Hamiltonian Simulation for Advection-Diffusion Equation with arbitrary transport field

Niladri Gomes[1], Gautam Sharma[2], and Jay Pathak[1]

[1]Ansys, Inc, San Jose, CA
[2]Ansys Software Pvt. Ltd., Pune, India

***Abstract*—We present a novel approach to solve the advection-diffusion equation under arbitrary transporting fields using a quantum-inspired 'Schrödingerisation' technique for Hamiltonian simulation. Although numerous methods exist for solving partial differential equations (PDEs), Hamiltonian simulation remains a relatively underexplored yet promising direction—particularly in the context of long-term, fault-tolerant quantum computing. Building on this potential, our quantum algorithm is designed to accommodate non-trivial, spatially varying transport fields and is applicable to both 2D and 3D advection-diffusion problems. To ensure numerical stability and accuracy, the algorithm combines an upwinding discretization scheme for the advective component and the central differencing for diffusion, adapted for quantum implementation through a tailored mix of approximation and optimization techniques. We demonstrate the algorithm's effectiveness on benchmark scenarios involving coupled rotational, shear, and diffusive transport in two and three dimensions. Additionally, we implement the 2D advection-diffusion equation using 16 qubits on IBM Quantum hardware, validating our method and highlighting its practical applicability and robustness.**

***Index Terms*—Hamiltonian Simulation, Schrödingerisation, Computational Fluid Dynamics, Advection-Diffusion Equation.**

I. Introduction

Achieving quantum speedup for the numerical solution of ordinary and partial differential equations (ODEs/PDEs), beyond those specific to the Schrödinger equation, holds considerable promise for advancing a wide range of scientific and engineering disciplines [1], [2], [3]. PDEs play a central role in modeling the dynamic behavior of physical systems and arise in numerous applications, including reservoir simulation, climate and ocean circulation, and wave propagation. In particular, computational fluid dynamics (CFD) relies fundamentally on the numerical solutions of PDEs that describe fluid flow phenomena [4]. While high-performance computing (HPC) has enabled significant strides in solving such complex models, the computational cost remains a persistent challenge, especially in problems involving large-scale domains, high dimensionality, or multiscale dynamics.

The scalar advection-diffusion equation provides a canonical building block in conventional numerical methods development for computational fluid dynamics [5]. It encapsulates two fundamental mechanisms of transport, advection, associated with bulk motion, and diffusion, representing molecular or turbulent mixing [6]. Despite not completely covering all complexities such as pressure-velocity coupling and nonlinearities expressible with the complete Navier-Stökes set of governing equations, the scalar advection-diffusion equation presents a model analytical framework to test and benchmark early algorithms. It captures both the hyperbolic and parabolic behaviours in a single framework, providing for a robust and amenable prototype problem class. Quantum algorithms for time-dependent problems governed by PDEs can generally be categorized into two categories: a) the variational quantum simulation methods [7], [8] and b) Hamiltonian simulation techniques [9]. The latter represents a more promising direction for long-term, fault-tolerant quantum computing [9], [10]. Hamiltonian simulation refers to the construction of a quantum circuit that models the time evolution of a quantum system, typically represented by the unitary operator $e^{-i\mathbf{H}t}$, where $\mathbf{H}$ denotes the system's Hamiltonian and t is the time increment. Studies have explored the use of Hamiltonian simulation by reformulating the governing equations of classical systems into an equivalent Schrödinger equation [11], [12], [13].

Owing to inherent algorithmic and hardware constraints, computational fluid dynamics (CFD) presents a compelling opportunity for exploring quantum acceleration in classical engineering applications and remains an active area of investigation [2], [14], [15], [16], [17], [18]. Steijl et al. [19] demonstrated that a Poisson solver used in a vortex-in-cell approach could be reformulated as a hybrid quantum algorithm through the application of the quantum Fourier transform. Gaitan [20] proposed a quantum algorithm for solving the steady-state, inviscid, one-dimensional, compressible Navier–Stokes equations (NSE) in a converging-diverging nozzle, leveraging techniques from quantum ODE solvers [21]. Extending this work, Oz et al. [22] implemented a quantum algorithm to address the nonlinear Burgers equation. In parallel, Suau et al. [23] showcased a quantum approach for solving the wave equation based on the Hamiltonian simulation framework introduced by Costa et al. [11]. A recent study by Lee et al. [24] proposed a multiple-circuit approach to reduce quantum resource requirements in the quantum lattice Boltzmann method, offering a practical direction for quantum computational fluid dynamics. Brearley et al. used an explicit time marching strategy for solving the advection equation by embedding numerical integrators into a series of Hamiltonian simulations [25]. While these findings indicate that quantum algorithms have the potential to achieve speedup in simulating classical systems, they either typically assume oracle access to the Hamiltonian, or simulate idealistic models[26] or may not apply to a general class of PDEs [25].

With a goal of building a long term general purpose PDE solver, here we adopt the Hamiltonian simulation approach with Schrödingerisation. In order to move forward with realistic calculations, we consider vortex-dominated flows, characterized by circular or nearly circular streamlines and are governed by the dynamics of vorticity, a fundamental quantity in fluid mechanics [4], [27]. Vorticity, defined as the curl of the spatially varying transport field, quantifies the local rotational behavior of fluid elements and provides insight into the structure and evolution of complex flow fields. In particular, we will consider a transport field with non-zero vorticity, where the direction of the vorticity vector denotes the axis of rotation, while its magnitude reflects the intensity of local fluid spin. Vorticity plays a critical role in the analysis and modeling of key fluid dynamic phenomena, including boundary layers, shear layers, wakes, and mixing regions [27].

Our key contributions in this work can be summarized as follows:

- **Algorithmic scaling upto 30 qubits of Hamiltonian simulation of non-trivial transport fields**: We present a new algorithm for solving the 2D/3D advection-diffusion equation with spatially varying transport fields using upwinding discretization scheme for the advective component and central difference scheme for the diffusive component that scales to 30 qubits on emulator.
- **First hardware implementation up to 16 qubits**: Using the same discretization scheme, we implemented evolution of advection-diffusion equation in 2D for a (256x256 grid) with a spatially varying transport field for 40 time steps.
- **Efficient circuits for upwinding scheme**: We introduce practical approximations to the upwinding scheme for the advective component that eliminates the need for ancilla qubits, thereby removing mid-circuit measurements and reducing the overall circuit depth by at least 30%.

II. Theory and Methods

In this section, we will describe the form of the advection-diffusion equation we want to solve and the Schrödingerisation technique applied in order to convert it into a Schrödinger-like equation.

A. *Advection-Diffusion Equation and Schrödingerisation*

A d-dimensional advection-diffusion equation is written as,

$$\begin{aligned}\frac{\partial u(t,\mathbf{x})}{\partial t} &= -v\cdot\boldsymbol{\nabla} u(t,\mathbf{x}) + D\boldsymbol{\nabla}^2 u(t,\mathbf{x})\\ &\equiv -\sum_{\alpha=1}^{d} v_\alpha \partial_\alpha u(t,\mathbf{x}) + D\boldsymbol{\nabla}^2 u(t,\mathbf{x}),\end{aligned} \tag{1}$$

where u is the scalar field moving with transport v in a medium with diffusivity $D \geq 0$. $\boldsymbol{\nabla}$ and $\boldsymbol{\nabla}^2$ are the spatial gradient and diffusion operators respectively. Applying the warped phase transformation $\mathbf{v}(t,p) = e^{-p}u(t,\mathbf{x})$, and subsequently applying a Fourier transform ($\mathcal{F}$) over p, $(\hat{\mathbf{v}}(t,\eta) = \mathcal{F}(\mathbf{v}(t,p)))$, a spatially discretized version of Eq. (1) can be written as

$$\frac{\partial \hat{\mathbf{v}}(t,\mathbf{x},\eta)}{\partial t} = \mathbf{H_D}\hat{\mathbf{v}}(t,\mathbf{x},\eta) + \mathbf{H}\hat{\mathbf{v}}(t,\mathbf{x},\eta), \tag{2}$$

where we refer to $\mathbf{H_D}$ as the diffusion Hamiltonian and $\mathbf{H}$ as the advection Hamiltonian.

The discretization process has been described in [28] with respect to heat and advection equations, such that the $\mathbf{H_D}$ comes from the diffusion term in the heat equation and $\mathbf{H}$ comes from the gradient term of advection equation. For the ease of readers, we describe the process briefly here. We first discretize the variable p over $N_p = 2^{n_p}$ grid points in the warped space $[-\pi R, \pi R]$ (for $R \in \mathbb{R}$) and the variable x_α over $N_\alpha = 2^n_\alpha$ grid points on the physical space $[0, L_\alpha]$ such that the grid size is $h_\alpha = L_\alpha/N_\alpha$. For simplicity, we choose all N_α and $L_\alpha(= L))$ to be the same and hence call all h_α to be $h = 1$. The discretized scalar field $u(t,\mathbf{x})$ is encoded in the quantum state $|u(t,\mathbf{x})\rangle$ using amplitude encoding such that

$$|u(t,\mathbf{x})\rangle = \sum_{j_1=0}^{L-1}\dots\sum_{j_d=0}^{L-1} u(t,x_{j_1}\dots,x_{j_d})\,|j_1\rangle\otimes\dots\otimes|j_d\rangle\,, \tag{3}$$

where u is a normalized array and $|j_i\rangle$ is the computational basis for the x_i axis. We further define the Fourier space variable $\eta_k = (k - N_p/2)$, $k = 0, 1, \dots, N_p - 1$. Using all the discretizing ingredients, the Hamiltonians $\mathbf{H_D}$ and $\mathbf{H}$ are given by,

$$\begin{aligned}\mathbf{H_D} &= \sum_{k=0}^{N_p}\left(k - \frac{N_p}{2}\right)\sum_{\alpha=1}^{d}\mathbf{H}_{D,\alpha}\otimes|k\rangle\langle k|\\ \mathbf{H} &= \sum_{k=0}^{N_p}\left(k - \frac{N_p}{2}\right)\sum_{\alpha=1}^{d}\mathbf{H}_{1,\alpha}\otimes|k\rangle\langle k| + \sum_{\alpha=1}^{d}\mathbf{H}_{2,\alpha}\otimes I^{\otimes N_p}.,\end{aligned} \tag{4}$$

where it should be noted that we use the same set of ancilla qubits for both $\mathbf{H_D}$ and $\mathbf{H}_{1,\alpha}$. In 3D we have $\alpha \in \{x, y, z\}$. Following [28], [26], we can expand $\mathbf{H}_{D,\alpha}$, $\mathbf{H}_{1,\alpha}$ and $\mathbf{H}_{2,\alpha}$ as,

$$\begin{aligned}\mathbf{H}_{D,\alpha} &= \gamma_D^\alpha\left(\left(\sigma_{01}^{\otimes n_\alpha} + \sigma_{10}^{\otimes n_\alpha}\right) + \sum_{j=1}^{n_\alpha}\left(s_j^- + s_j^+\right) - 2I^{\otimes n_\alpha}\right)\\ &\equiv DS_{D,\alpha}\end{aligned} \tag{5}$$

$$\begin{aligned}\mathbf{H}_{1,\alpha} &= \gamma_1^\alpha\left(\left(\sigma_{01}^{\otimes n_\alpha} + \sigma_{10}^{\otimes n_\alpha}\right) + \sum_{j=1}^{n_\alpha}\left(s_j^- + s_j^+\right) - 2I^{\otimes n_\alpha}\right)\\ &\equiv |v_\alpha| S_{1,\alpha}\end{aligned} \tag{6}$$

$$\begin{aligned}\mathbf{H}_{2,\alpha} &= -i\gamma_2^\alpha\left(\left(-\sigma_{01}^{\otimes n_\alpha} + \sigma_{10}^{\otimes n_\alpha}\right) + \sum_{j=1}^{n_\alpha}\left(s_j^- - s_j^+\right)\right)\\ &\equiv v_\alpha S_{2,\alpha}.\end{aligned} \tag{7}$$

where we have separated out the γ's so that we can build our theory on top of the theory of the constant transport model for the advection Hamiltonian. For later convenience, the total Hamiltonian for a given physical dimension labeled with $\alpha \in \{x, y, z\}$ is defined as

$$\mathbf{H}_{\mathbf{D},\alpha} + \mathbf{H}_{\mathbf{1},\alpha} + \mathbf{H}_{\mathbf{2},\alpha} = DS_{D,\alpha} + |v_\alpha| S_{1,\alpha} + v_\alpha S_{2,\alpha}.$$

Writing the γ's explicitly,

$$\gamma_D^\alpha = \frac{D}{h^2 R}, \gamma_1^\alpha = \frac{|v_\alpha|}{2hR}, \gamma_2^\alpha = \frac{v_\alpha}{2h}. \tag{8}$$

In the above equations, we have used 2×2 matrices,

$$\sigma_{01} = |0\rangle\langle 1| \, ; \; \sigma_{10} = |1\rangle\langle 0| \tag{9}$$
$$\sigma_{00} = |0\rangle\langle 0| \, ; \; \sigma_{11} = |1\rangle\langle 1| \tag{10}$$

and

$$s_j^+ = I^{\otimes(n_x - j)} \otimes \sigma_{10} \otimes \sigma_{01}^{\otimes(j)} \tag{11}$$
$$s_j^- = I^{\otimes(n_x - j)} \otimes \sigma_{01} \otimes \sigma_{10}^{\otimes(j)} \tag{12}$$

B. Circuit Implementation

The full unitary for the advection-diffusion evolution $U_{\text{adv-diff}}(t)$ is implemented via three Hamiltonian operators $\mathbf{H}_{\mathbf{D},\alpha}, \mathbf{H}_{\mathbf{1},\alpha}$, and $\mathbf{H}_{\mathbf{2},\alpha}$. This unitary can be approximated as

$$\begin{aligned} U_{\text{adv-diff}}(t) &= \exp\{i(\mathbf{H}_{\mathbf{D},\alpha} + \mathbf{H}_{\mathbf{1},\alpha} + \mathbf{H}_{\mathbf{2},\alpha})dt\} \\ &\approx \exp(i\mathbf{H}_{2,\alpha}dt)\exp(i\mathbf{H}_{1,\alpha}dt)\exp(i\mathbf{H}_{D,\alpha}dt) \\ &\approx \left(\tilde{V}_2(t) \otimes I^{\otimes n_p}\right) \times \left(\sum_{k=0}^{N_p-1} \tilde{V}_1^{k-N_p/2}(t) \otimes |k\rangle\langle k|\right) \\ &\times \left(\sum_{k=0}^{N_p-1} \tilde{V}_D^{k-N_p/2}(t) \otimes |k\rangle\langle k|\right), \end{aligned}$$

where we can obtain further simplification by

$$\sum_{k=0}^{N_p-1} \tilde{V}_1^k(t) \otimes |k\rangle\langle k| = \sum_{m=0}^{n_p-1} \tilde{V}_1^{2^m}(t) \otimes |1\rangle\langle 1| + \mathcal{I}^{n_\alpha} \otimes |0\rangle\langle 0|. \tag{13}$$

$$\sum_{k=0}^{N_p-1} \tilde{V}_D^k(t) \otimes |k\rangle\langle k| = \sum_{m=0}^{n_p-1} \tilde{V}_D^{2^m}(t) \otimes |1\rangle\langle 1| + \mathcal{I}^{n_\alpha} \otimes |0\rangle\langle 0|. \tag{14}$$

The resulting full circuit is given in Fig. 1 where the gates $\tilde{V}_D(t)$, $\tilde{V}_1(t)$, and $\tilde{V}_2(t)$ are defined as,

$$\tilde{V}_D(t) \approx \prod_{\alpha=1}^{d} \exp\{-i\mathbf{H}_{D,\alpha}t\}, \tag{15}$$
$$\tilde{V}_1(t) \approx \prod_{\alpha=1}^{d} \exp\{-i\mathbf{H}_{1,\alpha}t\}, \tag{16}$$
$$\tilde{V}_2(t) \approx \prod_{\alpha=1}^{d} \exp\{-i\mathbf{H}_{2,\alpha}t\} \tag{17}$$

Circuit implementation for $\tilde{V}_D(t)$, $\tilde{V}_1(t)$ and $\tilde{V}_2(t)$ are followed from [29] and have been defined in the Appendix A.

C. Spatially varying transport field

As mentioned in the introduction, we will consider the advection-diffusion equation with a spatially varying transport field. Notice that the diffusion component of the circuit plays no role here. To accurately capture such behavior in CFD models, it is essential to incorporate transport fields that preserve vorticity into the governing PDEs. Existing works on solving advection equations with Hamiltonian simulation have only explored a uniform transporting field ($v_x = v_y = c$) [28], [26], [30]. In this work, we will consider an implementation of a non-trivial transport field with spatial dependence. We will denote the qubit registers for x and y directions as $|x\rangle$ and $|y\rangle$, respectively and the scalar field would be represented as

$$|u(x, y, t)\rangle = \sum_{x,y} c_{x,y,t} |x\rangle |y\rangle. \tag{18}$$

We consider the transport field $v = (y, x)$ and work in the first quadrant $y \in [0, L]$ such that $v_x \geq 0$. Therefore, the advection Hamiltonian has the following form

$$\begin{aligned} \mathbf{H} &= \mathbf{H}_\mathbf{x} + \mathbf{H}_\mathbf{y} \\ &= y(S_{1,x} + S_{2,x}) + x(S_{1,y} + S_{2,y}) \\ &= yS_x + xS_y, \end{aligned} \tag{19}$$

where $S_x = S_{1,x} + S_{2,x}$ and $S_y = S_{1,y} + S_{2,y}$. To demonstrate how this Hamiltonian can be simulated in the upwinding scheme, we focus on the implementation of the x-component of the Hamiltonian $\mathbf{H}_\mathbf{x}$. The implementation for $\mathbf{H}_\mathbf{y}$ can be done in a similar way. Let us denote the operator $U(t) = e^{-iS_x t}$, for later convenience. Note that this is the evolution operator for the advection equation with transport $v_x = 1$, the implementation of which is already described in [28]. Using the fact that $y = \sum_{m=0}^{n_x-1} y_m 2^m$, where $y_m \in \{0, 1\}$, it can now be shown that

$$\begin{aligned} e^{-i\mathbf{H}_\mathbf{x}t} |x\rangle &= \sum_{y=0}^{N_y-1} \left(e^{-iyS_x t} \otimes |y\rangle\langle y|\right) |x\rangle \\ &= \sum_{y=0}^{N_y-1} \left(e^{-i(\sum_{m=1}^{n_y} y_m 2^m) S_x t} \otimes |y\rangle\langle y|\right) |x\rangle \\ &= \left(\prod_{m=0}^{n_y-1}{}' e^{-iy_m 2^m S_x t} \otimes \sum_{y_m} |y_m\rangle\langle y_m|\right) |x\rangle \\ &= \left(\prod_{m=0}^{n_y-1}{}' U(y_m 2^m t) \otimes \sum_{y_m} |y_m\rangle\langle y_m|\right) |x\rangle \\ &= \left(\prod_{m=0}^{n_y-1}{}' U(2^m t) \otimes |1\rangle\langle 1| + \mathcal{I}^{n_x} \otimes |0\rangle\langle 0|\right) |x\rangle, \end{aligned} \tag{20}$$

where the primed product $\prod'$ implies direct product with respect to $|x\rangle$ register and tensor product with the $|y\rangle$ register. The above equation can be implemented using the circuit in Fig. 2, where the unitary $U(t)$ is being controlled by $|y\rangle$ qubits.

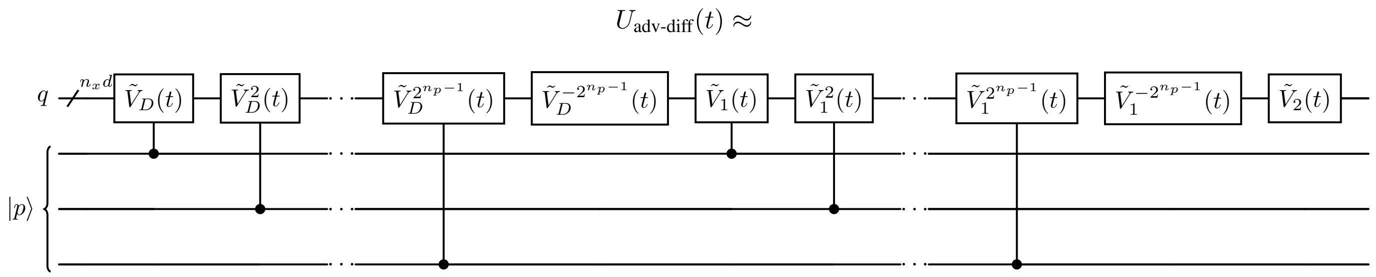
$U_{\text{adv-diff}}(t) \approx$
q
$n_x d$
$\tilde{V}_D(t)$
$\tilde{V}_D^2(t)$
$\tilde{V}_D^{2^{n_p-1}}(t)$
$\tilde{V}_D^{-2^{n_p-1}}(t)$
$\tilde{V}_1(t)$
$\tilde{V}_1^2(t)$
$\tilde{V}_1^{2^{n_p-1}}(t)$
$\tilde{V}_1^{-2^{n_p-1}}(t)$
$\tilde{V}_2(t)$
$|p\rangle$

Fig. 1. Quantum circuit for $U_{\text{adv-diff}}(t)$ for a constant (zero vorticity) transport advection evolution.

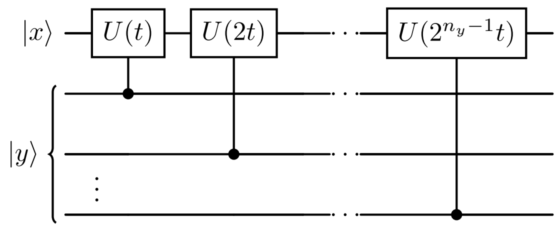

Fig. 2. Modified circuit for incorporating a linearly varying transport field $v = (y, 0)$. $U(2^j t)$ is the unitary that implements advection with transport $v_\alpha = 2^j$.

It should be noted that $U_{\text{adv}}(t) \approx U(t)$ for $v_x = 1$. The above derivation can be extended to transport fields, where $v_x = -y$, i.e., with a negative and linearly varying transport field. In this case,

$$\mathbf{H_x} = yS_{1,x} - yS_{2,x} = yS'_x,$$

where $S'_x = S_{1,x} - S_{2,x}$. By defining $U'(t) = e^{-iS'_x t}$, we can simply follow the previous derivation to obtain the corresponding circuit.

D. *Gate complexity of the modified circuit*

In Ref. [28], complexity analysis for the heat equation and advection equation with uniform transport field was done. Our algorithm is built on top of their method by observing that the heat equation operator is essentially the diffusion operator and appending it with the modified advection equation operator for the spatially varying transport field. Therefore, the gate complexity of the diffusion component Eq. (14) of our algorithm is the same as the heat equation from [28], which states that for $n_\alpha \geq 3$

$$\begin{aligned} 1 - qubit_{Diff} &= \mathcal{O}(N_p n_\alpha), \\ CNOT_{Diff} &= \mathcal{O}(N_p n_\alpha^2), \end{aligned} \tag{21}$$

where $1 - qubit_{Diff}$ and $CNOT_{Diff}$ represent the single qubit and CNOT gate counts respectively for the diffusion component. For d-dimensions the corresponding gate count is simply multiplied with d. Now, the advection component of the circuit that consists of $\tilde{V}_2(t) \otimes I^{\otimes n_p}$ and Eq. (13). For one-dimension, with $n_\alpha \geq 3$ the gate complexity is given by

$$\begin{aligned} 1 - qubit_{Adv} &= 6n_\alpha + 2 + 2^{n_p-1}(4_\alpha + 1) = \mathcal{O}(N_p n_\alpha), \\ CNOT_{Adv} &= (2^{n_p} - 1)(16n_\alpha^2 - 2n_\alpha - 30) \\ &\quad + (2^{n_p-1} + 1)9n_\alpha^2 - 15n_\alpha - 8 = \mathcal{O}(N_p n_\alpha^2), \end{aligned} \tag{22}$$

where $1 - qubit_{Adv}$ and $CNOT_{Adv}$ represent the single qubit and CNOT gate count respectively. For constant transport fields, if there are d-dimensions the corresponding number is obtained by multiplying with d. For spatially varying transport fields, let's consider $d = 1$, since in this case a simple multiplication with d won't give the correct gate count. For the modified circuit in Fig. 2 the corresponding CNOT gate count is given by

$$\begin{aligned} Adv_{CNOT} &= (2^{n_\alpha} - 1)(\mathcal{O}(N_p n_\alpha) + Adv_{CNOT} * 8) \\ &= \mathcal{O}(N_p N_\alpha n_\alpha^2) \approx \mathcal{O}(N_p N_\alpha). \end{aligned} \tag{23}$$

Therefore the CNOT gate count is now exponential with respect to both the system qubits n_α and ancilla qubits n_p. For the full circuit in Fig.1 and for a given dimension α, the corresponding gate complexity is obtained by adding Eq. (21) and Eq. (23), such that

$$\begin{aligned} 1_{qubit} &= \mathcal{O}(N_p n_\alpha), \\ CNOT &= \mathcal{O}(N_p n_\alpha^2) + \mathcal{O}(N_p N_\alpha) \approx \mathcal{O}(N_p N_\alpha). \end{aligned}$$

E. *Efficient circuit implementation without ancillae*

To be able to run our algorithm on currently available quantum hardware that have constraints on the qubit count and circuit depth, we need to make certain approximations to the advective component of the circuit. We will ignore the $c - V_1$ component of the circuit from Fig 1. This approximation is justified due the following to two reasons.

Following [28], the advection Hamiltonian can be written as a sum of the finite difference operators, $\mathbf{H} = \eta A_1 + A_2$ with

$$A_1 = \sum_{\alpha=1}^{d} \frac{|v_\alpha| h}{2} (D_P^{\Delta})_\alpha, \quad A_2 = \sum_{\alpha=1}^{d} \frac{v_\alpha}{i} (D_P^{\pm})_\alpha.$$

where $(D_P^{\Delta})_\alpha$ and $(D_P^{\pm})_\alpha$ represent the second-order and first-order derivatives using the central finite difference respectively.

The primary phenomenon of advection is implemented by the A_2 component which is the first-order derivative using the central difference scheme. Advection with A_2 alone is prone to numerical oscillations [31], [32], which is stabilized by the second-order diffusion term A_1. Evolutions due to $\mathbf{H}_{1,\alpha}$'s are approximated by A_1 and implemented using the quantum gate V_1 Eq. (16). To compare the contribution of the A_1 compared to A_2, consider the following

$$A_{1/2} = \frac{|A_1|}{|A_2|} = \frac{h|D_P^{\triangle}|}{2|D_P^{\pm}|} \approx h\frac{|u''(x)|}{|u'(x)|}. \tag{24}$$

$A_{1/2}$ is small when we have a finer grid, i.e. h is small, and also when the ratio of second-order derivative to first-order derivative is small. In our case, the grid size h is always 1, so it does not affect $A_{1/2}$. However, since we have a superposition of Gaussian states, this ratio is inversely proportional to the grid size, i.e. $A_{1/2} \propto \frac{1}{L^2}$. This happens because the σ is proportional to grid size L. Therefore, as the size of the grid increases, the ratio $A_{1/2}$ becomes increasingly small. Thus, any approximations that we make to the V_1 component (The block in Fig. 1 containing only the unitary gate V_1) of the circuit will not have a significant effect on the final output for a sufficiently large grid size.

Second, the only input to V_1 is $2\gamma_1 dt$. If this input is zero V_1 becomes identity, thus, if we make dt sufficiently small $c - V_1$ (controlled-V_1) will become much closer to identity. Therefore, we calculate the distance of the operator $\mathcal{U}_{cV1}$ (corresponding to the circuit $c - V_1$) from identity using the normalized Frobenius norm given by

$$D(\mathcal{U}_{cV1}, \mathcal{I}) = \frac{\|\mathcal{U}_{cV1} - I\|_F}{\sqrt{L}} = \frac{\sqrt{\sum_i^L \sum_j^L |\mathcal{U}_{ij} - \delta ij|^2}}{\sqrt{L}}.$$

This distance is obtained for $n_x = 8$, $R = 8$, $d = 1$, and $dt = 0.1$ for which we have done the hardware run. For $n_p = 1, 2$, the distance is of the order of $1e^{-2}$, whereas for a 2D problem, i.e., for $d = 2$, the corresponding distance is of the order $1e^{-3}$. Thus, to a good degree of approximation, we can assume that the $c - V_1$ component of the circuit is close to the identity and therefore can be ignored for our hardware implementation. Since the full quantum circuit for a single time step is made from three components, $c - V_1, V_1^{-1}$ and V_2, by ignoring $c - V_1$ at least one-third of the circuit depth can be reduced. In general, these approximations will hold true for smoothly varying functions where the second order is much smaller than the gradient.

III. Simulation Results

A. Simulation - Advection Equation

We first, present simulation results for the advection equation to confirm the expected advective transport behavior. For classical simulation, we have used CUDA-Q 0.9.1 [33], an open-source toolkit that can be used for building and executing quantum circuits on a local simulator. As the backend, the *Statevector* simulator was used to get the full quantum state.

1) Two dimensional Shear: First, we consider the two-dimensional advection equation with periodic boundary condition (PBC) and transport field $v = (\frac{y}{L/2}, 0)$, such that the Courant–Friedrichs–Lewy (CFL) condition is maintained for numerical stability. The advection equation for shear in 2D with PBC is written as,

$$\begin{cases} \frac{\partial u(x,y,t)}{\partial t} + \frac{y}{L/2}\frac{\partial u(x,y,t)}{\partial x} = 0, \\ u(x,y,t) = u(x+L,y,t), \quad u(x,y,t) = u(x,y+L,t). \end{cases} \tag{25}$$

where as $x, y \in [0, L]$. The initial state $u_{shear}(x, y, 0)$ is a 2D Gaussian,

$$u_{shear}(x,y,0) = \frac{1}{\mathcal{N}} \exp\left(-\frac{(x-\mu)^2 + (y-\mu)^2}{2\sigma^2}\right), \tag{26}$$

where $\mu = L/2$, $\sigma = L/4$, and $\mathcal{N}$ is the normalization factor. This is chosen so as to have a smooth initial state such that we can get very good accuracy with just one ancilla. If we let this evolve according to Eq. (25) for time T, the analytical solution is given by

$$u_{shear}(x,y,T) = \frac{1}{\mathcal{N}} \exp\left(-\frac{(x - \frac{y}{L/2}T - \mu)^2 + (y-\mu)^2}{2\sigma^2}\right). \tag{27}$$

Under the action of this transport field, different layers of the fluid move along the x-direction with velocities that vary linearly with respect to the y-direction. This can be implemented using the circuit from Eq. (20) with the only difference being that the transport field $v_x = y$ is normalized by dividing with $L/2$.

In Fig. 3(a-e), we show the simulation results of the evolution of the 2D scalar function Eq. (26) for $n_x = n_y = 10$, i.e, $1024 \times 1024 (L \times L)$ grid points. We have chosen $R = 4$ and $T = 128$ and run the simulation for different time stepsize $dt = 0.5$, $dt = 0.1$, $dt = 0.02$. It can be seen that by decreasing dt, the proposed method approaches the analytical solution, thereby validating our proposed method. For various values of dt, we also calculate the relative ℓ_2-norm $\mathbf{e}_{dt}$, defined as

$$\mathbf{e}_{dt} = \frac{\|u_{shear}(x,y,T) - u_Q(x,y,T)\|_2}{\|u_{shear}(x,y,T)\|_2} \tag{28}$$

where $u_Q(x, y, T)$ is the numerical solution obtained by the proposed algorithm. Here the denominator is always 1 since $u_{shear}(x, y, T)$ is normalized. For the different cases plotted in Fig. 3(c-e), the ℓ_2-norm errors are $\mathbf{e}_{0.5} = 2.88e^{-2}$, $\mathbf{e}_{0.1} = 5.89e^{-3}$ and $\mathbf{e}_{0.02} = 2.25e^{-3}$.

2) Two dimensional Rotation: We consider a 2D advection equation with periodic boundary conditions, and with a transport field $v = (\frac{y-L/2}{L/2}, \frac{-(x-L/2)}{L/2})$. The advection equation can be put together as

$$\begin{cases} \frac{\partial u(x,y,t)}{\partial t} + \frac{y-L/2}{L/2}\frac{\partial u(x,y,t)}{\partial x} - \frac{(x-L/2)}{L/2}\frac{\partial u(x,y,t)}{\partial y} = 0, \\ u(x,y,t) = u(x+L,y,t), \quad u(x,y,t) = u(x,y+L,t). \end{cases} \tag{29}$$

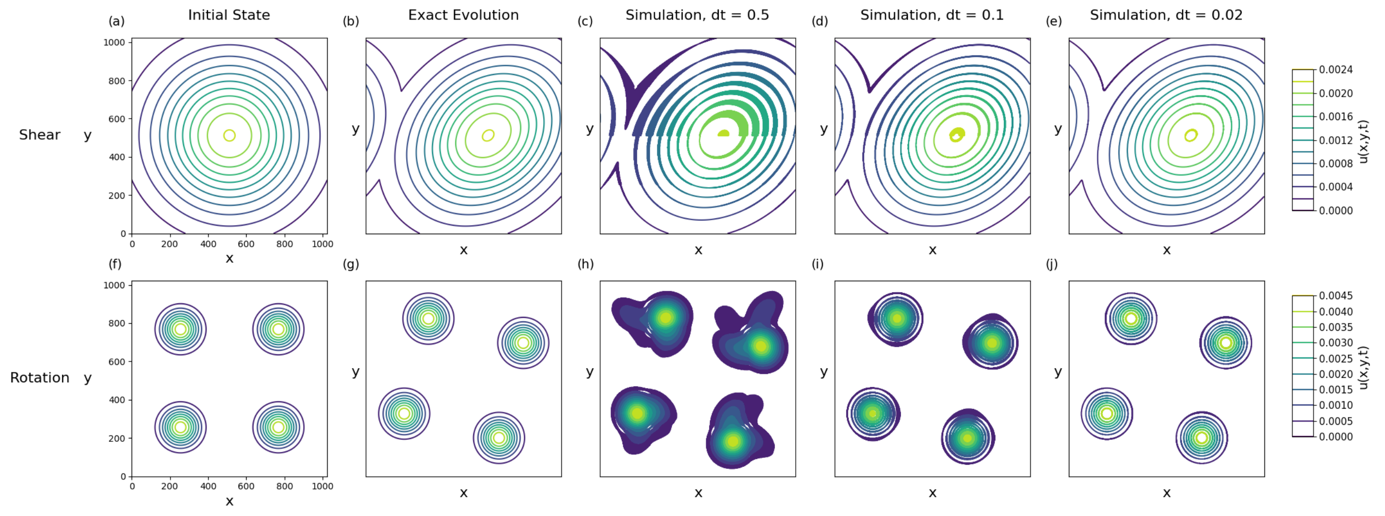

Fig. 3. **Simulation results of the advection equation with periodic BC in a 1024 $\times$ 1024 (20 qubits) lattice**: The top row represents the evolution of the scalar field $u_{shear}(x, y, t)$ with transport field $(\frac{y}{L/2}, 0)$. Plot (a) represents the initial state while (b) is the exact analytical solution at time $T = 128$. Plots (c), (d), and (e) are the solutions obtained from the proposed method at time $T = 128$, with $dt = 0.5, 0.1$, and 0.02 respectively. Similarly, the bottom row represents the evolution of the scalar field $u_{rotation}(x, y, t)$ with transport field $(\frac{y-L/2}{L/2}, \frac{-(x-L/2)}{L/2})$. For the initial state is given in (f), the analytical solution is plotted in (g) at time $T = 128$. Similar to shear, we have plotted the solutions from the quantum simulation for $dt = 0.5, 0.1$, and 0.02 in (h), (i), and (j) respectively. In both scenarios, we observe improvement in the numerical solution with a decrease in time increment size dt.

where $x, y \in [0, L]$. For the initial state $u_{rotation}(x, y, 0)$, we choose a superposition of 2D Gaussian states

$$u_{rotation}(x, y, 0) = \sum_{i,j=\{1,2\}} \frac{1}{\mathcal{N}} \exp\left(-\frac{(x-\mu_i)^2 + (y-\mu_j)^2}{2\sigma^2}\right) \tag{30}$$

where the constants have values $\mu_1 = L/2 + L/4$, $\mu_2 = L/2 - L/4$, $\sigma = L/16$, and $\mathcal{N}$ is the normalization factor. The choice of initial state helps us to observe the evolution implemented with this transport field which is a clockwise rotation around the point $(L/2, L/2)$. Generally, a clockwise rotation around $(0, 0)$ is implemented via the transport field of the form $v = (y, -x)$. This leads to a rotation around a corner in our implementation. In order to see the effect of the rotation more prominently, we shift the center for rotation to $(L/2, L/2)$. The analytical solution to this problem at time T is given by,

$$u_{rotation}(x, y, T) = \frac{1}{\mathcal{N}} \sum_{i,j=\{1,2\}} \exp\left(-\frac{(x'(T)-\mu_i)^2 + (y'(T)-\mu_j)^2}{2\sigma^2}\right), \tag{31}$$

where the rotated coordinates are

$$x'(T) = (x - \frac{L}{2})\cos\left(\frac{2T}{L}\right) - (y - \frac{L}{2})\sin\left(\frac{2T}{L}\right) + \frac{L}{2} + mL,$$
$$y'(T) = (x - \frac{L}{2})\sin\left(\frac{2T}{L}\right) + (y - \frac{L}{2})\cos\left(\frac{2T}{L}\right) + \frac{L}{2} + m'L,$$

where $m, m' \in \mathbb{Z}$. Implementing this shifted rotation with our method is a two-step application of Eq. (20) for every time step. The first step is the advection along $x-$direction with a transport $\frac{y}{L/2}$ followed by an advection with constant transport -1, which makes the resulting x velocities to be $\frac{y-L/2}{L/2}$. Similarly, the second component is an advection along the $y-$direction with transport $\frac{-x}{L/2}$ followed by an advection with constant transport 1, which results in $v_y = \frac{L/2-x}{L/2}$.

The corresponding numerical solution has been plotted in Fig. 3 (f-j). We choose the same problem size as the shear, i.e, for $n_x = n_y = 10$, i.e, $1024 \times 1024 (L \times L)$ grid points and $R = 4$ and $T = 128$. We have run the simulation for different time step sizes $dt = 0.5$, $dt = 0.1$, $dt = 0.02$. Again we observe that the results agrees closely with the analytical solution for $dt = 0.02$. Since the source of error in our proposed method is due to Trotterization, it can be improved by reducing the time step size, which is verified in Fig. 3(h-j), where we see the improvement in simulation results with decreasing dt. In this case the ℓ_2-norm of relative errors for different values of dt are $\mathbf{e}_{0.5} = 0.32$, $\mathbf{e}_{0.1} = 6.88e^{-2}$ and $\mathbf{e}_{0.02} = 1.89e^{-2}$.

B. *Simulation- Advection-Diffusion Equation*

1) Two-dimensional Shear and Rotation: So far, we have presented the simulation results for the advection equation only. In this section, we will present the simulation results by including the diffusion term as well. The advection-diffusion equations for the two-dimensional shear and rotation transport fields are obtained by adding a diffusion term to the right-hand side of the equations Eq. (25) and Eq. (29), respectively. The 2D diffusion term is of the following form

$$\nabla^2 u(x, y, t) = D\left(\frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2}\right), \tag{32}$$

where the diffusion coefficient has the value $D = 1$ for all the results presented in this section. We take the same

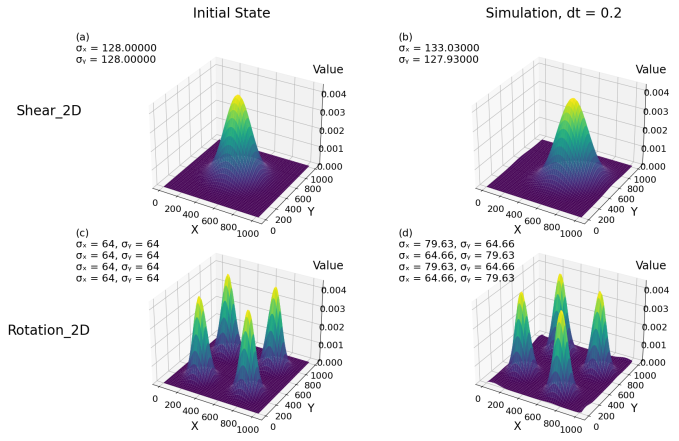

Fig. 4. **Simulation results of the advection-diffusion equation with periodic BC in a 1024×1024 lattice**: The initial states for shear and rotation are plotted in (a) and (c). We run both simulations for a time $T = 128$ with $dt = 0.2$. The final form of the evolved states is plotted in (b) and (d). It is readily seen that the evolved gaussian are more spread out due to diffusion. For brevity, we have mentioned the variance values for the corresponding plots, which tell about the degree of spread due to diffusion.

initial Gaussian states for the respective cases as in Eq. (26) and Eq. (30). For the 2D shear transport field, we change the variance $\sigma = L/8$ due to the fact that diffusion terms becomes visibly significant for a peaked Gaussian initial state. The corresponding analytical solutions in the presence of the diffusion term maintains the same mathematical form as in Eq. (27) and Eq. (31), with an enlarged variance, which at time T is given by

$$\sigma_T^2 = \sigma^2 + 2DT, \tag{33}$$

where σ_T^2 and σ^2 are the variances at time T and at the beginning of evolution. In Fig. 4, we have plotted the evolution for a 2D advection-diffusion equation with shear and rotation transport fields with periodic boundary conditions for a grid size $N_x = N_y = 1024$. The simulation runs for time $T = 128$ with $dt = 0.2$ and $R = 4$. The total qubits required to implement this evolution are 23, where qubits representing the grid are $n_x = n_y = 10$ and the ancilla qubits $n_p = 3$. The relative l_2-norm error for the 2D shear and rotational transport field are $\mathbf{e}_{0.2} = 8.39e^-3$ and $\mathbf{e}_{0.2} = 1.30e^-2$ respectively.

2) Three dimensional Shear: Lastly, we consider a three-dimensional (3D) advection-diffusion equation with periodic boundary conditions and transport field $v = (\frac{z}{L/2}, \frac{z}{L/2}, 0)$

$$\begin{cases} \frac{\partial u(x,y,z,t)}{\partial t} + \frac{z}{L/2}\frac{\partial u(x,y,t)}{\partial x} + \frac{z}{L/2}\frac{\partial u(x,y,t)}{\partial y} = D\nabla^2 u(x,y,z,t), \\ u(x,y,z,t) = u(x+L,y,z,t), \\ u(x,y,z,t) = u(x,y+L,z,t), \\ u(x,y,t) = u(x,y,z+Lt), \end{cases} \tag{34}$$

where the computation domain is given by $x, y, z \in [0, L]$ and the $D = 1$. The 3D diffusion term has the following form

$$\nabla^2 u(x,y,z,t) = D\left(\frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} + \frac{\partial^2 u(x,y,t)}{\partial z^2}\right). \tag{35}$$

We take the initial state as a 3D Gaussian

$$u_{3Dshear}(x,y,z,0) = \frac{1}{\mathcal{N}} \exp\left(-\frac{(x-\mu)^2 + (y-\mu)^2 + (z-\mu)^2}{2\sigma^2}\right), \tag{36}$$

where $\mu = L/2$ and $\sigma = L/8$. Under evolution with Eq. (34), the solution at time T is given by

$$u_{3Dshear}(x,y,z,T) = \frac{1}{\mathcal{N}} \exp\left(-\frac{(x - \frac{z}{L/2} - \mu)^2 + (y - \frac{z}{L/2} - \mu)^2 + (z-\mu)^2}{2\sigma_T^2}\right), \tag{37}$$

where the σ_T is defined from Eq. (33). The evolution due to this transport field results in advection along x and y axes with linearly varying velocities along z direction. This can be implemented with Eq. (20), using two such circuits, one along the x and y direction each with normalized velocities $\frac{y}{L/2}$ and $\frac{x}{L/2}$ respectively. We plot the 3D evolution results in Fig. 5, for $N_x = N_y = N_z = 9$, i.e, a grid size of $512 \times 512 \times 512$, $n_p = 3$, $R = 4$, $dt = 0.1$, and $T = 64$. To check the accuracy of the result of the simulated result with respect to the exact evolution, we calculated the relative ℓ_2-norm error, which comes out to be $\mathbf{e}_{0.1} = 1.12e^{-2}$. 30 total qubits are required to run this simulation of which 27 qubits are for the physical grid and 3 ancilla qubits.

Note: There are further improvements possible in the simulation results for all the cases presented above by having more mesh points both in x and p, which requires setting up a problem with a bigger system size with n_x and more ancilla qubits n_p. We refer the reader to [28] for the analysis on how the system and ancilla qubits affect the accuracy of the simulation since it is applicable in our case also.

IV. Hardware results

In order to demonstrate our algorithm, we run our computations for several time steps. This amounts to running the circuits for eq 15 in IBM's 156-qubit processor ibmq_fez based on the Heron architecture. Given the complexity of the upwinding scheme, the circuit depth for our quantum simulation even for one time step could be gigantic. In order to run them on the hardware, substantial transpilation and approximation were implemented without compromising much of the accuracy. We describe the methods here.

1) Circuit construction and error suppression: The initial state for the simulation is constructed using Qiskit's `prepare_state()`. When a Statevector argument is passed in form of an array, this function prepares the state based on the Isometry synthesis [34]. Like all Hamiltonian simulation algorithms, our approach also experiences increasing circuit

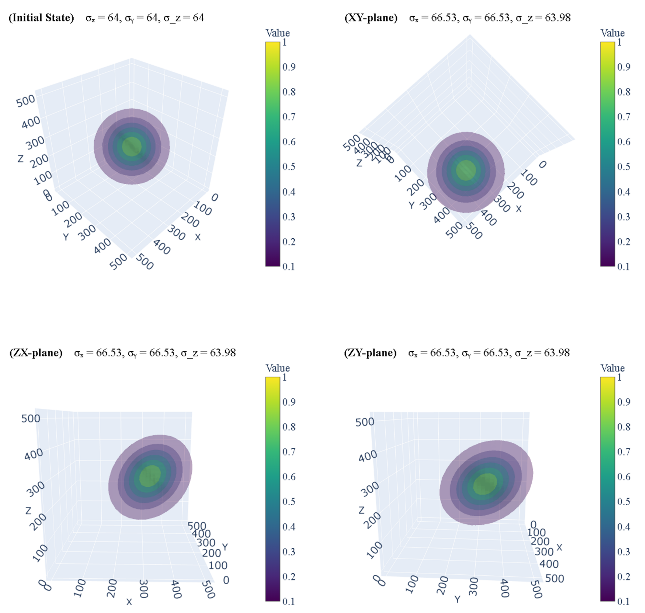
(Initial State) σₓ = 64, σᵧ = 64, σ_z = 64
(XY-plane) σₓ = 66.53, σᵧ = 66.53, σ_z = 63.98
(ZX-plane) σₓ = 66.53, σᵧ = 66.53, σ_z = 63.98
(ZY-plane) σₓ = 66.53, σᵧ = 66.53, σ_z = 63.98
Value
X
Y
Z

Fig. 5. **Simulation results of the advection equation with periodic BC in a $512 \times 512 \times 512$ (27 qubits) lattice**: The initial state is plotted in (a), which evolves according to the 3D advection-diffusion equation with transport field $(\frac{z}{L/2}, \frac{z}{L/2}, 0)$ with diffusion $D = 1$. The evolved state from different view angles are plotted in (b), (c), and (d), which are obtained by running the simulation for time $T = 64$ with $dt = 0.2$. The diffusion effect is seen in the enlarge value of the variance of the Gaussian which corresponds to the spread.

depth with each time step. For a 16-qubit problem (8×8 lattice), the circuit depth for a single time step reaches approximately $100,000$. To mitigate this issue, we first extract the statevector of the quantum circuit at each time step and reconstruct it using Qiskit's `initialize()` function. This approach helps keep the circuit depth limited to that of a single time step.

Once we construct the circuit for the full simulation, we apply an additional round of optimization using approximate quantum compilation with tensor networks (AQC-Tensor) to achieve a lower circuit depth than would typically be required for the entire evolution. Specifically, we generate a target tensor-network state for the full circuit using the Qiskit Aer matrix-product state (MPS) simulator, currently the only supported tensor network backend. We then construct a matrix-product representation of the state, which AQC approximates, which we call $|u\rangle_{AQC}$.

Next, we build a general, parameterized ansatz circuit using Qiskit's `RealAmplitude`, where the optimized parameters yield the tensor network circuit. To achieve this, we minimize the simplest possible cost function, `MaximizeStateFidelity`, using the L-BFGS optimizer from SciPy. We will call this ansatz state as $|\tilde{u}(\Theta)\rangle$, where $\Theta := \{\theta_0, \theta_1, ..\}$, $\theta_j \in \mathbb{R}$. For the results shown in this work, we use $|\langle\tilde{u}(\Theta)|u_{AQC}\rangle| > 0.99$. In other words, our ansatz has $> 99\%$ fidelity with the AQC approximate state. This step enables us to get down our circuit depth from $\sim 100,000$ to 500. Starting from these reconstructed circuits of $|u\rangle_{AQC}$, we compile the most efficient version using Qiskit's pass manager with an optimization level of two. Additionally, we apply dynamical decoupling using standard Qiskit tools by incorporating periodic gate sequences, aiming to mitigate undesired system-environment interactions affecting idle qubits [29].

2) Error mitigation: We implemented two error mitigation techniques based solely on the observed characteristics of the hardware output. The first approach stems from noticing that the histogram of noisy bitstrings includes contributions from less relevant states, which are not present in the ideal, noise-free distribution [35]. These extraneous contributions reduce the weight of more meaningful basis states. To address this, we eliminate bitstrings with counts below a threshold ϵ_{th} by setting their counts to zero and then redistribute those counts evenly among the more significant states. Furthermore, since we are analyzing a shear operation in two dimensions, we anticipate an increase in gate usage, and consequently, noise, as we move farther from the $X = 0$ axis. To account for this spatial variation, the threshold is not kept constant but scaled with position as $\epsilon_{th} = y\epsilon_{cut}$, where $\epsilon_{cut} = 1e^{-5}$.

Although the initial step improves the results, the mitigated data still suffers from the fact that the number of measurements (shots) for the full state tomography is finite. So, we apply the Savitzky–Golay technique to smooth the fluctuations [36]. Commonly used in digital signal processing, this filter smooths a dataset to improve accuracy without altering the overall shape of the signal. In our work, we apply the method in both x and y directions. The final results are discussed in the next section.

A. Results

We run our hardware experiments for a 256×256 grid lattice moving under a transport field $v = (y/L, 0)$. For a lattice of that size, we need only $8 + 8$ that is 16 qubits. We have used two ancilla qubits ($n_p = 2$) and for advection and $n_p = 3$ for advection-diffusion and $R = 8$ in order to incorporate the warped transformation. Following the discussion in section II-E, we can ignore the $c - V_1$ term and keep only the inverse of H_1 and H_2 in order to maintain the upwinding scheme for advection. However, for diffusion we keep all terms. We keep the effect of the number ancilla qubit on the accuracy for future research.

We assume PBC and an initial field which is a superposition of two gaussian in each direction,

$$u(x, y, 0) = \frac{1}{\mathcal{N}} \sum_{i,j=\{1,2\}} \exp\left(-\frac{(x-\mu_i)^2 + (y-\mu_j)^2}{2\sigma^2}\right),$$

where $\mathcal{N}$ is a normalization factor. We have chosen $\mu_1 = L/2 - L/4$, $\mu_2 = L/2 + L/4$ and $\sigma = L/8$. We run the simulation upto $T = 40$ for $dt = 1.0$, that is for 40 Trotter steps. The results are shown in Fig 6 (a-d). Figure 6 (a) shows the initial state $u(x, y, 0)$, (b) is the statevector result and (c) is the raw data obtained from a full state tomography from the hardware. Since u is real, the complex phase factor is

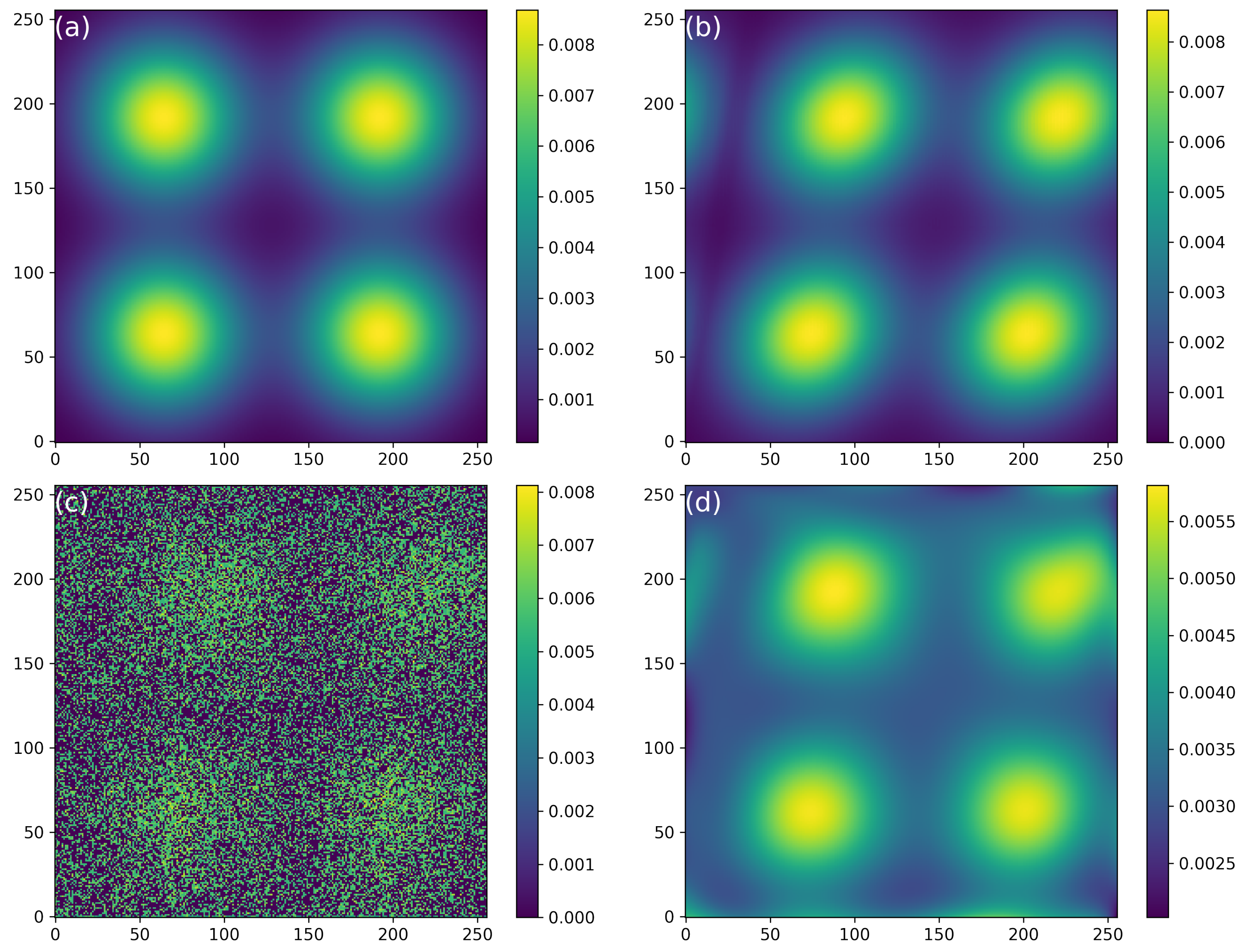

Fig. 6. **Hardware results for (256 × 256) grid points in two dimensions (16 qubits) :** . The color contour plot represents the values of the scalar field $u(t)$ moving with a transport $(x, 0)$ at each grid point. of the scalar field. (a) The initial scalar field at $t = 0$. Results for $u(t = 20.0)$, dt = 1.0 at statevector level, and hardware results are shown in (b) and (c), respectively. (d) is obtained after applying Savitzky–Golay filter to the noisy data in (c). It clearly shows that the scalar field could be reconstructed from finite number of shots (100,000).

irrelevant here. We take a square root and normalize the counts to obtain the coefficients $c_{x,y}$ in eq. (18) and plot them in (c) after reshaping them as a $(2^{n_x}, 2^{n_y})$ array. In (d) we show the hardware results after performing the error mitigation schemes described in section IV-2. The ℓ_2 norm error for our hardware result is $\mathbf{e} = 1.96$.

The results for advection-diffusion is shown in Fig 7. Following [28], we present the total energy $\|u\|^2$ by measuring the observable $\hat{O}$ for different final time $T \in \{0, 10, 30\}$. $\hat{O}$ is defined as,

$$\hat{O} = I^{\otimes(n_x+n_y)} \otimes |1\rangle\langle 1| \otimes |0\rangle\langle 0|^{\otimes(n_p-1)} \tag{38}$$

which projects the full quantum state (including the ancilla) $|\hat{\mathbf{v}}\rangle\,(T)$ on to $|u(T)\rangle$. The result is show in fig. 7.

V. Conclusion and Future Work

We have developed a new scalable quantum algorithm to include non-zero vorticity into the advection-diffusion equation. In the advection component of the equation, we incorporated non-trivial transport fields using an upwinding discretization scheme and demonstrated it for shear and rotational transport fields on an emulator. The simulation results scale up to 30 qubits grid size for the shear flow in 3D. We have also proposed a few approximations to the upwinding scheme that reduce the depth of the quantum circuit and avoid the use of ancilla qubits and mid-circuit measurements. This makes the algorithm more NISQ-friendly. Finally, we have demonstrated our method on a 16-qubit system which is equivalent to a grid of 256×256 lattice on a real quantum hardware for advection and a 19-qubit system to simulate advection and diffusion for the same grid size. Since diffusion is a non-conservative process we have used three ancilla qubits.

The outcome of a quantum computer simulation is ultimately encoded in a quantum state, as described in Eq. (3). In general, extracting full information—such as all quantum amplitudes—incurs exponential cost, making this approach

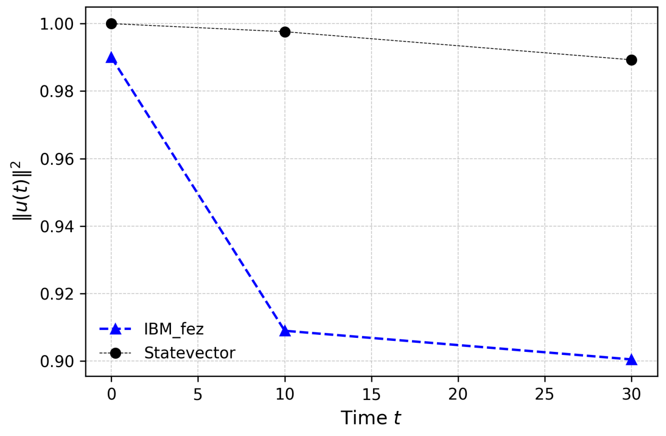

Fig. 7. Hardware results for advection diffusion for a 256×256 grid in 2D. We have used 19 qubits (16 system qubits and 3 ancilla qubits). The plot shows the evolution of $\||u(t)\rangle\|^2$ which is obtained by measuring the observable $\hat{O}$ in eq. (38)

impractical for large systems. Therefore, it is crucial to focus on 'macroscopic' observables, such as averages or higher moments of dynamic variables, which can be efficiently estimated from the final state. In future work, we plan to explore observables like flux or kinetic energy [28], whose expectation values can be accessed without such overhead. Additionally, the recently introduced concept of classical shadows [37] offers a promising framework for efficiently retrieving essential features from the quantum output.

As a natural progression of this work, we aim to extend our framework to accommodate more complex transport fields, including time-dependent and turbulent flows. Future efforts will also focus on scaling the algorithm to a larger number of qubits through advanced quantum circuit optimization techniques. Beyond fluid dynamics, we envision applying our methods to a broader class of partial differential equations, such as the heat equation, Maxwell's equations, and elasticity problems involving stress and strain. Ultimately, our long-term objective is to enable the simulation of general-purpose PDEs through Hamiltonian-based quantum algorithms.

Acknowledgement: The authors thank Chris Hill and Hamed Mohammadbagherpoor for useful discussions.

Appendix A
Circuits for implementing $\tilde{V}_D(\tau)$, $\tilde{V}_1(t)$ and $\tilde{V}_2(t)$

In this section we present the circuits required for implementing $\tilde{V}_D(\tau)$, $\tilde{V}_1(\tau)$ and $\tilde{V}_2(\tau)$. $\tilde{V}_1(\tau)$ and $\tilde{V}_D(\tau)$ have the same circuit, with the only difference being the input γ_D and γ_1 values. Due to this, we will present the circuits for $\tilde{V}_1(\tau)$ only. As described in the circuits in Fig. 8, $V_1(\tau)$ and $V_2(\tau)$ are implemented via the unitary operator $W_j(\gamma\tau, \lambda, x)$ that takes as input γ and $\tau = dt$, λ and $x \in \{0, 1\}$. The full circuit for $\tilde{V}_1(\tau)$ and $\tilde{V}_2(\tau)$ are then implemented as

$$\tilde{V}_1(\tau) := \prod_{\alpha=1}^{d} (V_1(\tau))_\alpha,$$

$$\tilde{V}_2(\tau) := \prod_{\alpha=1}^{d} (V_2(\tau))_\alpha.$$

For more details on how the individual circuits for $W_j(\gamma\tau, \lambda, x)$, $V_1(\tau)$, and $V_2(\tau)$ are obtained, we refer the reader to [28] from which these are inspired.

References

[1] Jeffrey Yepez. Quantum computation of fluid dynamics. In *NASA International Conference on Quantum Computing and Quantum Communications*, pages 34–60. Springer, 1998.

[2] Sachin S Bharadwaj and Katepalli R Sreenivasan. Quantum computation of fluid dynamics. *arXiv preprint arXiv:2007.09147*, 15:1–20, 2020.

[3] Mark Horowitz and Emily Grumbling. Quantum computing: progress and prospects. 2019.

[4] John F Wendt. *Computational fluid dynamics: an introduction*. Springer Science & Business Media, 2008.

[5] Randall R.J. J LeVeque. *Numerical Methods for Conservation Laws*, volume 54. 2002.

[6] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, 2000.

[7] Reuben Demirdjian, Daniel Gunlycke, Carolyn A Reynolds, James D Doyle, and Sergio Tafur. Variational quantum solutions to the advection–diffusion equation for applications in fluid dynamics. *Quantum Information Processing*, 21(9):322, 2022.

[8] Fong Yew Leong, Dax Enshan Koh, Wei-Bin Ewe, and Jian Feng Kong. Variational quantum simulation of partial differential equations: applications in colloidal transport. *International Journal of Numerical Methods for Heat & Fluid Flow*, 33(11):3669–3690, 2023.

[9] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[10] Koichi Miyamoto, Soichiro Yamazaki, Fumio Uchida, Kotaro Fujisawa, and Naoki Yoshida. Quantum algorithm for the vlasov simulation of the large-scale structure formation with massive neutrinos. *Physical Review Research*, 6(1):013200, 2024.

[11] Pedro CS Costa, Stephen Jordan, and Aaron Ostrander. Quantum algorithm for simulating the wave equation. *Physical Review A*, 99(1):012323, 2019.

[12] Ryan Babbush, Dominic W Berry, Robin Kothari, Rolando D Somma, and Nathan Wiebe. Exponential quantum speedup in simulating coupled classical oscillators. *Physical Review X*, 13(4):041041, 2023.

[13] Shi Jin, Nana Liu, and Yue Yu. Quantum simulation of partial differential equations: Applications and detailed analysis. *Physical Review A*, 108(3):032603, 2023.

[14] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[15] Jeffrey Yepez. Lattice-gas quantum computation. *International Journal of Modern Physics C*, 9(08):1587–1596, 1998.

[16] Nikita Gourianov, Michael Lubasch, Sergey Dolgov, Quincy Y van den Berg, Hessam Babaee, Peyman Givi, Martin Kiffner, and Dieter Jaksch. A quantum-inspired approach to exploit turbulence structures. *Nature Computational Science*, 2(1):30–37, 2022.

[17] Koji Fukagata. Towards quantum computing of turbulence. *Nature Computational Science*, 2(2):68–69, 2022.

[18] Furkan Oz, Omer San, and Kursat Kara. An efficient quantum partial differential equation solver with chebyshev points. *Scientific Reports*, 13(1):7767, 2023.

[19] René Steijl and George N Barakos. Parallel evaluation of quantum algorithms for computational fluid dynamics. *Computers & Fluids*, 173:22–28, 2018.

[20] Frank Gaitan. Finding flows of a navier–stokes fluid through quantum computing. *npj Quantum Information*, 6(1):61, 2020.

[21] Bolesław Kacewicz. Almost optimal solution of initial-value problems by randomized and quantum algorithms. *Journal of Complexity*, 22(5):676–690, 2006.

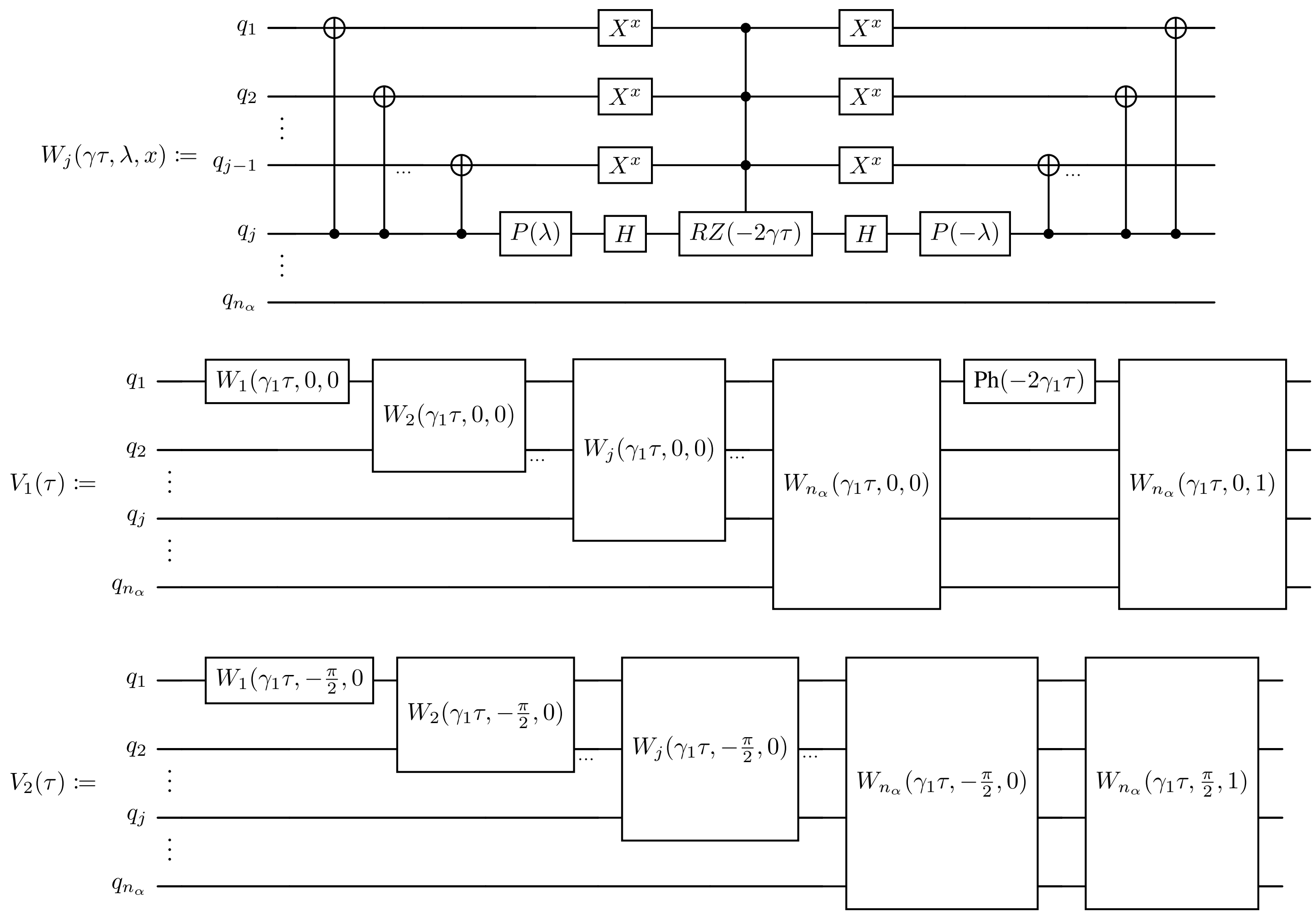
$W_j(\gamma\tau, \lambda, x) :=$
q_1
q_2
q_{j-1}
q_j
q_{n_α}
X^x
$P(\lambda)$
H
$RZ(-2\gamma\tau)$
$P(-\lambda)$
$V_1(\tau) :=$
$W_1(\gamma_1\tau, 0, 0$
$W_2(\gamma_1\tau, 0, 0)$
$W_j(\gamma_1\tau, 0, 0)$
$W_{n_\alpha}(\gamma_1\tau, 0, 0)$
$\mathrm{Ph}(-2\gamma_1\tau)$
$W_{n_\alpha}(\gamma_1\tau, 0, 1)$
$V_2(\tau) :=$
$W_1(\gamma_1\tau, -\frac{\pi}{2}, 0$
$W_2(\gamma_1\tau, -\frac{\pi}{2}, 0)$
$W_j(\gamma_1\tau, -\frac{\pi}{2}, 0)$
$W_{n_\alpha}(\gamma_1\tau, -\frac{\pi}{2}, 0)$
$W_{n_\alpha}(\gamma_1\tau, \frac{\pi}{2}, 1)$

Fig. 8. Quantum circuits for $W_j(\gamma\tau, \lambda, x), V_1(\tau)$ and $V_2(\tau)$.

[22] Furkan Oz, Rohit KSS Vuppala, Kursat Kara, and Frank Gaitan. Solving burgers' equation with quantum computing. *Quantum Information Processing*, 21(1):30, 2022.

[23] Adrien Suau, Gabriel Staffelbach, and Henri Calandra. Practical quantum computing: Solving the wave equation using a quantum approach. *ACM Transactions on Quantum Computing*, 2(1):1–35, 2021.

[24] Melody Lee, Zhixin Song, Sriharsha Kocherla, Austin Adams, Alexander Alexeev, and Spencer H Bryngelson. A multiple-circuit approach to quantum resource reduction with application to the quantum lattice boltzmann method. *arXiv e-prints*, pages arXiv–2401, 2024.

[25] Peter Brearley and Sylvain Laizet. Quantum algorithm for solving the advection equation using hamiltonian simulation. *Physical Review A*, 110(1):012430, 2024.

[26] Yuki Sato, Ruho Kondo, Ikko Hamamura, Tamiya Onodera, and Naoki Yamamoto. Hamiltonian simulation for hyperbolic partial differential equations by scalable quantum circuits. *Phys. Rev. Res.*, 6:033246, Sep 2024.

[27] George Keith Batchelor. *An introduction to fluid dynamics*. Cambridge university press, 2000.

[28] Junpeng Hu, Shi Jin, Nana Liu, and Lei Zhang. Quantum Circuits for partial differential equations via Schrödingerisation. *Quantum*, 8:1563, December 2024.

[29] Nic Ezzell, Bibek Pokharel, Lina Tewala, Gregory Quiroz, and Daniel A. Lidar. Dynamical decoupling for superconducting qubits: A performance survey. *Phys. Rev. Appl.*, 20:064027, Dec 2023.

[30] Peter Brearley and Sylvain Laizet. Quantum algorithm for solving the advection equation using hamiltonian simulation. *Phys. Rev. A*, 110:012430, Jul 2024.

[31] Charles R Molenkamp. Accuracy of finite-difference methods applied to the advection equation. *Journal of Applied Meteorology and Climatology*, 7(2):160–167, 1968.

[32] Bram Van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of computational Physics*, 32(1):101–136, 1979.

[33] The CUDA-Q development team. Cuda-q.

[34] Raban Iten, Roger Colbeck, Ivan Kukuljan, Jonathan Home, and Matthias Christandl. Quantum circuits for isometries. *Physical Review A*, 93(3):032318, 2016.

[35] Niladri Gomes, David B Williams-Young, and Wibe A de Jong. Computing the many-body green's function with adaptive variational quantum dynamics. *Journal of Chemical Theory and Computation*, 19(11):3313–3323, 2023.

[36] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[37] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020.