side of Eq. (H6) depends on both the pressure and velocity field, i.e., $f(t, |\phi(x, t_j)\rangle) \to f(t, |p(x, t_j)\rangle, |u(x, t_j)\rangle)$. In the quantum register, the pressure and velocity field are encoded as $|p(x, t_j)\rangle = \theta^0_n\hat{P}(\boldsymbol{\theta}_j)\,|0\rangle$ and $|u(x, t_j)\rangle = \theta^0_j\hat{U}(\boldsymbol{\theta}_j)\,|0\rangle$. The first four cost functions $C_m$ can be constructed from Eq. (6) and follow the scheme

$$\begin{aligned}
&C^p_m(\boldsymbol{\theta}_{j,m+1})\\
&= \| \, |p^*_{j,m+1}\rangle - c^{st}_m \, |p_{j,m}\rangle + c^{rk}_m \cdot dt(\rho \cdot c^2\hat{\nabla} \, |u_{j,m}\rangle\\
&\quad + \gamma(\hat{x}) \, |p_{j,m}\rangle - A_0 \sin(\omega t) \, |\delta(x)\rangle)\|^2\\
&= \big(\theta^0_{j,m+1}\big)^2 - 2\theta^0_{j,m+1}\theta^0_{j,m}\Re \, \langle 0| \, \hat{P}^\dagger(\boldsymbol{\theta}_{j,m+1})\hat{P}(\boldsymbol{\theta}_{j,m}) \, |0\rangle\\
&\quad + 2dt\rho c^2 f^u_{\hat{\nabla},j}\theta^0_{j,m+1}\theta^0_n\Re \, \langle 0| \, \hat{P}^\dagger(\boldsymbol{\theta}_{j,m+1})\hat{\nabla}_{\text{MPO}}\hat{U}(\boldsymbol{\theta}_{j,m}) \, |0\rangle\\
&\quad + 2f^p_{\hat{\gamma},j,m}\Re\theta^0_{j,m+1}\theta^0_{j,m} \, \langle 0| \, \hat{P}^\dagger(\boldsymbol{\theta}_{j,m+1})\hat{\gamma}_{\text{MPO}}\hat{P}(\boldsymbol{\theta}_{j,m}) \, |0\rangle\\
&\quad - 2A_0 \sin(\omega t)\Re\theta^0_{j,m+1}\theta^0_{j,m} \, \langle 0| \, \hat{P}^\dagger(\boldsymbol{\theta}_{j,m+1}) \, |\delta(x)\rangle\\
&\quad + \text{const.}
\end{aligned} \tag{H8}$$

and

$$\begin{aligned}
&C^u_m(\boldsymbol{\theta}_{j,m+1})\\
&= \bigg\| \, |u_{j,m+1}\rangle - c^{st}_m \, |u_{j,m}\rangle + c^{rk}_m \cdot dt\bigg(\frac{1}{\rho} \cdot \hat{\nabla} \, |p_{j,m}\rangle\\
&\quad + \hat{\gamma}(x) \, |u_{j,m}\rangle\bigg)\bigg\|^2\\
&= \big(\theta^0_{j,m+1}\big)^2 - 2\theta^0_{j,m+1}\theta^0_{j,m}\Re \, \langle 0| \, \hat{U}^\dagger(\boldsymbol{\theta}_{j,m+1})\hat{U}(\boldsymbol{\theta}_{j,m}) \, |0\rangle\\
&\quad + 2\frac{dt}{\bar{\rho}} f^p_{\hat{\nabla},j,m}\theta^0_{j,m+1}\theta^0_{j,m}\Re \, \langle 0| \, \hat{U}^\dagger(\boldsymbol{\theta}_{j,m+1})\hat{\nabla}_{\text{MPO}}\hat{P}(\boldsymbol{\theta}_{j,m}) \, |0\rangle\\
&\quad + 2f^u_{\hat{\gamma},j,m}\theta^0_{j,m+1}\theta^0_{j,m}\Re \, \langle 0| \, \hat{U}^\dagger(\boldsymbol{\theta}_{j,m+1})\hat{\gamma}_{\text{MPO}}\hat{U}(\boldsymbol{\theta}_{j,m}) \, |0\rangle\\
&\quad + \text{const.}
\end{aligned} \tag{H9}$$

The last cost functions $C^{u/p}_5$ only depend on one field, and hence do not differ from Eq. (H7).

[1] F. Black and M. Scholes, The pricing of options and corporate liabilities, J. Political Econ. **81**, 637 (1973).

[2] S. T. Lee and H.-W. Sun, Fourth-order compact scheme with local mesh refinement for option pricing in jump-diffusion model, Numer. Methods Partial **28**, 1079 (2012).

[3] X. Bian, C. Kim, and G. E. Karniadakis, 111 years of Brownian motion, Soft Matter **12**, 6331 (2016).

[4] X. Wu, Y. Zhang, and S. Mao, Learning the physics-consistent material behavior from measurable data via PDE-constrained optimization, Comput. Methods Appl. Mech. Eng. **437**, 117748 (2025).

[5] R. Courant, K. Friedrichs, and H. Lewy, On the partial difference equations of mathematical physics, IBM J. Res. Dev. **11**, 215 (1967).

[6] J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics*, 3rd ed. (Springer, Berlin, 2002).

[7] P. Wu, Recent advances in the application of computational fluid dynamics in the development of rotary

blood pumps, Med. Nov. Technol. Devices **16**, 100177 (2022).
[8] J. P. Slotnick, A. Khodadoust, J. J. Alonso, D. L. Darmofal, W. Gropp, E. A. Lurie, and D. J. Mavriplis, *CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences* (National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia, 2014).
[9] F. R. Menter, Two-equation Eddy-viscosity turbulence models for engineering applications, AIAA J. **32**, 1598 (1994).
[10] D. Wilcox, Turbulence modeling for CFD, *DCW Industries, La Canada* (DCW Industries, Inc., Canada, 1998).
[11] S. B. Pope, *Turbulent Flows* (Cambridge University Press, Cambridge, 2000).
[12] P. Sagaut, *Large Eddy Simulation for Incompressible Flows: An Introduction* (Springer, Berlin, Heidelberg, 2005).
[13] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, A dynamic subgrid-scale Eddy viscosity model, Phys. Fluids A: Fluid Dyn. **3**, 1760 (1991).
[14] G.-L. Long and Y. Sun, Efficient scheme for initializing a quantum register with an arbitrary superposed state, Phys. Rev. A **64**, 014303 (2001).
[15] 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).
[16] D. Jaksch, P. Givi, A. J. Daley, and T. Rung, Variational quantum algorithms for computational fluid dynamics, AIAA J. **61**, 1885 (2023).
[17] P. Givi, A. J. Daley, D. Mavriplis, and M. Malik, Quantum speedup for aeroscience and engineering, AIAA J. **58**, 3715 (2020).
[18] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett. **103**, 150502 (2009).
[19] A. Ambainis, Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations, arXiv:1010.4458.
[20] 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).
[21] J. Penuel, A. Katabarwa, P. D. Johnson, C. Farquhar, Y. Cao, and M. C. Garrett, Feasibility of accelerating incompressible computational fluid dynamics simulations with fault-tolerant quantum computers, arXiv:2406.06323.
[22] S. Lloyd, G. D. Palma, C. Gokler, B. Kiani, Z.-W. Liu, M. Marvian, F. Tennie, and T. Palmer, Quantum algorithm for nonlinear differential equations, arXiv:2011.06571.
[23] P. Brearley and S. Laizet, Quantum algorithm for solving the advection equation using Hamiltonian simulation, Phys. Rev. A **110**, 012430 (2024).
[24] P. Over, S. Bengoechea, P. Brearley, S. Laizet, and T. Rung, Quantum algorithm for the advection-diffusion equation with optimal success probability, Phys. Rev. A **112**, L010401 (2025).
[25] 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).
[26] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).
[27] M. Cerezo, K. Sharma, A. Arrasmith, and P. Coles, Variational quantum state eigensolver, npj Quantum Inf. **8**, 113 (2022).
[28] G. Q. AI and Collaborators, Quantum error correction below the surface code threshold, Nature (London) **638**, 920 (2025).
[29] B. W. Reichardt, D. Aasen, R. Chao, A. Chernoguzov, W. van Dam, J. P. Gaebler, D. Gresh, D. Lucchetti, M. Mills, S. A. Moses, B. Neyenhuis, A. Paetznick, A. Paz, P. E. Siegfried, M. P. da Silva, K. M. Svore, Z. Wang, and M. Zanner, Demonstration of quantum computation and error correction with a tesseract code, arXiv:2409.04628.
[30] W. van Dam, H. Liu, G. H. Low, A. Paetznick, A. Paz, M. Silva, A. Sundaram, K. Svore, and M. Troyer, End-to-end quantum simulation of a chemical system, arXiv:2409.05835.
[31] Quera computing releases a groundbreaking roadmap for advanced error-corrected quantum computers, pioneering the next frontier in quantum innovation, https://www.quera.com/press-releases/quera-computing-releases-a-groundbreaking-roadmap-for-advanced-error-corrected-quantum-computers-pioneering-the-next-frontier-in-quantum-innovation-0, visited on 2025-10-06.
[32] IBM technology atlas quantum roadmap, https://www.ibm.com/roadmaps/quantum.pdf, visited on 2024-10-10.
[33] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, Variational quantum algorithms for nonlinear problems, Phys. Rev. A **101**, 010301 (2020).
[34] M. Schilling, F. Preti, M. M. Müller, T. Calarco, and F. Motzoi, Exponentiation of parametric Hamiltonians via unitary interpolation, Phys. Rev. Res. **6**, 043278 (2024).
[35] W. J. Huggins, J. Lee, U. Baek, B. O'Gorman, and K. B. Whaley, A non-orthogonal variational quantum eigensolver, New J. Phys. **22**, 073009 (2020).
[36] M. Umer, E. Mastorakis, S. Evangelou, and D. G. Angelakis, Probing the limits of variational quantum algorithms for nonlinear ground states on real quantum hardware: The effects of noise, Phys. Rev. A **111**, 012626 (2025).
[37] A. J. Pool, A. D. Somoza, C. Mc Keever, M. Lubasch, and B. Horstmann, Nonlinear dynamics as a ground-state solution on quantum computers, Phys. Rev. Res. **6**, 033257 (2024).
[38] Y. Shao, F. Wei, S. Cheng, and Z. Liu, Simulating noisy variational quantum algorithms: A polynomial approach, Phys. Rev. Lett. **133**, 120603 (2024).
[39] E. Rosenberg, P. Ginsparg, and P. L. McMahon, Experimental error mitigation using linear rescaling for variational quantum eigensolving with up to 20 qubits, Quantum Sci. Technol. **7**, 015024 (2022).
[40] B. Jaderberg, A. Agarwal, K. Leonhardt, M. Kiffner, and D. Jaksch, Minimum hardware requirements for hybrid quantum–classical DMFT, Quantum Sci. Technol. **5**, 034015 (2020).
[41] B. Jaderberg, A. Eisfeld, D. Jaksch, and S. Mostame, Recompilation-enhanced simulation of electron–phonon dynamics on IBM quantum computers, New J. Phys. **24**, 093017 (2022).
[42] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, Evidence for the utility of quantum computing before fault tolerance, Nature (London) **618**, 500 (2023).

[43] I. V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. **33**, 2295 (2011).

[44] R. Orús, A practical introduction to tensor networks: Matrix product states and projected entangled pair states, Ann. Phys. **349**, 117 (2014).

[45] I. V. Oseledets, Constructive representation of functions in low-rank tensor formats, Constr. Approximation **37**, 1 (2013).

[46] E. Ye and N. F. G. Loureiro, Quantum-inspired method for solving the Vlasov-Poisson equations, Phys. Rev. E **106**, 035208 (2022).

[47] M. Kiffner and D. Jaksch, Tensor network reduced order models for wall-bounded flows, Phys. Rev. Fluids **8**, 124101 (2023).

[48] E. Kornev, S. Dolgov, K. Pinto, M. Pflitsch, M. Perelshtein, and A. Melnikov, Numerical solution of the incompressible Navier-Stokes equations for chemical mixers via quantum-inspired tensor train finite element method, arXiv:2305.10784.

[49] E. Ye and N. F. Loureiro, Quantized tensor networks for solving the Vlasov–Maxwell equations, J. Plasma Phys. **90**, 805900301 (2024).

[50] R. D. Peddinti, S. Pisoni, A. Marini, P. Lott, H. Argentieri, E. Tiunov, and L. Aolita, Quantum-inspired framework for computational fluid dynamics, Commun. Phys. **7**, 135 (2024).

[51] L. Hölscher, P. Rao, L. Müller, J. Klepsch, A. Luckow, T. Stollenwerk, and F. K. Wilhelm, Quantum-inspired fluid simulation of two-dimensional turbulence with GPU acceleration, Phys. Rev. Res. **7**, 013112 (2025).

[52] N. Gourianov, P. Givi, D. Jaksch, and S. B. Pope, Tensor networks enable the calculation of turbulence probability distributions, Sci. Adv. **11**, eads5990 (2025).

[53] J. Schachenmayer, B. P. Lanyon, C. F. Roos, and A. J. Daley, Entanglement growth in quench dynamics with variable range interactions, Phys. Rev. X **3**, 031015 (2013).

[54] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC'96 (Association for Computing Machinery, New York, 1996), pp. 212–219.

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

[56] D. Malz, G. Styliaris, Z.-Y. Wei, and J. I. Cirac, Preparation of matrix product states with log-depth quantum circuits, Phys. Rev. Lett. **132**, 040404 (2024).

[57] K. C. Smith, A. Khan, B. K. Clark, S. Girvin, and T.-C. Wei, Constant-depth preparation of matrix product states with adaptive quantum circuits, PRX Quantum **5**, 030344 (2024).

[58] M. Nibbi and C. B. Mendl, Block encoding of matrix product operators, Phys. Rev. A **110**, 042427 (2024).

[59] A. Termanova, A. Melnikov, E. Mamenchikov, N. Belokonev, S. Dolgov, A. Berezutskii, R. Ellerbrock, C. Mansell, and M. Perelshtein, Tensor quantum programming, New J. Phys. **26**, 123019 (2024).

[60] M. Goldack, Y. Atia, O. Alberton, and K. Jansen, Computing statistical properties of velocity fields on current quantum hardware, arXiv:2601.10166.

[61] F. Uchida, K. Miyamoto, S. Yamazaki, K. Fujisawa, and N. Yoshida, Quantum simulation of burgers turbulence: Nonlinear transformation and direct evaluation of statistical quantities, arXiv:2412.17206.

[62] E. Mastorakis, M. Umer, M. Guevara-Bertsch, J. Ulmanis, F. Rohde, and D. G. Angelakis, Resource-efficient Hadamard test circuits for nonlinear dynamics on a trapped-ion quantum computer, arXiv:2507.19250.

[63] A. Sarma, T. W. Watts, M. Moosa, Y. Liu, and P. L. McMahon, Quantum variational solving of nonlinear and multidimensional partial differential equations, Phys. Rev. A **109**, 062616 (2024).

[64] P. Over, S. Bengoechea, T. Rung, F. Clerici, L. Scandurra, E. de Villiers, and D. Jaksch, Boundary treatment for variational quantum simulations of partial differential equations on quantum computers, Comput. Fluids **288**, 106508 (2025).

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

[66] M. Ben-Dov, D. Shnaiderov, A. Makmal, and E. G. Dalla Torre, Approximate encoding of quantum states using shallow circuits, npj Quantum Inf. **10**, 65 (2024).

[67] V. A. Kazeev and B. N. Khoromskij, Low-rank explicit QTT representation of the Laplace operator and its inverse, SIAM J. Matrix Anal. Appl. **33**, 742 (2012).

[68] I. V. Oseledets, Approximation of $2^d \times 2^d$ matrices using tensor decomposition, SIAM J. Matrix Anal. Appl. **31**, 2130 (2010).

[69] H. Zhao, M. Bukov, M. Heyl, and R. Moessner, Making trotterization adaptive and energy-self-correcting for NISQ devices and beyond, PRX Quantum **4**, 030319 (2023).

[70] R. Puig, M. Drudis, S. Thanasilp, and Z. Holmes, Variational quantum simulation: A case study for understanding warm starts, PRX Quantum **6**, 010317 (2025).

[71] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[72] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99**, 032331 (2019).

[73] J. Spall, An overview of the simultaneous perturbation method for efficient optimization, Johns Hopkins Apl Technical Digest **19**, 482 (1998).

[74] A. J. Chorin and J. E. Marsden, *A Mathematical Iintroduction to Fluid Mechanics* edited by J.E. Marsden (Caltech), L. Sirovich (Brown University), M. Golubitsky (University of Houston) (Springer Science+Business Media, New York, 1979).

[75] R. Peric and M. Abdel-Maksoud, Reliable damping of free surface waves in numerical simulations, Sh. Technol. Res. **63** (2015).

[76] F. H. Harlow and J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids **8**, 2182 (1965).

[77] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary ed. (Cambridge University Press, Cambridge, 2010).

[78] R. Haghshenas, J. Gray, A. C. Potter, and G. K.-L. Chan, Variational power of quantum circuit tensor networks, Phys. Rev. X **12**, 011047 (2022).

[79] D. Wang, O. Higgott, and S. Brierley, Accelerated variational quantum eigensolver, Phys. Rev. Lett. **122**, 140504 (2019).

[80] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, Iterative quantum amplitude estimation, npj Quantum Inf. **7**, 52 (2021).

[81] Y. Li, E. Perlman, M. Wan, Y. Yang, C. Meneveau, R. Burns, S. Chen, A. Szalay, and G. Eyink, A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence, J. Turbul. **9**, N31 (2008).

[82] E. Perlman, R. Burns, Y. Li, and C. Meneveau, Data exploration of turbulence simulations using a database cluster, in *Proceedings of the 2007 ACM/IEEE Conference on Supercomputing (SC '07)*, 1-11, https://api.semanticscholar.org/CorpusID:11021323.

[83] John Hopkins turbulence database—isotropic turbulence.

[84] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. **326**, 96 (2011).

[85] F. G. S. L. Brandão, A. W. Harrow, and M. Horodecki, Local random quantum circuits are approximate polynomial-designs, Commun. Math. Phys. **346**, 397 (2016).

[86] K. Goswami, P. Schmelcher, and R. Mukherjee, Qudit-based scalable quantum algorithm for solving the integer programming problem, arXiv:2508.13906.

[87] V. V. Shende, I. L. Markov, and S. S. Bullock, Minimal universal two-qubit controlled-not-based circuits, Phys. Rev. A **69**, 062321 (2004).

[88] M. Lubasch, P. Moinier, and D. Jaksch, Multigrid renormalization, J. Comput. Phys. **372**, 587 (2018).

[89] L. Slattery, B. Villalonga, and B. K. Clark, Unitary block optimization for variational quantum algorithms, Phys. Rev. Res. **4**, 023072 (2022).

[90] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem, arXiv:quant-ph/9511026.

[91] S. Fomichev, K. Hejazi, M. S. Zini, M. Kiser, J. Fraxanet, P. A. M. Casares, A. Delgado, J. Huh, A.-C. Voigt, J. E. Mueller, and J. M. Arrazola, Initial state preparation for quantum chemistry on quantum computers, PRX Quantum **5**, 040339 (2024).

[92] P. Siegl, G. S. Reese, T. Hashizume, N.-L. van Hülst, and D. Jaksch, Dataset tensor-programmable quantum circuits for solving differential equations (version 3) (2025), http://doi.org/10.25592/uhhfdm.18138.

[93] V. Bergholm *et al.*, PennyLane: Automatic differentiation of hybrid quantum-classical computations, arXiv:1811.04968.

[94] A. Paszke *et al.*, PyTorch: An imperative style, high-performance deep learning library, arXiv:1912.01703.

[95] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, JAX: Composable transformations of Python + NumPy programs (2018), https://github.com/jax-ml/jax.

[96] I. A. Luchnikov, M. E. Krechetov, and S. N. Filippov, Riemannian geometry and automatic differentiation for optimization problems of quantum physics and quantum technologies, New J. Phys. **23**, 073006 (2021).

[97] J. Li, F. Li, and S. Todorovic, Efficient Riemannian optimization on the Stiefel manifold via the Cayley transform, arXiv:2002.01113.

[98] G. Strang, *Introduction to Linear Algebra* (Wellesley, Cambridge, 2016).

[99] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 4812 (2018).

[100] A. A. Mele, G. B. Mbeng, G. E. Santoro, M. Collura, and P. Torta, Avoiding barren plateaus via transferability of smooth solutions in a Hamiltonian variational ansatz, Phys. Rev. A **106**, L060401 (2022).